I. INTRODUCTION

Java Essential Dynamics (JED) is a java library (a package of programs) for analyzing protein trajectories. The trajectories may be derived from MD, FIRST/FRODA, or any other dynamic simulations that output a trajectory as a set of PDB files. The program can handle single chain PDB files with no chain identifier as well as multi chain PDB files that use chain IDs. The user may specify the set of residues to be considered for the analysis, and this set need not be contiguous. A variety of utility tools are provided that use **Principal Component Analysis** (PCA) that are not found in MD-simulation packages or other stand alone PCA software, especially in regards to comparative analysis of multiple trajectories. JED is capable of running on any platform with a suitable Java Runtime Environment (JRE).

Expected Input to JED:

Ideally, each PDB structure must follow standard PDB-format. Note that some deviations from standard will often work fine, but JED expects standard format. Moreover, it is required that the structures have been prepared in such a way that there are no gaps in residue labeling. If residues or consecutive regions of residues are missing, these need to be fixed, or the residue labeling has to be altered in order to remove gaps in residue labels. The first residue label must start at 1 or higher. No 0 or negative numbers are allowed for residue labels. All preprocessing of the PDB files must be done with external software before using JED. It is convenient to label PDB files using leading zeros in the name of the files to simplify tracking time progression. For example, if a simulation generates 100,000 frames in the trajectory, it is best to name the PDB files like <file_name_000000>, <file_name_000001>, … <file_name_100001>, which specifies that relative to the starting structure 100,000 frames are generated in successive order. Although this naming scheme is not required, it is highly recommended because it allows the user to track time order easily on operating systems that sort order by literal alphabetic-characters, rather than interpreting 34 is less than 100, for example.

JED Preprocessing Output:

As a pre-processing step, JED reads in all PDB files in a specified directory and aligns all the structures in the trajectory to a specified reference structure using a quaternion alignment algorithm. A matrix of the **read PDB coordinates,** obtained from all the residues in the input PDB files, is created so that it can be used for all subsequent JED runs. A list of all the residues (**residue list**) found in the PDB files (along with the chain IDs when appropriate) is generated. The original and transformed **conformation rmsd** are determined for each member structure in the trajectory relative to the specified reference structure. The **residue rmsd** (also commonly referred to as **RMSF**) is determined from the entire trajectory. An **edited PDB file** is also generated where the B-factors are replaced with the **residue rmsd** values for visualization purposes. This output automatically happens and is non-optional.

Carbon Alpha Atoms:

The current implementation of JED only considers Cα atoms. As such, we speak about residues because the information is tied to the Cα atoms, which represents the dynamics of the residue at some coarse grained level of description. For example, the distance between two residues is modeled in JED as the distance between the two Cα atoms associated with the two residues. This choice of working only with Cα atoms allows the labeling of the Cα atoms to be associated with residue labels. For a single chain protein, this is a simple 1 to 1 mapping. For multiple chain proteins, JED also tracks the chain ID.

Different Types of PCA:

The core element of essential dynamics is to perform PCA. JED implements two variations of PCA. The first and most common method is based on Cartesian coordinates (**cPCA**). The second method is based on internal coordinates using either all-to-all residue distances (**dPCA**) and/or residue-pair distances (**dpPCA**). The dPCA is much more computationally expensive and the interpretation can be difficult when the number of residues in the subset exceeds ten. Note that dPCA using ***n*** residues will yield eigenvectors having ***n*(n-1)/2** components, each corresponding to one set of inter-residue distances. Thus, for five residues, one obtains ten pairs of interactions, and this becomes difficult to interpret.

Both the cPCA and dPCA and dpPCA can be constructed as a **covariance matrix** (**Q**) and/or **correlation matrix** (**R**). The correlation matrix is a normalized version of the covariance matrix. The results obtained from **Q** and **R** generally differ somewhat due to the inherent statistical biases in each approach. The current implementation does both.

Conditioning of the Q and R Matrices:

JED handles the removal of outliers prior to the PCA analyses with two approaches. First, the user can specify the **percent** of the structures to be removed based on the conformation rmsd. The most deviant structures are tagged as outliers and subsequently removed prior to the PCA analysis. In this first method, frames that are identified as an outlier are thrown out from the sample. Second, the user can specify a **z-score cutoff** such that when the value of a PCA variable (either a Cartesian or internal distance coordinate) has a |deviation| from the variable mean that exceeds the z-score cutoff, it is identified as an outlier. For each PCA-entry that is identified as an outlier, it is replaced with its mean. This process is done per variable over all frames, and each PCA-entry is treated independently. In this second method, a frame is never thrown away, but some entries within a frame may be modified. Both methods are intended to reduce the condition numbers of **Q** and **R**. While the first method of conditioning is most commonly employed in the protein field (if at all), the second method of conditioning is most commonly used in the field of statistics, and is the preferred method due to its superior effectiveness. It should be noted here that without applying the conditioning, the interpretation of the results of PCA can be completely wrong --- depending on the quality of the sampling. It is highly recommended to use the z-score cutoff conditioning method in all applications to avoid misinterpreting the PCA results.

Visualization of cPCA modes:

JED computes the **PCA modes** (RMSD and MSD, with and without weight by the corresponding eigenvalue) from the Cartesian eigenvectors so that they may be mapped to the residue set. As noted above, sets of structures can be generated to visually inspect the cPCA modes. Eigenvectors from dPCA cannot be mapped to the residue set in any simple way, so no mapping or visualization is attempted. The user can specify the number of Cartesian modes to visualize. Mode visualization is done by creating a set of 20 PDB files that capture the displacement of the alpha carbons for the given mode. A scale-factor parameter is adjustable to control the amount of displacement in the modes. A Pymol© script is generated to animate the frames.

Dimension Reduction Level:

The primary purpose of applying PCA to capture the essential dynamics of a protein is to reduce the large dimension of variables to a much smaller number of variables that captures the greatest variance in protein motion. The **Q** and **R** matrices, once diagonalized, provide a set of eigenvalues and eigenvectors. The eigenvalues for proteins typically fall off fast for the first several modes, out of possibly thousands of modes. The number of dimensions needed to provide a fair assessment of the essential dynamics in a protein is system-dependent. The user can specify any number (say 20, which typically is more than needed) to obtain results for all possible selections ranging from 1 up to the value selected. In this way, the user can see how the added dimensions help glean more information, albeit making it harder to interpret the greater number of dimensions. Eventually, the user must decide, based on their purpose/goals, the optimal number of dimensions to use for representing the essential dynamics.

Displacement Vectors:

A set of **displacement vectors** (**DV**s) based on the full conformational space is calculated using a specified reference structure. Those **DV**s are then projected onto a set of eigenvector directions to create delta vector projections (**DVP**s), which are similar to principle components (**PC**s). The **PC**s are delta vector projections, but according to the standard definition used in statistics, they are always relative to the mean conformation position as defined in the construction of the **Q** or **R** matrix. In studying the essential dynamics of a protein, it is common to use a reference structure that has a particular physical or biochemical meaning, which is why we call these displacements **DVP**s, and not **PC**s. The DVPs are very useful to have for visualizing protein motions. For example, if the first two eigenvector directions are selected (those eigenvectors associated with the highest and second highest eigenvalues, or variance) the **DVP**s can be plotted for each frame to construct the trajectory in conformational space projected onto a two dimensional cross-section. Other eigenvector directions can be specified, allowing the user to investigate how the trajectory projects into the space

defined by each eigenvector. The **DVP**s are given using un-normalized and normalized inner products, as well as weighted by the corresponding eigenvalue.

Post PCA Comparative Subspace Analysis:

JED performs a subspace analysis (**SSA**) on two different sets of eigenvectors of equal numbers generated from the **Q** or **R** variants of PCA. The results provide comparisons for all subspace dimensions up to the dimension chosen by the user when selecting the number of Cartesian or Distance modes to process in an iterative fashion. This allows one to quantitatively determine how different the PCA results are due only to the choice of PCA model, while also assessing the size of the essential subspace. Additional analysis can be done using the Subspace Analysis classes or the associated driver programs. To perform these kinds of tests, it is a best practice to first generate equidimensional sets of eigenvectors from each trajectory of interest, as well as from a pooled trajectory to use as a reference set. Subspace analysis is done by comparing the sets of eigenvectors, directly or iteratively, and determining the root mean square inner products (**RMSIP**s), Principal Angles (**PA**s), cumulative overlap (**CO**s), cosine products (**CP**s), vectorial angular sum (**VAS**), and the maximum angle between subspaces of the given vector space.

(In this tutorial, code, file paths, and text file content are shown in dark blue 9 point Consolas)

II. Understanding JED

JED Install Instructions:

Java is **platform independent** and JREs exist for all common architectures. The machine on which JED is to be run should have **JRE version 1.6 or higher** installed. The programs can be run from compiled source or from the provided executable jar files. While JED can be installed in any directory that is part of your Java classpath, the sources must be compiled on the local machine to insure runtime integrity. When compiling from source, be sure to compile the JAMA MATRIX package as JED uses that library. Alternatively, no source code or compilation is needed to run the executable jar files. These can be placed in any directory that is on the Java classpath. For most applications, a **64bit OS is required** to address the amount of memory needed for the analyses. It is critical that the environment variable Java **CLASSPATH** be correctly set to run Java programs at the command prompt. Alternatively, you can always add the **-cp** option to the **java** command, which allows you to specify the path that contains your Java classes.

Expected Memory Requirements:

On high performance computer clusters make sure the 64 bit JRE is installed. Memory use is demanding because JED holds the complete covariance matrix (among other data structures) and matrix diagonalization scales as $O(N^3)$. Typically 15 to 35 GB of RAM will be needed depending on the size of the protein. For very large proteins consisting of thousands of residues and/or many tens of thousands of frames, make available as much memory per node as possible (~ 1TB).

Two Kinds of JED Drivers:

There are two driver programs for JED: One (**JED_Driver**) runs a single job using parameters specified in the input file, and the other (**JED_Batch_Driver**) runs a batch of jobs sequentially. The first is suited for running a single job at the command line or when using submit scripts on computer cluster resources. This can be implemented using job arrays so that your jobs run in parallel rather than sequentially. The second is suited for running multiple jobs on a single computer so that a user can submit a batch of jobs, and come back a few hours later with many different jobs finished without having to launch each one separately.

Input File and Data for JED Driver:

JED requires an input file for job parameters. The format of this file will be described below. The run command takes only one argument, which is the name of the input file that includes the absolute path to the file. If no argument is specified, then JED assumes that the default input file name is used and the file is located in the same directory from which the JVM was called. The default input file names are:
JED_Driver.txt for JED_Driver.java (or .jar file)
JED_Batch_Driver.txt for JED_Batch_Driver.java (or .jar file)

Each job should be assigned to its own directory, which must contain either the PDB files to read or the coordinate file to process, along with the reference PDB file and residue lists for specifying either the Cartesian or Distance subsets.

JED Command Line format:

To run **JED_Driver** at the command prompt or within a PBS script, you can use one of the following commands:

```
java -d64 JED_Driver "/path/to/your/input/file.txt" (runs the compiled java program)
java -jar -d64 JED_Driver.jar "/path/to/your/input/file.txt" (runs the executable jar file)
```

To run **JED_Batch_Driver** at a command prompt or in a PBS script, you can use one of the following commands:

```
java -d64 JED_Batch_Driver "/path/to/your/input/file.txt")
java -jar -d64 JED_Batch_Driver.jar "/path/to/your/input/file.txt"
```

Organization of Output Files:

Output files from JED are written to subdirectories within the working directory, structured to organize the multitude of files produced in a meaningful manner. The top level of this directory tree is named "JED_RESULTS_**$description**", where **$description** is a user set parameter that succinctly describes the job. Limbs of the tree separate Cartesian PCA (**cPCA**), Distance PCA (**dPCA**), and Mode Visualization analyses (**VIZ**), when present. Each of these in turn contains limbs for **Q** (**COV**) and **R** (**CORR**) compartmentalization, which also contains a subdirectory for the subspace analysis (**SSA**). Most of the output file names include the **number of residues** in the selected subset for reference plus a description of the file contents.

Current Limitations:

Initial input of the protein trajectory must be done using PDB files that are expected to conform to the standard format, or a matrix of PDB coordinates containing the alpha carbon atomic positions only (see below for a description of this file). Only carbon-alpha atomic positions are used to create a **Q** or **R** matrix for essential dynamic analysis. Each PDB file must have the exact same number of residues. JED cannot process a PDB file with missing residues or have alternate conformations within a given frame based on fractional **occupancy** values. Only a single conformation per frame is allowed.

III. Step by Step Instructions to Run JED

IN ALL CASES:

A preliminary run with NO PCA must be performed to generate the JED formatted coordinate matrix file for all the alpha carbons in the PDB files. This makes subsequent subset analyses much faster to perform. It also serves to guarantee that the specified residues for subset selection are correctly chosen. After this initialization step, the PDB files can be deleted or archived, with the exception of the reference PDB file. Once the coordinate matrix is created, it should be used for all subsequent analyses using different residue subsets and different job parameters.

The name of the coordinate file matrix produced from the PDB files is: `"original_PDB_coordinates.txt"`
The matrix packing is as follows:
Rows are coordinate variables and columns are frames.
For N residues, there are 3N rows: N x-coordinates, N y-coordinates, and N-z coordinates, stacked in that order.

The file to use in all subsequent JED analyses is the original_PDB_coordinates matrix.

Note: The most critical step in preparing to run JED is in the creation of the input file. The input file must have the correct format (shown in examples below) and the entries must be accurate. If either of these conditions is violated, the program will crash, or worse, the results will be corrupt.

Common Causes for JED to Crash

If **any directory** cannot be found or if **any file** cannot be found, JED will crash.

If unexpected format is found in **any** of the input files, JED will crash.

IV. JED DRIVER:

A. The Preliminary Run

i. Run Command:

```
java -jar -d 64 JED_Driver.jar "/path/JED_Driver.txt"
```

The PDB files (including the PDB reference file) must be in the working directory.
JED input file may be in the working directory.

This pre-processing step will read all PDB files in the working directory, but will perform **no PCA.**
The purpose of **no PCA** this is to generate the matrix of coordinates for performing subset analyses efficiently.

ii. Standard NO-PCA Output Files:

These are written to the **root** of the JED Results directory tree:
`/working/directory/JED_Results_Description/`

JED LOG providing a summary of the job parameters and results:

`JED_Log.txt`

PDB READ LOG listing all the PDB files read, in the order they were read:

`PDB_READ_Log.txt`

coordinates matrix from all the alpha carbon coordinates in the PDB files:

`original_PDB_coordinates.txt`

transformed coordinates matrix, which aligns all the frames to the reference frame :

`ss_$num_res_transformed_PDB_coordinates.txt`

list of all residues found in the PDB files for subsequent editing and use:

`All_PDB_Residues_JED.txt`

`All_PDB_Residues_Multi_JED.txt (for Multi runs)`

original and transformed conformation RMSDs:

`ss_$num_res_original_conformation_rmsds.txt`

`ss_$num_res_conformation_rmsds.txt`

residue RMSDs (RMSF):

`ss_$num_res_residue_rmsd.txt`

edited PDB file containing all the residues with the RMSF replacing B-factors:

`ss_$num_res_edidited.PDB`

coordinates z-score matrix from all the alpha carbon coordinates in the PDB files:

`ss_$num_res_coordinate_Z_scores.txt`

percent of the frames to remove based on conformation RMSD (OPTIONAL)

z-cutoff for adjusting coordinate outliers to their mean values (OPTIONAL)

If **percent** or **z-cutoff** are set to a value other than **zero**, additional output files are generated with the results. However, the appropriate time to handle outliers is during runs that actually perform PCA and it is recommended that for pre-processing runs, these be set to zero.

iii. Single Chain PDB files: The Preliminary Run

```
-----------------------------------------------------------------------------------------------
0.00
0.00
1     0
0     0     0     0
/working/directory/job/
Description     reference_PDB_file.pdb
-----------------------------------------------------------------------------------------------
```

Notes: This is a whitespace separated file.

Line 1 Field 1: specifies the **percent** (double) of frames to remove from the data: 0 →NONE

Line 2 Field 1: specifies the **z-cutoff** (double) for adjusting outliers: 0 →NO ADJUSTMENT

Line 3 Field 1 specifies whether to **read PDB files** (0 or 1) →1 = yes
Field 2: specifies if the PDB files are **Multi** Chain (0 or 1) → 0 = no

Line 4 Field 1: specifies the **number of Cartesian** (integer) modes to process →0 = none
Field 2: specifies the **number of all-to-all Distance modes** (integer) to process →0 = none
Field 3: specifies the **number of residue-pair Distance modes** (integer) to process →0 = none
Field 4: specifies the **number of Cartesian modes to Visualize**(integer) →0 = none

Line 5 Field 1: specifies the **working directory** (String)

Line 6 Field 1: specifies the **description** (String) for the requested job
Field 2: specifies the **reference PDB** (String) for the requested job

---

Key Points:

The **read** flag must be set to 1 and the **Multi** flag must be set to 0 for Single Chain PDBs with no Chain IDs.

When the **read** flag is set to 1, all other PCA analyses are turned off.

All PDB files must have the same number of residues. The matrix of alpha carbon coordinates is determined from the first PDB file read. If other files in the working directory do not match exactly, then the array sizes will not match and the program will crash. *IF JED crashes during the reading of PDBs, this is probably the reason*.

In subsequent analyses, it is critical that no residues are requested that do not actually exist in the PDB files!
JED maps the specified residue list to an internal list that is aligned to the columns of the coordinates matrix.

The file to use in all subsequent JED analyses is the original_PDB_coordinates matrix.

This matrix contains all the residues in the PDB files and thus can be used for any subset of those residues. When a subset is chosen, a new correspondence set is generated and a new transformation is done to optimize the alignment of the structures. This removes overall translation and rotation.

iv. Multi-Chain PDB files: The Preliminary Run

Differences from Single-Chain analysis shown highlighted in amber

```
-----------------------------------------------------------------------------------------------
0.00
0.00
1     1     2     A     B     795     151     0     0
0     0     0     0
/working/directory/job/
Description     reference_PDB_file.pdb
-----------------------------------------------------------------------------------------------
```

Notes: This is a whitespace delineated file.

Line 1 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.00 →NONE

Line 2 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 0.00 →NO ADJUSTMENT

Line 3 Field 1 specifies whether to **read PDB files** (0 or 1) →1 = yes
Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 1 = yes
Field 3 specifies the number of chains → (2)
Field 4 specifies the first chain ID → (A)
Field 5 specifies the second chain ID → (B)
Field 6 specifies the number of residues in chain A→ (795)
Field 7 specifies the number of residues in chain B→ (151)
Field 8 specifies the offset of Chain A → (0)
Field 9 specifies the offset of Chain B → (0)

Line 4 Field 1 specifies the **number of Cartesian** (integer) modes to process →0 = none
Field 2: specifies the **number of all-to-all Distance modes** (integer) to process →0 = none
Field 3: specifies the **number of residue-pair Distance modes** (integer) to process →0 = none
Field 4: specifies the **number of Cartesian modes to Visualize**(integer) →0 = none

Line 5 Field 1 specifies the **working directory** (String)

Line 6 Field 1 specifies the job **description** (String) for job one
Field 2 specifies the **reference PDB** (String) for job one

Key Points:

The **read** flag must be set to 1 and the **Multi** flag must be set to 1 for Multi Chain PDBs.
When the **read** flag is set to 1, all other PCA analyses are turned off.

All PDB files must have the same number of residues. The matrix of alpha carbon coordinates is determined from the first PDB file read. If other files in the working directory do not match exactly, then the array sizes will not match and the program will crash. IF JED crashes during the reading of PDBs, this is probably the reason.

Multi Chain PDBs must have unique chain identifiers for every chain. A missing chain identifier will cause JED to crash.

In subsequent analyses, it is critical that no residues are requested that do not actually exist in the PDB files!

JED maps the specified residue list to an internal list that is aligned to the columns of the coordinates matrix.

The file to use in all subsequent JED analyses is the original_PDB_coordinates matrix.

This matrix contains all the residues in the PDB files and thus can be used for any subset of those residues. When a subset is chosen, a new correspondence set is generated and a new transformation is done to optimize the alignment of the structures. This removes overall translation and rotation.

B. Debugging Crashes Part I:

Things that will generally make your life miserable…

i. Simple mistakes:

Does the path to the input file exist?
Does the input file exist in the proper location?
Does the input-file start on the first line?

Is the **number format** correct? (20.0 will NOT parse as an integer)
Are the **Read** and **Multi** flags set correctly?
Did you forget a parameter declaration?

Does the working directory string end in "**/**" or "****"?
Does the working directory exist?
Does the working directory contain PDB files of different sizes?
Does the working directory contain the reference PDB file?

ii. More subtle mistakes:

The directory contains PDB files with 2 chains, but no chain IDs.

If the PDB file names are sorted in a different order than how they were generated, then the conformation RMSD results will not reflect what actually occurred in the simulation. Naming the PDB files appropriately by padding the numbers with leading zeros will ensure proper sorting to prevent this problem caused by the operating system.

If the conformation RMSD is very different from what you expect, then you may be using PDB files that contain occupancy information. JED does not use that information. Your results will not be accurate.

If you have pooled data, make sure the combined matrix is constructed the way you think it is and the reference column is the frame you think it is.

C. Performing Cartesian PCA

i. run command:

```
java -jar -d64 JED_Driver.jar "/path/JED_Driver.txt"
```

The working directory must contain: The coordinates matrix, the PDB reference file, and the residue list.
JED input file may be in the working directory.

The purpose of this type of run is to perform Essential Dynamics using cPCA based on **Q** and **R**. The user specifies the subset of interest for the analysis, which may be the entire protein or a sub-region, which can be non-contiguous, by providing a residue list file. This task is simplified since JED has already created a list of all the residues in the protein. The user can simply edit this file. The cPCA results are written to the sub-directory "cPCA" and the Visualizations of the top modes (when selected) are written to the subdirectory "VIZ". The directory cPCA has sub directories for the **Q** and **R** analysis, as does the VIZ directory.

ii. Standard Output Files:

These are written to the **root** of the JED Results directory tree:
`/working/directory/JED_Results_Description/`

JED LOG providing a summary of the job parameters and results:

`JED_Log.txt`

subset transformed coordinates matrix:

`ss_$num_res_transformed_PDB_coordinates.txt`

original and transformed conformation RMSDs:

`ss_$num_res_original_conformation_rmsds.txt`

`ss_$num_res_conformation_rmsds.txt`

residue RMSDs (RMSF):

`ss_$num_res_residue_rmsd.txt`

subset edited PDB file with the RMSF replacing B-factors:

`ss_$num_res_edidited.PDB`

subset coordinates Z-Score matrix:

`ss_$num_res_coordinate_Z_scores.txt`

list of frames removed, based on the **percent** parameter:

`ss_$num_res_ Removed_Conformation_Outliers.txt`

trimmed transformed coordinate matrix:

`ss_$num_res_ trimmed_$percent_percent_PDB_coordinates_COLS.txt`

list of coordinate variables adjusted, based on the **z-cutoff** parameter:

`ss_$num_res_ adjustments_per_variable.txt`

adjusted transformed coordinate matrix:

`ss_$Z_threshold_$z-cutoff_adjusted_PDB_coordinates_ROWS.txt`

iii. Standard cPCA Output Files:

These are written to the **/cPCA subdirectory** of the JED Results directory tree:
`/working/directory/JED_Results_Description/cPCA/`

centroids (means) of the variables:

`ss_$num_res_centroids_of_variables.txt`

standard deviations of the variables:

`ss_$num_res_std_devs_of_centered_variables.txt`

displacement vectors:

`ss_$num_res_delta_vectors.txt`

The Q output files are written to the /COV subdirectory of /cPCA

`/working/directory/JED_Results_Description/cPCA/COV/`

The R output files are written to the /CORR subdirectory of /cPCA

`/working/directory/JED_Results_Description/cPCA/CORR/`

covariance matrix:

`ss_$num_res_COV_matrix.txt`

eigenvalues:

`ss_$num_res_eigenvalues_COV.txt`

top eigenvalues:

`ss_$num_res_top_$num_of_cart_modes_eigenvalues_COV.txt`

top eigenvectors:

`ss_$num_res_top_$num_of_cart_modes_eigenvectors_COV.txt`

top pca modes and top weighted pca modes:

`ss_$num_res_top_$num_of_cart_modes_pca_modes_COV.txt`
`ss_$num_res_top_$num_of_cart_modes_weighted_pca_modes_COV.txt`

top square pca modes and top weighted square pca modes:

`ss_$num_res_top_$num_of_cart_modes_square_pca_modes_COV.txt`
`ss_$num_res_top_$num_of_cart_modes_weighted_square_pca_modes_COV.txt`

top PCs and top weighted PCs:

`ss_$num_res_top_$num_of_cart_modes_PCs_COV.txt`
`ss_$num_res_top_$num_of_cart_modes_weighted_PCs_COV.txt`

top PCs and top weighted PCs:

`ss_$num_res_top_$num_of_cart_modes_normed_PCs_COV.txt`
`ss_$num_res_top_$num_of_cart_modes_weighted_normed_PCs_COV.txt`

iv. SSA Output Files:

These are written to the **/SSA subdirectory** of /cPCA:
`/working/directory/JED_Results_Description/cPCA/SSA/`

The Fast SSA Iterated Log:

`JED_FSSA_Iterated_log.txt`

The SSA Log:

`JED_SSA_dim_$top_num_cart_modes_log.txt`

The Random SSA Log

`JED_Random_SSA_log.txt`

There are many more files in the /SSA directory that are flat files of the results reported in the log files:

RMSIPs

PAs

COs

Cosine Products

Vectorial Sum of Angles

v. Single-Chain PDB files: Performing Cartesian PCA

```
-----------------------------------------------------------------------------------------
0.01
3.00
0     0
20    0     0     2     1.0
/working/directory/
Description           reference_PDB_file.pdb       0
residues.txt
original_PDB_Coordinates.txt        0
-----------------------------------------------------------------------------------------
```

Notes:

Line 1 Field specifies the **percent** (double) of frames to remove from the data: 0.01 → 1%

Line 2 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|

Line 3 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no

Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 0 = no

Line 4 Field 1 specifies the **number of Cartesian** (integer) modes to process → 20 = top 20 modes
Field 2: specifies the **number of all-to-all Distance modes** (integer) to process →0 = none
Field 3: specifies the **number of residue-pair Distance modes** (integer) to process →0 = none
Field 4: specifies the **number of Cartesian modes to Visualize**(integer) →0 = none
Field 5 specifies the **scale factor** for the visualizations (double) → 1.0 = default value

Line 5 Field 1 specifies the **working directory** (String)

Line 6 Field 1 specifies the job **description** (String)

Field 2 specifies the **reference PDB** (String)

Field 3 specifies the **offset** for the residue numbering (integer)

Line 7 Field 1 specifies the **Cartesian residue list** (String)

Line 8 Field1 specifies the **coordinate matrix** (String)

Field 2 specifies the **reference column** (integer)

Key Points:

Both the **Read** PDB file flag and the **Multi** flag must be set to 0 to perform a JED analysis on Single chain PDBs.

The number of cPCA modes must not be 0.

There must be a Cartesian residue list specified.

The number of dPCA modes and dpPCA modes must be zero.

There must not be a Distance residue list specified.

If the subset you have chosen contains N residues, then you must NOT request more than 3N modes.

If you request N cPCA modes then you can only visualize up to N top modes.

If number of cPCA modes > 0, then there must be a Cartesian residue list file!

vi. Multi-Chain PDB files: Performing Cartesian PCA

Differences from Single-Chain analysis shown highlighted in amber

```
-------------------------------------------------------------------------------------------
0.01
3.00
0     1     2     A     B     795     151     0     0
20    0     0     2     1.0
/working/directory/
Description           reference_PDB_file.pdb       (no offset needed here)
residues.txt
original_PDB_Coordinates.txt        0
-------------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.01 → 1%

Line 2 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|

Line 3 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no
Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 1 = yes
Field 3 specifies the number of chains → (2)
Field 4 specifies the first chain ID → (A)
Field 5 specifies the second chain ID → (B)
Field 6 specifies the number of residues in chain A→ (795)
Field 7 specifies the number of residues in chain B→ (151)
Field 8 specifies the offset of Chain A → (0)
Field 9 specifies the offset of Chain B → (0)

Line 4 Field 1 specifies the **number of Cartesian** (integer) modes to process → 20 = top twenty modes
Field 2: specifies the **number of all-to-all Distance modes** (integer) to process →0 = none
Field 3: specifies the **number of residue-pair Distance modes** (integer) to process →0 = none
Field 4: specifies the **number of Cartesian modes to Visualize**(integer) →0 = none
Field 5 specifies the **scale factor** for the visualizations (double) → 1.0 = default value

Line 5 Field 1 specifies the **working directory** (String)

Line 6 Field 1 specifies the job **description** (String)
Field 2 specifies the **reference PDB** (String)

Line 7 Field 1 specifies the **residue list** (String)

Line 8 Field 1 specifies the **coordinate matrix** (String)
Field 2 specifies the **reference column** (integer)

Key Points:

The Read PDB file flag must be set to 0.

The Multi flag must be set to 1 for JED to perform the analysis on Multi Chain PDBs.

Chain Offsets are specified on line 3, not after the reference pdb file on line 6.

The number of cPCA modes must be > 0.

There must be a Cartesian residue list specified.

The number of dPCA modes must be 0.

There must not be a distance residue list specified.

If the subset you have chosen contains N residues, then you must not request more than 3N modes.

If you request ***N*** cPCA modes then you can only visualize up to ***N*** top modes.

****

The format of the residue list file is TWO columns for Multi Chain Analysis: Chain ID, residue number.

The format of the residue list file is ONE column for Single Chain Analysis: residue number.

****

If number of cPCA modes > 0, then there must be a Cartesian residue list file!

D. Performing Distance PCA: dPCA

i. run command:

```
java -jar -d64 JED_Driver.jar "/path/JED_Driver.txt"
```

The working directory must contain: The coordinates matrix, the PDB reference file, and the residue list.
JED input file may be in the working directory.

The purpose of this type of run is to perform Essential Dynamics using dPCA based on **Q** and **R**. The user specifies the subset of interest for the analysis, which is typically less than 10 residues, by providing a residue list file. This task is simplified since JED has already created a list of all the residues in the protein. The user must simply edit this file. The dPCA results are written to the sub-directory "dPCA". Note that for dPCA no transform is needed since internal distances are used for coordinates and no visualization can be done in JED for the distance modes. The directory dPCA has sub directories for the **Q** and **R** analysis as well as for the subspace analysis (SSA). Choosing more than ten residues for the dPCA analysis makes the interpretation of the results challenging as each component of the distance eigenvectors corresponds to an inter-residue distance pair. Often subsets with three or four residues can be used to investigate experimental findings in critical areas like binding pockets or clefts.

ii. Standard Output Files:

These are written to the **root** of the JED Results directory tree:
`/working/directory/JED_Results_Description/`

JED LOG providing a summary of the job parameters and results:

`JED_Log.txt`

subset transformed coordinates matrix:

`ss_$num_res_transformed_PDB_coordinates.txt`

original and transformed conformation RMSDs:

`ss_$num_res_original_conformation_rmsds.txt`

`ss_$num_res_conformation_rmsds.txt`

residue RMSDs (RMSF):

`ss_$num_res_residue_rmsd.txt`

subset edited PDB file with the RMSF replacing B-factors:

`ss_$num_res_edidited.PDB`

subset coordinates Z-Score matrix:

`ss_$num_res_coordinate_Z_scores.txt`

list of frames removed, based on the **percent** parameter:

`ss_$num_res_ Removed_Conformation_Outliers.txt`

trimmed transformed coordinate matrix:

`ss_$num_res_ trimmed_$percent_percent_PDB_coordinates_COLS.txt`

list of coordinate variables adjusted, based on the **z-cutoff** parameter:

`ss_$num_res_ adjustments_per_variable.txt`

adjusted transformed coordinate matrix:

`ss_$Z_threshold_$z-cutoff_adjusted_PDB_coordinates_ROWS.txt`

iii. Standard dPCA Output Files:

These are written to the **/dPCA subdirectory** of the JED Results directory tree:

`/working/directory/JED_Results_Description/dPCA/`

subset distance residue stats from all the alpha carbon coordinates in the **subset**:

`ss_$num_res_distance_residue_stats.txt`

subset distance Z-Score matrix from all the alpha carbon coordinates in the **subset**:

`ss_$num_res_distance_Z_scores.txt`

subset all-to-all distances matrix from all the alpha carbon coordinates in the **subset**:

`ss_$num_res_all_to_all_distances.txt`

subset distance variables adjusted:

`ss_$num_res_ outliers_per_variable.txt`

subset centroids (means) of the variables:

`ss_$num_res_centroids_of_variables.txt`

subset standard deviations of the variables:

`ss_$num_res_std_devs_of_centered_variables.txt`

subset displacement vectors:

`ss_$num_res_delta_vectors.txt`

The Q output files are written to the /COV subdirectory of /dPCA (Shown below)

`/working/directory/JED_Results_Description/cPCA/COV/`

The R output files are written to the /CORR subdirectory of /dPCA

covariance matrix:

`ss_$num_res_distance_COV_matrix.txt`

eigenvalues:

`ss_$num_res_distance_eigenvalues_COV.txt`

top eigenvalues:

`ss_$num_res_top_$num_of_dist_modes_distance_eigenvalues_COV.txt`

top eigenvectors:

`ss_$num_res_top_$num_of_dist_modes_distance_eigenvectors_COV.txt`

top pca modes and top weighted pca modes:

`ss_$num_res_top_$num_of_dist_modes_distance_pca_modes_COV.txt`
`ss_$num_res_top_$num_of_dist_modes_weighted_distance_pca_modes_COV.txt`

top square pca modes and top weighted square pca modes:

`ss_$num_res_top_$num_of_dist_modes_square_distance_pca_modes_COV.txt`
`ss_$num_res_top_$num_of_dist_modes_weighted_square_distance_pca_modes_COV.txt`

top PCs and top weighted PCs:

`ss_$num_res_top_$num_of_dist_modes_PCs_COV.txt`
`ss_$num_res_top_$num_of_dist_modes_weighted_PCs_COV.txt`

top normed PCs and top weighted normed PCs:

`ss_$num_res_top_$num_of_dist_modes_normed_PCs_COV.txt`
`ss_$num_res_top_$num_of_dist_modes_weighted_normed_PCs_COV.txt`

iv. Standard SSA Output Files:

These are written to the **/SSA subdirectory** of /cPCA:
`/working/directory/JED_Results_Description/cPCA/SSA/`

The Fast SSA Iterated Log:

`JED_FSSA_Iterated_log.txt`

The SSA Log:

`JED_SSA_dim_$top_num_cart_modes_log.txt`

The Random SSA Log

`JED_Random_SSA_log.txt`

There are many more files in the /SSA directory that are flat files of the results reported in the log files:

RMSIPs

PAs

COs

Cosine Products

Vectorial Sum of Angles

v. Single-Chain PDB files: Performing dPCA

```
-----------------------------------------------------------------------------------------
0.00
3.00
0     0
0     3     0     0
/working/directory/
Description           reference_PDB_file.pdb       0
residues_dist.txt
original_PDB_Coordinates.txt        0
-----------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.00 = none
Line 2 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|
Line 3 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no
Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 0 = no
Line 4 Field 1 specifies the **number of Cartesian** (integer) modes to process → 0 = none
Field 2: specifies the **number of all-to-all Distance modes** (integer) to process →3 = top 3 modes
Field 3: specifies the **number of residue-pair Distance modes** (integer) to process →0 = none
Field 4: specifies the **number of Cartesian modes to Visualize**(integer) →0 = none
Line 5 Field 1 specifies the **working directory** (String)
Line 6 Field 1 specifies the job **description** (String)
Field 2 specifies the **reference PDB** (String)
Field 3 specifies the **offset** (integer)
Line 7 Field 1specifies the **residue list** (String)
Line 8 Field 1 specifies the **coordinate matrix** (String)
Field 2 specifies the **reference column** (integer)

Key Points:
Both the **Read** PDB file flag and the **Multi** flag MUST be set to **ZERO**.
This tells JED to perform the analysis on Single Chain PDBs.

The number of cPCA modes MUST be zero.
There must be no Cartesian residue list specified.
The number of dPCA modes MUST NOT be zero.
The number of dpPCA modes MUST be zero.

There must be a Distance residue list specified.
If the subset you have chosen contains ***N*** residues, then you must NOT request more than ***N(N-1)/2*** modes.

If number of dPCA modes > 0, then there must be a Distance residue list file!

vi. Multi-Chain PDB files: Performing Distance PCA

Differences from Single-Chain analysis shown highlighted in amber

```
-------------------------------------------------------------------------------------------
0.00
3.00
0     1     2     A     B     795     151     0     0
0     3     0
/working/directory/
Description           reference_PDB_file.pdb       (no offset here)
residues_dist.txt
original_PDB_Coordinates.txt        0
-------------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.00 → none

Line 2 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|

Line 3 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no

Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 1 = yes

Field 3 specifies the number of chains → (2);

Field 4 specifies the first chain ID → (A);

Field 5 specifies the second chain ID → (B);

Field 6 specifies the number of residues in chain A→ (795);

Field 7 specifies the number of residues in chain B→ (151);

Field 8 specifies the offset of Chain A → (0);

Field 9 specifies the offset of Chain B → (0);

Line 4 Field 1 specifies the **number of Cartesian** (integer) modes to process → 0

Field 2: specifies the **number of all-to-all Distance modes** (integer) to process →3 = top 3 modes

Field 3: specifies the **number of residue-pair Distance modes** (integer) to process →0 = none

Field 4: specifies the **number of Cartesian modes to Visualize**(integer) →0 = none

Line 5 Field 1 specifies the **working directory** (String)

Line 6 Field 1 specifies the job **description** (String)

Field 2 specifies the **reference PDB** (String)

Line 7 Field 1 specifies the **residue list** (String)

Line 8 Field 1 specifies the **coordinate matrix** (String)

Field 2 specifies the **reference column** (integer)

Key Points:

The **Read** PDB file flag MUST be set to **ZERO**.

The **Multi** flag MUST be set to **ONE**;

This tells JED to perform the analysis on MULTI Chain PDBs.

The number of cPCA modes MUST be zero.

There must be no Cartesian residue list specified.

The number of dPCA modes MUST NOT be zero.

There must be a Distance residue list specified.

The number of dpPCA modes MUST be zero.

If the subset you have chosen contains ***N*** residues, then you must NOT request more than ***N(N-1)/2*** modes.

Remember that the format of the residue list file is TWO columns for Multi Chain Analysis: **Chain ID, residue number** while for Single Chain Analysis, the file has ONE column: **residue number**.

If number of dPCA modes > 0, then there must be a Distance residue list file!

D. Performing Distance PCA: dpPCA

i. run command:

```
java -jar -d64 JED_Driver.jar "/path/JED_Driver.txt"
```

The working directory must contain: The coordinates matrix, the PDB reference file, and the residue list.
JED input file may be in the working directory.

The purpose of this type of run is to perform Essential Dynamics using dpPCA based on **Q** and **R**. The user specifies the set of residue pairs of interest for the analysis, by providing a residue pair list file. This file is a two column file for Single Chain PDBs in which the pairs of interest are listed. However, for Multi Chain PDBs, the file is four columns, the first two for the chain ID and residue number of residue one, and the third and fourth columns for the chain ID and residue number of the second residue. The dpPCA results are written to the sub-directory "dpPCA". Note that for dpPCA no transform is needed since internal distances are used for coordinates and no visualization can be done in JED for the distance modes. The residue pair method is much easier to interpret as the number of components in the eigenvectors is equal to the number of residue pairs specified. The directory dpPCA has sub directories for the **Q** and **R** analysis as well as for the subspace analysis (SSA). Very large subsets can be used to investigate experimental findings in critical areas like binding pockets or clefts. Unfortunately, like the dPCA results, the dpPCA results cannot be visualized as no simple mapping can be made to the residues.

ii. Standard Output Files: Same as for dPCA

v. Single-Chain PDB files: Performing dpPCA

```
-------------------------------------------------------------------------------------------
0.00
3.00
0     0
0     0     10     0
/working/directory/
Description           reference_PDB_file.pdb       0
residues_pairs.txt
original_PDB_Coordinates.txt        0
-------------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.00 = none
Line 2 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|
Line 3 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no
Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 0 = no
Line 4 Field 1 specifies the **number of Cartesian** (integer) modes to process → 0 = none
Field 2: specifies the **number of all-to-all Distance modes** (integer) to process → 0 = none
Field 3: specifies the **number of residue-pair Distance modes** (integer) to process → 10 = top 10 modes
Field 4: specifies the **number of Cartesian modes to Visualize**(integer) →0 = none
Line 5 Field 1 specifies the **working directory** (String)
Line 6 Field 1 specifies the job **description** (String)
Field 2 specifies the **reference PDB** (String)
Field 3 specifies the **offset** (integer)
Line 7 Field 1specifies the **residue list** (String)
Line 8 Field 1 specifies the **coordinate matrix** (String)
Field 2 specifies the **reference column** (integer)

Key Points:
Both the **Read** PDB file flag and the **Multi** flag MUST be set to **ZERO**.
This tells JED to perform the analysis on Single Chain PDBs.

The number of cPCA modes MUST be zero.
There must be no Cartesian residue list specified.
The number of dPCA modes MUST be zero.
The number of dpPCA modes MUST NOT be zero.

There must be a Distance Residue Pair List specified.
The file format is a two column list of integers: residue number-----residue number

If the you have chosen ***N*** residue pairs, then you must NOT request more than ***N*** modes.

If number of dpPCA modes > 0, then there must be a Distance Residue Pair List file!

vi. Multi-Chain PDB files: Performing dpPCA

Differences from Single-Chain analysis shown highlighted in amber

```
-------------------------------------------------------------------------------------------
0.00
3.00
0     1     2     A     B     795     151     0     0
0     0     10     0
/working/directory/
Description           reference_PDB_file.pdb       (no offset needed here)
residues_dist.txt
original_PDB_Coordinates.txt        0
-------------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.00 → none

Line 2 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|

Line 3 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no

Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 1 = yes

Field 3 specifies the number of chains → (2);

Field 4 specifies the first chain ID → (A);

Field 5 specifies the second chain ID → (B);

Field 6 specifies the number of residues in chain A→ (795);

Field 7 specifies the number of residues in chain B→ (151);

Field 8 specifies the offset of Chain A → (0);

Field 9 specifies the offset of Chain B → (0);

Line 4 Field 1 specifies the **number of Cartesian** (integer) modes to process → 0

Field 2: specifies the **number of all-to-all Distance modes** (integer) to process → 0 = none

Field 3: specifies the **number of residue-pair Distance modes** (integer) to process → 10 = top 10 modes

Field 4: specifies the **number of Cartesian modes to Visualize**(integer) →0 = none

Line 5 Field 1 specifies the **working directory** (String)

Line 6 Field 1 specifies the job **description** (String)

Field 2 specifies the **reference PDB** (String)

Line 7 Field 1 specifies the **residue list** (String)

Line 8 Field 1 specifies the **coordinate matrix** (String)

Field 2 specifies the **reference column** (integer)

Key Points:

The **Read** PDB file flag MUST be set to **ZERO**.

The **Multi** flag MUST be set to **ONE**;

This tells JED to perform the analysis on MULTI Chain PDBs.

The number of cPCA modes MUST be zero.

There must be no Cartesian residue list specified.

The number of dPCA modes MUST be zero.

There must be a Distance Residue Pair List specified.

The number of dpPCA modes MUST NOT be zero.

If the subset you have chosen contains ***N*** Residue Pairs, then you must NOT request more than ***N*** modes.

Remember that the format of the Residue Pair List file is FOUR columns for Multi Chain Analysis:

Chain ID, residue number-----Chain ID, residue number

If number of dpPCA modes > 0, then there must be a Distance Residue Pair List file!

E. Debugging Crashes Part II:

Things that will generally make your life miserable…

i. Dumb mistakes:
Did you set the Read and Multi flags correctly?
Did you request cPCA but not specify a Cartesian residue list?
Did you request dPCA but not specify a Distance residue list?
Did you request dpPCA but not specify a Distance Residue Pair List?
Did you set the number of modes apropriately?
Are you requesting to read PDBs when you should be specifying a coordinate matrix?

ii. More subtle mistakes:
If the PDB file residue numbering starts at 5, then the chain offset is not 0, it is 4.

Did you request more modes than actually exist? For example, if your Cartesian subset contains 12 residues, but you asked for 50 modes, then you are going to crash JED because there are only 36 Cartesian modes in total.

Also, if your Distance subset contains 3 residues and you request 5 modes, then you are going to crash JED because there are only 3 distance modes in total.

If your trajectory has not equilibrated, then you must address the problem of outliers. If you do not, then the Q and R matrices will be highly ill-conditioned and may cause the eigenvalue decomposition to fail. You can check the variables in statistics packages that compute the Kaiser-Myer-Olkin (KMO) statistic as well as the Measure of Sampling Adequacy (MSA) for each coordinate variable to critically assess your data. If it is not well suited for PCA, you can condition the variables by setting the z-cutoff in JED between 2.0 and 3.0 when running your jobs. This type of conditioning is by far not very sophisticated, but it has the effect of lowering the condition numbers of Q and R as well as un-dilating the high and low regions of the eigenspectrum. In particular, it does not alter the ordinality of the eigenspectrum but does correct the distortion that arises from under sampling when trying to estimate the population covariance matrix from the sample covariance matrix.

F. Performing Cartesian and Distance PCA with Mode Visualization

The working directory must contain: The coordinates matrix, the PDB reference file, and both residue lists.
It may also contain the JED input file

JED is capable of doing both cPCA and dPCA, using both Q and R, and generating cPCA mode visualizations simultaneously. All outputs are delivered as discussed for the individual components.

The outputs to this type of job include the outputs for both the cPCA and dPCA analyses, as well as all the structures for the top modes chosen for visualization. JED will permute the reference structure for a given subset along the top eigenvectors selected for visualization and output structures (PDBs) that capture one cycle of this motion. The amplitude of the motion is determined by the value of the **$scale_factor**, whose default value is 1.0, and can be adjusted as necessary. Setting the value too high will cause Visualization software like Pymol to break the ribbon diagrams of the structures. Ultimately, this is controlled by the magnitude of the eigenvector components for any given residue. Setting the scale factor between 1 and 3 is usually safe, but for proteins with highly mobile regions like loops, you may need to choose a smaller scale factor. This is done for both the top Q and R modes. Additionally, Pymol scripts are generated to animate those structures into a movie for better analysis of the physical meanings of the top modes.

These files will be located in the /VIZ subdirectory of the root of the JED results tree:
`/working/directory/JED_Results_Description/VIZ/`

The Q results will be in the subdirectory /COV and the R results will be in the subdirectory /CORR.

One huge advantage of JED is that it is highly configurable and can perform many types of Essential Dynamics analysis concurrently. Combined with the cluster resources or just using the batch feature allows a user to process a great deal of data efficiently.

i. Single-Chain PDB files: Performing Cartesian and Distance PCA with Mode Visualization

```
-----------------------------------------------------------------------------------------
0.00
3.00
0     0
20    3     0     2     1.0
/working/directory/
Description           reference_PDB_file.pdb       0
residues_cartesian.txt
residues_dist.txt
original_PDB_Coordinates.txt        0
-----------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.00 = none

Line 2 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|

Line 3 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no

Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 0 = no

Line 4 Field 1 specifies the **number of Cartesian** (integer) modes to process → 20 = top twenty modes

Field 2: specifies the **number of all-to-all Distance modes** (integer) to process → 3 = top 3 modes

Field 3: specifies the **number of residue-pair Distance modes** (integer) to process → 0 = none

Field 4: specifies the **number of Cartesian modes to Visualize**(integer) → 2 = top 2 modes

Field 5: specifies the **mode amplitude** (double) → 1.0 = default

Line 5 Field 1 specifies the **working directory** (String)

Line 6 Field 1 specifies the job **description** (String)

Field 2 specifies the **reference PDB** (String)

Field 3 specifies the **offset** (integer)

Line 7 Field 1 specifies the **Cartesian residue list** (String)

Line 8 Field 1 specifies the **Distance residue list** (String)

Line 9 Field 1 specifies the **coordinate matrix** (String)

Field 2 specifies the **reference column** (integer)

Key Points:

Both the Read PDB file flag and the Multi flag MUST be set to ZERO.

The number of cPCA modes MUST be > zero.

The number of dPCA modes MUST be > zero.

There must be a Cartesian residue list specified.

There must be a Distance residue list specified.

Residue Lists MUST be in the CORRECT ORDER: Cartesian first, then Distance

For Cartesian subsets containing ***N*** residues, you must NOT request more than ***3N*** modes.

For Distance subsets containing ***N*** residues, you must NOT request more than ***N(N-1)/2*** modes.

ii. Multi-Chain PDB files: Performing Cartesian and Distance PCA with Mode Visualization

Differences from Single-Chain analysis shown highlighted in amber

```
-------------------------------------------------------------------------------------------
0.00
3.00
0     1     2     A     B     795     151     0     0
20    3     0     2     1.0
/working/directory/
Description           reference_PDB_file.pdb       (no offset here)
residues_cartesian.txt
residues_dist.txt
original_PDB_Coordinates.txt        0
-------------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.00 → None

Line 2 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|

Line 3 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no

Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 1 = yes

Field 3 specifies the number of chains → (2);

Field 4 specifies the first chain ID → (A);

Field 5 specifies the second chain ID → (B);

Field 6 specifies the number of residues in chain A→ (795);

Field 7 specifies the number of residues in chain B→ (151);

Field 8 specifies the offset of Chain A → (0);

Field 9 specifies the offset of Chain B → (0);

Line 4 Field 1 specifies the **number of Cartesian** (integer) modes to process → 20 = top twenty modes

Field 2: specifies the **number of all-to-all Distance modes** (integer) to process → 3 = top 3 modes

Field 3: specifies the **number of residue-pair Distance modes** (integer) to process → 0 = none

Field 4: specifies the **number of Cartesian modes to Visualize**(integer) → 2 = top 2 modes

Field 5: specifies the **mode amplitude** (double) → 1.0 = default

Line 5 Field 1 specifies the **working directory** (String)

Line 6 Field 1 specifies the job **description** (String)

Field 2 specifies the **reference PDB** (String)

Line 7 Field 1 specifies the **Cartesian residue list** (String)

Line 8 Field 1 specifies the **Distance residue list** (String)

Line 9 Field 1 specifies the **coordinate matrix** (String)

Field 2 specifies the **reference column** (integer)

Key Points:

The read PDB file flag MUST be set to ZERO.

The Multi flag must be set to ONE; This tells JED to perform the analysis on MULTI Chain PDBs.

The number of cPCA modes MUST be > zero.

The number of dPCA modes MUST be > zero.

There must be a Cartesian residue list specified.

There must be a Distance residue list specified.

Residue Lists MUST be in the CORRECT ORDER: Cartesian first, then Distance

For cPCA subsets containing ***N*** residues, you must NOT request more than ***3N*** modes.

For dPCA subsets containing ***N*** residues, you must NOT request more than ***N(N-1)/2*** modes.

Remember that the format of the residue list file is TWO columns for Multi Chain Analysis: **Chain ID, residue number** while for Single Chain Analysis, the file has ONE column: **residue number**.

If number of cPCA modes > 0, then there must be a Cartesian residue list file!

If number of dPCA modes > 0, then there must be a Distance residue list file!

F. Performing Cartesian and Distance PCA with Mode Visualization

The working directory must contain: The coordinates matrix, the PDB reference file, and three residue lists.
It may also contain the JED input file

JED is capable of doing both cPCA, dPCA, and dpPCA, using both Q and R, and generating cPCA mode visualizations simultaneously. All outputs are delivered as discussed for the individual components.

Be sure that there is a corresponding input for each type of analysis that you turn on.

i. Single-Chain PDB files: Performing cPCA, dPCA, and dpPCA with Mode Visualization

```
-----------------------------------------------------------------------------------------
0.00
3.00
0     0
20    5     10     2     1.0
/working/directory/
Description           reference_PDB_file.pdb       0
residues_cartesian.txt
residues_dist.txt
residue_pairs.txt
original_PDB_Coordinates.txt        0
-----------------------------------------------------------------------------------------
```

Notes:
There are 20 cPCA modes
There are 5 dPCA modes
There are 10 dpPCA modes
2 cPCA modes will be visualized
There are three residue lists, in order, cPCA, dPCA, dpPCA.

If number of cPCA modes > 0, then there must be a Cartesian residue list file!
If number of dPCA modes > 0, then there must be a Distance residue list file!
If number of dpPCA modes > 0, then there must be a Distance Residue Pair List file!

ii. Multi-Chain PDB files: Performing cPCA, dPCA, and dpPCA with Mode Visualization

Differences from Single-Chain analysis shown highlighted in amber

```
-----------------------------------------------------------------------------------------
0.00
3.00
0     1     2     A     B     795     151     0     0
20    5     10     2     1.0
/working/directory/
Description           reference_PDB_file.pdb       (no offset here)
residues_cartesian.txt
residues_dist.txt
residue_pairs.txt
original_PDB_Coordinates.txt        0
-----------------------------------------------------------------------------------------
```

Notes:
There are 20 cPCA modes
There are 5 dPCA modes
There are 10 dpPCA modes
2 cPCA modes will be visualized
There are three residue lists, in order, cPCA, dPCA, dpPCA.

If number of cPCA modes > 0, then there must be a Cartesian residue list file!
If number of dPCA modes > 0, then there must be a Distance residue list file!
If number of dpPCA modes > 0, then there must be a Distance Residue Pair List file!

V. USING JED BATCH DRIVER

The batch version is identical to the non-batch version with the exception of the format of the input file.

i. Single Chain PDB files: The Preliminary Run

```
-------------------------------------------------------------------------------------------------------
$num_of_jobs
0.00
0.00
1     0
0     0     0     0
**************************************************************************
/working/directory/job1/
Description1                reference_PDB_file1.pdb
**************************************************************************
/working/directory/job2/
Description2                reference_PDB_file2.pdb
**************************************************************************
-------------------------------------------------------------------------------------------------------
```

Notes: This is a whitespace delineated file.

Line 1 Field 1 specifies the **$num_of_jobs** (integer) for the batch.
Line 2 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.00 →NONE
Line 3 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 0.00 →NO ADJUSTMENT
Line 4 Field 1 specifies whether to **read PDB files** (0 or 1) →1 = yes
Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 0 = no
Line 5 Field 1 specifies the **number of Cartesian** (integer) modes to process →0 = none
Field 2: specifies the **number of all-to-all Distance modes** (integer) to process → 0 = none
Field 3: specifies the **number of residue-pair Distance modes** (integer) to process → 0 = none
Field 4: specifies the **number of Cartesian modes to Visualize**(integer) → 0 = none
Line 6 is a **separator line** between the batch parameters and the job parameters **********
Line 7 Field 1 specifies the **working directory** (String) for job one
Line 8 Field 1 specifies the job **description** (String) for job one
Field 2 specifies the **reference PDB** (String) for job one
Line 9 is a **separator line** between sets of job parameters **********
Line 10 Field 1 specifies the **working directory** (String) for job one
Line 11 Field 1 specifies the job **description** (String) for job two
Field 2 specifies the **reference PDB** (String) for job two
Line 12 is a **separator line** between sets of job parameters **********

Key Points:

For single jobs, it is preferred to use JED_Driver.

Be sure that the number of jobs matches the number of job inputs.

Make sure to use separator lines after the batch parameters, between jobs, and after the last job.

ii. Multi-Chain PDB files: The Preliminary Run

Differences from Single-Chain analysis shown highlighted in amber

```
-------------------------------------------------------------------------------------------------
$num_of_jobs
0.00
0.00
1     1     2     A     B     795     151     0     0
0     0     0     0
*********************************************************************************
/working/directory/job1/
Description1                  reference_PDB_file1.pdb
*********************************************************************************
/working/directory/job2/
Description2                  reference_PDB_file2.pdb
*********************************************************************************
-------------------------------------------------------------------------------------------------
```

Notes: This is a whitespace delineated file.

Line 1 Field 1 specifies the **$num_of_jobs** (integer) for the batch.
Line 2 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.00 →NONE
Line 3 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 0.00 →NO ADJUSTMENT
Line 4 Field 1 specifies whether to **read PDB files** (0 or 1) →1 = yes
Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 1 = yes
Field 3 specifies the number of chains → (2);
Field 4 specifies the first chain ID → (A);
Field 5 specifies the second chain ID → (B);
Field 6 specifies the number of residues in chain A→ (795);
Field 7 specifies the number of residues in chain B→ (151);
Field 8 specifies the offset of Chain A → (0);
Field 9 specifies the offset of Chain B → (0);
Line 5 Field 1 specifies the **number of Cartesian** (integer) modes to process →0 = none
Field 2: specifies the **number of all-to-all Distance modes** (integer) to process → 0 = none
Field 3: specifies the **number of residue-pair Distance modes** (integer) to process → 0 = none
Field 4: specifies the **number of Cartesian modes to Visualize**(integer) → 0 = none
Line 6 is a **separator line** between the batch parameters and the job parameters **********
Line 7 Field 1 specifies the **working directory** (String) for job one
Line 8 Field 1 specifies the job **description** (String) for job one
Field 2 specifies the **reference PDB** (String) for job one
Line 9 is a **separator line** between sets of job parameters **********
Line 10 Field 1 specifies the **working directory** (String) for job two
Line 11 Field 1 specifies the job **description** (String) for job two
Field 2 specifies the **reference PDB** (String) for job two
Line 12 is a **separator line** between sets of job parameters **********

Key Points:
For single jobs, it is preferred to use JED_Driver.

Be sure that the number of jobs matches the number of job inputs.

Make sure to use separator lines after the batch parameters, between jobs, and after the last job.

iii. Single-Chain PDB files: Performing Cartesian PCA

```
-------------------------------------------------------------------------------------------
$num_of_jobs
0.01
3.00
0     0
20    0     0     2     1.0
*****************************************************************************
/working/directory1/
Description1          reference_PDB_file1.pdb      0
residues1.txt
original_PDB_Coordinates.txt        0
*****************************************************************************
/working/directory2/
Description2          reference_PDB_file2.pdb      0
residues2.txt
original_PDB_Coordinates.txt        0
*****************************************************************************
-------------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **$num_of_jobs** (integer) for the batch
Line 2 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.01 → 1%
Line 3 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|
Line 4 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no
Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 0 = no
Line 5 Field 1 specifies the **number of Cartesian** (integer) modes to process → 20 = top twenty modes
Field 2: specifies the **number of all-to-all Distance modes** (integer) to process → 0 = none
Field 3: specifies the **number of residue-pair Distance modes** (integer) to process → 0 = none
Field 4: specifies the **number of Cartesian modes to Visualize**(integer) → 2 = top 2 modes
Field 5 specifies the **scale factor** for the visualizations (double) → 1.0 = default value
Line 6 is a **separator line** between the batch parameters and the job parameters **********
Line 7 Field 1 specifies the **working directory** (String) for job one
Line 8 Field 1 specifies the job **description** (String) for job one
Field 2 specifies the **reference PDB** (String) for job one
Field 3 specifies the **offset** (integer) for job one
Line 9 Field 1 specifies the **residue list** (String) for job one
Line 10 Field 1 specifies the **coordinate matrix** (String) for job one
Field 2 specifies the **reference column** (integer) for job one
Line 11 is a **separator line** between sets of job parameters **********
Line 12 Field 1 specifies the **working directory** (String) for job two
Line 13 Field 1 specifies the job **description** (String) for job two
Field 2 specifies the **reference PDB** (String) for job two
Field 3 specifies the **offset** (integer) for job two
Line 14 Field 1 specifies the **residue list** (String) for job two
Field 2 specifies the **offset** (integer) for job two
Line 15 Field 1 specifies the **coordinate matrix** (String) for job two
Field 2 specifies the **reference column** (integer) for job two
Line 16 is a **separator line** between sets of job parameters **********

---

iv. Multi-Chain PDB files: Performing Cartesian PCA

Differences from Single-Chain analysis shown highlighted in amber

```
-------------------------------------------------------------------------------------
$num_of_jobs
0.01
3.00
0     1     2     A     B     795     151     0     0
20    0     0     2     1.0
*************************************************************************************************
/working/directory1/
Description1           reference_PDB_file1.pdb        (no offset here)
residues1.txt
original_PDB_Coordinates.txt           0
*************************************************************************************************
/working/directory2/
Description2           reference_PDB_file2.pdb        (no offset here)
residues2.txt
original_PDB_Coordinates.txt           0
*************************************************************************************************
-------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **$num_of_jobs** (integer) for the batch

Line 2 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.01 → 1%

Line 3 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|

Line 4 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no

Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 1 = yes

Field 3 specifies the number of chains → (2);

Field 4 specifies the first chain ID → (A);

Field 5 specifies the second chain ID → (B);

Field 6 specifies the number of residues in chain A→ (795);

Field 7 specifies the number of residues in chain B→ (151);

Field 8 specifies the offset of Chain A → (0);

Field 9 specifies the offset of Chain B → (0);

Line 5 Field 1 specifies the **number of Cartesian** (integer) modes to process → 20 = top twenty modes

Field 2: specifies the **number of all-to-all Distance modes** (integer) to process → 0 = none

Field 3: specifies the **number of residue-pair Distance modes** (integer) to process → 0 = none

Field 4: specifies the **number of Cartesian modes to Visualize**(integer) → 2 = top 2 modes

Field 5 specifies the **scale factor** for the visualizations (double) → 1.0 = default value

Line 6 is a **separator line** between the batch parameters and the job parameters **********

Line 7 Field 1 specifies the **working directory** (String) for job one

Line 8 Field 1 specifies the job **description** (String) for job one

Field 2 specifies the **reference PDB** (String) for job one

Line 9 Field 1 specifies the **residue list** (String) for job one

Line 10 Field 1 specifies the **coordinate matrix** (String) for job one

Field 2 specifies the **reference column** (String) for job one

Line 11 is a **separator line** between sets of job parameters **********

Line 12 Field 1 specifies the **working directory** (String) for job two

Line 13 Field 1 specifies the job **description** (String) for job two

Field 2 specifies the **reference PDB** (String) for job two

Line 14 Field 1 specifies the **residue list** (String) for job two

Line 15 Field 1 specifies the **coordinate matrix** (String) for job two

Field 2 specifies the **reference column** (String) for job two

Line 16 is a **separator line** between sets of job parameters **********

---

v. Single-Chain PDB files: Performing dPCA

```
-------------------------------------------------------------------------------------------
$num_of_jobs
0.01
3.00
0     0
0     3     0     0     (no scale factor)
*************************************************************************
/working/directory1/
Description1          reference_PDB_file1.pdb      0
residues_dist1.txt
original_PDB_Coordinates.txt        0
*************************************************************************
/working/directory2/
Description2          reference_PDB_file2.pdb      0
residues_dist2.txt
original_PDB_Coordinates.txt        0
*************************************************************************
-------------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **$num_of_jobs** (integer) for the batch
Line 2 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.01 → 1%
Line 3 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|
Line 4 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no
Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 0 = no
Line 5 Field 1 specifies the **number of Cartesian** (integer) modes to process → 0 = none
Field 2: specifies the **number of all-to-all Distance modes** (integer) to process → 3 = top 3 modes
Field 3: specifies the **number of residue-pair Distance modes** (integer) to process → 0 = none
Field 4: specifies the **number of Cartesian modes to Visualize**(integer) → 0 = none
Line 6 is a **separator line** between the batch parameters and the job parameters **********
Line 7 Field 1 specifies the **working directory** (String) for job one
Line 8 Field 1 specifies the job **description** (String) for job one
Field 2 specifies the **reference PDB** (String) for job one
Field 3 specifies the **offset** (integer) for job one
Line 9 Field 1 specifies the **distance residue list** (String) for job one
Line 10 Field 1 specifies the **coordinate matrix** (String) for job one
Field 2 specifies the **reference column** (integer) for job one
Line 11 is a **separator line** between sets of job parameters **********
Line 12 Field 1 specifies the **working directory** (String) for job two
Line 13 Field 1 specifies the job **description** (String) for job two
Field 2 specifies the **reference PDB** (String) for job two
Field 3 specifies the **offset** (integer) for job two
Line 14 Field 1 specifies the **distance residue list** (String) for job two
Field 2 specifies the **offset** (integer) for job two
Line 15 Field 1 specifies the **coordinate matrix** (String) for job two
Field 2 specifies the **reference column** (integer) for job two
Line 16 is a **separator line** between sets of job parameters **********

---

vi. Multi-Chain PDB files: Performing dPCA

Differences from Single-Chain analysis shown highlighted in amber

```
-------------------------------------------------------------------------------------------
$num_of_jobs
0.01
3.00
0     1     2     A     B     795     151     0     0
0     3     0     0     (no scale factor)
*******************************************************************************************
/working/directory1/
Description1          reference_PDB_file1.pdb      (no offset here)
residues_dist1.txt
original_PDB_Coordinates.txt        0
*******************************************************************************************
/working/directory2/
Description2          reference_PDB_file2.pdb      (no offset here)
residues_dist2.txt
original_PDB_Coordinates.txt        0
*******************************************************************************************
-------------------------------------------------------------------------------------------
```

Notes:

Line 1 Field 1 specifies the **$num_of_jobs** (integer) for the batch

Line 2 Field 1 specifies the **percent** (double) of frames to remove from the data: 0.01 → 1%

Line 3 Field 1 specifies the **z-cutoff** (double) for adjusting outliers: 3.00 → values beyond |3.0|

Line 4 Field 1 specifies whether to **read PDB files** (0 or 1) → 0 = no

Field 2 specifies if the PDB files are **Multi** Chain (0 or 1) → 1 = yes

Field 3 specifies the number of chains → (2);

Field 4 specifies the first chain ID → (A);

Field 5 specifies the second chain ID → (B);

Field 6 specifies the number of residues in chain A→ (795);

Field 7 specifies the number of residues in chain B→ (151);

Field 8 specifies the offset of Chain A → (0);

Field 9 specifies the offset of Chain B → (0);

Line 5 Field 1 specifies the **number of Cartesian** (integer) modes to process → 0 = none

Field 2: specifies the **number of all-to-all Distance modes** (integer) to process → 3 = top 3 modes

Field 3: specifies the **number of residue-pair Distance modes** (integer) to process → 0 = none

Field 4: specifies the **number of Cartesian modes to Visualize**(integer) → 0 = none

Line 6 is a **separator line** between the batch parameters and the job parameters **********

Line 7 Field 1 specifies the **working directory** (String) for job one

Line 8 Field 1 specifies the job **description** (String) for job one

Field 2 specifies the **reference PDB** (String) for job one

Line 9 Field 1 specifies the **distance residue list** (String) for job one

Line 10 Field 1 specifies the **coordinate matrix** (String) for job one

Field 2 specifies the **reference column** (String) for job one

Line 11 is a **separator line** between sets of job parameters **********

Line 12 Field 1 specifies the **working directory** (String) for job two

Line 13 Field 1 specifies the job **description** (String) for job two

Field 2 specifies the **reference PDB** (String) for job two

Line 14 Field 1 specifies the **distance residue list** (String) for job two

Line 15 Field 1 specifies the **coordinate matrix** (String) for job two

Field 2 specifies the **reference column** (String) for job two

Line 16 is a **separator line** between sets of job parameters **********

---

vii. Single-Chain PDB files: Performing cPCA, dPCA, and Visualization

```
-------------------------------------------------------------------------------------------
$num_of_jobs
0.01
3.00
0     0
20    3     0     2     1.0
*******************************************************************************
/working/directory1/
Description1          reference_PDB_file1.pdb      0
residues1.txt
residues_dist1.txt
original_PDB_Coordinates.txt        0
*******************************************************************************
/working/directory2/
Description2          reference_PDB_file2.pdb      0
residues2.txt
residues_dist2.txt
original_PDB_Coordinates.txt        0
*******************************************************************************
-------------------------------------------------------------------------------------------
```

viii. Multi-Chain PDB files: Performing cPCA, dPCA, and Visualization

```
-------------------------------------------------------------------------------------------
$num_of_jobs
0.01
3.00
0     1     2     A     B     795     151     0     0
20    3     2     1.0
*********************************************************************************************
/working/directory1/
Description1          reference_PDB_file1.pdb      (no offset here)
residues1.txt
residues_dist1.txt
original_PDB_Coordinates.txt        0
*********************************************************************************************
/working/directory2/
Description2          reference_PDB_file2.pdb      (no offset here)
residues2.txt
residues_dist2.txt
original_PDB_Coordinates.txt        0
*********************************************************************************************
-------------------------------------------------------------------------------------------
```

vii. Single-Chain PDB files: Performing cPCA, dPCA, dpPCA, and Visualization

```
-------------------------------------------------------------------------------------------
$num_of_jobs
0.00
2.00
0     0
20    3     10     2     1.0
*****************************************************************************
/working/directory1/
Description1          reference_PDB_file1.pdb      0
residues1.txt
residues_dist1.txt
residue_pairs1.txt
original_PDB_Coordinates.txt        0
*****************************************************************************
/working/directory2/
Description2          reference_PDB_file2.pdb      0
residues2.txt
residues_dist2.txt
residue_pairs2.txt
original_PDB_Coordinates.txt        0
*****************************************************************************
-------------------------------------------------------------------------------------------
```

viii. Multi-Chain PDB files: Performing cPCA, dPCA, dpPCA, and Visualization

```
-------------------------------------------------------------------------------------------
$num_of_jobs
0.00
2.00
0     1     2     A     B     795     151     0     0
20    3     10    2     1.0
*********************************************************************************************
/working/directory1/
Description1          reference_PDB_file1.pdb      (no offset here)
residues1.txt
residues_dist1.txt
residue_pairs1.txt
original_PDB_Coordinates.txt        0
*********************************************************************************************
/working/directory2/
Description2          reference_PDB_file2.pdb      (no offset here)
residues2.txt
residues_dist2.txt
residue_pairs1.txt
original_PDB_Coordinates.txt        0
*********************************************************************************************
-------------------------------------------------------------------------------------------
```

VI. Additional Types of Analysis

A. Pooling Data:

It is often useful to pool trajectory statistics. This can be done in JED by combining coordinate files and then performing the usual analysis. To combine the coordinate files, there is a utility program called **JED_Pool_Data.java** that will combine multiple matrices into one. Each matrix is appended to the last column of the preceding matrix. Of courses, the number of rows in the coordinate files must match. The matrices to combine are specified by an input file called **pool.txt** that the user must construct correctly.

i. Run Command:

```
java -jar -d 64 JED_Pool_Data.jar "/path/to/pool.txt"
```

ii. Input File format:

Line #1 specifies the number of jobs (integer)
Then for each job you must specify the following:
The number of matrices to combine (integer)
The output directory (string)
The path to the first matrix (String)

Below is a sample pool.txt file:

```
-----------------------------------------------------------------------------------------------
2
*****************************************
2
/output/directory/
/path/to/first/matrix/matrix_1.txt
/path/to/first/matrix/matrix_2.txt
*****************************************
2
/output/directory/
/path/to/first/matrix/matrix_1.txt
/path/to/first/matrix/matrix_2.txt
*****************************************
-----------------------------------------------------------------------------------------------
```

Notes:
Line #1 specifies the number of jobs (integer)
Line #2 Separator Line (Required)
Line #3 specifies the number of matrices to combine for Job 1 (integer)
Line #4 specifies the output directory for Job 1 (string)
Line #5 specifies the path to the first matrix for Job 1 (String)
Line #6 specifies the path to the second matrix for Job 1 (String)
Line #7 Separator Line (Required)
etc...

---

iii. Output File format:

The output is a single, augmented matrix with the same number of rows as the composite matrices and columns equal to the sum of all columns in the composite matrices.

The output file name is: **"Pooled_Coordinates_Matrix_$*number_of_input_matrices*.txt"**

B. Subspace Analysis:

Once JED Driver has been run on multiple trajectories as well as pooled trajectories, an analysis can be done to compare how similar the essential subspaces derived from those trajectories are to each other. JED contains a program called **Subspace_Analysis.java** along with 3 driver programs that perform those functions. The core program takes as input two matrices of eigenvectors derived from PCA (or NMA, ANM, etc.). The matrices must have the same number of rows and columns, meaning the vectors being compared come from the same vector space and that the subspaces have the same dimensions. For example, in an analysis of lysozyme you might choose to process 20 cPCA modes while examining 10 different experimental conditions plus pooled data. As long as all the subsets in the analysis are the same then all the 20 dimensional subspaces can be compared.

Like most of the JED programs, the subspace analysis program driver reads an input file called **SSA.txt** to obtain runtime information. This file must be constructed properly by the user to perform the analysis. The three driver programs are **SSA_Driver.java**, **FSSA_Driver.java**, and **FSSA_Iterated_Driver.java** and are different in how much analysis is requested. The SSA_Driver gives full outputs for non-iterated subspace comparison including both log files and individual flat files. The FSSA_Driver is a light-weight version with only RMSIP and PA output in the log files. The Iterated version performs a recursive variation of the above where all equidimensional subspaces are compared up to the size that was provided, for example, from 1 to 20 by step-size 1 for a 20 column input file.

i. Run Commands:

```
java -jar -d 64 SSA_Driver.jar “/path/to/SSA.txt”
java -jar -d 64 FSSA_Driver.jar “/path/to/SSA.txt”
java -jar -d 64 FSSA_Iterated_Driver.jar “/path/to/SSA.txt”
```

ii. Input File format:

ALL three drivers use the same input file, only the outputs are different.

The format for the file shown below is:

LINE 1: Number_of_Jobs (integer)
LINE 2: Output_Directory (string ending in "/" or "\\")
LINE 3: Batch_Decription (string)
LINE 4: Separator Line (Required)
THEN FOR EACH JOB:
Description (string)
Directory1 (string ending in "/" or "\\") Name1 (string) (eigenvectors1)
Directory2 (string ending in "/" or "\\") Name2 (string) (eigenvectors2)
Separator Line (Required)

Below is a sample SSA.txt file:

```
--------------------------------------------------------------------------------------------
4
/output/directory/
Single_Combo_SSA
********************************************************************************************
All-vs-A
/Users/physicslabs/                  all_combo_SS_75_top_20_eigenvectors.txt
/Users/physicslabs/                  1a6n_combo_SS_75_top_20_eigenvectors.txt
********************************************************************************************
All-vs-B
/Users/physicslabs/                  all_combo_SS_75_top_20_eigenvectors.txt
/Users/physicslabs/                  1wit_combo_SS_75_top_20_eigenvectors.txt
********************************************************************************************
All-vs-A+B
/Users/physicslabs/                  all_combo_SS_75_top_20_eigenvectors.txt
/Users/physicslabs/                  1ubq_combo_SS_75_top_20_eigenvectors.txt
********************************************************************************************
All-vs-A_B
/Users/physicslabs/                  all_combo_SS_75_top_20_eigenvectors.txt
/Users/physicslabs/                  1ypi_combo_SS_75_top_20_eigenvectors.txt
********************************************************************************************
--------------------------------------------------------------------------------------------
```

APPENDIX

A. Sample JED Log file:

```
Java Essential Dynamics
Job Description: Single_cPCA_dPCA_Viz
Working directory: C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Single\\
Output directory: C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Single\\JED_Results_Single_cPCA_dPCA_Viz/

Performing cPCA: 20 modes.
Performing dPCA: 3 modes.
Performing mode visualization on top  2  cPCA modes
Alpha carbon coordinates were obtained from file:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Single\\original_PDB_coordinates.txt
The number of residues read: 151
The number of conformations read: 2001
The dimension of the coordinates matrix is: 453  by  2001
The transformed PDB coordinates were obtained by quaternion least-squares alignment.
PDB reference file is 1A6N.pdb
Reference conformation (column) in the coordinate matrix: 0
Residue list for Cartesian subset:  residues.txt
Number of residues in Cartesian subset: 81
The transformed data was trimmed by removing 10.0 percent of the samples based on conformation RMSD.
The coordinates were 'conditioned' by adjusting outliers with Z-scores beyond 3.0 to their mean value.
Condition number of the covariance matrix (Q): 52,101,318,291,300,056
LOG of the Q Condition Number: 17
Trace of Q: 69
Condition number of the correlation matrix (R): 35,505,888,101,013,780
LOG of the R Condition Number: 17
Trace of R: 243
PDB file with B-factors replaced by residue RMSDs:  Single_cPCA_dPCA_Viz_SS_81.pdb
Number of residues in distance subset: 12
The coordinates were 'conditioned' by adjusting outliers with Z-scores beyond |3.0| to their mean value.
Condition number of the distance Covariance matrix (Q_dist): 80,427,675
LOG of Q_dist: 8
Trace of Q_dist: 2
Condition number of the distance Correlation matrix (R_dist): 183,847
LOG of R_dist: 5
Trace of R_dist: 66
MEANs and STD_DEVs for the residue distances:
Res1            Res2            Mean                Std_Dev
1               2               3.780               0.004
1               3               6.388               0.245
1               4               9.460               0.425
1               5               11.239              0.242
1               6               9.326               0.284
1               7               9.790               0.511
1               8               13.292              0.427
1               9               13.982              0.305
1               10              13.117              0.431
1               11              15.209              0.647
1               12              17.872              0.479
2               3               3.808               0.005
2               4               6.632               0.067
2               5               7.994               0.080
2               6               5.818               0.115
2               7               6.416               0.145
2               8               9.854               0.134
2               9               10.503              0.138
2               10              9.922               0.150
2               11              12.086              0.152
2               12              14.519              0.149
3               4               3.783               0.009
3               5               5.308               0.010
3               6               5.113               0.011
```

3	7	6.309	0.011
3	8	8.648	0.012
3	9	10.012	0.012
3	10	10.692	0.012
3	11	12.367	0.012
3	12	14.171	0.014
4	5	3.804	0.004
4	6	5.409	0.005
4	7	5.070	0.003
4	8	6.172	0.005
4	9	8.766	0.006
4	10	9.846	0.006
4	11	10.448	0.006
4	12	12.007	0.008
5	6	3.784	0.003
5	7	5.458	0.004
5	8	5.064	0.003
5	9	6.444	0.004
5	10	8.797	0.005
5	11	10.033	0.006
5	12	10.526	0.006
6	7	3.809	0.006
6	8	5.357	0.006
6	9	5.167	0.004
6	10	6.341	0.006
6	11	8.741	0.007
6	12	9.919	0.007
7	8	3.823	0.005
7	9	5.407	0.006
7	10	4.992	0.003
7	11	6.075	0.004
7	12	8.364	0.007
8	9	3.815	0.004
8	10	5.431	0.004
8	11	5.248	0.004
8	12	5.868	0.005
9	10	3.806	0.005
9	11	5.620	0.005
9	12	5.210	0.004
10	11	3.777	0.004
10	12	5.545	0.005
11	12	3.787	0.006

Sets of structures were generated to animate each of the top 2 cPCA modes using both Q and R PCA methods.
MODE AMPLITUDE: 1.000

Analysis completed: 2013-12-16 02:25:55

B. Sample PDB READ Log file:

```
1A6N.pdb
1A6N_froda_00000001.pdb
1A6N_froda_00000002.pdb
1A6N_froda_00000003.pdb
1A6N_froda_00000004.pdb
1A6N_froda_00000005.pdb
1A6N_froda_00000006.pdb
1A6N_froda_00000007.pdb
1A6N_froda_00000008.pdb
1A6N_froda_00000009.pdb
1A6N_froda_00000010.pdb
1A6N_froda_00000011.pdb
1A6N_froda_00000012.pdb
1A6N_froda_00000013.pdb
1A6N_froda_00000014.pdb
1A6N_froda_00000015.pdb
1A6N_froda_00000016.pdb
1A6N_froda_00000017.pdb
1A6N_froda_00000018.pdb
1A6N_froda_00000019.pdb
1A6N_froda_00000020.pdb
1A6N_froda_00000021.pdb
1A6N_froda_00000022.pdb
1A6N_froda_00000023.pdb
1A6N_froda_00000024.pdb
1A6N_froda_00000025.pdb
```

C. Sample Single Chain PDB Residue List file:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```

D. Sample Multi Chain PDB Residue List file:

```
A	1
A	2
A	3
A	4
A	5
A	6
A	7
A	8
A	9
A	10
A	11
A	12
A	13
A	14
A	15
A	16
A	17
A	18
A	19
A	20
A	21
A	22
A	23
A	24
A	25
B	1
B	2
B	3
B	4
B	5
B	6
B	7
B	8
B	9
B	10
B	11
B	12
B	13
B	14
B	15
B	16
B	17
B	18
B	19
B	20
```

E. Sample Single Chain PDB Residue-Pair List file:

```
1       3
2       5
3       8
4       9
5       12
6       22
7       28
8       45
9       48
10      90
```

F. Sample Multi Chain PDB Residue-Pair List file:

```
A       1       A       3
A       2       A       7
A       3       A       14
A       4       A       22
A       5       A       45
B       6       A       70
B       7       A       81
B       8       B       30
B       9       B       56
B       10      B       72
```

G. Sample SSA Log File:

```
Top_COV_Eigenvectors
Rows: 45
Cols: 3
Top_CORR_Eigenvectors
Rows: 45
Cols: 3

Projections file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_dPCA_V
iz_projections_dim_3.txt

Cumulative overlaps 1 --> 2 file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_dPCA_V
iz_CO_1_2_dim_3.txt

Cumulative overlaps 2 --> 1 file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_dPCA_V
iz_CO_2_1_dim_3.txt

Principle Angles file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_dPCA_V
iz_PA_dim_3.txt

Cosine Products file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_dPCA_V
iz_Cosine_Products_dim_3.txt

Vectorial sums of angles file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_dPCA_V
iz_Vector_sums_of_Angles_dim_3.txt

The absolute projections of each vector in subspace 1 with each vector in subspace 2 are:

    0.76    0.52    0.07
    0.31    0.69    0.47
    0.06    0.37    0.70

The cumulative overlaps CO_3 for each vector in subspace 1 with all the vectors in subspace 2 are:
Vector 1       0.919
Vector 2       0.890
Vector 3       0.797

The cumulative overlaps CO_3 for each vector in subspace 2 with all the vectors in subspace 1 are:
Vector 1       0.820
Vector 2       0.936
Vector 3       0.850

The RMSIP score is 0.870

The principle angles (in degrees) are:
PA1            6
PA2            21
PA3            31

The cosine products (in degrees) are:
CP1            6
CP2            30
CP3            51

The vectorial sums of angles (in degrees) are:
VS1            6
VS2            22
VS3            38

Maximum possible angle between two subspaces of this dimension is 90 degrees

Analysis completed: 2013-12-16 02:15:05
```

H. Sample FSSA Iterated Log File:

```
Principle Angle Spectra file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_
dPCA_Viz_PA_Spectra.txt

RMSIPs file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_
dPCA_Viz_RMSIPs.txt


RMSIPs:
Dim 1          0.758
Dim 2          0.839
Dim 3          0.870

The PA spectra for the range of subspaces are:

   29    0    0
   14   32    0
    6   21   31

Analysis completed: 2013-12-16 02:15:05
```

I. Sample Random SSA Log File:

```
Average RMSIPs file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_
dPCA_Viz_avg_random_RMSIPs.txt

Average PAs file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_
dPCA_Viz_avg_random_PAs.txt

Average COs file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_
dPCA_Viz_avg_random_COs.txt

RMSIP Std Devs file written to:
C:\\Users\\Charles\\workspace\\JED_1.0\\JED_Test\\Multi\\JED_results_Multi_cPCA_dPCA_Viz/dPCA/SSA/Multi_cPCA_
dPCA_Viz_random_RMSIP_std_devs.txt

The dimension of the vector space is  45
SS_DIM  avg_RMSIP      std_dev
1               0.883          0.094
2               0.647          0.069
3               0.557          0.064

The avg random PA spectra for the range of subspaces are:

   27    0    0
   26   71    0
   25   61   75


The avg random CO scores for the range of subspaces are:

   0.794   0.000   0.000
   0.803   0.176   0.000
   0.810   0.242   0.156

Analysis completed: 2013-12-16 02:15:05
```