**Java Essential Dynamics (JED)**

**User Manual and Tutorial**

**Acknowledgement**

Dr. Charles David wrote the JED program during his Ph.D. studies in Bioinformatics and Computational Biology at UNC Charlotte under the direction of Dr. Donald Jacobs. Partial support for this work came from NIH grants (GM073082 and HL093531), from the Center of Biomedical Engineering and Science, and the Department of Physics and Optical Science.

**GNU General Public License Agreement**

If you choose to use JED software, you agree to cite the references listed below on all publications that present results based on the JED analysis, and you agree to abide by the GNU General Public License Agreement (version 3). The GNU General Public License Agreement can be found at <http://www.gnu.org/licenses/gpl.html>

**Citation**

Charles C. David and Donald J. Jacobs. *JED: Java Essential Dynamics*. BMC (publication information).

**Table of Contents:**

1. **Introduction 2**
2. **Using JED 4**
3. **Overview of Using JED 6**
4. **Using JED Driver 8**
5. **The Preliminary Run 8**
6. **Debugging Crashes Part I 10**
7. **Performing Only cPCA 11**
8. **Performing Only dpPCA 16**
9. **Debugging Crashes Part II 20**
10. **Performing Cartesian Mode Visualization 21**
11. **Performing Multiple PCAs 22**
12. **Using JED Batch driver 23**
    1. **JED Batch Driver Input File Format: The Preliminary Run 23**
    2. **JED Batch Driver Input File Format: Only cPCA 24**
    3. **JED Batch Driver Input File Format: Only dpPCA 25**
    4. **JED Batch Driver Input File Format: Combined cPCA, dpPCA, VIZ 26**
    5. **Debugging Batch Crashes 26**
13. **Additional Types of Analysis 27**
    1. **Pooling Data 27**
    2. **Subspace Analysis 28**

**Appendix 30**

**I. Input File Formats 30**

**II. Output File Formats 33**

1. **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. The Z-scores for the variables is also calculated. 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 a 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**). Note that cPCA using ***n*** **residues** will yield eigenvectors having ***3n*** components, each corresponding to one Cartesian coordinate. The second method is based on internal coordinates using residue-pair distances (**dpPCA**). The dpPCA, using ***n*** **residue-pairs,** will yield eigenvectors having ***n*** components, each corresponding to one of inter-residue distance pairs. For small subsets, an all-to-all comparison could be specified.

**PCA Models:**

Both of the PCA methods are performed using a **covariance matrix** (**Q**), a **correlation matrix** (**R**) and a **partial correlation matrix (PC)**. 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 partial correlation matrix is obtained from the inversion of the covariance matrix, with subsequent normalization. The current implementation implements all three methods and compares the results.

**Conditioning of the sample Q Matrix:**

JED handles the removal of outliers prior to the PCA analyses with two approaches. First, the user can specify the **percent** (a decimal [0,1]) 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. Note that we do NOT recommend this approach as it discards data, but it is included because it is considered a standard practice. Second, the user can specify a **z-score cutoff** (a decimal ≥ 0) 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 number of **Q** and make it a better estimator for the population covariance matrix. 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 results of a PCA can be highly skewed due to the presence of outliers since such analyses are highly dependent on the quality of the sampling. It is strongly 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 dpPCA 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 each residue's atoms for each requested mode using a sine function to perturb the associated eigenvector component. A scale-factor parameter is used 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,** **R,** and **PC** 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 maximum value that is 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. Note that for the Q and R methods, the largest eigenvalue-eigenvector pairs are used, but for the PC method, the smallest magnitude pairs are used (closest to zero).

**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, R,** or **PC** 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. The different methods highlight the structure of the data and provide scaling for visualization. Note that weighted DVPs from the PC method will be essentially zero.

**Free energy calculation:**

When two or more modes are requested, JED will use the first two DVPs as order parameters to calculate free energy using a 2-D kernel density estimate (KDE) derived from Gaussian kernels. This output can be used to plot a free energy surface with respect to the first two PCA modes. This is done for both cPCA and dpPCA, and for each PCA model.

**Post PCA Comparative Subspace Analysis:**

JED performs a subspace analysis (**SSA**) on the two equidimensional sets of eigenvectors generated from the **Q, R,** and **PC** variants of PCA. The results provide a detailed comparison for the chosen subspace as well as 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 driver programs for the Subspace Analysis class. 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, while ensuring that the subsets of residues analyzed are identical. 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. JED produces summary log files for both of these analyses.

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

1. **Using 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 1.7 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 loads the complete covariance matrix (among other data structures) and performs a full matrix diagonalization, which scales as O(N3). Typically 8 to 32 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. On most platforms, Java can be optimized by specifying parameters at runtime for heap space, etc.

**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.

Note: The input file formats for the two driver programs are NOT equivalent.

**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 Java Virtual Machine (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 (for Pre-Processing runs) or the **Coordinates Matrix** to process (for all Analytical runs), along with the **reference PDB file** and **residue lists** for specifying the subsets of interest: Cartesian, and/or Distance Pairs.

**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”

***Remember to include command line switches to optimize the Java runtime environment for your jobs.***

**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-Pair PCA (**dpPCA**), and Mode Visualization Analysis (**VIZ**), when present. Each of these in turn contains limbs for **Q** (**COV**), **R** (**CORR**), and **PC (P\_CORR)** compartmentalization. The PCA directories also contains 3 subdirectories for the subspace analysis (**SSA**). The output file names include the **number of residues or residue pairs** 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 the coordinates matrix for essential dynamic analysis.

Each PDB file must have the exact 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 PDB files, this is probably the reason*.

While JED can process a PDB file with missing residues and various numbering schemes, it can NOT interpret files that have alternate conformations within a given frame based on fractional **occupancy** values. Only a single conformation per frame is allowed. Note that the original residue coordinates in the PDB files are mapped to the rows of the coordinates matrix.

1. **Overview of Using JED**

A **Preliminary Run** 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 represented in matrix form. 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.**

*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 for each subset chosen.*

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 rows of the coordinates matrix. JED generates a residue list file for all residues it finds in the PDB files that it reads. This file should be edited with care when specifying residue subsets.

Note: The most critical step when using 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. JED provides abundant error feedback during most crashes so that the problems can be addressed.

**Common Causes for JED to Crash**

* If **any** **specified** **directory** cannot be found or if **any specified** **file** cannot be found, JED will crash.
* If unexpected format is found in **any** of the input files, JED will crash.

Note that the JED driver programs employ many checks during the reading of the input files and the execution of the program. There are checks to validate the number formats of numeric data. There are checks to ensure that enough parameters were specified for the particular job. There are checks to ensure that the input files have the correct format/number of columns. There are checks to ensure that the number of modes requested does not exceed the actual number of modes available. JED also verifies that directories and files exist before performing any analysis. In many cases, missing or problematic parameter settings are set to a default value. The developers have attempted to provide meaningful information when the program crashes to facilitate making the necessary corrections. The specified input file is echoed to ***standard out***, as are the assignment of parameters, while any detected problems or errors have their messages directed to ***standard error***. In the case that a Java runtime exception is thrown, a stack trace will also be sent to standard error. Please refer to the **Appendix** for help in creating properly formatted input files.

1. **Using JED DRIVER:**
2. **The Preliminary Run**
3. **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 this is to generate the matrix of coordinates for performing subset analyses efficiently.

1. **Root 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 (for Single Chain PDBs)

All\_PDB\_Residues\_Multi\_JED.txt (for Multi Chain PDBs)

**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

**variables z-score matrix** for all the alpha carbon coordinates in the PDB files:

ss\_**$num**\_res\_Variable\_Z\_scores.txt

1. **JED Driver Input File Format: The Preliminary Run**

-----------------------------------------------------------------------------------------------------

1 $multi

$working\_directory

$description $reference\_PDB\_file.pdb

-----------------------------------------------------------------------------------------------------

Notes: This is a **whitespace** separated file with **6 lines**.

Line 1 Field 1 specifies **read flag**, whether to **read PDB files** (0 or 1) 🡪 1 = yes

Field 2: specifies **multi flag**, if the PDB files are **Multi** **Chain** (0 or 1)

Line 2 Field 1: specifies the **working directory** (String)

Line 3 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** to perform the pre-processing runs
  + When the **Read** flag is set to **1**, all PCAs are turned off
* The **Multi** flag must be set to **0** for Single Chain PDBs with no Chain IDs
* The **Multi** flag must be set to **1** or "multi" for Multi Chain PDBs with Chain IDs
  + **Multi Chain PDBs must have unique chain identifiers for every chain**
  + **Missing chain identifiers will cause JED to crash**
* **The file to use in all subsequent JED analyses is the original\_PDB\_coordinates matrix**

1. **Debugging Crashes Part I:**

Things that will generally make your life miserable…

1. **Simple mistakes:**

Are the **Read** and **Multi** flags set correctly?

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)

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?

Does the reference PDB file exist?

Does the reference PDB file correspond to the trajectory?

1. **More subtle mistakes:**

The directory contains PDB files in non-standard format.

The directory contains PDB files with fractional occupancy data.

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

The directory contains PDB files with missing 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.

1. **Performing Only cPCA**
2. **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 Cartesian residue list.***

The purpose of this type of run is to perform Essential Dynamics using cPCA based on **Q**, **R,** and **PC**. 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, R** and **PC** analysis, as does the VIZ directory.

1. **Root 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

1. **cPCA Output Files:**

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

/working/directory/JED\_Results\_Description/cPCA/

**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\_edited.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

**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 (shown below)**

/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/

* **The PC output files are written to the /PCORR subdirectory of /cPCA**

/working/directory/JED\_Results\_Description/cPCA/PCORR/

**covariance matrix:**

ss\_**$num\_res**\_covariance\_matrix.txt

**inverse covariance matrix:**

ss\_**$num\_res**\_inverse\_covariance\_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

**deltaG free energy from top 2 modes:**

ss\_**$num\_res**\_top\_**2**\_DVPs\_deltaG\_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

1. **SSA Output Files:**

These are written to the **/SSA subdirectory**  of /cPCA:

/working/directory/JED\_Results\_Description/cPCA/SSA/

* **The CORR versus PCORR output files are written to the /CORR\_VS\_PCORR subdirectory of /SSA (shown below)**

/working/directory/JED\_Results\_Description/cPCA/SSA/CORR\_vs\_PCORR/

* **The CORR versus COV output files are written to the /CORR\_VS\_COV subdirectory of /SSA**

/working/directory/JED\_Results\_Description/cPCA/SSA/CORR\_vs\_COV/

* **The COV versus PCORR output files are written to the /COV\_VS\_PCORR subdirectory of /SSA**

/working/directory/JED\_Results\_Description/cPCA/SSA/COV\_vs\_PCORR/

**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 additional files in the /SSA/CORR\_vs\_PCORR directory that contain the results reported in the log files:**

**RMSIPs**

**PAs**

**COs**

**Cosine Products**

**Vectorial Sum of Angles**

1. **JED Driver Input File Format: Only cPCA**

------------------------------------------------------------------------------------------

0 $multi

$working\_directory/

$description $reference\_PDB\_file.pdb

$percent $z\_cutoff

$num\_cPCA\_modes 0 0

residues.txt

original\_PDB\_Coordinates.txt $ref\_col

------------------------------------------------------------------------------------------

Notes: This is a **whitespace** delineated file with **7 lines**.

Line 1 Field 1 specifies **read flag**, whether to **read PDB files** (0 or 1) 🡪 **0 = no**

Field 2: specifies **multi flag**, if the PDB files are **Multi** **Chain** (0 or 1)

Line 2 Field 1 specifies the **working directory** (String)

Line 3 Field 1 specifies the job **description** (String)

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

Line 4 Field 1 specifies the **percent** (double)

Field 2 specifies the **z\_cutoff** (double)

Line 5 Field 1 specifies the **number of cPCA modes** (integer)

Field 2: specifies the **number of dpPCA modes** (integer) 🡪 **0 = none**

Field 3: specifies the **number of Cartesian modes to Visualize**(integer) 🡪 **0 = none**

Line 6 Field 1 specifies the **Cartesian** **residue list** (String)

Line 7 Field1 specifies the **coordinate matrix** (String)

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

------------------------------------------------------------------------------------------

**Key Points**:

* The **Read** flag **MUST** be set to **0** 
  + When the **Read** flag is set to **1**, all PCAs are turned off
* The **Multi** flag must be set to **0** for Single Chain PDBs with no Chain IDs
* The **Multi** flag must be set to **1** or "multi" for Multi Chain PDBs with Chain IDs
  + **Multi Chain PDBs must have unique chain identifiers for every chain**
  + **Missing chain identifiers will cause JED to crash**
* **The number of cPCA modes must NOT be 0.**
* **There must be a Cartesian residue list specified.**
* The number of dpPCA modes must be zero.
* There must not be a Distance or Distance Pair 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 modes.
* **If number of cPCA modes > 0, then there must be a Cartesian residue list file!**

1. **Residue List Format for cPCA**

***The easiest way to create the residue list for cPCA is to edit the JED generated file that lists all residues found in the PDB files, in the proper format:***

* 1. **Single Chain PDBs: "*All\_PDB\_Residues\_JED.txt*"**

**ONE column of integers: residue numbers**

* 1. **Multi Chin PDBs: "*All\_PDB\_Residues\_Multi\_JED.txt*"**

**TWO columns: Column 1 Strings, Column 2 integers (tab separated): Chain IDs, residue numbers**

***Note that all entries in the residue list are checked against the reference PDB file.***

***If a requested residue cannot be found in the reference file, then JED will crash with an error message stating that a requested residue could not be found.***

1. **Performing Only dpPCA**
2. **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, R,** and **PC** models. 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 easy 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, R,** and **PC** analysis, as well as for the subspace analysis (parallel to the cPCA method). Very large subsets can be used to investigate experimental findings in critical areas like binding pockets or clefts.

Note: Unfortunately, the dpPCA results cannot be visualized as no simple mapping can be made to the residues.

1. **Root 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

1. **dpPCA Output Files:**

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

/working/directory/JED\_Results\_Description/dpPCA/

**variables Z-Score matrix**:

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_distance\_Z\_scores.txt

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

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_outliers\_per\_variable.txt

**distance residue stats** from all the distance pairs in the **pair list**:

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_distance\_residue\_stats.txt

**distance Z-Score matrix** from all the distance pairs in the **pair list**:

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_distance\_Z\_scores.txt

**distances matrix** from all the distance pairs in the **pair list**:

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_distances.txt

**centroids (means) of the variables:**

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_centroids\_of\_variables.txt

**standard deviations of the variables:**

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_std\_devs\_of\_centered\_variables.txt

**displacement vectors:**

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_delta\_vectors.txt

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

/working/directory/JED\_Results\_Description/dpPCA/COV/

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

/working/directory/JED\_Results\_Description/dpPCA/CORR/

* **The PC output files are written to the /PCORR subdirectory of /dpPCA**

/working/directory/JED\_Results\_Description/dpPCA/PCORR/

**covariance matrix:**

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_distance\_pair\_covariance\_matrix.txt

**inverse covariance matrix:**

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_distance\_pair\_inverse\_covariance\_matrix.txt

**eigenvalues:**

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_eigenvalues\_COV.txt

**top eigenvalues:**

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_top\_**$num\_of\_dist\_modes**\_eigenvalues\_COV.txt

**top eigenvectors:**

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_top\_**$num\_of\_dist\_modes**\_eigenvectors\_COV.txt

**deltaG free energy from top 2 modes:**

ss\_**$num\_res\_pairs**\_Distance\_Pairs\_top\_**2**\_DVPs\_deltaG\_COV.txt

**top pca modes and top weighted pca modes:**

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_top\_**$num\_of\_dist\_modes**\_pca\_modes\_COV.txt

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_top\_**$num\_of\_dist\_modes**\_weighted\_pca\_modes\_COV.txt

**top square pca modes and top weighted square pca modes:**

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_top\_**$num\_of\_dist\_modes**\_square\_pca\_modes\_COV.txt

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_top\_**$num\_of\_dist\_modes**\_weighted\_square\_pca\_modes\_COV.txt

**top DVPs and top weighted DVPs:**

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_top\_**$num\_of\_dist\_modes**\_DVPs\_COV.txt

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_top\_**$num\_of\_dist\_modes**\_weighted\_DVPs\_COV.txt

**top normed DVPs and top weighted normed DVPs:**

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_top\_**$num\_of\_dist\_modes**\_normed\_DVPs\_COV.txt

ss\_$**num\_res\_pairs**\_Distance\_Pairs\_top\_**$num\_of\_dist\_modes**\_weighted\_normed\_DVPs\_COV.txt

1. **SSA Output Files: Same as for cPCA**

These are written to the **/SSA subdirectory**  of /dpPCA:

/working/directory/JED\_Results\_Description/dpPCA/SSA/

1. **JED Driver Input File Format: Only dpPCA**

------------------------------------------------------------------------------------------

0 $multi

$working\_directory/

$description $reference\_PDB\_file.pdb

$percent $z\_cutoff

0 $num\_dpPCA\_modes 0

residue\_pairs.txt

original\_PDB\_Coordinates.txt $ref\_col

------------------------------------------------------------------------------------------

Notes: This is a **whitespace** delineated file with **7 lines**.

Line 1 Field 1 specifies **read flag**, whether to **read PDB files** (0 or 1) 🡪 **0 = no**

Field 2: specifies **multi flag**, if the PDB files are **Multi** **Chain** (0 or 1)

Line 2 Field 1 specifies the **working directory** (String)

Line 3 Field 1 specifies the job **description** (String)

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

Line 4 Field 1 specifies the **percent** (double)

Field 2 specifies the **z\_cutoff** (double)

Line 5 Field 1 specifies the **number of cPCA modes** (integer) 🡪 **0 = none**

Field 2: specifies the **number of dpPCA modes** (integer) 🡪10 = top 10 modes

Field 3: specifies the **number of Cartesian modes to Visualize**(integer) 🡪 **0 = none**

Line 6 Field 1 specifies the **Distance Pairs residue list** (String)

Line 7 Field1 specifies the **coordinate matrix** (String)

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

------------------------------------------------------------------------------------------

**Key Points**:

* The **Read** flag **MUST** be set to **0** 
  + When the **Read** flag is set to **1**, all PCAs are turned off
* The **Multi** flag must be set to **0** for Single Chain PDBs with no Chain IDs
* The **Multi** flag must be set to **1** or "multi" for Multi Chain PDBs with Chain IDs
  + **Multi Chain PDBs must have unique chain identifiers for every chain**
  + **Missing chain identifiers will cause JED to crash**
* The number of cPCA modes must be 0.
* There must be no Cartesian residue list specified.
* **The number of dpPCA modes must NOT be zero.**
* **There MUST be a Distance Pair residue list specified.**
* If the subset you have chosen contains ***N*** pairs, then you must NOT request more than ***N*** modes.
* **If number of dpPCA modes > 0, then there must be a Distance Pair Residue list file!**

1. **Residue List Format for dpPCA**

***Residue pairs are specified one to a line in the residue pair list file:***

1. **Single Chain PDBs: Two columns of integers, tab separated:**

**residue number1 residue number2**

1. **Multi Chin PDBs: Four columns of strings and integers, tab separated:**

**ChainID1 Res\_Number1 Chain ID2 Res\_Number2**

***Note that all entries in the residue list are checked against the reference PDB file.***

***If a requested residue cannot be found in the reference file, then JED will crash with an error message stating that a requested residue could not be found.***

1. **Debugging Crashes Part II:**

Things that will generally make your life miserable…

1. **Simple mistakes:**

Did you set the **Read** and **Multi** flags correctly?

Are you requesting to read PDBs when you are doing an analytical run?

Did you request cPCA but not specify a Cartesian residue list?

Did you request dpPCA but not specify a Distance Residue Pair List?

Did you set the number of modes appropriately?

Are you requesting residues that are not in the reference PDB?

1. **Subtle mistakes:**

Did you request more PCA modes than actually exist?

For example (cPCA):

if your Cartesian subset contains 12 residues and you ask for 50 modes, then you are going get error messages:

Because there are only 36 Cartesian modes in total.

For example (dpPCA):

if your Distance Pairs List contains 5 pairs and you request 10 modes, then you are going to get error messages:

Because there are only 5 distance-pair modes in total.

In the above cases, JED will attempt to reset the offending value.

If your trajectory has not equilibrated, then you must address the problem of outliers. If you do not, then the covariance matrix 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 eigenvalues, but does correct the distortion that arises from under sampling when trying to estimate the population covariance matrix from a poor sample covariance matrix.

Examination of the correlation matrix, in conjunction with the partial correlation matrix, can provide insight into the amount of correlation between the variables, and how many variables are conditionally independent.

Note: The KMO and MSA are determined by using the correlations and partial correlations.

1. **Performing Cartesian Mode Visualization**

To activate cPCA mode visualization, you must be running a job with cPCA selected. To generate the output files, you only need to **set the number of modes VIZ > 0** and **≤ number of available cPCA modes**, and **set the mode amplitude**.

Recommended values for the mode amplitude are from 1.0 to 3.0 depending on the system.

If the mode amplitude is not set, it will be automatically set to **the default value of 1.5**

Note: Setting the modes VIZ flag to zero turns off the visualization feature.

The outputs to this type of job include all the structures for the top modes chosen for visualization. JED will permute the reference structure for a given subset along the top eigenvectors (using a sine function) selected for visualization and output 20 structures (PDBs) that capture one cycle of this motion. The amplitude of the motion is determined mostly by the value of the **$mode\_amplitude**, whose default value is 1.5, and can be adjusted as necessary. Setting the value too high could cause Visualization software like Pymol to break the ribbon diagrams of the structures, or yield bizarre distortions of the molecule. Ultimately, the magnitude of the displacements is controlled by the magnitude of the eigenvector components for any given residue. Setting the mode amplitude between 1 and 3 is usually safe, but for proteins with highly mobile regions like loops, you may need to adjust the mode amplitude. This is done for the modes from all three models. 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

The **R** results will be in the subdirectory /CORR.

The **PC** results will be in the subdirectory /PCORR.

1. **Performing Multiple PCAs**

***The working directory must contain: The coordinates matrix, the PDB reference file, and the residue lists.***

JED is capable of doing cPCA (with or without visualization), and dpPCA simultaneously.

All outputs are delivered as discussed for the individual components.

JED expects the input file to follow the following format regarding the order of the residue lists:

if cPCA then Cartesian Residue List

if dpPCA then Distance Pair Residue List

An important advantage of JED is that it is highly configurable to perform many types of Essential Dynamics analysis concurrently. Combined with cluster resources or just using the batch feature (discussed in the next section) allows a user to process a great deal of data efficiently.

1. **JED Driver Input File Format: Combined cPCA, dpPCA, VIZ**

------------------------------------------------------------------------------------------

0 $multi

$working\_directory/

$description $reference\_PDB\_file.pdb

0.00 3.00

$num\_cPCA\_modes $num\_dpPCA\_modes $num\_viz\_modes $mode\_amplitude

residues\_cartesian.txt

residue\_pairs.txt

original\_PDB\_Coordinates.txt $ref\_col

------------------------------------------------------------------------------------------

**Key Points**:

* The **Read** flag **MUST** be set to **0**
* The **Multi** flag must be set to **0** for Single Chain PDBs with no Chain IDs
* The **Multi** flag must be set to **1** or "multi" for Multi Chain PDBs with Chain IDs
* If the number of cPCA modes > 0, then there must be a Cartesian residue list specified
* If the number of dpPCA modes > 0, then there must be a Distance Pair residue list specified

**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.**

1. **JED Batch Driver Input File Format: The Preliminary Run**

-----------------------------------------------------------------------------------------------------

$num\_of\_jobs

1 $multi

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$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 **read flag**, whether to **read PDB files** (0 or 1) 🡪 1 = yes

Field 2 specifies the **multi flag**, if the PDB files are **Multi** Chain (0 or 1) 🡪 0 = no

Line 3 is a **separator line** between the batch parameters and the job parameters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Line 4 Field 1 specifies the **working directory** (String) for job one

Line 5 Field 1 specifies the job **description** (String) for job one

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

Line 6 is a **separator line** between sets of 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 two

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

Line 9 is a **separator line** between sets of job parameters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-----------------------------------------------------------------------------------------------------

**Key Points**:

* **Be sure that the number of jobs matches the number of job inputs.**
* The **Read** flag **MUST** be set to **1** or **"read"**
* The **Multi** flag must be set to **0** for Single Chain PDBs with no Chain IDs
* The **Multi** flag must be set to **1** or "multi" for Multi Chain PDBs with Chain IDs
* **Make sure to use separator lines after the batch parameters, between jobs, and after the last job.**

1. **JED Batch Driver Input File Format: Only cPCA**

------------------------------------------------------------------------------------------

2

0 0

0.01 3.00

20 0 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$working/directory1/

$description1 $reference\_PDB\_file1.pdb

$residues1.txt

original\_PDB\_Coordinates.txt 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$working/directory2/

$description2 $reference\_PDB\_file2.pdb

$residues2.txt

original\_PDB\_Coordinates.txt 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------

Notes:

Line 1 Field 1 specifies the **$num\_of\_jobs** (integer) for the batch 🡪 **2**

Line 2 Field 1 specifies the **read flag**, whether to **read PDB files** (0 or 1) 🡪 **0 = no**

Field 2 specifies the **multi flag**, if the PDB files are **Multi** Chain (0 or 1) 🡪 **0 = no**

Line 3 Field 1 specifies the **percent** (double) of frames to remove from the data: **0.01 🡪 1%**

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

Line 4 Field 1 specifies the **number of cPCA** modes (integer) 🡪 **20 = top 20 modes**

Field 2 specifies the **number of dpPCA** modes (integer) 🡪 **0 = none**

Field 3 specifies the **number of Cartesian modes to Visualize**(integer) 🡪 **0 = none**

Line 5 is a **separator line** between the batch parameters and the job parameters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Line 6 Field 1 specifies the **working directory** (String) for job one

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

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

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

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

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

Line 10 is a **separator line** between sets of job parameters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Line 11 Field 1 specifies the **working directory** (String) for job two

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

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

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

Line 14 Field 1 specifies the **coordinate matrix** (String) for job two

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

Line 15 is a **separator line** between sets of job parameters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------------------------------------------------------------------

1. **JED Batch Driver Input File Format: Only dpPCA**

------------------------------------------------------------------------------------------

2

0 1

0.01 3.00

0 3 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$working/directory1/

$description1 $reference\_PDB\_file1.pdb

$residues\_pairs1.txt

original\_PDB\_Coordinates1.txt $ref\_col

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$working/directory2/

$description2 $reference\_PDB\_file2.pdb

$residues\_pairs2.txt

original\_PDB\_Coordinates2.txt $ref\_col

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------

Notes:

Line 1 Field 1 specifies the **$num\_of\_jobs** (integer) for the batch

Line 2 Field 1 specifies the **read flag**, whether to **read PDB files** (0 or 1) 🡪 **0 = no**

Field 2 specifies the **multi flag**, if the PDB files are **Multi** Chain (0 or 1) 🡪 **1 = yes**

Line 3 Field 1 specifies the **percent** (double) of frames to remove from the data: **0.01 🡪 1%**

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

Line 4 Field 1 specifies the **number of cPCA** modes (integer) 🡪 **0 = none**

Field 2 specifies the **number of dpPCA** modes (integer) 🡪 **3 = top 3 modes**

Field 3 specifies the **number of Cartesian modes to Visualize**(integer) 🡪 **0 = none**

Line 5 is a **separator line** between the batch parameters and the job parameters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Line 6 Field 1 specifies the **working directory** (String) for job one

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

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

Line 8 Field 1 specifies the **residue pair list** (String) for job one

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

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

Line 10 is a **separator line** between sets of job parameters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Line 11 Field 1 specifies the **working directory** (String) for job two

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

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

Line 13 Field 1 specifies the **residue pair list** (String) for job two

Line 14 Field 1 specifies the **coordinate matrix** (String) for job two

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

Line 15 is a **separator line** between sets of job parameters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------------------------------------------------------------------

1. **JED Batch Driver Input File Format: Combined cPCA, dpPCA, VIZ**

------------------------------------------------------------------------------------------

$num\_of\_jobs

0 $multi

0.00 3.00

20 3 2 1.5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$working/directory1/

$description1 $reference\_PDB\_file1.pdb

$residues1.txt

$residues\_pairs1.txt

original\_PDB\_Coordinates.txt $ref\_col

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$working/directory2/

$description2 $reference\_PDB\_file2.pdb

$residues2.txt

$residues\_pairs2.txt

original\_PDB\_Coordinates.txt $ref\_col

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------

**Key Points**:

* **Be sure that the number of jobs matches the number of job inputs**
* The **Read** flag **MUST** be set to **0**
* The **Multi** flag must be set to **0** for Single Chain PDBs with no Chain IDs
* The **Multi** flag must be set to **1** or "multi" for Multi Chain PDBs with Chain IDs
* Make sure to use **separator lines** after the batch parameters, between jobs, and after the last job
* Make sure to specify the residue lists in the **correct order**: **cPCA first, dpPCA second**
* Make sure to specify the desired **$mode\_amplitude** if you request cPCA mode visualization

1. **Debugging Batch Crashes:**

Handling batch problems is more difficult than for single jobs. Any problem in any job can cause a crash.

Thus, it is good to track the standard out and errors streams to record the cause of any problems.

Also, you will know which jobs ran successfully so that you can edit the batch input file to finish the undone jobs.

1. **Simple mistakes:**

Are the **Read** and **Multi** flags set correctly?

Is the **number format** correct? (20.0 will NOT parse as an integer)

Do ALL the working directories, residue lists, and PDB reference files **exist**?

Do ALL the working directories end in “**/**” or “**\\**”?

Does the working directory contain **ALL of the required files**? (2 Types of PCA = 2 residue list input files)

1. **More subtle mistakes:**

Do any of the jobs have residue sets that will not allow the batch parameters to apply?

For example, Too few residues for the number of modes specified.

Does the directory contain PDB files in **non-standard format** or with fractional **occupancy** data?

Does the reference PDB file correspond to the trajectory? (JED must map the ref pdb to the coords matrix)

Does the directory contain Multi Chain PDB files with missing chain IDs?

1. **Additional Types of Analysis**
2. **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 course, 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.

* 1. **Run Command:**

java -jar -d64 JED\_Pool\_Data.jar “/path/to/pool.txt”

* 1. **Input File format:**

**LINE 1 specifies the number of jobs (integer)**

**LINE 2 is a separator line \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Then for each job you must specify the following:**

**The number of matrices to combine (integer)**

**The job description (String)**

**The output directory (string ending in "/" or "\\")**

**The path to each matrix (String)**

**A separator line at end of job declaration \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Sample "pool.txt":**

------------------------------------------------------------------------------------------

2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3

description1

/output/directory1/

/path/to/first/coords/matrix/original\_PDB\_Coordinates.txt

/path/to/second/coords/matrix/original\_PDB\_Coordinates.txt

/path/to/third/coords/matrix/original\_PDB\_Coordinates.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4

description2

/output/directory2/

/path/to/first/coords/matrix/original\_PDB\_Coordinates.txt

/path/to/second/coords/matrix/original\_PDB\_Coordinates.txt

/path/to/third/coords/matrix/original\_PDB\_Coordinates.txt

/path/to/fourth/coords/matrix/original\_PDB\_Coordinates.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------

Notes:

This file specifies 2 jobs, with 3 matrices to combine for job 1, and 4 matrices to combine for job 2.

Be sure that each path and matrix file exists.

------------------------------------------------------------------------------------------

* 1. **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 for job 1 is: **Pooled Coordinates Matrix $description1 $*number\_of\_input\_matrices*.txt**

(Pooled\_Coordinates\_Matrix\_$description\_3.txt for the example)**.**

1. **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 to perform the analysis correctly. 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.

1. **Run Commands:**

java -jar -d64 SSA\_Driver.jar “/path/to/SSA.txt”

java -jar -d64 FSSA\_Driver.jar “/path/to/SSA.txt”

java -jar -d64 FSSA\_Iterated\_Driver.jar “/path/to/SSA.txt”

1. **Input File format:**

**ALL three drivers use the same input file (only the outputs are different)**

The format for SSA.txt (shown below) is:

**LINE 1: Number\_of\_Jobs (integer)**

**LINE 2: Output\_Directory (string ending in "/" or "\\")**

**LINE 3: Batch\_Decription (string)**

**LINE 3: Separator Line \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**THEN FOR EACH JOB:**

**Description (string)**

**$directory1 (string ending in "/" or "\\") $eigenvectors1 (string)**

**$directory2 (string ending in "/" or "\\") $eigenvectors2 (string)**

**Separator Line \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Sample "SSA.txt":**

------------------------------------------------------------------------------------------

5

/output/directory1/

MV\_PCA\_Model\_Test

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MV1

/path/to/first/eigenvector/matrix/ ss\_946\_top\_20\_eigenvectors\_COV.txt

/path/to/second/eigenvector/matrix/ ss\_946\_top\_20\_eigenvectors\_CORR.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MV2

/path/to/first/eigenvector/matrix/ ss\_946\_top\_20\_eigenvectors\_COV.txt

/path/to/second/eigenvector/matrix/ ss\_946\_top\_20\_eigenvectors\_CORR.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MV3

/path/to/first/eigenvector/matrix/ ss\_946\_top\_20\_eigenvectors\_COV.txt

/path/to/second/eigenvector/matrix/ ss\_946\_top\_20\_eigenvectors\_CORR.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MV4

/path/to/first/eigenvector/matrix/ ss\_946\_top\_20\_eigenvectors\_COV.txt

/path/to/second/eigenvector/matrix/ ss\_946\_top\_20\_eigenvectors\_CORR.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MV5

/path/to/first/eigenvector/matrix/ ss\_946\_top\_20\_eigenvectors\_COV.txt

/path/to/second/eigenvector/matrix/ ss\_946\_top\_20\_eigenvectors\_CORR.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------

**APPENDIX**

1. **Input File Formats**
2. **JED\_Driver.txt (Pre-Processing Runs)**

------------------------------------------------------------------------------------------

$read $multi

$working­\_directory

$description $reference\_PDB

------------------------------------------------------------------------------------------

1. **JED\_Driver.txt (Analytical Runs)**

------------------------------------------------------------------------------------------

$read $multi

$working­\_directory

$description $reference\_PDB

$percent $z-cutoff

$num\_cPCA\_modes $num\_dpPCA\_modes $num\_viz\_modes $mode\_amplitude

$residues\_cartesian (Needed for cPCA)

$residue\_pairs.txt (Needed for dpPCA)

$PDB\_Coordinates $ref\_col

------------------------------------------------------------------------------------------

1. **JED\_Batch\_Driver.txt (Pre-Processing Runs)**

------------------------------------------------------------------------------------------

$num\_jobs

$read $multi

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$working­\_directory

$description $reference\_PDB

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------

1. **JED\_Batch\_Driver.txt (Analytical Runs)**

------------------------------------------------------------------------------------------

$num\_jobs

$read $multi

$percent $z-cutoff

$num\_cPCA\_modes $num\_dpPCA\_modes $num\_viz\_modes $mode\_amplitude

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$working­\_directory

$description $reference\_PDB

$residues\_cartesian

$residue\_pairs.txt

$PDB\_Coordinates $ref\_col

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------

1. **Residue List file for cPCA: Single Chain PDBs**

------------------------------------------------------------------------------------------

1

2

3

7

8

9

10

23

24

25

------------------------------------------------------------------------------------------

1. **Residue List file for cPCA: Multi Chain PDBs**

------------------------------------------------------------------------------------------

A 1

A 2

A 3

A 7

A 8

A 9

A 10

B 1

B 2

B 3

B 4

B 5

------------------------------------------------------------------------------------------

1. **Residue Pair List file for dpPCA: Single Chain PDBs**

------------------------------------------------------------------------------------------

2 27

5 32

12 57

18 80

36 101

------------------------------------------------------------------------------------------

1. **Residue Pair List file for dpPCA: Multi Chain PDBs**

------------------------------------------------------------------------------------------

A 2 A 27

A 5 A 32

A 12 B 57

B 18 B 80

B 36 B 101

------------------------------------------------------------------------------------------

1. **pool.txt**

------------------------------------------------------------------------------------------

$num\_jobs --> (repeat job declaration $num\_jobs times)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$num\_of\_matrices\_to\_combine

$description

$output\_directory

$path\_to\_coords\_matrix --> (repeat line $num\_of\_matrices\_to\_combine times)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------

1. **SSA.txt**

------------------------------------------------------------------------------------------

$num\_jobs --> (repeat job declaration $num\_jobs times)

$output\_directory

$batch\_description

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$job\_description

$path\_to\_first\_eigenvector\_file

$path\_to\_second\_eigenvector\_file

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------------------------

1. **Output File Formats**
2. **Sample JED Log file: cPCA, dpPCA, VIZ**

------------------------------------------------------------------------------------------

JED: Java Essential Dynamics version 1.0

Job Description: TEST

Working directory: C:\\Users\\Charles\\workspace\\JED\_2.0\\JED\_Test\\Multi\\

Output directory: C:\\Users\\Charles\\workspace\\JED\_2.0\\JED\_Test\\Multi\\JED\_RESULTS\_TEST/

READ PDBs = false

MULTI CHAIN PDBs = true

The alpha carbon coordinates were obtained from coordinates matrix file: original\_PDB\_Coordinates.txt

The dimension of the coordinates matrix is = 2838 by 101

Total number of residues in matrix = 946

Total number of conformations in matrix = 101

Transformed PDB coordinates obtained by quaternion least-squares alignment to the reference structure.

PDB reference structure is: MVb\_A\_B\_ATP.pdb

Reference Column in matrix = 0

PERFORMING cPCA, Computing Top 5 modes.

Residue list for Cartesian subset: residues.txt

Number of residues in Cartesian subset: 50

No samples were removed from the data

No coordinate outliers were adjusted.

Trace of the Covariance Matrix = 3

Condition Number of the Covariance Matrix = 5,806,065

Determinant of the Covariance Matrix = 0.0

Rank of the Covariance Matrix = 150

Trace of the Correlation Matrix = 150

Trace of the Partial Correlation Matrix = -150

PDB file with B-factors replaced by residue RMSDs: ss\_50\_RMSF\_edited.pdb

The DVPs (PCs) from the 3 different models were calculated using:

Standard dot product(dp), normed dp, weighted dp (by eigenvalue), and weighted normed dp

The Free energy (delta\_G) was determined using the first 2 standard DVPs as order parameters.

Probabilities were calculated using 2D Kernel Density Estimation with Gaussian kernels.

Subspace analysis was done comparing the top vector spaces from the 3 different models.

Comparators include RMSIP and Principle Angles, for the essential subspace and iterated comparisons from dim 1 to 5

Additional log files can be found in the /SSA directory tree.

PERFORMING dpPCA, Computing Top 3 modes.

Residue Pair list: residue\_pairs.txt

Number of residues pairs: 5

No coordinate outliers were adjusted.

Trace of the Covariance Matrix = 0.000

Condition Number of the Covariance Matrix = 51

Determinant of the Covariance Matrix = 0

Rank of the Covariance Matrix = 5

Trace of the Correlation Matrix = 5

Trace of the Partial Correlation Matrix = -5

MEANs and STANDARD DEVIATIONs for the Residue Pair Distances:

Res1 Res2 Mean Std\_Dev

A225 A294 32.651 0.000

A294 A525 41.435 0.000

A325 A525 44.517 0.002

A525 A795 105.495 0.001

A525 B52 80.534 0.000

The DVPs (PCs) from the 3 different models were calculated using:

Standard dot product(dp), normed dp, weighted dp (by eigenvalue), and weighted normed dp.

The Free energy (delta\_G) was determined using the first 2 standard DVPs as order parameters.

Probabilities were calculated using 2D Kernel Density Estimation with Gaussian kernels.

Subspace analysis was done to compare the top vector spaces from the 3 different models.

Comparators include RMSIP and Principle Angles, for the essential subspace and iterated comparisons from dim 1 to 3

Additional log files can be found in the /SSA directory tree.

Performing Cartesian Mode Visualization on Top 3 cPCA modes.

Sets of 20 structures were generated to animate each selected cPCA mode, for the COV, CORR, and PCORR PCA models.

Atoms of each residue were perturbed along the mode eigenvector using a sine function ranging from 0 to 2PI.

A Pymol(TM) script was generated for each mode to play the mode structures as a movie.

MODE AMPLITUDE = 2.500

Analysis completed: 2016-08-15 07:07:13

------------------------------------------------------------------------------------------

1. **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

1. **Sample SSA Log File:**

------------------------------------------------------------------------------------------

Top\_COV\_Eigenvectors:

Rows: 150

Cols: 5

Top\_CORR\_Eigenvectors:

Rows: 150

Cols: 5

Output Directory: C:\\Users\\Charles\\workspace\\JED\_2.0\\JED\_Test\\Multi\\JED\_RESULTS\_TEST/cPCA/SSA/CORR\_vs\_PCORR/

Projections file written to: Projections\_dim\_5.txt

Cumulative overlaps 1 --> 2 file written to: CO\_1\_2\_dim\_5.txt

Cumulative overlaps 2 --> 1 file written to: CO\_2\_1\_dim\_5.txt

Principle Angles file written to: PAs\_dim\_5.txt

Cosine Products file written to: Cosine\_Products\_dim\_5.txt

Vectorial sums of angles file written to: Vector\_Sums\_of\_Angles\_dim\_5.txt

The Inner Products of each vector in subspace 1 with each vector in subspace 2 are:

-0.995 0.015 -0.004 0.003 -0.009

0.016 0.993 -0.002 0.011 -0.028

0.003 0.001 -0.985 -0.003 -0.014

-0.004 0.008 0.004 -0.978 0.067

0.025 -0.040 0.001 -0.059 -0.946

The cumulative overlaps CO\_5 for each vector in subspace 1 with all the vectors in subspace 2 are:

Vector 1 0.996

Vector 2 0.994

Vector 3 0.985

Vector 4 0.980

Vector 5 0.949

The cumulative overlaps CO\_5 for each vector in subspace 2 with all the vectors in subspace 1 are:

Vector 1 0.996

Vector 2 0.994

Vector 3 0.985

Vector 4 0.979

Vector 5 0.949

The RMSIP score is 0.981

The principle angles (in degrees) are:

PA 1 4

PA 2 6

PA 3 10

PA 4 12

PA 5 19

The cosine products (in degrees) are:

CP 1 4

CP 2 7

CP 3 12

CP 4 16

CP 5 25

The vectorial sums of angles (in degrees) are:

VS 1 4

VS 2 7

VS 3 12

VS 4 17

VS 5 25

Maximum possible angle between two subspaces of this dimension is 201 degrees

Analysis completed: 2016-08-15 07:07:09

------------------------------------------------------------------------------------------

1. **Sample FSSA Iterated Log File:**

------------------------------------------------------------------------------------------

Output Directory: C:\\Users\\Charles\\workspace\\JED\_2.0\\JED\_Test\\Multi\\JED\_RESULTS\_TEST/cPCA/SSA/CORR\_vs\_PCORR/

Principle Angle Spectra file written to: Iterated\_PAs.txt

RMSIPs file written to: Iterated\_RMSIPs.txt

RMSIPs:

Dim 1 0.995

Dim 2 0.995

Dim 3 0.991

Dim 4 0.988

Dim 5 0.981

The PA spectra for the range of subspaces are:

6 0 0 0 0

5 7 0 0 0

5 6 10 0 0

5 6 10 12 0

4 6 10 12 19

Analysis completed: 2016-08-15 07:07:09

------------------------------------------------------------------------------------------