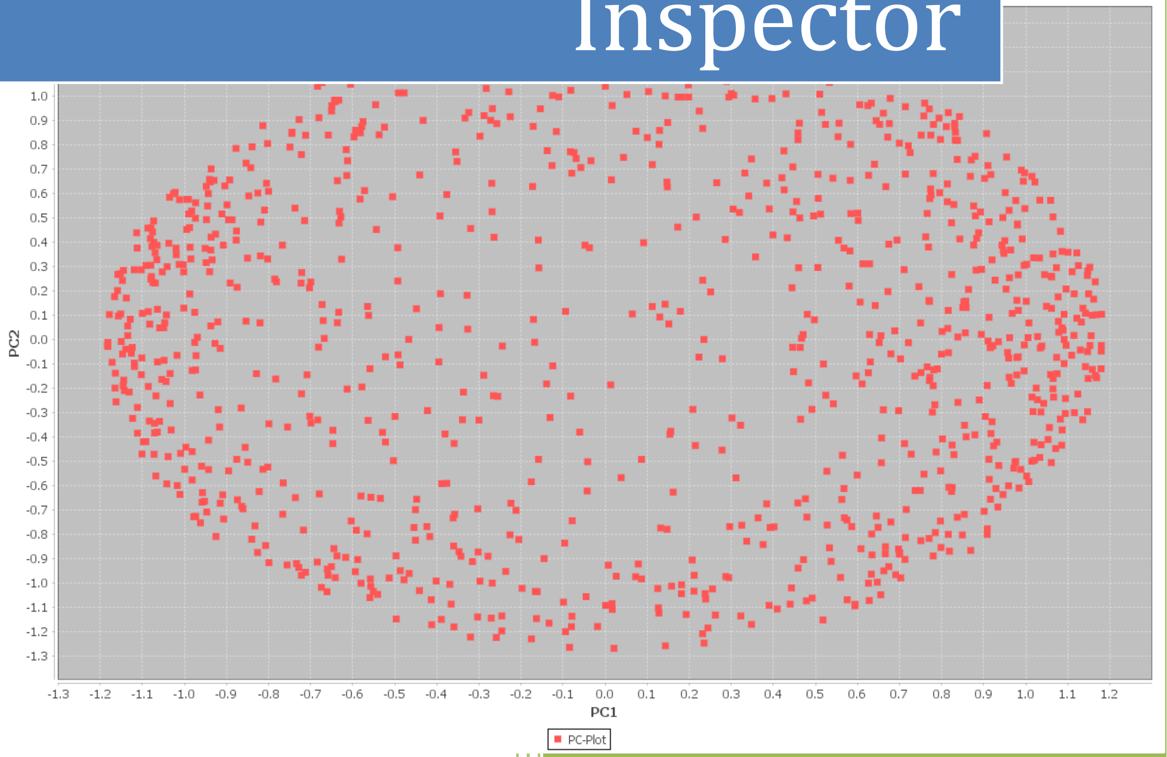
JEDi 2019

Java Essential Dynamics Inspector



Charles David, Donald Jacobs

UNC Charlotte, BMPG

JEDi 2019

Java Essential Dynamics Inspector (JEDi)

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. JEDi is the updated version of JED.

Citation

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

Supplemental Material: JEDi: Java Essential Dynamics Inspector, Charles David and Donald Jacobs

Table of Contents

Supplemental Material: JEDi: Java Essential Dynamics Inspector, Charles David and Donald Jacobs

I. Introduction

Java Essential Dynamics Inspector (JEDi) 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) and **Kernel Principle Component Analysis** (KPCA) that are not found in MD-simulation packages or other stand-alone PCA software, especially in regards to comparative analysis of multiple trajectories. JEDi is capable of running on any platform with a suitable Java Runtime Environment (JRE).

A. Expected Input to JEDi:

Ideally, each PDB structure must follow standard PDB-format. Note that some deviations from standard will often work fine, but JEDi expects standard format. Moreover, it is required that every PDB file used in the creation of the coordinates matrix be identical, meaning specifically, that they must all contain the same atoms. Additionally, JEDi distinguishes between PDB files that are single chain and those that are multi chain. A multi chain PDB file must have unique chain IDs for every atom in the file. We highly recommend that users review their PDB files carefully and ensure that they are all-atom and consistent with all formatting guidelines. All preprocessing of the PDB files must be done with external software before using JEDi. 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_000001>, <file_name_000002>, … <file_name_100000>, 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 as 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).

B. JEDi Pre-Processing Output:

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

C. Levels of Coarse-Graining:

The current implementation of JEDi offers four levels of coarse graining: all-atom, heavy atom, back-bone-atom, and alpha carbon atoms. The all atom level includes every atom, the heavy atom level includes all atoms except hydrogen, the backbone level considers only the backbone atoms, and the alpha carbon level uses the alpha carbon atom to represent the entire residue.

D. Types of PCA:

The core element of essential dynamics is to perform PCA. JEDi implements four variations of PCA. The first and most common method is based on <u>C</u>artesian coordinates (**cPCA**). Note that cPCA using ***n* atoms** will yield eigenvectors having ***3n*** components, each corresponding to one Cartesian coordinate. The second method is based on internal coordinates using atom-pair distances (**dpPCA**). The dpPCA method, using ***n* atom-pairs,** will yield eigenvectors having ***n*** components, each corresponding to one of inter-atom distance pairs. For small subsets, an all-to-all comparison could be specified. The third method is <u>D</u>isplacement PCA (**dPCA**) and is based on successive displacements between sampled frames. Whereas standard Cartesian PCA uses only one reference structure, the mean, the Displacement method uses each frame as a reference for the next frame. A data matrix ***n*** atoms and ***m*** frames will yield ***m-1*** displacements, and like standard Cartesian PCA, will yield eigenvectors having ***3n*** components. The fourth method is Hierarchical (**hPCA**), and is done by performing Cartesian PCA on each atom of every residue in the chosen subset, yielding a set of eigenvectors and principle components. Those residue eigenvectors are then convoluted with the overall eigenvectors derived from the factoring of the residue principle components. This method is available for the all-atom and heavy-atom levels, and yields results approximating those from standard Cartesian PCA on those subsets, in much less time.

E. PCA Models:

All of the PCA methods except hierarchical are performed using a **covariance matrix** (**Q**), and optionally, 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 release of JEDi implements all three models, and compares the results when selected.

F. Conditioning of the sample Q Matrix:

JEDi handles the removal of outliers prior to the PCA analyses with a simple, naive approach. The user can specify a **z-score cutoff** (a decimal ≥ 0) such that when the value of a PCA variable 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 process, a frame is never thrown away, but some entries within a frame may be modified. This method is intended to reduce the condition number of **Q** and make it a better estimator for the population covariance matrix. This 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. The method can be turned off by setting the z-score cutoff to zero.

A **stability threshold** is also provided that can be used to threshold the covariance matrix for very small numbers. The threshold is specified so that values less than the threshold but greater than 1.00E-16 are set to the threshold. Values less than 1.00E-16 are set to zero. This value should be tuned to the magnitude of the entries in the covariance matrices, to prevent skewing of real covariance. The value of this method is that extremely small numbers are removed from the covariance matrix before performing the eigenvalue decomposition, speeding up the calculation and stabilizing the spectral decomposition.
The method can be turned off by setting the threshold to zero.

G. Defining Subsets of Residues and Atoms:

The user can submit distinct residue lists for the following levels and types of analysis:

- All-Atom (Cartesian and/or Displacement)
- All-Atom Hierarchical
- All-Atom Local Residue (Cartesian PCA of each residue using local alignment)
- Heavy-Atom (Cartesian and/or Displacement)
- Heavy-Atom Hierarchical
- Backbone Atom (Cartesian and/or Displacement)
- Alpha Carbon (Cartesian and/or Displacement)
- Atom-Pairs (Distance Pair)

The format of the residue lists files is described below and differs between single and multi chain PDB files.

H. Visualization of PCA modes:

JEDi computes the **PCA modes** (RMSD and MSD, with and without weight by the corresponding eigenvalue) from the Cartesian/Displacement/Hierarchical eigenvectors so that they may be mapped to the subset of atoms. Sets of structures (PDB files) can be generated to visually inspect the cPCA, dPCA, and hPCA eigenvectors and 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 modes to visualize, beginning with mode one. Mode visualization is done by creating a set of 10 PDB files that capture the displacement of each atom, for each requested mode, using a sine function to perturb the associated PDB coordinate with 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. JEDi also produces a set of 30 frames to capture the Essential Modes, using a superposition of the top selected modes (up to five). This set of frames captures 3 periods of the fundamental mode.

I. 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 **P** 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.

J. Delta Vectors, Displacement Vector Projections, and Principle Components:

A set of **displacement vectors** (**DV**s) 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 <u>similar</u> 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** 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 square root of the corresponding eigenvalue (giving units of angstroms or angstrom squared). The different methods highlight the structure of the data and provide scaling for visualization.

When the **doPLOT** Boolean switch is set to **true**, JEDi will automatically plot the top two PCs or DVPs, producing a scatter plot with resolution 1200x800 in PNG format. This allows for immediate review of the data structure in each analysis.

K. Post PCA Kernel PCA Processing:

JEDi offers a Boolean option of performing kernel PCA (**doKPCA**) on the reduced data from the PCA analyses done. If selected, JEDi will perform KPCA, using multiple kernels, on the top two PCs or DVPs. A separate KPCA driver is available for fine tuning the KPCA analysis, but we find that using the top two PCA modes allows quick calculation times and strong separation of the data. The current kernels include mutual information based on 2D KDE, Gaussian, Neural Net (based on tanh function), Euclidean, Polynomial (with degrees 2, 3, and 4, difference of cubes, difference of squares), and Sine^2-Cosine^2.

When the **doPLOT** Boolean switch is set to **true**, JEDi will automatically plot the top two KPCs, producing a scatter plot with resolution 1200x800 in PNG format. This allows for immediate review of the data structure in each analysis.

L. Post PCA Comparative Subspace Analysis:

When multiple models are selected, JEDi performs inter-model subspace analysis (**SSA**) on the two equidimensional sets of eigenvectors generated from the **Q, R,** and **P** variants of PCA. The results provide a detailed comparison for the chosen essential subspace, as well as comparisons for all subspace dimensions up to the dimension chosen by the user (when selecting the number of Cartesian/Displacement/Hierarchical 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. Moreover, JEDi performs an inter-type subspace analysis when results from identical subsets are obtained using different types of PCA (the subspaces must have the same dimensions).

To provide a base-line for the reported RMSIP values, JEDi computes a RMSIP score for **two random subspaces** of the representative vector space, iterated, so that one can assess the affect of choice of subspace dimension on the results. This is only done for the COV model. 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. JEDi produces summary log files for both of these analyses.

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

II. Using JEDi

A. JEDi Install Instructions:

Java is **platform independent** and JREs exist for all common architectures. The machine on which JEDi is to be run should have **JRE version 1.8 or higher** installed. The programs can be run from compiled source or from the provided executable jar files. While JEDi 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.

- Dependencies: (these are in the **`/library`** folder)
    - The JAMA matrix package: `Jama-1.0.3.jar`
    - The Java Commons package: `jcommon-1.0.23.jar`
    - The JFreeChart package: `jfreechart-1.0.19.jar`, `jfreechart-1.0.19-experimental.jar`, `jfreechart-1.0.19-swt.jar`

When compiling from source, be sure to compile the **Jama, Java Commons**, and **JFree** packages as JEDi depends on those libraries, or ensure that the package JAR files are on the Java classpath. Alternatively, no source code or compilation is needed to run the executable jar files. These contain all source code and dependencies and 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.

B. Expected Memory Requirements:

On high performance computer clusters make sure the 64 bit JRE is installed. Memory use is demanding because JEDi loads the complete covariance matrix (among other data structures) and performs a full matrix diagonalization, which scales as $O(N^3)$. Typically 8 to 32 GB of RAM will be needed depending on the size of the protein. For very large proteins consisting of tens of thousands of atoms and/or many tens of thousands of frames, select a high memory node and request as much RAM as possible. On most platforms, Java can be optimized by specifying parameters at runtime for heap space, etc.

C. The JEDi Driver:

The JEDi programs are run by a single class, JEDi_Driver.java. This class contains the main method and instantiates all analysis. The key to running JEDi is in specifying the parameters for the analyses. This is made simple by having the driver class read in an input file that contains all needed information.

D. Input File for JEDi Driver:

JEDi requires an input file that specifies job parameters. The format of this file is a simple list of **KEY=VALUE** pairs. There must be no spaces in the declaration. Lines starting with a "#" are ignored. The run command takes only one (optional) argument, which is the name of the input file that includes the absolute path to the file. If no argument is specified, then JEDi assumes that the default input file name is used and that the file is located in the same directory from which the Java Virtual Machine (JVM) was called.

The default input file name is: **`JEDi_Parameters.txt`**

Below is an annotated version of the input file:

```
# ------------------------------------------------------------------------------------------------------------------------
#     DIRECTORY, DESCRIPTION, REFERENCE_PDB
# ------------------------------------------------------------------------------------------------------------------------
DIRECTORY=/working/directory/     ----->     Working directory, ends with a File.separator.char (\ or /)
DESCRIPTION=TEST                  ----->     Job Description
REFERENCE_PDB=pdb_file.pdb        ----->     The name of the PDB reference file (must be in the working directory)
# ------------------------------------------------------------------------------------------------------------------------
#     LOGICAL SWITCHES
# ------------------------------------------------------------------------------------------------------------------------
doREAD=false            ----->     Set to true when doing a PreProcessing Run ONLY
doMULTI=false           ----->     Set to true for Multi-Chain PDB files (each chain ID must be unique and non-empty)
doCARTESIAN=true        ----->     Set to true to perform the Cartesian Based Analysis
doDISPLACEMENT=true     ----->     Set to true to perform the Displacement Based Analysis
doCORR=true             ----->     Set to true to perform the Correlation PCA models
doPCORR=false           ----->     Set to true to perform the partial correlation PCA models
doREDUCE=false          ----->     Set to true to perform the matrix reductions of the PCA C-matrices
doKPCA=false            ----->     Set to true to perform kernel PCA analysis using the top weighted DVPs or PCs
doFES=false             ----->     Set to true to calculate Free Energy Surface files using the top two normed DVPs or PCs
doModeViz=false         ----->     Set to true to Visualize the Individual PCA modes in addition to the Essential Modes
doPLOT=true             ----->     Set to true to Plot the top two DVPs or PCs and save as PNG files
# ------------------------------------------------------------------------------------------------------------------------
#     NUMBER OF MODES
# ------------------------------------------------------------------------------------------------------------------------
NUMBER_OF_MODES_LOCAL=0              ----->     The number of modes for the Local Alignment Residue analysis
NUMBER_OF_MODES_HIERARCHICAL_AA=9    ----->     The number of modes for the All-Atom Hierarchical analysis
NUMBER_OF_MODES_HIERARCHICAL_HA=6    ----->     The number of modes for the Heavy Atom Hierarchical analysis
NUMBER_OF_MODES_ALL_ATOM=5           ----->     The number of modes for the All-Atom Cartesian/Displacement analysis
NUMBER_OF_MODES_HEAVY_ATOM=5         ----->     The number of modes for the Heavy-Atom Cartesian/Displacement analysis
NUMBER_OF_MODES_BACKBONE=5           ----->     The number of modes for the Backbone-Atom Cartesian/Displacement analysis
NUMBER_OF_MODES_ALPHA_CARBON=5       ----->     The number of modes for the Alpha-Carbon Cartesian/Displacement analysis
NUMBER_OF_MODES_DISTANCE_PAIRS=0     ----->     The number of modes for the Distance Pair analysis
NUMBER_OF_MODES_VIZ=3                ----->     The number of modes to use for visualization (Essential and Individual)
# ------------------------------------------------------------------------------------------------------------------------
#     RESIDUE LISTS (These files must be in the working directory)
# ------------------------------------------------------------------------------------------------------------------------
RESIDUE_LIST_ALL_ATOM=residues.txt             ----->     The residue list for the All-Atom analysis
RESIDUE_LIST_HEAVY_ATOM=residues.txt           ----->     The residue list for the Heavy-Atom analysis
RESIDUE_LIST_BACKBONE=residues.txt             ----->     The residue list for the Backbone analysis
RESIDUE_LIST_ALPHA_CARBON=residues.txt         ----->     The residue list for the Alpha-Carbon analysis
RESIDUE_LIST_LOCAL=residues.txt                ----->     The residue list for the Local Residue analysis
RESIDUE_LIST_HIERARCHICAL_AA=residues.txt      ----->     The residue list for the Global All-Atom Hierarchical analysis
RESIDUE_LIST_HIERARCHICAL_HA=residues.txt      ----->     The residue list for the Global Heavy-Atom Hierarchical analysis
ATOM_PAIRS_LIST=atom_pairs.txt                 ----->     The atom pairs list for the Distance-Pair analysis
# ------------------------------------------------------------------------------------------------------------------------
#     PCA and VIZ SETTINGS
# ------------------------------------------------------------------------------------------------------------------------
Z_SCORE_CUTOFF=3.00           ----->     z-score cutoff for removing variable outliers (0.00 to disable)
STABILITY_THRESHOLD=1.00E-12  ----->     stability threshold for small numbers in the covariance matrix (0.00 to disable)
VIZ_MODE_SCALE_FACTOR=0.25    ----->     mode amplitude scale factor for mode visualizations (suggest values near 0.25)
# ------------------------------------------------------------------------------------------------------------------------
#     ALL-ATOM COORDS FILE (This file must be in the working directory)
# ------------------------------------------------------------------------------------------------------------------------
ORIGINAL_PDB_COORDS=original_PDB_Coordinates_AA.txt   ----->     all-atom PDB coordinates file from the PreProcessing step
# ------------------------------------------------------------------------------------------------------------------------
```

Lines starting with a "#" are comments, and ignored.

NOTE: The working input file has no annotations, these are for reference only.

III. Running JEDi Driver:

Each job should be assigned to its own working 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.

A. JEDi Command Line format:

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

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

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

- `-d64, -server, -Xms, -Xmx, -XX:MaxGCPauseMillis, -XX:+UseLargePages, -XX:+AlwaysPreTouch, -XX:+DisableExplicitGC`
    - ➢ `Set: Xms = Xmx, -XX:+AlwaysPreTouch, -XX:MaxGCPauseMillis=10000, and -XX:+DisableExplicitGC to eliminate heap resizing and back virtual memory with physical memory, turn off explicit GC, and increase GC pauses.`

B. Organization of Output Files:

Output files from JEDi 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 "JEDi_RESULTS_**$description**", where **$description** is a user set parameter that succinctly describes the job. Limbs of the tree separate the types of PCA analysis, KPCA, FES, SSA, and VIZ, when present. Each of these in turn contains limbs for the models of PCA used, **Q** (**COV**), **R** (**CORR**), and **P (PCORR)** compartmentalization. The output file names include the **number of atoms or residues or atom-pairs** in the selected subset for reference, plus a description of the file contents.

C. 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 all-atom atomic positions only (see below for a description of this file). in JED, only carbon-alpha atomic positions were used to create the coordinates matrix for essential dynamic analysis.
In JEDi, all-atom atomic positions are used to create the coordinates matrix for essential dynamic analysis.

During pre-processing, be sure that each PDB file has the exact same number of residues and atoms. If other files in the working directory do not match exactly, then the array sizes will not match and the program will crash. *If JEDi crashes during the reading of PDB files, this is probably the reason*. During analytical runs, the indices for addressing the matrix of coordinates is determined from the specified Reference PDB file. JEDi will attempt to verify that the two match.

While JEDi 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.
A block XYZ packing is done:
For ***N*** *atoms*, there will be ***N x-coordinate*** rows, ***N y-coordinate*** rows, and ***N z-coordinate*** rows, in that order.

IV. Getting Started:

A. The Pre-Processing Run:

A Preliminary **Pre-Processing Run** must be performed to generate the JEDi formatted coordinate matrix file for all the atoms in the PDB files. This makes subsequent subset analyses much faster to perform. It also serves to guarantee that the specified atoms/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_AA.txt"`
The matrix packing is as follows:
Rows are coordinate variables and columns are frames.
For N *atoms*, 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 JEDi analyses is the original_PDB_coordinates_AA matrix.**

This matrix contains all the atoms in the PDB files and thus can be used for any subset of 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 file.
JEDi maps the specified residue list to an internal list that is aligned to the rows of the coordinates matrix.
JEDi 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.

B. Common Causes for JEDi to Crash

- A **KEY=VALUE PAIR** is missing
- A key in a **KEY=VALUE PAIR** is missing its value
- A specified **directory** or **file** cannot be found
- If **unexpected format** is found in **any** of the input files (reference PDB, residue list)

Note that the JEDi 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 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. JEDi 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 input file is logged and 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.

V. Using JEDi DRIVER

A. The Preliminary Run

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.

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

i. Parameters to set in the parameters file:

```
# -------------------------------------------------------------------------------------------------------------------
#     DIRECTORY, DESCRIPTION, REFERENCE_PDB
# -------------------------------------------------------------------------------------------------------------------
DIRECTORY=C:\Users\workspace\JEDi\test\BL\
DESCRIPTION=TEST
REFERENCE_PDB=1ERM_TEM1.cleaned.pdb
# -------------------------------------------------------------------------------------------------------------------
#     LOGICAL SWITCHES
# -------------------------------------------------------------------------------------------------------------------
doREAD=true
doMULTI=false
# -------------------------------------------------------------------------------------------------------------------
```

- **KEY=VALUE PAIRS SHOWN IN RED ARE RELEVENT TO THE PRE-PROCESSING RUN**

ALL OTHER PARAMETERS ARE IGNORED (but still need to be in the file)

Key Points:

- The **doREAD** parameter **MUST** be set to **true** to perform the pre-processing runs
  - When the **doREAD** parameter is set to **true**, all PCAs are turned off

- The **doMULTI** parameter must be set to **false** for Single Chain PDBs with no Chain IDs

- The **doMULTI** flag must be set to **true** for Multi Chain PDBs with Chain IDs
  - **Multi Chain PDBs must have unique chain identifiers for every chain**
  - **Missing chain identifiers will cause JEDi to crash**

ii. Output Files:

These are written to the **root** of the JEDi Results directory tree: `/working/directory/JEDi_Results_Description/`

- **JEDi LOG** providing a summary of the job parameters and results: `JEDi_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 all-atom coordinates in the PDB files: `original_PDB_coordinates_AA.txt`
- **A list of all residues** found in the PDB files for subsequent editing and use:
  - `All_PDB_Residues_JEDi.txt (for Single Chain PDBs)`
  - `All_PDB_Residues_Multi_JEDi.txt (for Multi Chain PDBs)`
- **Transformed conformation RMSDs**: `ss_$num_res_conformation_rmsds.txt`
- **Residue RMSF**: `ss_$num_res_residue_rmsf.txt`
- **Edited PDB file** containing all the atoms with the RMSF replacing B-factors: `ss_$num_res_edidited.PDB`
- **Variables z-score matrix** for all the atoms coordinates in the PDB files: `ss_$num_res_Variable_Z_scores.txt`

B. Performing Production Runs

The working directory <u>must</u> contain: The coordinates matrix, the PDB reference file, and the residue lists.

i. <u>Parameters to set in the parameters file (example):</u>

```
# -----------------------------------------------------------------------------------------------------------------------
#     DIRECTORY, DESCRIPTION, REFERENCE_PDB
# -----------------------------------------------------------------------------------------------------------------------
DIRECTORY=C:\Users\workspace\JEDi\test\BL\
DESCRIPTION=TEST
REFERENCE_PDB=1ERM_TEM1.cleaned.pdb
# -----------------------------------------------------------------------------------------------------------------------
#     LOGICAL SWITCHES
# -----------------------------------------------------------------------------------------------------------------------
doREAD=false
doMULTI= true
doCARTESIAN=true
doDISPLACEMENT=true
doCORR=true
doPCORR= true
doREDUCE= true
doKPCA= true
doFES= true
doModeViz= true
doPLOT=true
# -----------------------------------------------------------------------------------------------------------------------
#                                            NUMBER OF MODES
# -----------------------------------------------------------------------------------------------------------------------
NUMBER_OF_MODES_LOCAL=5
NUMBER_OF_MODES_HIERARCHCAL_AA=9
NUMBER_OF_MODES_HIERARCHCAL_HA=6
NUMBER_OF_MODES_ALL_ATOM=5
NUMBER_OF_MODES_HEAVY_ATOM=5
NUMBER_OF_MODES_BACKBONE=5
NUMBER_OF_MODES_ALPHA_CARBON=5
NUMBER_OF_MODES_DISTANCE_PAIRS=5
NUMBER_OF_MODES_VIZ=5
# -----------------------------------------------------------------------------------------------------------------------
#      DECIMAL SETTINGS
# -----------------------------------------------------------------------------------------------------------------------
Z_SCORE_CUTOFF=3.000
STABILITY_THRESHOLD=1.000E-12
VIZ_MODE_SCALE_FACTOR=0.250
# -----------------------------------------------------------------------------------------------------------------------
#     RESIDUE LISTS FOR SUBSETS
# -----------------------------------------------------------------------------------------------------------------------
RESIDUE_LIST_ALL_ATOM=residues.txt
RESIDUE_LIST_HEAVY_ATOM=residues.txt
RESIDUE_LIST_BACKBONE=residues.txt
RESIDUE_LIST_ALPHA_CARBON=residues.txt
RESIDUE_LIST_LOCAL=residues.txt
RESIDUE_LIST_HIERARCHICAL_AA=residues.txt
RESIDUE_LIST_HIERARCHICAL_HA=residues.txt
ATOM_PAIRS_LIST=atom_pairs.txt
# -----------------------------------------------------------------------------------------------------------------------
#     COORDS FILE
# -----------------------------------------------------------------------------------------------------------------------
ORIGINAL_PDB_COORDS=original_PDB_Coordinates_AA.txt
# -----------------------------------------------------------------------------------------------------------------------
```

<u>Key Points</u>:

- The **doREAD** parameter **<u>MUST</u>** be set to **false** to perform production runs
- The **doMULTI** parameter must be set to **false** for Single Chain PDBs with no Chain IDs
- The **doMULTI** flag must be set to **true** for Multi Chain PDBs with Chain IDs
- Set the logical switches to **true** to perform the selected action
- Set the Number of modes for any analysis **greater than zero** to turn it on
- For each subset, specify the residue list
- **Make sure there are no spaces in the KEY=VALUE declarations**

ii. Root Output Files:

These are written to the **root** of the JEDi Results directory tree: `/working/directory/JEDi_Results_Description/`

- **The JEDi LOG** providing a summary of the job parameters and results:
  - `JEDi_LOG.txt`
- Numbers of atoms in residues: `numbers_Of_Atoms_in_Residues.txt`
- Numbers of heavy atoms in residues: `numbers_Of_Heavy_Atoms_in_Residues.txt`
- Reference coordinates for chosen subsets: ex) `subset_reference_PDB_coordinates_cartesian_AA.txt`
- RMSF edited PDB files: ex) `ss_xxx_HA_RMSF_edited.pdb`

iii. Process Dependent Output Files:

- PCA analyses are written to the **/type_PCA/model_PCA subdirectories** of the JEDi Results directory tree
- KPCA analyses are written to the **/KPCA/type_PCA/model_PCA subdirectories** of the JEDi Results directory tree
- FES analyses are written to the **/FES /type_PCA/model_PCA subdirectories** of the JEDi Results directory tree
- VIZ outputs are written to the **/VIZ /type_PCA/model_PCA subdirectories** of the JEDi Results directory tree
- SSA outputs are written to the **/SSA /type_PCA/model_PCA subdirectories** of the JEDi Results directory tree

iv. Residue List Format

The easiest way to create a residue list is to edit the JEDi generated file that lists all residues found in the PDB files

(This is already in the proper format)

a. Single Chain PDBs: "*All_PDB_Residues_JEDi.txt*"

ONE column of integers: residue numbers

b. Multi Chin PDBs: "*All_PDB_Residues_Multi_JEDi.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 JEDi will crash with an error message stating that a requested residue could not be found.

v. Residue List Format for Distance Pairs

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

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

   residue number1 residue number2

b. **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 JEDi will crash with an error message stating that a requested residue could not be found.

C. Debugging Crashes: Fixing and Preventing Problems:

i. Simple mistakes:

a) General:

1) Does every **KEY** have its **VALUE**?
2) Are the **doREAD** and **doMULTI** parameters set correctly?
3) Are you requesting to read PDBs when doing an analytical run?
4) Are you really working with multi-chain PDB files?
5) Does the specified path to the input file exist?
6) Is the input file actually in the specified location?
7) Is the **number format** correct? (20.0 is NOT an integer)
8) Does the reference PDB file correspond to the trajectory?

b) Pre-Processing Runs:

1) Does the working directory exist?
2) Does the working directory end with the file separator character ("**/**" or "****")?
3) Does the working directory contain PDB files of different sizes?
4) Does the working directory contain the reference PDB file?
5) Does the working directory contain the residue list files?

c) Production Runs:

1) Does the working directory contain the residue list files?
2) Are you requesting residues that are not in the reference PDB?
3) Are you requesting atoms that are not in the reference PDB?
4) Are you requesting analyses that exceed the available compute resources?

ii. More subtle issues:

- PDB files in non-standard format
- PDB files with fractional occupancy data
- PDB files with 2 chains, but no chain IDs or 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. JEDi 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.

Did you request more PCA modes than actually exist?

- For example (cPCA or dPCA):

  If your residue subset contains 12 atoms and you ask for 50 modes, then you are going get error messages: ***Because there are only 36 modes available.***

- 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 available.***

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

iii. Very subtle issues:

If your trajectory has not equilibrated, then you must address the problems of under-sampling and outliers. If you do not, then the covariance matrix will be highly ill-conditioned and may cause the eigenvalue decomposition to fail. Even if the EVD does not fail, your results will be highly suspect. It is advisable to examine the JEDi log file for the **rank** of the covariance matrix. If Q is not full rank, you may need to provide additional samples for analyzing your chosen subsets.

If these types of problems arise, we suggest that you check the variables in your data using a statistics package that computes the Kaiser-Myer-Olkin (KMO) statistic and the Measure of Sampling Adequacy (MSA) for each coordinate variable to critically assess your data. If the data is not well suited for PCA and additional samples are not available, you can condition the variables by setting the z-cutoff in JEDi 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 number of Q 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 data and reveal how many variables are independent, conditionally independent, and dependent.

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

D. Performing Mode Visualization

To activate the Essential PCA mode visualization, you need to:

- set **NUMBER_OF_MODES_VIZ > 0**

To activate the individual mode visualization, you need to:

- set **doModeViz = true**

To adjust the amplitude of the mode displacements, you need to:

- set the **mode amplitude** (default value = 0.25)

Recommended values for the mode amplitude are **near 0.25** depending on the system. JEDi uses a linear scaling in assigning the amount of mode displacement that is directly proportional to the number of atoms.

Note: Setting the **NUMBER_OF_MODES_VIZ** parameter to **zero** turns off the Essential Mode visualization feature.
Note: Setting the **doModeViz** parameter to **false** turns off the individual mode visualization feature.

The VIZ outputs include all the structures for the Essential Mode visualization and the top individual modes chosen for visualization. JEDi will permute the reference structure for a given subset along the top eigenvectors (using a sine function) selected for visualization and output 10 structures (PDBs) that capture one cycle of this motion. Additionally, JEDi will also compute the superposition of the top modes deemed 'essential' (up to five).

The amplitude of the motion is determined mostly by the value of the **mode_amplitude**, whose default value is **0.25**, 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 0.25 and 0.50 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 JEDi results tree:
`/working/directory/JEDi_Results_Description/VIZ/type_of_PCA/Model_of_PCA`

VI. Additional Types of Analysis

A. Pooling Data:

It is often useful to pool trajectory statistics. This can be done in JEDi by combining coordinate files and then performing the usual analysis. To combine the coordinate files, there is a utility program called **POOL_Driver.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.

i. Run Command:

```
java -d64 Pool_Driver.java "/path/to/POOL.txt"
java -jar -d64 Pool_Driver.jar "/path/to/POOL.txt"
```

ii. 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.

----

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 for job 1 is: **Pooled Coordinates Matrix $description1 $*number_of_input_matrices*.txt**

(`Pooled_Coordinates_Matrix_$description_3.txt` for the example)**.**

B. Subspace Analysis:

Once JEDi 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. JEDi 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 JEDi 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.

i. 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"
```

ii. Input File format:

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

The format for SSA.txt is shown below:

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/directory/
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
****************************************************************************************************
--------------------------------------------------------------------------------
```

C. Free Energy Surface:

JEDi contains a program called FES_Driver.java that takes two DVPs as order parameters (OP) to calculate free energy (FE) using a 2-D kernel density estimate (KDE) derived from Gaussian kernels. The output file is three columns: OP1, OP2, FE. This output can be used to plot a free energy surface with respect to the selected PCA modes. The FES can be done for both cPCA and dpPCA, and for each PCA model. The user specifies the OPs, the number of conformations to use, the offset into the points (which determines the points to use in the KDE), and the size of the 2D grid elements for making the KDE. All required information is specified in the input file FES.txt

i. Run Commands:

```
java -jar -d64 FES_Driver.jar “/path/to/FES.txt”

java -d64 FES_Driver.java “/path/to/FES.txt”
```

ii. Input File format:

The format for FES.txt is shown below:

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 "\")
$delta_vectors (string)
$OP1 (int) $OP2 (int) $num_points (int) $offset (int) $cellsize (double)
Separator Line ***

<u>Sample "FES.txt":</u>

```
--------------------------------------------------------------------------------------------------
2
/output/directory/
Test_FES_Pooled_Data
****************************************************************************************************
Test_First_Half
/working/directory1/
ss_151_top_2_DVPs_CORR.txt
0     1     1001     0     0.00
****************************************************************************************************
Test_Second_Half
/working/directory1/
ss_151_top_2_DVPs_CORR.txt
0     1     1000     1000     0.00
****************************************************************************************************
--------------------------------------------------------------------------------------------------
```

<u>Notes</u>:
Recommended values for cellsize is [0.01, 0.05]
Set $size = 0 to determine automatically, using the Max (Range X, Range Y)/256

D. Kernel PCA:

JEDi contains a program called KPCA_Driver.java that takes as input the top PCs or DVPs from the PCA analyses. This program runs a new PCA analysis using a selection of kernels on the dimension-reduced generalized coordinates. The user selects the number of inputs, the kernels to use, and the number of output kernel PCs.
All required information is specified as **KEY=VALUE** pairs in the in the input file: KPCA_Parameters.txt

i. Run Commands:

```
java -jar -d64 KPCA_Driver.jar /path/to/KPCA_Parameters.txt

java -d64 KPCA_Driver.java /path/to/KPCA_Parameters.txt
```

ii. Input File format:

The format for KPCA_Parameters.txt is shown below:

```
-------------------------------------------------------------------------------------
DIRECTORY=C:\Users\workspace\JEDi\test\BL\JEDi_RESULTS_TEST_BL\Alpha_Carbon_PCA\COV\
OUT_DIRECTORY=C:\Users\workspace\JEDi\test\BL\KPCA\
DESCRIPTION=TEST_KPCA
PROJECTIONS=ss_263_top_5_normed_DVPs_Alpha_Carbon_PCA.txt
NUMBER_KPCs=3
do_linear_kernel=false
do_Degree_2_Poly=false
do_Degree_3_Poly=false
do_Degree_4_Poly=false
do_XY_Poly=false
do_Poly_Diff_Sq=false
do_Poly_Diff_Cubes=false
do_Euclidean=false
do_Gaussian=true
do_Neural_Net=true
do_MI=true
do_MI_KDE=true
do_SinCos=false
SLOPE=100
SIGMA=1.000
-------------------------------------------------------------------------------------
```

Notes:
Set SLOPE and SIGMA according to the variance in the data.
Make sure that the output directory exists prior to running the program.

iii. Output

The output includes the kernel PCs and scatterplots of the top two KPCs.

E. Essential Mode Visualization:

JEDi contains a program called VIZ_Driver.java that allows the user to view not only individual modes, but the superposition of all the modes deemed 'essential'. A starting mode is specified so that the user can window over sets of modes. The number of modes to visualize determines the size of that window. Also, the user can specify a threshold (low/high) that can be adjusted to enhance the log coloring scheme. With thresholding, a percentage of frames are set to the minimum and maximum values, broadening the coloring of the inter-threshold range. The mode amplitude parameter determines the amount of displacement from equilibrium for visualizing the modes. Selecting the number of frames and number of cycles controls the level of detail captured in the movies.
All required information is specified in the input file VIZ.txt

i. Run Commands:

```
java -jar -d64 VIZ_Driver.jar "/path/to/VIZ.txt"
java -d64 VIZ_Driver.java "/path/to/VIZ.txt"
```

ii. Input File format:

The format for VIZ.txt is shown below:

LINE 1: Number_of_Jobs (integer)
LINE 2: Separator Line **
THEN FOR EACH JOB:
$start_mode $num_modes_viz $mode_amplitutude $threshold_low $threshold_high $num_frames $num_cycles $do_individual $type

$ref_PDB

$eigenvalues

$eigenvectors

$sq_modes

$mode_maxes

$mode_mins

$out_dir (string ending in "/" or "\\")

Separator Line **

Sample "VIZ.txt":

```
--------------------------------------------------------------------------------
2
*******************************************************************************************************
1     1     3.0     .05     .05     100     5     0     PCORR
/path/to/ss_151_RMSF_edited.pdb
/path/to/ss_151_eigenvalues_PCORR.txt
/path/to/ss_151_top_20_eigenvectors_PCORR.txt
/path/to/ss_151_top_20_square_pca_modes_PCORR.txt
/path/to/ss_151_top_20_square_pca_mode_MAXES_PCORR.txt
/path/to/ss_151_top_20_square_pca_mode_MINS_PCORR.txt
/output/directory/VIZ_TEST_PCORR/
*******************************************************************************************************
1     1     3.0     .05     .05     100     5     0     CORR
/path/to/ss_151_RMSF_edited.pdb
/path/to/ss_151_eigenvalues_CORR.txt
/path/to/ss_151_top_20_eigenvectors_CORR.txt
/path/to/ss_151_top_20_square_pca_modes_CORR.txt
/path/to/ss_151_top_20_square_pca_mode_MAXES_CORR.txt
/path/to/ss_151_top_20_square_pca_mode_MINS_CORR.txt
/output/directory/VIZ_TEST_CORR/
*******************************************************************************************************
--------------------------------------------------------------------------------
```

iii. Output:

The output is a set of **$num_frames** PDBs for the individual modes, if requested by setting **doModeViz** to **true**.

A set of 30 PDBs for the Essential Modes

A PyMol^{TM} script is generated to animate each set of PDB files (.pml files).

Make sure that the output directory exists prior to running the program.

VII. Appendix 1 - Input File Formats

A. JEDi_Parameters.txt (example for pre processing run on single chain PDB files)

```
# ---------------------------------------------------------------------------------------------------------------------
#     DIRECTORY, DESCRIPTION, REFERENCE_PDB
# ---------------------------------------------------------------------------------------------------------------------
DIRECTORY=working/directory/
DESCRIPTION=description
REFERENCE_PDB=pdb_file.pdb
# ---------------------------------------------------------------------------------------------------------------------
#     LOGICAL SWITCHES
# ---------------------------------------------------------------------------------------------------------------------
doREAD=true
doMULTI=false
doCARTESIAN=true
doDISPLACEMENT=true
doCORR=true
doPCORR=false
doREDUCE=false
doKPCA=false
doFES=false
doModeViz=false
doPLOT=true
# ---------------------------------------------------------------------------------------------------------------------
#                                           NUMBER OF MODES
# ---------------------------------------------------------------------------------------------------------------------
NUMBER_OF_MODES_LOCAL=5
NUMBER_OF_MODES_HIERARCHCAL_AA=9
NUMBER_OF_MODES_HIERARCHCAL_HA=6
NUMBER_OF_MODES_ALL_ATOM=5
NUMBER_OF_MODES_HEAVY_ATOM=5
NUMBER_OF_MODES_BACKBONE=5
NUMBER_OF_MODES_ALPHA_CARBON=5
NUMBER_OF_MODES_DISTANCE_PAIRS=5
NUMBER_OF_MODES_VIZ=5
# ---------------------------------------------------------------------------------------------------------------------
#      PCA and Visualization SETTINGS
# ---------------------------------------------------------------------------------------------------------------------
Z_SCORE_CUTOFF=3.000
STABILITY_THRESHOLD=1.000E-12
VIZ_MODE_SCALE_FACTOR=0.250
# ---------------------------------------------------------------------------------------------------------------------
#     RESIDUE LISTS FOR SUBSETS
# ---------------------------------------------------------------------------------------------------------------------
RESIDUE_LIST_ALL_ATOM=residues.txt
RESIDUE_LIST_HEAVY_ATOM=residues.txt
RESIDUE_LIST_BACKBONE=residues.txt
RESIDUE_LIST_ALPHA_CARBON=residues.txt
RESIDUE_LIST_LOCAL=residues.txt
RESIDUE_LIST_HIERARCHICAL_AA=residues.txt
RESIDUE_LIST_HIERARCHICAL_HA=residues.txt
ATOM_PAIRS_LIST=atom_pairs.txt
# ---------------------------------------------------------------------------------------------------------------------
#     COORDS FILE
# ---------------------------------------------------------------------------------------------------------------------
ORIGINAL_PDB_COORDS=original_PDB_Coordinates_AA.txt
# ---------------------------------------------------------------------------------------------------------------------
```

B. Residue List file for Single Chain PDBs

```
----------------------------------------------------------------------------------------------------
1
2
3
7
8
9
10
23
24
25
----------------------------------------------------------------------------------------------------
```

C. Residue List file for 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
----------------------------------------------------------------------------------------------------
```

D. Atom Pair List file for Single Chain PDBs

```
----------------------------------------------------------------------------------------------------
2             27
5             32
12            57
18            80
36            101
----------------------------------------------------------------------------------------------------
```

E. Atom Pair List file for Multi Chain PDBs

```
----------------------------------------------------------------------------------------------------
A             2             A             27
A             5             A             32
A             12            B             57
B             18            B             80
B             36            B             101
----------------------------------------------------------------------------------------------------
```

F. POOL.txt

```
------------------------------------------------------------------------------------------------------
$num_jobs --> (repeat job declaration $num_jobs times)
******************************************************************************************
$num_of_matrices_to_combine
$description
$output_directory
$path_to_coords_matrix1 --> (repeat line $num_of_matrices_to_combine times)
******************************************************************************************
---------------------------------------------------------------------------------------------------------
```

G. 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
******************************************************************************************
---------------------------------------------------------------------------------------------------------
```

H. FES.txt

```
---------------------------------------------------------------------------------------------------------
$num_jobs --> (repeat job declaration $num_jobs times)
$out_dir
$batch_description
***********************************************************
$job_description
$directory
$delta_vectors
$op1     $op2     $num_points     $offset     $cell_size
***********************************************************
---------------------------------------------------------------------------------------------------------
```

I. KPCA_Parameters.txt

```
----------------------------------------------------------------------------------------------------
DIRECTORY=C:\Users\workspace\JEDi\test\BL\JEDi_RESULTS_TEST_BL\Alpha_Carbon_PCA\COV\
OUT_DIRECTORY=C:\Users\workspace\JEDi\test\BL\KPCA\
DESCRIPTION=TEST_KPCA
PROJECTIONS=ss_263_top_5_normed_DVPs_Alpha_Carbon_PCA.txt
NUMBER_KPCs=3
do_linear_kernel=false
do_Degree_2_Poly=false
do_Degree_3_Poly=false
do_Degree_4_Poly=false
do_XY_Poly=false
do_Poly_Diff_Sq=false
do_Poly_Diff_Cubes=false
do_Euclidean=false
do_Gaussian=true
do_Neural_Net=true
do_MI=true
do_MI_KDE=true
do_SinCos=false
SLOPE=100
SIGMA=1.000
-----------------------------------------------------------------------------------------------------
```

J. VIZ.txt

```
-----------------------------------------------------------------------------------------------------------

$num_jobs --> (repeat job declaration $num_jobs times)
**********************************************************************************************************
$start_mode   $num_modes_viz   $mode_amplitutude   $threshold_low   $threshold_high   $num_frames   $num_cycles   $do_individual   $type

$ref_PDB
$eigenvalues
$eigenvectors
$sq_modes
$mode_maxes
$mode_mins
$out_dir
**********************************************************************************************************
---------------------------------------------------------------------------------------------------------
```

VIII. Appendix 2 - Output File Formats:

A. Sample JEDi Log file:

```
---------------------------------------------------------------------------------------------------
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
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.
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
----------------------------------------------------------------------------------------------------
```

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 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
----------------------------------------------------------------------------------------
```

D. 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
----------------------------------------------------------------------------------------------------
```

IX. Appendix 3 - For the Impatient...

A. All you need to do a JEDi Pre-Processing Run is a working directory with the following:

- **The JEDi_Parameters.txt** file
- The **reference PDB** file
- All the **PDB files** to be processed

B. All you need to do a JEDi Production Run is a working directory with the following:

- **The JEDi_Parameters.txt** file
- The **reference PDB** file
- The **residue lists**
- The **coordinates file**.

  ➢ You get the all atom coordinates by doing the pre-processing run on your PDB files.

Notes on the parameters file:

- **Please look at the file 'JEDi_Parameters.dat' to learn what the parameters control**
  - **NO SPACES in the KEY=VALUE declarations.**
  - **DO NOT remove or add any key value pairs, they need to be handled programmatically.**
  - **DO NOT have any blank values: no KEY= or JEDi will crash.**
  - **The format of the file NEVER changes, only the VALUES.**
  - **Lines starting with a hash (#) are ignored**

C. The JEDi Pre-Processing Workflow:

1. Define the working directory (with the PDB files), job description, and reference PDB file: For example,

    - DIRECTORY=C:\Users\workspace\JEDi\test\PDB\
    - DESCRIPTION=TEST_PP
    - REFERENCE_PDB=1ERM_TEM1.cleaned.pdb

2. Set the Logical switches for your job: The only relevant ones here are:

    - **doREAD=true**
    - doMULTI=false

- All other parameters are ignored (but need to be in the parameters file!)

D. The JEDi Production Workflow:

1. Define the working directory, job description, and reference PDB file: For example,

    - DIRECTORY=C:\Users\workspace\JEDi\test\
    - DESCRIPTION=TEST
    - REFERENCE_PDB=1ERM_TEM1.cleaned.pdb

2. Set the Logical switches for your job: For example,

    - **doREAD=false**
    - doMULTI=false
    - doCARTESIAN=true
    - doDISPLACEMENT=true
    - doCORR=true
    - doPCORR=false
    - doREDUCE=false
    - doKPCA=true
    - doFES=true
    - doModeViz=true
    - doPLOT=true

3. Choose the levels of coarse-graining by setting the number of modes greater than zero: For example,

   - NUMBER_OF_MODES_LOCAL=5
   - NUMBER_OF_MODES_HIERARCHICAL_AA=9
   - NUMBER_OF_MODES_HIERARCHICAL_HA=6
   - NUMBER_OF_MODES_ALL_ATOM=5
   - NUMBER_OF_MODES_HEAVY_ATOM=5
   - NUMBER_OF_MODES_BACKBONE=5
   - NUMBER_OF_MODES_ALPHA_CARBON=5
   - NUMBER_OF_MODES_DISTANCE_PAIRS=5
   - NUMBER_OF_MODES_VIZ=5

4. Specify the residue lists and/or atom list for the analysis you are requesting: For example,

   - RESIDUE_LIST_ALL_ATOM=residues5.txt
   - RESIDUE_LIST_HEAVY_ATOM=residues10.txt
   - RESIDUE_LIST_BACKBONE=residues25.txt
   - RESIDUE_LIST_ALPHA_CARBON=residues100.txt
   - RESIDUE_LIST_LOCAL=residues5.txt
   - RESIDUE_LIST_HIERARCHICAL_AA=residues100.txt
   - RESIDUE_LIST_HIERARCHICAL_HA=residues100.txt
   - ATOM_PAIRS_LIST=atom_pairs.txt

5. Specify the PCA and VIZ parameters: For example,

   - Z_SCORE_CUTOFF=3.000
   - STABILITY_THRESHOLD=1.000E-12
   - VIZ_MODE_SCALE_FACTOR=0.250

6. Specify the all-atom PDB coordinates file: For example,

   - ORIGINAL_PDB_COORDS=original_PDB_Coordinates_AA.txt
