

## 1 Interactive essential dynamics

2 John Mongan

3 *Bioinformatics Program; Medical Scientist Training Program; NSF Center for Theoretical Biological*  
4 *Physics, University of California at San Diego, La Jolla, CA 92093-0365, USA (jmongan@mccam-*  
5 *mon.ucsd.edu)*

6 Received 13 May 2004; accepted in revised form 28 September 2004

7 *Key words:* essential dynamics, graphical user interface, interactive, molecular dynamics, principal com-  
8 ponent analysis, visualization

### 9 Summary

10 Essential dynamics(ED) is a useful method for analyzing trajectories generated by molecular dynamics  
11 (MD), but current tools are awkward to use, limiting the usefulness of the technique. This paper describes a  
12 new interactive graphical interface for visualization of ED results, including filtering a trajectory on an  
13 arbitrary set of eigenvectors and manipulation of a structure's projection along any eigenvector.

14 *Abbreviations:* ED – essential dynamics; IED – Interactive Essential Dynamics; MD – molecular  
15 dynamics; VMD – Visual Molecular Dynamics

### 17 Introduction

18 Trajectories generated from molecular dynamics  
19 (MD) simulations provide a means to identify and  
20 study motions crucial for protein function [1].  
21 Separating functionally important motions from  
22 random thermal fluctuations is a major challenge  
23 in analyzing MD trajectories. Principal component  
24 analysis of MD trajectory data, often called  
25 *essential dynamics* (ED) [2, 3], is frequently used to  
26 separate large-scale correlated motions from local  
27 harmonic fluctuations [4–9].

28 ED analysis constructs a new orthogonal basis  
29 set for the atomic coordinates in a trajectory, such  
30 that the greatest variance occurs along the first  
31 vector, with monotonically decreasing variance  
32 along successive vectors. These vectors are often  
33 called principal components or eigenvectors, since  
34 their derivation involves an eigen decomposition.  
35 The eigenvalues from the eigen decomposition  
36 represent the relative amount of molecular motion  
37 that occurs along each eigenvector. The eigenvalue  
38 spectrum is sharply peaked for molecular trajectory

data, indicating that most of the molecular motion 39  
can be described by displacements along the first few 40  
eigenvectors [2, 4–6, 9]. A trajectory can be 41  
projected onto a subset of selected eigenvectors so 42  
only motion along the selected vectors is allowed. 43  
The most commonly selected subset is the first  $n$  44  
eigenvectors such that a given percentage of the 45  
molecular motion occurs within the subspace 46  
formed by the selected eigenvectors. Projection onto 47  
these vectors filters out thermal noise, making the 48  
functionally interesting motions easier to appreci- 49  
ate. Smaller subsets may be selected to isolate a 50  
particular aspect of the molecule's motion. One can 51  
also examine the functional meaning of a single 52  
eigenvector by generating a trajectory with atomic 53  
positions interpolated between extreme projections 54  
on the selected eigenvector. 55

ED is a standard method of analysis that is 56  
widely implemented in molecular simulation 57  
packages [10–13]. Tools in these packages take 58  
trajectory and eigenvector files as input and pro- 59  
duce a new trajectory as output, which must be 60  
loaded into an integrated [11] or external [10, 12,13] 61

viewer. A more flexible approach [14], implemented within a limited viewer, is not widely available. In the available tools, a separate trajectory file of interpolations between extreme projections must be generated for each eigenvector, and a separate filtered trajectory file must be generated for each set of eigenvectors selected for filtering. Some tools [11, 12] are limited to filtering along a single eigenvector at a time, which may be problematic since rotational motion cannot be adequately represented with a single eigenvector.

Generating and loading a separate trajectory file for each aspect of the ED results is cumbersome and discourages complete understanding of the ED analysis. Interactive essential dynamics (IED) is a new program that addresses these problems, providing fully interactive analysis of ED results through a graphical interface. Filtering eigenvectors can be rapidly added or removed from within the viewer, even while the trajectory is being played. The functional meaning of an eigenvector can be examined from within the viewer by dragging the atomic positions along the eigenvector using a slider control. Arrows representing an atom's motion along an eigenvector can be drawn to provide a static representation of an eigenvector, as in the work of Huitema and van Liere. [14] IED can calculate eigenvectors and projections directly, or read the results of calculations performed in GROMACS [10] or the ptraj module of AMBER 8 [13]. IED also allows sets of vectors that do not have accompanying projections to be loaded so results of normal modes analysis performed by AMBER or GROMACS can be visualized. The Python scripting interface of visual molecular dynamics (VMD) [15] is used for display. The extensive visualization, animation, rendering and analysis capabilities of VMD remain available while using IED.

## 101 Theory and methods

102 To perform ED, coordinate data from each time-  
103 step is fitted to a reference structure to remove  
104 translational and rotational motion. The fitted  
105 trajectory data are used to construct a covariance  
106 matrix  $C$  according to equation 1

$$C = \langle (x - \langle x \rangle)(x - \langle x \rangle)^T \rangle \quad (1)$$

where  $\langle \rangle$  represents the mean across all timesteps, and the T superscript represents transpose. An eigen decomposition (or diagonalization) of the symmetric matrix  $C$  is performed to identify  $\Lambda$ , a diagonal matrix of eigenvalues and  $T$ , a matrix of column eigenvectors forming a new orthonormal basis set [2], satisfying

$$C = T \Lambda T^T \quad (2)$$

A zero-mean trajectory matrix,  $X$ , can be constructed by subtracting  $\langle x \rangle$  from the coordinate vector for each timestep to form the rows of  $X$ . The matrix of the projections of each timestep onto each eigenvector,  $P$ , is obtained by multiplying the trajectory matrix,  $X$  by  $T$

$$P = X T \quad (3)$$

For use with IED, these calculations may be performed using the AMBER or GROMACS suites. IED is also capable of performing these calculations itself, but is less efficient than AMBER or GROMACS.

The trajectory matrix,  $X$ , can be reconstructed from the eigenvectors and projection matrices  $T$  and  $P$ , by right multiplying Equation 3 by  $T^T$

$$P T^T = X T T^T = X I = X \quad (4)$$

where  $I$  is an identity matrix. More usefully, a matrix of filtered trajectory data,  $F$ , can be calculated by multiplying a subset of the (column) projection vectors in  $P$  by the corresponding subset of the eigenvectors in  $T^T$ . This way  $F$  contains only motions that occur along the eigenvectors selected from  $P$  and  $T$ , since motions along other eigenvectors are represented by projections omitted from the calculation of  $F$ . IED employs this method to calculate filtered trajectories, adding  $\langle x \rangle$  to the coordinate vector in each row of  $F$  to translate the coordinates back to their original origins. When a single eigenvector is to be examined by interpolation between extreme projections, coordinates are calculated by varying the (scalar) projection value for the selected eigenvector at the current time step and recalculating the appropriate row from  $F$  for each value of the projection.

When IED calculates ED directly, it can operate on trajectory data in any format that VMD is able to load. When loading results of ED analysis carried out in GROMACS, it requires a molecular topology file (in any VMD acceptable format), an

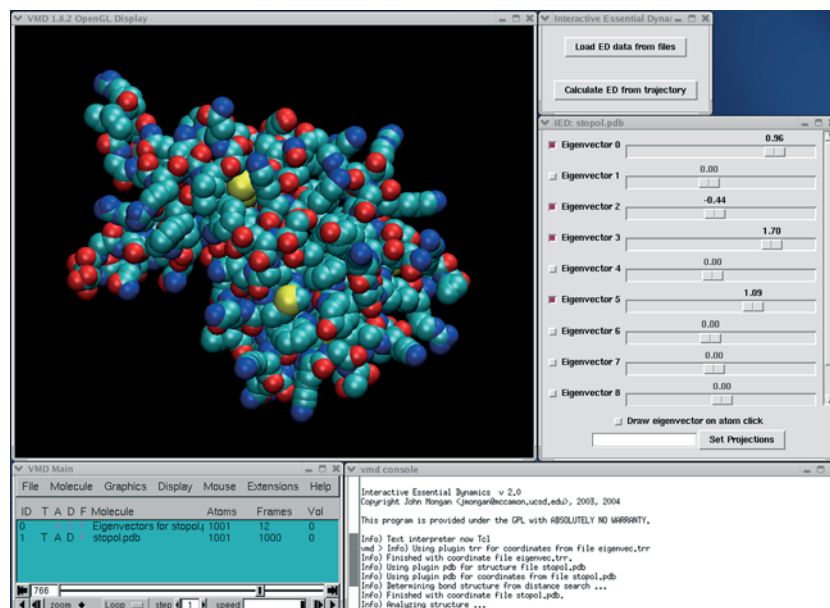


Figure 1. Screen shot of IED. Top right window is the main IED window, window immediately below contains the checkboxes and sliders for selecting eigenvectors and manipulating projections. Remaining windows are VMD windows: main window and animation controls at bottom left, console at bottom right and molecular display at top left.

155 eigenvectors file in GROMACS TRR binary format generated by `g_covar`, and a projections file  
 156 generated by `g_anaeig`. The first timestep of the  
 157 eigenvectors file is ignored, the second contains the  
 158 molecules average coordinates over the trajectory  
 159 and the remaining timesteps contain eigenvectors  
 160 in decreasing order of their eigenvalues. The pro-  
 161 jections file is formatted as text input to Grace, a  
 162 plotting tool. Each eigenvector has a separate  
 163 block of projection data within the file; within each  
 164 block there is one projection per line, consisting of  
 165 a time value followed by a projection value, sepa-  
 166 rated by whitespace. When loading ED results  
 167 from AMBER, the requirements are similar: a  
 168 topology file, an eigenvectors file and a projections  
 169 file. The eigenvectors file and projections file are  
 170 both produced by `ptraj` and are in text format. The  
 171 eigenvectors file contains two header lines, which  
 172 are ignored, the average coordinates, followed by  
 173 the eigenvectors. Each eigenvector has a two line  
 174 header consisting of a line containing 4 asterisks  
 175 (\*\*\*\*), followed by a line giving the ordinal  
 176 number of the eigenvector and its eigenvalue.  
 177 Numeric data for the average coordinates and ei-  
 178 genvectors are whitespace delimited, with 7 values  
 179 per line. The projections file has a two line header  
 180 which is ignored. Each successive line contains a

timestep number, followed by projection values  
 onto each eigenvector for that timestep. The values  
 are whitespace delimited.

Internally, IED represents the eigenvector data  
 in a VMD trajectory object and the projection  
 data in a Python Numeric array object. IED is an  
 open source application, and is easily extended to  
 other file formats by writing parsing routines to  
 read data into the aforementioned data structures.

## User interface

192 IED is started either by selecting a trajectory in  
 193 VMD for ED analysis, or by loading files con-  
 194 taining the results of an ED analysis previously  
 195 performed in `ptraj` or GROMACS. Once the ED  
 196 data are loaded, a window is displayed with a  
 197 checkbox and slider for each eigenvector (see  
 198 Figure 1). When necessary, the eigenvector slider  
 199 area of the window can be scrolled to allow for  
 200 arbitrarily large numbers of eigenvectors. Selecting  
 201 a checkbox allows motion along the corresponding  
 202 eigenvector and activates the eigenvector's slider,  
 203 setting its position to the projection on the eigen-  
 204 vector for the current frame of the trajectory.  
 205 Check boxes can be selected independently,

206 allowing simultaneous analysis of any combina-  
 207 tion of eigenvectors. When the VMD animation  
 208 controls are used to play the trajectory, the  
 209 molecular display shows the filtered trajectory: the  
 210 projection of the trajectory on the currently se-  
 211 lected eigenvectors. Slider positions corresponding  
 212 to selected eigenvectors are updated as each frame  
 213 of the trajectory is displayed. The movement of the  
 214 sliders provides an animated, graphical represen-  
 215 tation of the projection of the trajectory on each  
 216 eigenvector. When the animation is stopped, the  
 217 sliders for any selected eigenvector can be moved  
 218 manually, which temporarily changes the projec-  
 219 tion value on the eigenvector for the displayed  
 220 frame. The molecular display is updated as the  
 221 slider is moved, making it easy to appreciate any  
 222 eigenvector's contribution to the molecular mo-  
 223 tion. A comma delimited list of projections can be  
 224 entered in the text box near the bottom of the  
 225 window to rapidly set the projections along all  
 226 eigenvectors.

227 Interactive manipulation of the molecule's  
 228 projection along an eigenvector provides the  
 229 clearest visualization of the eigenvector, but is not  
 230 possible in cases where a static image is required  
 231 for publication or presentation. Static visualiza-  
 232 tions can be produced by selecting a single eigen-  
 233 vector and clicking on representative atoms. An  
 234 arrow is drawn through the clicked atoms, with the  
 235 arrow's head representing the atom's position at  
 236 the most positive projection and the tail repre-  
 237 senting the most negative projection.

238 When IED is used to visualize normal modes  
 239 data, there is no associated trajectory, and no  
 240 projections file is loaded. In this case trajectory  
 241 playing and filtering features are disabled, but all  
 242 other features are available.

## 243 Conclusions

244 IED allows interactive visualization and manipu-  
 245 lation of projections of protein motion on selected  
 246 eigenvectors and easy selection and filtering on  
 247 different discontinuous sets of eigenvectors. It in-  
 248 creases efficiency in working with ED results and  
 249 enables appreciation of aspects of the dynamics  
 250 that might be missed with more limited tools.

251 IED is freely available under the Gnu Public  
 252 License (GPL) at <http://mccammon.ucsd.edu/>

software. html. The language, applications and  
 libraries on which it depends are also freely  
 available.

## Acknowledgements

I thank Prof. J. Andrew McCammon for helpful  
 suggestions and Dr. Justin Gullingsrud for advice  
 on VMD and Python. I thank the Taft family for  
 their support through the Taft Fellowship, and  
 The Burroughs Wellcome Fund for their support  
 through the La Jolla Interfaces in Science Pre-  
 doctoral Fellowship. This work has been sup-  
 ported in part by grants from NSF, NIH, the NSF  
 Center for Theoretical Biological Physics, the  
 National Biomedical Computing Resource, and  
 Accelrys, Inc.

## References

- Shen, T., Tai, K. Henchman, R.H. and Mc Cammon J.A., *Acc.*  
*Chem. Res.* 35(6) (2002) 332. 269 270
- Amadei, A., Linssen, A.B.M. and Berendsen, H.J.C., *Proteins:*  
*Struct. Funct. Genet.* 17(4) (1993) 412. 271 272
- Garcia, A.E., *Phys.Rev.Lett.* 68(17) (1992), 2696. 273
- Hamelberg, D., Mongan, J. and McCammon, J.A., *J. Chem.*  
*Phys.* 120(24), (2004) 11919. 274 275
- de Groot, B.L., Daura, X., Mark, A.E. and Grubmuller, H.,  
*J. Mol. Biol.* 309(1) (2001) 299. 276 277
- Xiong, B., Huang, X.Q., Shen, L.L., Shen, J.H., Luo, X.M.,  
 Shen, X., Jiang, H.L. and Chen, K.X., *Acta Pharmacol.*  
*Sin.* 25(6) (2004) 705. 278 279 280
- Lee, J., Suh, S.W. and Shin, S., *J. Biomol. Struct. Dyn.* 18(2)  
 (2000) 297. 281 282
- Crabbe, M.J., Cooper, L.R. and Corne, D.W., *Comput. Biol.*  
*Chem.* 27(4-5) (2003) 507. 283 284
- Arcangeli, C., Bizzarri, A.R. and Cannistraro, S., *Biophys.*  
*Chem.* 90(1) (2001) 45. 285 286
- Lindahl, E., Hess, B. and van der Spoel, D., *J. Mol. Model.* 7(8)  
 (2001) 306. 287 288
- Vriend, G., *J. Mol. Graph.* 8 (1990) 52. 289
- Kendall, R.A., Apra, E., Bernholdt, D.E., Bylaska, E.J.,  
 Dupuis, M., Fann, G.I., Harrison, R.J., Ju, J., Nichols,  
 J.A., Nieplocha, J., Straatsma, T.P., Windus, T.L. and  
 Wong, A.T., *Comput. Phy. Commun.* 128 (2000) 260. 290 291 292 293
- Case, D., Darden, T., Cheatham, I.T.E., Simmerling, C., Wang,  
 J., Duke, R., Luo, R., Merz, K., Wang, B., Pearlman, D.,  
 Crowley, M., Brozell, S., Tsui, V., Gohlke, H., Mongan,  
 J., Hornak, V., Cui, G., Beroza, P., Schafmeister, C.,  
 Caldwell, J., Ross, W. and Kollman, P., *AMBER 8,*  
 (2004). 294 295 296 297 298 299
- Huitema, H. and van Liere, R., *IEEE Visualization 2000*  
*Proceedings 11,* (2000) 300 301
- Humphrey, W., Dalke, A. and Schulten, K., *J. Mol. Graph.*  
 14(1) (1996) 33. 302 303 304