

Structural bioinformatics

MEPSA: minimum energy pathway analysis for energy landscapes

Iñigo Marcos-Alcalde¹, Javier Setoain², Jesús I. Mendieta-Moreno^{1,3},
Jesús Mendieta^{1,4} and Paulino Gómez-Puertas^{1,*}

¹Molecular Modelling Group, CBMSO (CSIC-UAM), ES-28049 Madrid, Spain, ²Departamento de Arquitectura de Computadores y Automática, UCM, ES-28040 Madrid, Spain, ³Departamento de Física Teórica de la Materia Condensada and Condensed Matter Physics Center (IFIMAC), UAM and ⁴Biomol-Informatics SL, Campus UAM, ES-28049 Madrid, Spain

*To whom correspondence should be addressed.

Associate Editor: Anna Tramontano

Received on June 12, 2015; revised on July 20, 2015; accepted on July 24, 2015

Abstract

Summary: From conformational studies to atomistic descriptions of enzymatic reactions, potential and free energy landscapes can be used to describe biomolecular systems in detail. However, extracting the relevant data of complex 3D energy surfaces can sometimes be laborious. In this article, we present MEPSA (Minimum Energy Path Surface Analysis), a cross-platform user friendly tool for the analysis of energy landscapes from a transition state theory perspective. Some of its most relevant features are: identification of all the barriers and minima of the landscape at once, description of maxima edge profiles, detection of the lowest energy path connecting two minima and generation of transition state theory diagrams along these paths. In addition to a built-in plotting system, MEPSA can save most of the generated data into easily parseable text files, allowing more versatile uses of MEPSA's output such as the generation of molecular dynamics restraints from a calculated path.

Availability and implementation: MEPSA is freely available (under GPLv3 license) at: <http://bioweb.cbm.uam.es/software/MEPSA/>

Contact: pagomez@cbm.csic.es

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

The development of efficient conformational space sampling methodologies, coupled with a dramatic increase in the computational capacities and capabilities, has made the calculation of energy landscapes significantly more accessible (Bernardi *et al.*, 2015; Mendieta-Moreno *et al.*, 2014).

A 3D energy surface may provide a lot of detailed information about biomolecular processes such as protein or nucleotide folding (Hori *et al.*, 2009; Shcherbakova *et al.*, 2008), ligand binding (Bai *et al.*, 2013) or enzymatic reactions (Mendieta-Moreno *et al.*, 2015); but the extraction and analysis of such data are often cumbersome tasks. MEPSA (Minimum Energy Path Surface Analysis) provides a

GUI-based tool to analyse these landscapes from a transition state theory point of view, making the analysis of 3D energy landscapes agile.

2 MEPSA software

MEPSA is an open-source program written in Python (compatible with both Python 2.7.x and 3.4.x) that describes the connectivity of the minima (called nodes in MEPSA) present in a given energy landscape (called map in MEPSA). The graphic interface is built with TKinter (Shipman, 2010), plots are drawn using Matplotlib (Hunter, 2007) and NumPy (Van der Walt *et al.*, 2011) is employed for the calculations.

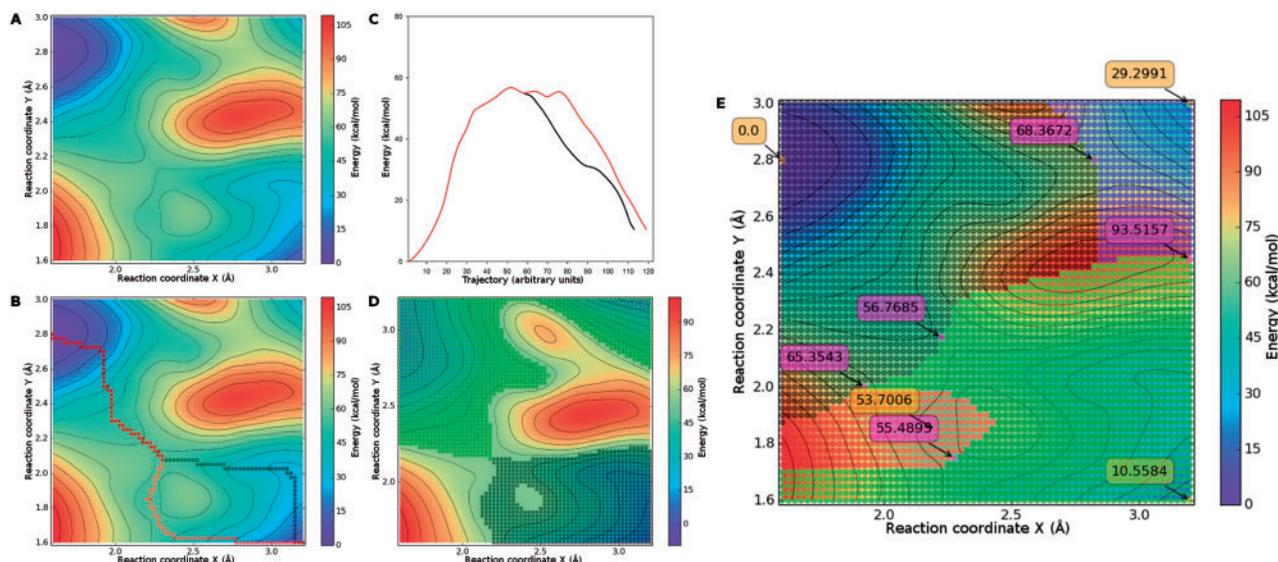


Fig. 1. Some results of an analysis with MEPSA. (A) Plot of the test map. (B) Comparison of two alternative paths connecting the origin [1.6, 2.8] and the target [3.2, 1.6] nodes. (C) Energy profile of both paths. (D) Well sampling analysis showing the points sampled to reach the barrier from the origin (green) and the target (black). (E) Global connectivity analysis output, highlighting the energy values of the points of interest (minima in orange and saddle points in purple)

MEPSA supports three column formatted plain text files as input and can save most of the generated data in column formatted plain text files as well. There is no restriction related to technique with which the map has been generated, apart from being rectangular and uniformly distributed (e.g. calculation of the minimum elevation path connecting Geneva and Turin is included in [Supplementary Material](#)). In addition, any plot obtained (e.g. Fig. 1) can be directly saved in many different image formats (png, eps, pdf...).

The GUI structure consists of one main window to load, unload and plot maps and two secondary windows to manage specific tasks. The ‘connectivity analyses’ window gives access to the ‘global connectivity analysis’, ‘path generator’ and ‘well sampling analysis’ tools; and the ‘map editor’ window gives access to the ‘modify map’, ‘smooth map’ and ‘invert map’ functionalities.

‘Global connectivity analysis’ simultaneously samples the whole map, starting from every node and iteratively occupying the lowest energy points available. The node where each particular propagation comes from is kept in memory in order to assign a domain to each node. As the lowest energy points in the borders between domains are necessarily the barriers connecting the nodes, all the minima and barriers of the map are identified at once (Fig. 1E).

‘Path generator’ detects the lowest energy path connecting two points (named origin and target). The resulting trajectory only depends on the points selected and not on the path direction. MEPSA offers two sampling modes: ‘global’ and ‘node by node’. ‘Global’ mode uses an approach similar to Dijkstra’s algorithm (Dijkstra, 1959), with small differences in the sampling and the trace back. The algorithm samples the map from the origin point, propagating to the lowest energy points available on each iteration, until the target point is reached. The iteration in which each node has been occupied is stored in memory and, after the target node is reached, a trace-back is performed from the target, iteratively selecting the points with lowest iteration counters. ‘Node by node’ mode uses a ‘global’ mode sampling to define the order in which the nodes are visited, then performing a series of runs connecting all the consecutive nodes in a pairwise fashion. This way the path is forced to pass through every node whose domain is crossed by the shortest lowest energy path. The paths generated with the ‘path generator’ can be

stored in memory using the ‘path stack’, enabling the comparison of several paths at once (Fig. 1B and C), even if those were generated using different maps.

‘Well sampling analysis’ determines the area of the map that has to be sampled from the origin node to reach the closest barrier to the target node and *vice versa* (Fig. 1D).

The data obtained can be plotted in several ways, the path trajectories can be smoothed and most of the data generated (even the ‘path stack’ as a whole) can be stored into text files to be plotted or analysed with other software (e.g. parsing path files in order to generate molecular dynamics restraints, see [Supplementary Material](#)).

In the ‘map editor window’, ‘modify map’ performs simple modifications of the energy values in a defined region of the map, which can be useful, e.g. to block favourable paths in order to evaluate alternative ones (Fig. 1B and C); ‘smooth map’ applies a simple running average smoothing to the map, allowing the user to remove unwanted local minima in noisy maps; and ‘invert map’ changes the sign of the energy values, enabling the characterization of maxima edge profiles (see [Supplementary Material](#)).

Acknowledgements

Support from the ‘Fundación Severo Ochoa’ and the ‘Centro de Computación Científica CCC-UAM’ is gratefully acknowledged.

Funding

Grant IPT2011-0964-900000 (Government of Spain). Work at Biomol-Informatics was financed by the European Social Fund.

Conflict of Interest: none declared.

References

- Bai, F. et al. (2013) Free energy landscape for the binding process of Huperzine A to acetylcholinesterase. *Proc. Natl. Acad. Sci. USA.*, **110**, 4273–4278.
- Bernardi, R.C. et al. (2015) Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochim. Biophys. Acta*, **1850**, 872–877.

- Dijkstra, E.W. (1959) A note on two problems in connection with graphs. *Numerische Mathematik*, **1**, 269–271.
- Hori, N. *et al.* (2009) Folding energy landscape and network dynamics of small globular proteins. *Proc. Natl. Acad. Sci. USA.*, **106**, 73–78.
- Hunter, J.D. (2007) Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.*, **9**, 22–30.
- Mendieta-Moreno, J.I. *et al.* (2014) FIREBALL/AMBER: an efficient local-orbital DFT QM/MM method for biomolecular systems. *J. Chem. Theor. Comput.*, **10**, 2185–2193.
- Mendieta-Moreno, J.I. *et al.* (2015) A practical quantum mechanics molecular mechanics method for the dynamical study of reactions in biomolecules. *Adv. Protein Chem. Struct. Biol.*, **100**, 67–88.
- Shcherbakova, I. *et al.* (2008) Energy barriers, pathways and dynamics during folding of large, multi-domain RNAs. *Curr. Opin. Chem. Biol.* **12**, 655–666.
- Shipman, J.W. (2010) *Tkinter Reference: a GUI for Python*. New Mexico Tech Computer Center, Socorro, New Mexico.
- Van der Walt, S. *et al.* (2011) The NumPy Array: a structure for efficient numerical computation. *Comput. Sci. Eng.*, **13**, 22–30.