TRITON: in silico construction of protein mutants and prediction of their activities

Martin Prokop, Jiří Damborský and Jaroslav Koča*

Laboratory of Biomolecular Structure and Dynamics, Masaryk University, Kotlarska 2, 611 37 Brno, Czech Republic

Received on July 7, 1999; revised on November 2, 1999; accepted on November 8, 1999

Abstract

Motivation: One of the objectives of protein engineering is to propose and construct modified proteins with improved activity for the substrate of interest. Systematic computational investigation of many protein variants requires the preparation and handling of a large number of data files. The type of the data generated during the modelling of protein variants and the estimation of their activities offers the possibility of process automatization.

Results: The graphical program TRITON has been developed for modelling protein mutants and assessment of their activities. Protein mutants are modelled from the wild type structure by homology modelling using the external program MODELLER. Chemical reactions taking place in the mutants active site are modelled using the semi-empirical quantum mechanic program MOPAC. Semi-quantitative predictions of mutants activities can be achieved by evaluating the changes in energies of the system and partial atomic charges of active site residues during the reaction. The program TRITON offers graphical tools for the preparation of the input data files, for calculation and for the analysis of the generated output data.

Availability: The program TRITON can run under operating systems IRIX, Linux and NetBSD. The software is available at http://www.chemi.muni.cz/lbsd/triton.html.

Contact: triton@chemi.muni.cz

Introduction

The construction of modified proteins with improved properties is the ultimate goal of many protein engineering projects. Proteins are usually engineered for higher stability, broader specificity or elevated catalytic activity. Computational methods are being developed with the aim of assisting during the design process (Hellenga, 1998). Available computer programs in the field of protein engineering are mainly focused on the process of substrate binding, i.e. serving for the engineering of substrate specificity (Wilson et al., 1991; Hellenga and Richards, 1991; Honda et al., 1996). However, we are not aware of any computational tool for the systematic engineering of the catalytic efficiency of proteins. The purpose of this project was to develop and evaluate a graphically oriented computer program for the modelling of single point and multiple mutants of a protein with subsequent analysis of their activities by calculation of the reaction pathway (Damborský et al., 1998).

Methods

Modelling of protein mutants

The 3D structures of mutants are generated by the MODELLER program (Sali and Blundell, 1993). This approach uses the method of the satisfaction of spacial restraints for model building with the structure from the structural database as a template and the amino acid sequence of the studied protein with the desired substitution as the target sequence.

Preparation of input data for reaction pathway calculation

Only the active site residues, substrate, co-substrate and co-factors are included in the calculation. All the other atoms of the enzyme are removed. The active site residues are kept in their original geometry, and hydrogen atoms are added onto the C and N terminals. In order to mimic the situation in the enzyme, the positions of all backbone atoms are fixed during the calculation.

Reaction pathway calculation

The reaction pathway is calculated using the semiempirical quantum chemistry program MOPAC (Stewart, 1990). The subroutine DRIVER (Cernohorsky et al., 1996) is used for reaction pathway mapping. The idea is that the proposed reaction coordinate is changed and fixed step by step, and all other internal coordinates are subsequently optimized. The modelling approach has been described in detail in the literature (Damborský et al., 1997; Kuty et al., 1998).
Computer program TRITON

Construction of protein mutants

The input data consists of the template structure of the enzyme and the information concerning which residues will be mutated. TRITON then generates all the possible combinations of mutated residues. For each of them, the input for MODELLER is generated and the 3D structure is calculated by the MODELLER. By default, each calculation is started automatically. A UNIX script is also generated for each mutant, so the calculation may be started by the user. If the calculation runs on a local machine then TRITON allows the user to monitor the calculations directly within the graphic environment. It is indicated which mutant is currently being calculated, which jobs are pending, which are completed successfully, and which jobs were not completed due to error. TRITON can also be used as the simple graphical interface of the program MODELLER—one of the most widely used pieces of software for homology modelling.

Calculation of reaction pathway

The input data is required only once for the entire series of mutants. It includes information about which atoms will be taken into account in the calculation, which atoms will be fixed, and which internal coordinate will serve as a reaction coordinate. All this information is entered by pointing at the corresponding atoms and clicking the mouse. Each mutant generated by homology modelling is processed independently in the following way. The 3D structure of the mutant is read into the program together with the template structure. All atoms of the mutant which will not be included in the calculation are removed, the remaining atoms of the mutant are collectively called cavity. Free valences are then saturated by hydrogens. The substrate is added to the cavity in such a way that it best fits the situation in the template enzyme/substrate system and can be manually adjusted further within TRITON. The initial input information which was given for the template system is now recalculated to fit the situation in the mutant. Since each mutant may have a different number of atoms than the template system, the atoms must often be renumbered and a new coordinate system must be created to meet all the requirements for constraining atoms, internal coordinates and the reaction coordinate. The input files include the internal coordinates of the system (*.dat) and also information about the starting and final values of the reaction coordinate (usually the distance between two atoms) together with the step by which the reaction coordinate will be changed (*.drv). An additional file (*.prj) is created in which further important information about the system is stored, for example the identification of the substrate molecule. Finally, a covering script is generated (*.sh) which runs all the scripts created for the single mutants. The entire procedure is fully automated and requires only very basic knowledge of the operating system. The entire calculation can be run directly from within the TRITON environment or by running scripts on a distant machine. If the entire enzymatic reaction is composed of several steps than each of them is modelled as described above. The last energy minimum from one reaction-step is typically set as an initial structure for the calculation of the next reaction-step.

Visualization of output data

TRITON enables us to display all the 3D structures formed along the reaction pathway together with the energy curve. The feature is extended by an animation option which allows the user to follow the changes in geometry during the reaction very simply. The energy profile is used to calculate the thermodynamics of the reaction and to estimate the activation energy. The graph of charge developments on selected atoms may also be displayed. After completion of all calculations within one project, a file containing a list of all activation energies may be generated.

Implementation

TRITON is written in C++ for the UNIX operating system. It is currently ported on IRIX, Linux and NetBSD. The X-window system is used with OpenGL libraries which are emulated by the Mesa library within Linux and NetBSD.

References


