Future Generation Computer Systems 67 (2017) 430–440

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Multi-GPU-based detection of protein cavities using critical points

Sérgio E.D. Dias [a,b,*], Quoc T. Nguyen [b], Joaquim A. Jorge [c], Abel J.P. Gomes [a,b]

[a] Instituto de Telecomunicações, Portugal
[b] Universidade da Beira Interior, Portugal
[c] INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

HIGHLIGHTS

- CriticalFinder is the first multi-GPU-based cavity detection algorithm.
- CriticalFinder is the first surface-based cavity detection algorithm that produces a meaningful, coarse cavity segmentation.
- CriticalFinder sustains on the theory of critical points (i.e., Morse theory).

ARTICLE INFO

Article history:
Received 14 August 2015
Received in revised form
16 May 2016
Accepted 19 July 2016
Available online 1 August 2016

Keywords:
Protein cavity detection
Protein pocket detection
Critical point
Molecular surface
CUDA

ABSTRACT

Protein cavities are specific regions on the protein surface where ligands (small molecules) may bind. Such cavities are putative binding sites of proteins for ligands. Usually, cavities correspond to voids, pockets, and depressions of molecular surfaces. The location of such cavities is important to better understand protein functions, as needed in, for example, structure-based drug design. This article introduces a geometric method to detecting cavities on the molecular surface based on the theory of critical points. The method, called CriticalFinder, differs from other surface-based methods found in the literature because it directly uses the curvature of the scalar field (or function) that represents the molecular surface, instead of evaluating the curvature of the Connolly function over the molecular surface. To evaluate the accuracy of CriticalFinder, we compare it to other seven geometric methods (i.e., LIGSITE^{CS}, GHECOM, ConCavity, POCASA, SURFNET, PASS, and Fpocket). The benchmark results show that CriticalFinder outperforms those methods in terms of accuracy. In addition, the performance analysis of the GPU implementation of CriticalFinder in terms of time consumption and memory space occupancy was carried out.

1. Introduction

Many biological processes in life sciences, in particular those involving drug interactions and protein docking, occur in water. The interaction between water and molecule can tell much information about the shape of a molecule, including the location of its binding sites. As Mezey noted in [1], this is of great importance to research in chemistry, biophysics, medicine, and nano-technology. A better interpretation and identification of such regions on a molecular surface can greatly help in discovering new drugs. Hence, the identification of those binding sites is often the first step in the study of protein functions, as in the structure-based drug design.

However, many small molecules (i.e., ligands) can bind to a given protein, depending on the number of binding sites on its molecular surface. It happens that, as noted by Henrich et al. [2], checking whether a certain molecule can bind to a particular protein takes a lot of time in lab. While, in general, binding sites correspond to concave, cleft or tunnel-shaped regions on a protein surface (cf. Kawabata and Go [3]), called pockets or cavities, not all cavities end up being binding sites for small ligands. Thus, detecting binding sites depends on efficient computational algorithms to locate all cavities on the molecular surface.

So, in this paper, we describe a method to identify the cavities on the protein surface as tentative binding sites for ligands. The novelty of the algorithm lies in *directly* evaluating the curvature of the scalar field (or function) that describes the molecular surface, instead of evaluating the curvature of the Connolly function [4] or the Mitchell–Kerr–Eyck function [5] over the molecular

* Corresponding author at: Universidade da Beira Interior, Portugal.
E-mail addresses: sergioduartedias@sapo.pt (S.E.D. Dias), quochcmvn@gmail.com (Q.T. Nguyen), jorgej@tecnico.ulisboa.pt (J.A. Jorge), agomes@di.ubi.pt (A.J.P. Gomes).

http://dx.doi.org/10.1016/j.future.2016.07.009
0167-739X/

surface. This provides us with an advantage over state-of-the-art techniques. In fact, the technique is more robust in identifying candidate cavities because the curvature can be evaluated not only on the protein surface, but also at any point of the domain of the scalar field from eigenvalues of the Hessian matrix; hence, we are able to identify the critical points of the scalar field.

Indeed, CriticalFinder is the first *surface-based method* to succeed in finding a meaningful segmentation of a molecular surface into cavities and saliences. More specifically, our method relies on the theory of critical points (also called Morse theory) to identify cavities on the protein surface. While some research works have already tried to use curvature information (see, for example, Natarajan et al. [6]), the resulting segmentations did not prove effective for cavity detection purposes, because their charts (or segments) do not necessarily match protein cavities as tentative binding sites. Furthermore, to the best of our knowledge, CriticalFinder is the first cavity detection algorithm to take advantage of a loosely-coupled GPU cluster of computers equipped with Nvidia Tesla K40 graphics cards, over a local area network (LAN), to identify cavities on protein surfaces.

The remainder of our paper is organized as follows. Section 2 briefly surveys the most closely related work published in the literature. Section 3 describes the fundamentals of scalar field theory and theory of critical points underlying our algorithm. Section 4 describes our algorithm in detail, as well as its implementation. Section 5 briefly describes our technique to triangulate and visualize protein surfaces. Section 6 discusses the theoretical complexity of the algorithm. Section 7 describes the methodology followed in the optimization of the CUDA code. Section 8 contains the most relevant results produced by our method, including a comparison to other well-known algorithms found in the literature. Section 9 discusses the main conclusions, while providing relevant hints for future work.

2. Prior work

Intuitively, cavities (also called pockets) are concavities on protein surfaces, although their geometrical definition is not straightforward [3]. Indeed, cavities range from small spherical invaginations to deep curved or linear clefts in the protein [7]. Interestingly, researchers have observed that ligands (drugs, in particular) commonly bind into the largest and/or deepest concavity on the protein surface [8]. On average, such cavity might be three times as large as the ligand, which shows how hard it is to characterize what a protein cavity (its boundary) really is.

In spite of this ambiguity in defining the boundaries of protein cavities, most works in the literature call attention to two main families of methods to detect cavities of proteins: energy-based and geometry-based [9]. *Energy-based methods* calculate the energy that results from the interaction between protein atoms and a small-molecule probe, whose value dictates the existence or not of a cavity. In turn, *geometry-based methods* aim at detecting solvent accessible regions of the protein surface using geometric criteria. Interestingly, as noted by Schmidtke et al. [10], both families of methods perform quite well detecting around 95% of the known cavities. Additionally, the geometry-based methods are faster and more robust against structural variations or missing atoms/residues in the input data concerning proteins than the energy-based algorithms, particularly in a context of a large-scale prediction of potential binding cavities [10]. However, methods based on geometry are hardly able to distinguish between different types of binding sites, and tend to fail when the larger cavities do not correspond to binding sites [11]. This agrees with Laskowski et al. [12], who noted that the ligand binds to the largest cavity in over 83% of the proteins.

In fact, the geometric methods are agnostic in relation to the type of cavity, since they all assume that cavities are depressions of the protein surface; as a consequence, they focus on the depth of the cavity – where the solvent lies in – not on the type of cavity. Since CriticalFinder is a geometric method, we will only discuss geometry-based methods from now on. In general, geometry-based methods can be divided into three main categories: grid-based, sphere-based, and tessellation-based [13]. *Grid-based methods* use an axis-aligned 3D grid embedded in the domain $D \in \mathbb{R}^3$ that encloses a given molecule, as well as an integer density map that determines if each grid node (or voxel) is *outside*, *inside*, or *on* the protein surface. Then, they use voxel clustering to collect relevant voxels into cavities; more specifically, a cavity is a cluster of *outside* voxels that are bracketed by *on* voxels in a number of directions. We refer the reader to [14–19] for further details about grid-based methods. In *sphere-based methods*, probe spheres play the same role as voxels in grid-based methods. Therefore, a cavity is conceptualized as a set of probe spheres which remain inside without slipping out, as described in [20–24]. *Tessellation-based methods* build up on the concept of tessellation, as is usual in computational geometry. Tessellation is a generalization of triangulation (e.g., Delaunay triangulation), in the sense that it also embraces the concept of Voronoi diagram, also called Voronoi tessellation. These methods have their roots in the theory of alpha shapes [25–27]. An alpha shape of a molecule is a triangulation that uniquely decomposes the space occupied by its atoms [28], while capturing the shape of the molecule itself [29].

On the other hand, the *surface-based methods* constitute a less known, but emergent, category of methods that explore the geometric properties of the protein surface to identify its cavities, such as curvature. But, using curvature to identify protein cavities is not an easy task, largely because the detection of cavities as putative binding sites requires a zonal – rather than local – shape analysis that goes beyond the neighborhood of each surface point. In the past, the Connolly function was used to decompose the molecular surface into convex, concave, and saddle patches [4]. However, the resulting surface segmentation was too fine, i.e., far from the coarse surface segmentation into binding cavities. In order to solve this problem, Natarajan et al. [6] introduced a new Morse theory-based method for segmentation of molecular surfaces, which uses a variant of the atomic density function originally introduced by Mitchell et al. [5]. By simplifying such atomic density function, neighbor segments merge into larger segments, so that cavities (and also saliences) become noticeable at progressively coarser resolutions. However, there is no evidence that the identified cavities correspond to protein cavities provided by any ground-truth database of binding sites (e.g., LigASite at http://ligasite.org/). Likewise, Exner et al. [30] tried to take advantage of the global curvatures due to Zachmann et al. [31] to describe larger surface segments. Still, it is not clear whether the discovered segments match ground-truth binding sites.

In a marked distinction to previous work, we believe that the curvature analysis must be applied to the scalar field (or function) that describes the surface, rather than to the Connolly or Mitchell–Kerr–Eyck functions defined over the molecular surface. Our argument is that cavities are in the vicinity of critical points (i.e., minimum, maximum, and saddle points) of the scalar field that features the molecular surface; hence we named our method CriticalFinder. It is a surface-based algorithm that has been designed to entirely run on a loosely-coupled GPU cluster based on the CUDA architecture. Apart from CriticalFinder, the only automatic cavity detection algorithm that entirely runs on GPU is due to Lo et al. [32], but it uses the Connolly function for segmentation of the molecular surface, with all the inherent shortcomings aforementioned. Furthermore, their method runs on a single GPU, while ours can scale to many GPUs.