UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

# Geometric Algorithms for Cavity Detection on Protein Surfaces

**Tiago Miguel Carrola Simões**

Tese para obtenção do Grau de Doutor em

**Engenharia Informática**

(3º ciclo de estudos)

Orientador: Prof. Doutor Abel João Padrão Gomes

**Covilhã, Junho de 2019**

*To my lovely family*

This doctoral thesis was developed at Instituto de Telecomunicações, MediaLab, Universidade da Beira Interior and supported by the Portuguese research council, Fundação para a Ciência e a Tecnologia, through the grant contract SFRH/BD/99813/2014 under the programme QREN POPH — Type 4.1 — Advanced Training, co-funded by the European Social Fund and by national funds from the Portuguese Ministry for Education and Science (Ministério da Educação e Ciência).

# Acknowledgements

This thesis would not have been possible without the support and collaboration of several people and entities.

First, my deepest thanks to my advisor, professor Abel Gomes, without his guidance, support, expertises, and research insights this thesis would not have been possible. Professor Abel Gomes is the example of personal, scientific, and professional qualities that I inspire to be throughout my life.

Second, I would also like to thank for all the support given by Instituto de Telecomunicações, University of Beira Interior, the Portuguese research council, Fundação para a Ciência e a Tecnologia, through the grant contract SFRH/BD/99813/2014 under the programme QREN POPH — Type 4.1 — Advanced Training, co-funded by the European Social Fund and by national funds from the Portuguese Ministry for Education and Science (Ministério da Educação e Ciência). Without the opportunity given by these entities, this thesis would also not have been possible.

Third, to my family, in particular to my grandmother that passed away while I was concluding this thesis, for all the years of love, thank you grandma. Wherever you are, grandpa is sure to be with you. I would also like to thank to my parents, brother, parents-in-law, brother-in-law, grandparents-in-law, and all my closest friends (Ricardo Graça, Catarina Morais, and Ricardo Venâncio). They have always encouraged me to follow my dreams. Additionally, a warm thank you to my father-in-law that passed away before the start of this thesis.

Last but not least, to my wife Ana. There are no words to describe her, she will always be the light of my path, thank you for never giving up on me. During the long hours of work her immeasurable patience and unconditional support made possible the conclusion of this thesis.

# Resumo

Estruturas macromoleculares tais como as proteínas potencializam processos ou funções celulares. Estas funções resultam das interações entre proteínas e peptídeos, substratos catalíticos, nucleótideos, ou até mesmo substâncias químicas produzidas pelo homem. Assim, há vários tipos de interacções: proteína-ligante, proteína-proteína, proteína-DNA e assim por diante. Além disso, estas interações geralmente ocorrem em regiões conhecidas como locais de ligação (*binding sites*, do inglês) e só acontecem sob condições de complementaridade química e de forma. É também importante referir que uma proteína pode ser estruturada em quatro níveis. A estrutura primária que consiste em sequências de aminoácidos (ou cadeias), a estrutura secundária que compreende essencialmente por hélices $\alpha$ e folhas $\beta$, que são subsequências (ou subdomínios) dos aminoácidos da estrutura primária, a estrutura terciária que resulta da composição de subdomínios em domínios, que por sua vez representa a forma geométrica da proteína, e por fim a estrutura quaternária que é o resultado da agregação de duas ou mais estruturas terciárias. Este último nível estrutural é frequentemente conhecido por um complexo proteico.

Esta tese enquadra-se no âmbito da conceção de fármacos baseados em estrutura e no acoplamento de proteínas. Mais especificamente, aborda-se o problema fundamental da deteção e identificação de cavidades que são frequentemente vistos como possíveis locais de ligação (*putative binding sites*, do inglês) para os seus ligantes (*ligands*, do inglês). De forma geral, os algoritmos de identificação de cavidades dividem-se em três categorias principais: baseados em energia, geometria ou evolução. Os métodos evolutivos baseiam-se em estimativas de conservação das sequências evolucionárias. Isto é, estes métodos permitem detectar locais funcionais através do cálculo da conservação evolutiva das posições dos aminoácidos das proteínas. Em relação aos métodos baseados em energia estes baseiam-se no cálculo das energias de interação entre átomos da proteína e do ligante. Por fim, os algoritmos geométricos baseiam-se na análise da forma geométrica da proteína para identificar cavidades. Esta tese foca-se nos métodos geométricos.

Apresentamos nesta tese três novos algoritmos geométricos para detecção de cavidades em proteínas. A principal contribuição desta tese está no uso de técnicas de computação gráfica na análise e reconhecimento de cavidades em proteínas, muito no espírito da modelação e visualização molecular. Como pode ser visto mais à frente, estas técnicas incluem o *field-of-view* (FoV), *voxel ray casting*, *back-face culling*, funções de diâmetro de forma, a teoria de Morse, e os pontos críticos. A ideia principal é segmentar a proteína, à semelhança do que acontece na segmentação de malhas em computação gráfica. Na prática, os algoritmos de detecção de cavidades não são nada mais que algoritmos de segmentação de proteínas.

# Palavras-chave

Computação gráfica
Modelação e visualização molecular
Biologia molecular
Proteína
Interação proteina-ligante
Local de acoplamento
Cavidade
Superficie molecular
Campo escalar Gaussiano
Triangulação
Algoritmo geométrico
*Field-of-view*
*Back-face culling*
*Voxel ray casting*
Segmentação de malha
Ponto crítico

# Abstract

Macromolecular structures such as proteins heavily empower cellular processes or functions. These biological functions result from interactions between proteins and peptides, catalytic substrates, nucleotides or even human-made chemicals. Thus, several interactions can be distinguished: protein-ligand, protein-protein, protein-DNA, and so on. Furthermore, those interactions only happen under chemical- and shape-complementarity conditions, and usually take place in regions known as binding sites. Typically, a protein consists of four structural levels. The primary structure of a protein is made up of its amino acid sequences (or chains). Its secondary structure essentially comprises $\alpha$-helices and $\beta$-sheets, which are sub-sequences (or sub-domains) of amino acids of the primary structure. Its tertiary structure results from the composition of sub-domains into domains, which represent the geometric shape of the protein. Finally, the quaternary structure of a protein results from the aggregate of two or more tertiary structures, usually known as a protein complex.

This thesis fits in the scope of structure-based drug design and protein docking. Specifically, one addresses the fundamental problem of detecting and identifying protein cavities, which are often seen as tentative binding sites for ligands in protein-ligand interactions. In general, cavity prediction algorithms split into three main categories: energy-based, geometry-based, and evolution-based. Evolutionary methods build upon evolutionary sequence conservation estimates; that is, these methods allow us to detect functional sites through the computation of the evolutionary conservation of the positions of amino acids in proteins. Energy-based methods build upon the computation of interaction energies between protein and ligand atoms. In turn, geometry-based algorithms build upon the analysis of the geometric shape of the protein (i.e., its tertiary structure) to identify cavities. This thesis focuses on geometric methods.

We introduce here three new geometric-based algorithms for protein cavity detection. The main contribution of this thesis lies in the use of computer graphics techniques in the analysis and recognition of cavities in proteins, much in the spirit of molecular graphics and modeling. As seen further ahead, these techniques include field-of-view (FoV), voxel ray casting, back-face culling, shape diameter functions, Morse theory, and critical points. The leading idea is to come up with protein shape segmentation, much like we commonly do in mesh segmentation in computer graphics. In practice, protein cavity algorithms are nothing more than segmentation algorithms designed for proteins.

# Keywords

Computer graphics
Molecular graphics and modeling
Molecular biology
Protein
Protein-ligand interaction
Protein cavity
Binding site
Molecular surface
Gaussian scalar field
Triangulation
Geometric-based algorithm
Field-of-view
Back-face culling
Voxel ray casting
Mesh segmentation
Critical point

# Resumo Alargado

Esta tese enquadra-se no âmbito da modelação e visualização molecular, abrangendo tópicos de computação geométrica, computação gráfica, e visualização aplicada à detecção e reconhecimento de cavidades. A deteção e a identificação de cavidades é um passo essencial na docagem em proteínas e na conceção de fármacos baseada em estrutura. Essencialmente, a detecção e o reconhecimento de cavidades em proteínas é um problema que envolve a segmentação da superfície da proteína (ou o seu envelope) em regiões concernentes a cavidades e seu complemento.

## Enquadramento da Tese

As proteínas desempenham um papel fundamental nas funções bioquímicas dos organismos vivos. As proteínas interagem com outras entidades na célula, em particular, com entidades tais como os ácidos nucleicos (por exemplo, o DNA), peptídeos, substratos catalíticos e produtos químicos fabricados pelo homem. Portanto, existem diferentes locais nas superficies das proteínas onde interações proteína-ligante, proteína-proteína, proteína-DNA podem ocorrer.

Esta tese foca-se nas interações entre proteínas e os seus ligantes; mais especificamente, é de interesse identificar os locais de ligação das proteínas onde os ligantes supostamente se ligam. Em termos gerais, isto é o que se encontra por detrás do desafio da conceção de fármacos baseada em estrutura e na docagem em proteínas. Isto visa não só a identificação dos locais onde pequenas moléculas podem acoplar mas também na avaliação do impacto que os ligantes poderão ter na função da proteína.

A compreensão do processo de ligação proteína-ligante teve progressos significativos desde a formulação da hipótese "fechadura-e-chave" em 1984 por Hermann Fischer. Este sugeriu que a ligação de um substrato a uma enzima é análoga à inserção de uma chave numa fechadura. Este modelo de complementaridade da forma rígida (também chamado de complementaridade geométrica) entre um ligante e um receptor evoluiu desde então no sentido de considerar a flexibilidade do receptor e do ligante. Após isto, modelos mais dinâmicos de ligação surgiram, como por exemplo, o modelo de *zipper*, o modelo de ajuste induzido (*induced-fit model*, do inglês) e o modelo de seleção conformacional (*conformational-selection model*, do inglês) (ver [FW08] para uma análise mais detalhada). Contudo foi observado por Kahraman et al. [KMLT07] que a complementaridade da forma é um requisito necessário mas não suficiente no acoplamento do ligante. O acoplamento molecular também depende da complementaridade físico-química, que têm que ver com, por exemplo, com as interações eletrostáticas, as ligações de hidrogénio, hidrofóbicas e também das interações mediadas por solventes entre o ligante e a proteína.

Um requisito importante para desenhar novos métodos de deteção de cavidades em proteínas é concordar com uma definição do que é realmente uma cavidade. Como discutido por [SLD$^+$17], não existe um consenso na definição de cavidade. No entanto, nesta tese iremos utilizar a definição apresentada por Simões et al. [SLD$^+$17]: "Uma cavidade é um componente conexo do espaço complementar da proteína dentro do seu invólucro convexo". Esta definição geral enfatiza a tridimensionalidade de uma cavidade. Uma cavidade de uma proteína é também frequentemente chamada de local de ligação putativo ou possível (*putative binding site*, do inglês). Contudo, um local de ligação é definido como um local funcional para o qual se conhece um ligante, ao passo que, uma cavidade pode não corresponder necessariamente a um local de ligação.

Em geral, os métodos de detecção de cavidades podem ser divididos em três categorias principais: métodos energéticos, evolutivos e geométricos [NSG12]. É conhecido que o desempenho destas três categorias de métodos é muito semelhante em termos de acuidade (*accuracy*, do inglês). No entanto, os algoritmos baseados em geometria são conhecidos por serem mais rápidos do que aqueles baseados em energia e em evolução, com a vantagem de serem mais robustos face a variações estruturais ou à falta de átomos / resíduos da proteína considerada [SSE$^+$10]. É também importante notar que os métodos evolutivos dependem do número de sequências disponíveis e da qualidade da ferramenta de alinhamento. Sendo assim, esta tese concentra-se nos algoritmos baseados em geometria.

## Descrição do Problema

Os algoritmos de deteção de cavidades baseados em geometria abrangem uma variedade de técnicas, incluindo o particionamento da *bounding box* que encapsula a proteína, a colocação de esferas no espaço vazio da proteína e a utilização de triangulações de Delaunay e esferas $\alpha$ [SSE$^+$10, KKL$^+$16, SLD$^+$17].

Os problemas principais destes métodos são a acuidade (*accuracy*, do inglês) e o desempenho. Nesta tese, abordamos principalmente a problemática relacionada com a acuidade (*accuracy*, do inglês), embora, o desempenho, que se relaciona com o tempo de execução dos métodos, seja também uma questão crítica que deverá ser analisada futuramente. Por ora, o nosso entendimento é que o problema do desempenho pode ser resolvido recorrendo à computação baseada em GPU. Acontece que nenhuma computação baseada em GPU foi usada para desenhar e implementar os métodos descritos nesta tese, de maneira que, não existe a necessidade colocar a ênfase no desempenho e na velocidade de execução.

A acuidade (*accuracy*, do inglês) tem muito a ver com os quatro problemas abaixo descritos, sendo estes extensivamente discutidos no *survey* do segundo capítulo desta tese:

- *Localização de cavidades assistidas pelo utilizador.* Este problema poderá surgir em qualquer categoria de métodos. Aparece tipicamente quando é solicitado ao utilizador, de forma interativa, a localização aproximada de uma cavidade na proteína. Cabe ao método determinar a extensão da mesma.

- *Sensibilidade ao espaçamento da grelha.* O valor do espaçamento é um parâmetro crucial em métodos baseados em grelha, ou qualquer outro método que aproveita a grelha para particionar a *bounding box* onde a proteína se encontra. É de notar, que ao variar o espaçamento podemos obter um número distinto de cavidades. Esta questão é bastante difícil de controlar, no sentido de que não é fácil encontrar um compromisso entre acuidade (*accuracy*, do inglês) e desempenho.

- *Ambiguidade na definição das entradas e saídas das cavidades.* Esta ambiguidade é definida como um problema geral uma vez que poderá surgir em qualquer método. Tendo em conta que existe a necessidade de saber onde uma cavidade começa e termina, este problema é essencialmente relacionado com a incapacidade do método de delimitar a região referente às entradas e saídas de cada cavidade.

- *Sensibilidade à orientação da proteína.* O problema surge em métodos que resultam da gridificação ou particionamento do espaço que envolve a proteína. Como é mostrado no segundo capítulo, métodos baseados em grelha são dependentes da rotação da proteína, a menos que ferramentas adicionais sejam utilizadas (por exemplo, o invólucro convexo). Esta questão relaciona-se com a anterior, porque a sensibilidade à orientação da proteína pode ser superada através da colocação de tampões nas entradas e saídas das cavidades.

Por fim, é importante notar que em termos de acuidade (*accuracy*, do inglês) a grande maioria dos métodos propostos ao longo dos anos continuam a apresentar taxas de sucesso relativamente baixas na detecção de cavidades em proteínas. Contudo, estes métodos são ferramentas fundamentais na descoberta de novos locais de ligação em proteínas. Isto é, cavidades para as quais não existem fármacos às quais se liguem.

## Hipótese de Investigação

Esta tese visa introduzir técnicas geométricas da computação gráfica no campo de modelação molecular, em particular, desenvolver algoritmos geométricos para detectar cavidades em proteínas que sejam mais precisos do que aqueles existentes na literatura. É importante mencionar que à parte daqueles problemas específicos identificados na secção anterior, os problemas de longo prazo dos métodos de detecção de cavidades são a *acuidade* (*accuracy*, do inglês) e o *desempenho*. No presente trabalho de doutoramento, o principal objectivo é encontrar formas de melhorar a acuidade (*accuracy*, do inglês) dos métodos sem perder desempenho de forma notória. Assim sendo, este trabalho é sustentado por uma revisão cuidadosa da literatura (o segundo capítulo desta tese), com uma compreensão aprofundada da natureza de cada método geométrico, dos seus pontos fortes e das sua fraquezas. Esta revisão da literatura levou o candidato a estudar ferramentas de segmentação em computação gráfica e computação geométrica, por forma a encontrar a inspiração e o discernimento necessários à resolução dos problemas anteriormente mencionados.

Neste pressuposto, o presente trabalho de investigação levou à seguinte hipótese de investigação (*thesis statement*, do inglês):

> *É possível desenhar métodos de deteção de cavidades que sejam mais precisos do que aqueles do estado da arte através de conceitos geométricos, descritores de forma, e técnicas tradicionalmente utilizadas em computação gráfica.*

Nesta tese, pretende-se explorar a formulação matemática da superfície (molecular) Gaussiana, descritores de forma como a função de diâmetro de forma (SDF) [SSCO08], análise espetral (os valores próprios e a teoria de Morse) e outros conceitos semelhantes para detetar cavidades em proteínas. Para além disto, pretende-se também estudar os fatores não geométricos que determinam a acuidade (*accuracy*, do inglês) global dos métodos de deteção de cavidades em proteínas.

## Plano de Investigação

Seguindo a hipótese de investigação anterior, o plano de investigação que sustenta esta tese de doutoramento foi organizada nas seguintes tarefas:

- *Revisão da literatura.* A primeira tarefa a ser completada durante o programa de doutoramento foi a escrita de um *survey* sobre métodos de deteção de cavidades em proteínas. Esta foi uma tarefa fundamental na tomada de consciência relativamente aos problemas existentes na deteção de cavidades em proteínas.

- *Teste, comparação e validação.* Esta tarefa teve como objetivo comparar e avaliar a acuidade (*accuracy*, do inglês) dos algoritmos de deteção de cavidades que se encontram publicamente disponíveis. Como será visto mais à frente, esta tarefa é transversal a todo o trabalho de investigação que conduziu ao desenho e à implementação dos métodos propostos nesta tese. Esta tarefa originou também a construção de uma ferramenta de análise comparativa denominada de CavBench (ver Secção 1.6). Esta ferramenta permite comparar métodos de deteção de cavidades relativamente a um conjunto de locais de ligação conhecidos (*ground-truth dataset of binding sites*, do inglês).

- *Um método de grelha e superfície para detectar cavidades através de ferramentas e técnicas de computação gráfica.* Esta tarefa originou "um método de deteção de cavidades chamado CavVis que combina a voxelização e a formulação analítica da superfície Gaussiana. Este método baseia-se na visibilidade dos pontos da superfície molecular para encontrar cavidades. Especificamente, o critério de visibilidade combina três conceitos da computação gráfica, o *field-of-view* (FOV) de cada ponto da superfície, o *voxel ray casting* e o *back-face culling*". (Baseado no resumo do artigo do Capítulo 3).

- *Um método de superfície para detetar cavidades utilizando uma segmentação baseada na SDF.* Nesta tarefa abordou-se o problema de encontrar cavidades nas superficies das proteínas como um problema de segmentação de malha como é usual em computação gráfica. Com esta técnica, a superfície da proteína foi segmentada utilizando "o conceito de *shape diameter function* (SDF). A SDF é uma função escalar que mede o diâmetro do volume interior de uma superfície fechada na vizinhança de cada um dos seus pontos. Curiosamente, quando aplicada a superfícies Gaussianas, que são modeladas pela soma das funções Gaussianas positivas, a SDF acaba por medir o diâmetro do volume exterior da região externa da proteína, a partir do qual podemos inferir a localização e a extensão das cavidades. Ao contrário de outros métodos de deteção de cavidades em proteínas, a SDF é independente da posição da proteína, apresentando valores distintos em diferentes cavidades, permitindo assim identificar cavidades através do agrupamento (*clustering*, do inglês) de valores SDF semelhantes". (Baseado no resumo do artigo do Capítulo 4).

- *Um método de grelha para detetar cavidades utilizando a análise espetral como é usual em computação gráfica.* Esta tarefa levou à construção de um novo algoritmo de deteção de cavidades designado por CavSeeker. Foi desenvolvido a partir da ideia chave que "as cavidade estão localizadas em torno de pontos críticos específicos do campo de densidade eletrónica da proteína. Basicamente, o CavSeeker encontra aqueles vóxeis que são transversais a duas isosuperfícies do campo de densidade eletrónica da proteína, entre as quais se podem encontrar os pontos críticos e as cavidades correspondentes." (Baseado no resumo do artigo do Capítulo 5). É importante também lembrar que na análise espetral, os pontos críticos são aqueles pontos do campo escalar onde a função da primeira derivada é nula ou não definida. Em particular, as cavidades correspondem às localizações dos pontos de sela e dos mínimos, que por sua vez são determinados através dos valores próprios da matriz Hessiana.

- *Escrita da tese.* A presente tese é constituída por artigos científicos. Tais artigos correspondem ao segundo, terceiro, quarto e quinto capítulos. A tese também inclui um capítulo introdutório, bem como um capítulo final onde se apresentam as principais conclusões e sugestões para trabalho futuro.

Como observação final, é importante mencionar que esta tese foca-se na resolução de problemas relacionados com a acuidade (*accuracy*, do inglês) na deteção de cavidades em proteínas. Os problemas de desempenho não foram considerados tão relevantes quanto aqueles relacionados com a acuidade (*accuracy*, do inglês).

## Principais Contribuições

As principais contribuições do trabalho de investigação que sustenta esta tese de doutoramento resultaram da necessidade de satisfazer a hipótese de investigação descrita anteriormente. As contribuições científicas são as seguintes:

- *Descritores de forma*. Como se verá ao longo da tese, o recurso a descritores de forma e técnicas de segmentação de malhas utilizados em computação gráfica para realizar a segmentação de proteínas em cavidades e seu complemento acaba por ser razoável e viável. Aparentemente, esta é uma linha de investigação para explorar ainda mais no futuro.
- *Aglomeração*. A acuidade (*accuracy*, do inglês) dos métodos baseados em computação gráfica tende a aumentar significativamente se estes incorporarem técnicas de aglomeração adequadas na formação das cavidades. Portanto, parece óbvio que a utilização de técnicas de aprendizagem automática para ajudar a delinear melhor os aglomerados ou agrupamentos (i.e., cavidades) é uma possível avenida de investigação a perseguir no futuro.
- *Filtragem*. A acuidade (*accuracy*, do inglês) dos métodos mencionados anteriormente aumenta se estes aproveitarem técnicas de filtragem baseadas em volume, descartando cavidades pequenas onde os ligantes não conseguem entrar. Parece também claro que é necessário investigar novas técnicas de filtragem para além daquelas baseadas no volume das cavidades.

De forma resumida, conseguimos agora ter uma compreensão melhor de que existe um longo caminho a percorrer para criar métodos de segmentação de proteínas que tirem partido da panóplia de técnicas de segmentação propostas no âmbito da computação gráfica nos últimos vintes anos. Caso contrário, será bastante difícil atingir taxas de acuidade (*accuracy*, do inglês) superiores a 90% na detecção de cavidades em proteínas.

## Publicações

O trabalho de investigação por detrás desta tese de doutoramento originou as seguintes publicações:

- **Geometric detection algorithms for cavities on protein surfaces in molecular graphics: a survey**
  Tiago Simões, Daniel Lopes, Sérgio Dias, Francisco Fernandes, João Pereira, Joaquim Jorge, Chandrajit Bajaj, and Abel Gomes.
  *Computer Graphics Forum*, Vol. 36, No. 8, pp. 643-683, June 2017.
  DOI: `https://doi.org/10.1111/cgf.13158`

- **CavVis — A field-of-view geometric algorithm for protein cavity detection**
  Tiago Simões and Abel Gomes.
  *Journal of Chemical Information and Modeling*, Vol. 59, No. 2, pp. 786-796, January 2019.
  DOI: `https://doi.org/10.1021/acs.jcim.8b00572`

- **CavShape — A cavity detection algorithm through the multivariate shape diameter function**
  Tiago Simões and Abel Gomes.
  *PLOS Computational Biology* (submitted for publication).

- **CavSeeker — Identifying protein cavities by locating critical points of the scalar field generated from summation of Gaussians**
  Tiago Simões and Abel Gomes.
  *Bioinformatics* (submitted for publication).

Outras publicações não incluídas nesta tese:

- ***CavBench: A benchmark for protein cavity detection methods***
  Sérgio E.D. Dias, Tiago M.C. Simões, Francisco Fernandes, Ana Mafalda Martins, Alfredo Ferreira, Joaquim A. Jorge, and Abel J.P. Gomes.
  *PLOS ONE* (submitted for publication).

## Organização da Tese

Esta tese de doutoramento está organizada da seguinte forma:

**Capítulo 1**: Este capítulo fornece a síntese do trabalho de investigação que levou à criação da presente tese de doutoramento, incluindo a motivação, a descrição do problema, a hipótese de investigação, o plano de investigação, e a organização da tese. Adicionalmente, são também apresentadas as principais contribuições e as publicações resultantes do plano de investigação referente ao programa de doutoramento, bem como algumas pistas para trabalho futuro.

**Capítulo 2**: Este capítulo revê os métodos geométricos para a detecção de cavidades em proteínas. Esta revisão da literatura baseia-se na definição matemática do que é uma cavidade, bem como numa taxinomia de cavidades em proteínas. Em seguida, as diferentes famílias ou categorias de métodos são abordadas e discutidas. Em concreto, são discutidas aqueles problemas relacionados com a localização de cavidades assistidas pelo utilizador, sensibilidade ao espaçamento da grelha, ambiguidade na definição das entradas e saídas das cavidades, e ainda a sensibilidade à orientação da proteína.

**Capítulo 3**: Este capítulo descreve um novo método baseado em grelha-e-superfície chamado CavVis. Este método utiliza conceitos e técnicas da computação gráfica como *field-of-view*, o *voxel ray casting*, e o *back-face culling*, em conjunção com a formulação analítica da superfície molecular Gaussiana para prever a localização das cavidades (ou locais putativos de ligação) em proteínas.

**Capítulo 4**: Este capítulo propõe um novo método baseado em superfície, chamado CavShape. Este introduz um descritor de forma designado por *multivariate shape diameter function* (mSDF), que é uma variante do descritor de forma SDF utilizado na esqueletização e segmentação de malhas no âmbito da computação gráfica [SSCO08]. O mSDF é utilizado para encontrar as cavidades em proteínas, ou seja, segmentar a proteína em cavidades e seu complemento.

**Capítulo 5**: Este capítulo apresenta um novo método de grelha chamado CavSeeker. Este método baseia-se na análise de valores próprios para determinar os mínimos e os pontos de sela (*saddle*, do inglês) do campo escalar Gaussiano da proteína. Tais pontos críticos indicam a localização das cavidades no exterior da proteína. De certa forma, é possível afirmar que este método tira partido da teoria de Morse em geometria diferencial.

**Capítulo 6**: O capítulo final desta tese de doutoramento reforça as principais contribuições desta tese, as conclusões mais relevantes e aponta algumas sugestões para trabalho futuro.

## Conclusões Finais e Trabalho Futuro

Esta tese enquadra-se no âmbito da docagem em proteínas e na conceção de fármacos baseada na estrutura. Em termos mais específicos, o seu trabalho de investigação subjacente focou-se em métodos de deteção de cavidades de proteínas. A sua principal contribuição científica está no uso de descritores de forma — comummente utilizados em computação — no contexto da modelação e visualização molecular, aqui especificamente aplicados à deteção de cavidades de proteínas.

## Contexto de Investigação

As proteínas desempenham um papel importante no funcionamento dos organismos vivos devido às suas interações com outras moléculas. Este papel está relacionado com a hipótese “fechadura-e-chave”, formulada em 1984 por Hermann Fischer, ou seja, a complementaridade geométrica entre uma proteína receptora e um ligante. O trabalho de investigação descrito nesta tese tem muito a ver com a deteção das “fechaduras” das proteínas, às quais chamamos de cavidades (ou locais putativos de ligação), e não tanto com as “chaves” (ou ligantes).

Esta tese foca-se no desenho e desenvolvimento de novos métodos geométricos de detecção de cavidades em proteínas, para o que se teve em consideração o seguinte:

- *Descritores de forma*. Como é usual em segmentação de objectos 3D no âmbito da computação gráfica, as segmentações de proteínas propostas nesta tese baseiam-se em descritores de forma. Por exemplo, o método CavShape (veja-se Capítulo 4) baseia-se na *multivariate shape diameter function* (mSDF) para realizar a segmentação da superfície molecular.

- *Aglomeração*. Como é descrito nesta tese, o agrupamento dos elementos (por exemplo, triângulos) de uma cavidade é uma tarefa fundamental nos métodos de detecção de cavidades em proteínas. Normalmente, utilizam-se critérios baseados na proximidade para agrupar tais elementos em cavidades.

- *Filtragem*. A maioria dos métodos de detecção de cavidades tendem a produzir um número excessivo de cavidades. No entanto, é sabido que alguns deles consideram apenas as três cavidades maiores no processo de detecção. Isto é, as cavidades mais pequenas são habitualmente descartadas na fase final do método de detecção.

No seu todo, este trabalho de investigação foi elaborado de forma a que se consiga chegar a métodos de detecção de cavidades mais precisos do que os existentes no estado atual da arte, ainda que garantindo que não exista uma perda perceptível de desempenho.

## Questões de Investigação

Para validar a hipótese de investigação, várias questões de investigação tiveram de ser respondidas:

- *É possível de forma precisa detetar e delimitar cavidades em proteínas utilizando conceitos da computação gráfica (por exemplo, o field-of-view) sem usar descritores de forma?*
  O método CavVis não utiliza descritores de forma para segmentar a proteína em

cavidades. No entanto, a sua acuidade (*accuracy*, do inglês) é mais elevada ou comparável a outros métodos analisados nesta tese (ver Capítulo 3). É importante lembrar que o método CavVis combina três ferramentas e conceitos importantes utilizados em computação gráfica, nomeadamente: o *field-of-view* (FoV), o *voxel ray casting* e, por fim, o *back-face culling*.

- *É possível de forma precisa detetar e delimitar cavidades em proteínas utilizando descritores de forma?*
  Ambos os métodos descritos nos Capítulos 4 (CavShape) e 5 (CavSeeker) utilizam descritores de forma para determinar a localização das cavidades. O método CavShape utiliza um descritor de forma chamado *multivariate shape diameter function* (mSDF) para encontrar cavidades, enquanto que o método CavSeeker utiliza outro descritor de forma baseado na análise dos valores próprios. Especificamente, as cavidades encontradas pelo CavSeeker correspondem às localizações dos mínimos e dos pontos de sela (*saddle*, do inglês) do campo escalar de densidade eletrónica (*electron density scalar field*, do inglês) da proteína.

Posto isto, a hipótese de investigação é aqui positivamente validada. Em particular, chegou-se à conclusão que é bastante promissor desenhar algoritmos precisos de detecção de cavidades utilizando descritores de forma do âmbito da computação gráfica. Para além disso, a principal contribuição desta tese é muito provavelmente a utilização de descritores de forma para segmentar proteínas.

## Limitações e Trabalho Futuro

Olhando para o trabalho de investigação já realizado durante o programa de doutoramento, pode-se antever três linhas de investigação a seguir no futuro.

Primeiro, a acuidade (*accuracy*, do inglês) dos métodos de detecção de cavidades está longe de ser um problema resolvido, uma vez que esta ainda se encontra abaixo dos 90%. Portanto, é necessário investigar outros tipos de descritores de forma que sejam capazes de atingir valores de acuidade (*accuracy*, do inglês) mais elevados.

Segundo, a velocidade dos métodos de detecção de cavidades ainda é relativamente lenta se se utilizar o modelo de computação sequencial (*single-thread*) para processar um número de átomos superior a algumas dezenas de milhares. Este fenómeno é particularmente perceptível quando é utilizada a soma das funções Gaussianas ou outras funções semelhantes. Por outras palavras, a velocidade do método é dependente da formulação matemática da superfície da proteína. Sendo assim, a questão que se coloca é se existe ou não uma formulação matemática alternativa para superfícies moleculares.

Por fim, os métodos actuais baseados em geometria são essencialmente estáticos, isto é, apenas consideram uma configuração da proteína no processo de detecção de cavidades. Uma tendência actual no âmbito da modelação e visualização molecular é o desenho e implementação de métodos de deteção que têm em consideração a geometria dinâmica (e topologia) das proteínas e das suas cavidades; por exemplo, uma cavidade ôca interna pode evoluir para uma cavidade de superfície exposta ao exterior e vice-versa.

## Referências

[FW08] A. Feldman-Salit and R. C. Wade. Molecular recognition: computational analysis and modelling. *Wiley Encyclopedia of Chemical Biology*, pages 1–10, 2008. xiii, 1

[KKL$^{+}$16] M. Krone, B. Kozlíková, N. Lindow, M. Baaden, D. Baum, J. Parulek, H.-C. Hege, and I. Viola. Visual analysis of biomolecular cavities: state of the art. In *Computer Graphics Forum*, volume 35, pages 527–551. Wiley Online Library, 2016. xiv, 2

[KMLT07] A. Kahraman, R. J. Morris, R. A. Laskowski, and J. M. Thornton. Shape variation in protein binding pockets and their ligands. *Journal of Molecular Biology*, 368(1):283–301, 2007. xiii, 1

[NSG12] B. Nisius, F. Sha, and H. Gohlke. Structure-based computational analysis of protein binding sites for function and druggability prediction. *Journal of Biotechnology*, 159(3):123–134, 2012. xiv, 2

[SLD$^{+}$17] T. Simões, D. Lopes, S. Dias, F. Fernandes, J. Pereira, J. Jorge, C. Bajaj, and A. Gomes. Geometric detection algorithms for cavities on protein surfaces in molecular graphics: a survey. In *Computer Graphics Forum*, volume 36, pages 643–683. Wiley Online Library, 2017. xiv, 2

[SSCO08] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249, 2008. xvi, xx, 4, 7

[SSE$^{+}$10] P. Schmidtke, C. Souaille, F. Estienne, N. Baurin, and R. T. Kroemer. Large-scale comparison of four binding site detection algorithms. *Journal of Chemical Information and Modeling*, 50(12):2191–2200, 2010. xiv, 2

# Contents

# List of Figures

**Chapter 2**

Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey.

**Chapter 3**

CavVis — A Field-of-View Geometric Algorithm for Protein Cavity Detection.

**Chapter 4**
CavShape — A Cavity Detection Algorithm Through the Multivariate Shape Diameter Function.

**Chapter 5**
CavSeeker — Identifying Protein Cavities by Locating Critical Points of the Scalar Field Generated from Summation of Gaussians.

**Chapter 6**
Conclusions and Future Work.

# List of Tables

**Chapter 2**

Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey.

**Chapter 3**

CavVis — A Field-of-View Geometric Algorithm for Protein Cavity Detection.

**Chapter 4**

CavShape — A Cavity Detection Algorithm Through the Multivariate Shape Diameter Function.

**Chapter 5**

CavSeeker — Identifying Protein Cavities by Locating Critical Points of the Scalar Field Generated from Summation of Gaussians.

# Acronyms

| | |
|---|---|
| OS | Operating system |
| CPU | Central processing unit |
| GPU | Graphics processing unit |
| CUDA | Compute unified device architecture |
| GLSL | OpenGL shading language |
| OpenCL | Open computing language |
| FoV | Field-of-view |
| 3D | Three-dimensional |
| PDB | Protein data bank file |
| SBDD | Structure-based drug design |
| DNA | Deoxyribonucleic acid |
| MC | Marching cubes |
| SAS | Solvent-accessible surface |
| SES | Solvent-excluded surface |
| GS | Gaussian surface |
| CH | Convex hull |
| vdW | Van der Waals surface |
| SA | Set of atoms |
| LES | Ligand-excluded surface |
| MOA | Mouth-opening ambiguity |
| GSS | Grid-spacing sensitivity |
| POS | Protein-orientation sensitivity |
| UACL | User-assisted cavity location |
| NMA | Normal mode analysis |
| MD | Molecular dynamics |
| PCA | Principal component analysis |
| ED | Essential dynamics |
| CBM | Constraint-based methods |
| ASP | Active site point |
| NSA | Nearest surface atom |
| VMD | Visual molecular dynamics |
| CCD | Cavity conformational distribution |
| GP | Geometric potential |
| EAT | Empirical alpha tuning |
| AD | Apollonius diagram |
| DT | Delaunay triangulation |
| MA | Medial axis |
| ET | Enveloping triangulation |
| SDF | Shape diameter function |
| mSDF | Multivariate shape diameter function |

# Chapter 1

# Introduction

This thesis fits in the scope of molecular graphics and modeling, spanning topics from geometric computing, computer graphics, and visualization applied to protein cavity detection and recognition. The discovery and identification of cavities in proteins is an essential step in protein docking and structure-based drug design. Essentially, protein cavity detection and recognition is a problem whose solution involves the segmentation of a protein surface (or its envelope) into cavity regions and their complement.

## 1.1 Motivation

Proteins play a fundamental role in the biochemical functions of most living organisms. Proteins interact with other entities in the cell, particularly with macro-entities like nucleic acid molecules (e.g., DNA), peptides, catalytic substrates, and human-made chemicals. Therefore, there exist a variety of sites on protein surfaces, where protein-ligand, protein-protein, protein-DNA interactions may take place.

This thesis focuses on the interactions between proteins and their ligands; more specifically, we are interested in identifying the protein binding sites where ligands supposedly bind. In general terms, this is behind the primary challenge of structure-based drug design (SBDD) and molecular docking, which aim at not only correctly predicting which small molecules would bind to a specific protein, but also assessing the impact that such ligand might have on protein function.

The understanding of the protein-ligand binding process has known significant progress since the formulation of the lock-and-key hypothesis in 1984 by Hermann Fischer, who suggested binding a substrate to an enzyme is analogous to inserting a key into a lock. This model of “rigid” shape complementarity (also called geometric complementarity) between the ligand and receptor has since evolved to consider the flexibility of both, the receptor and the ligand. Hence, the appearance of more dynamic models of binding such as, for example, the zipper model, the induced-fit model, and the conformational-selection model (see [FW08] for a more comprehensive review). But, as noted by Kahraman et al. [KMLT07], shape complementarity is a necessary but not sufficient requirement for ligand binding. Indeed, molecular recognition also depends on physicochemical complementarity, which has to do with, for example, the electrostatic, hydrogen-bonding, hydrophobic, and solvent-mediated interactions between the protein and ligand.

An important requirement to design a new protein cavity detection method is to agree on the definition of what is a cavity after all. As discussed in [SLD$^+$17], there is no consensual definition about what protein cavity is. Nevertheless, this thesis we will use the definition put forward by Simões et al. [SLD$^+$17], namely: “A cavity is a connected component of the complement space of the protein inside its convex hull”. This general definition emphasizes the three-dimensionality of a cavity. A protein cavity is quite often called *putative* binding site. However, a binding site is defined as an already known functional site that binds a ligand, while a protein cavity does not necessarily correspond to a binding site.

In general, cavity prediction methods can be split into three main categories: energetic, evolutionary, and geometric methods [NSG12]. As known, the performance of these three categories of methods is very similar in terms of accuracy. Nevertheless, geometry-based algorithms are known to be faster than energy- and evolution-based ones, with the advantage that they are more robust against structural variations or missing atoms/residues of the input structure [SSE$^+$10]. Furthermore, evolutionary methods depend on the number of available sequences and the quality of the alignment tool. Hence, this thesis focuses on geometry-based algorithms.

## 1.2 Problem Definition

Geometry-based cavity detection algorithms cover a variety of techniques, including space partitioning of the bounding box enclosing the protein, fitting of probe spheres into the solvent-accessible gaps of the protein, and using Delaunay triangulations and $\alpha$-spheres [SSE$^+$10, KKL$^+$16, SLD$^+$17].

As said elsewhere, the main two problems of these methods are the following: accuracy and performance. In this thesis, we chiefly tackle the problem of accuracy somehow, although running speed is another critical issue to take into consideration in the future. Our understanding at the moment is that performance can be addressed using massive GPU computing. It happens that, no GPU computing resources were used to design and implement the methods described in this thesis, so there is no need to focus on time performance or running speed.

Accuracy has much to do with the following four issues extensively discussed in the survey included in this thesis as second chapter:

- *User-assisted cavity location* (UACL). This issue may arise in any category of methods. It comes up when the user is required to assist in the location of a protein cavity, who roughly indicates the position of the cavity in an interactive manner. It is up the method to then determine the extent of such cavity.

- *Grid-spacing sensitivity* (GSS). Grid spacing is a crucial parameter in grid-based methods or any other method that leverages the use of a grid to partition the bounding box where the protein lies in. By varying the grid spacing, one may obtain a distinct number of cavities. This issue is rather difficult to control in the sense that it is not easy to find a grid spacing tradeoff between accuracy and speed.

- *Mouth-opening ambiguity* (MOA). This is a general issue in the sense that potentially it may arise in any method. The issue is related with a method's inability to delineate the mouth openings of each cavity; that is, it is required to know where a cavity starts and ends, though it may possess various entries or exits.
- *Protein-orientation sensitivity* (POS). This issue mostly arises in methods built upon the gridification or partitioning of the space embedding the protein. As shown in the second chapter, grid-based methods are not rotation-invariant, unless we use additional tools to mitigate the protein-orientation sensitivity (POS) as, for example, the convex hull or the like. This issue is closely related to the MOA issue because POS can be overcome by correctly placing stopgaps at the entrances and exits of cavities.

It is worth noting that most methods proposed over the years continue to present relatively low rates of accuracy in the detection of cavities (i.e., putative binding sites) in proteins. Such detection methods are fundamental tools in the discovery of new binding sites in proteins; that is, cavities for which there are not any known binding drugs.

## 1.3 Research Hypothesis

This thesis aims to introduce geometric techniques borrowed from computer graphics into the field of molecular graphics and modeling, in particular, to come up with more accurate geometric methods to detect cavities in proteins than those extant in the literature. Let us mention that, apart from those particular issues (UACL, GSS, MOA, and POS), the long-standing problems in protein cavity methods are *accuracy* and *performance*. In the present doctoral research work, the primary objective is to find ways to improve the accuracy of methods, without noticeably losing performance. Therefore, this work sustains on a careful review of the literature (see the second chapter of the present thesis), with a deep understanding of the nature of each geometric method, their strengths, and weaknesses. This literature review led the candidate to the study of the part segmentation tools in computer graphics and geometric computing, looking for inspiration and insights, as needed to solve the problems mentioned above.

Thus, the research work here described builds upon the following *thesis statement*:

> *It is feasible to design protein cavity methods that are more accurate than the state-of-the-art methods using geometric concepts, shape descriptors, and techniques commonly used in computer graphics.*

In this thesis, one intends to explore mathematical formulations of (molecular) Gaussian surface, shape descriptors like shape diameter function (SDF) [SSCO08], and spectral analysis (i.e., eigenvalues and Morse theory), and the like, to detect cavities in proteins. Furthermore, one also intends to study the non-geometric factors that determine the overall accuracy of each protein cavity method.

## 1.4 Research Plan

Following the thesis statement above, the research plan underpinning this thesis was organized into the following tasks:

- *Survey.* Writing a survey on protein cavity detection methods was the first task to be completed during the doctoral programme. It is was instrumental to be aware of the outstanding problems (or issues) in the recognition of cavities in proteins.
- *Testing, Benchmarking, and Validation.* This task aimed at evaluating and comparing the accuracy of publicly available cavity detection methods. As seen further ahead, this task pervades the entire research work that has led to the design and implementation of methods proposed in this thesis. This task has also originated the building of a benchmark tool, called CavBench (see Section 1.6), for comparing detection methods relative to a ground-truth dataset of known binding sites.
- *A grid-and-surface method to detect protein cavities through computer graphics tools and techniques.* This task has originated a "new protein cavity method, called CavVis, which combines voxelization (i.e., a grid of voxels) and an analytic formulation of Gaussian surfaces that approximates the solvent-excluded surface (SES). This method builds upon visibility of points on protein surface to find its cavities. Specifically, the visibility criterion combines three concepts we borrow from computer graphics, the field-of-view (FoV) of each surface point, voxel ray casting, and back-face culling." (Taken from the abstract of the article included in Chapter 3).
- *A surface-based method to detect protein cavities using the concept of SDF-based mesh segmentation as usual in computer graphics.* This task has addressed the problem of finding cavities on protein surfaces as a mesh segmentation problem as usual in computer graphics. With this segmentation idea in mind, the protein surface was segmented using "the concept of shape diameter function (SDF). SDF is a scalar function that measures the diameter of the interior volume of a closed surface in the neighborhood of each one of its points. Interestingly, when applied to Gaussian surfaces, which are modeled by the sum of Gaussian functions that are positive everywhere, SDF ends up measuring the diameter of the exterior volume of the protein exterior, from which we can infer the location and extent of protein

cavities. Unlike other cavity detection methods, SDF is largely independent of pose changes of the protein and holds similar values in separate cavities, allowing us to identify such cavities using clustering of points with similar SDF values." (Taken from the abstract of the article included in Chapter 4).

- *A grid-based method to detect protein cavities using spectral analysis as usual in computer graphics.* This task has led to the construction of a new protein cavity detection method called CavSeeker. It was developed from the key idea that "cavities are located around specific critical points of the electron density field of the protein. Basically, CavSeeker finds the voxels that are transverse to two iso-surfaces of the electron density field of the protein, between which one can find the critical points and their corresponding cavities." (Taken from the abstract of the article included in Chapter 5). Recall that in spectral analysis, the critical points are those points at which the gradient of the scalar field or function vanishes; in particular, cavities correspond to locations of saddles and minima, which are determined from the eigenvalues of the Hessian matrix.
- *Thesis Writing.* The present thesis consists of a collection of scientific articles. Such articles correspond to the second, third, fourth, and fifth chapters. This thesis also includes an introductory chapter, as well as the final chapter, where one draws the main conclusions and points out some hints for future work.

As a final remark about the research plan, let us mention this thesis has been mainly focused on solutions addressing issues related to the accuracy in the detection of cavities in proteins. Time performance issues have not been considered as much relevant as accuracy issues.

## 1.5 Main Contributions

The main contributions of the research work that underpins this thesis have resulted from the need in responding to the thesis statement above. Such scientific contributions are the following:

- *Shape descriptors.* As seen throughout the thesis, borrowing shape descriptors and mesh segmentation techniques from the computer graphics field to carry out protein segmentation into cavities and their complement ends up being reasonable and feasible. Seemingly, this is a research track to further explore in the future.
- *Clustering.* The accuracy of computer graphics-based methods tends to significantly increase if they incorporate adequate clustering techniques in the formation of cavities. Therefore, it seems now straightforward that using machine learning techniques to get better delineated clusters (or cavities) is a research avenue to pursue in the future.

- *Filtering.* Also, the accuracy of such methods further increases if they take advantage of volume-based filtering techniques, discarding small cavities when ligands cannot get in. It seems also clear that it is necessary to investigate new filtering techniques in addition to those based on cavity volumes.

Summing up, we now have a better understanding there is a long way to trek to come up with protein segmentation methods that leverage the panoply of mesh segmentation techniques developed in computer graphics in the last twenty years. Otherwise, it will be rather difficult to reach accuracy rates over 90 percent in the detection of protein cavities.

## 1.6 Publications

The doctoral research work behind this thesis has originated the following publications:

- **Geometric detection algorithms for cavities on protein surfaces in molecular graphics: a survey**
  Tiago Simões, Daniel Lopes, Sérgio Dias, Francisco Fernandes, João Pereira, Joaquim Jorge, Chandrajit Bajaj, and Abel Gomes.
  *Computer Graphics Forum*, Vol. 36, No. 8, pp. 643-683, June 2017.
  DOI: `https://doi.org/10.1111/cgf.13158`

- **CavVis — A field-of-view geometric algorithm for protein cavity detection**
  Tiago Simões and Abel Gomes.
  *Journal of Chemical Information and Modeling*, Vol. 59, No. 2, pp. 786-796, January 2019.
  DOI: `https://doi.org/10.1021/acs.jcim.8b00572`

- **CavShape — A cavity detection algorithm through the multivariate shape diameter function**
  Tiago Simões and Abel Gomes.
  *PLOS Computational Biology* (submitted for publication).

- **CavSeeker — Identifying protein cavities by locating critical points of the scalar field generated from summation of Gaussians**
  Tiago Simões and Abel Gomes.
  *Bioinformatics* (submitted for publication).

Other publications not included in this thesis:

- ***CavBench: A benchmark for protein cavity detection methods***
  Sérgio E.D. Dias, Tiago M.C. Simões, Francisco Fernandes, Ana Mafalda Martins, Alfredo Ferreira, Joaquim A. Jorge, and Abel J.P. Gomes.
  *PLOS ONE* (submitted for publication).

## 1.7 Organization of the Thesis

This doctoral thesis is organized as follows:

**Chapter 1**: This chapter provides the "big picture" of the research work that has led to the present thesis, including the motivation, problem definition, research hypothesis, research plan, and thesis organization. Additionally, the main contributions and publications are also listed, as well as some avenues for future work.

**Chapter 2**: This chapter surveys geometric-based methods for the detection of cavities in proteins. This survey builds upon a mathematical definition of what a protein cavity is, as well as a taxonomy for protein cavities. Then, different families are approached and discussed relative to issues like user-assisted cavity location (UACL), mouth-opening ambiguity (MOA), grid-spacing sensitivity (GSS), and protein-orientation sensitivity (POS).

**Chapter 3**: This chapter describes a new grid-and-surface-based method, called CavVis. This method takes advantage of computer graphics concepts and techniques like the field-of-view, voxel ray casting, and back-face culling, in conjunction with the analytical formulation of the Gaussian molecular surface to predict the location of protein cavities or putative binding sites.

**Chapter 4**: This chapter proposes a new surface-based method, called CavShape. It introduces the shape descriptor named *multivariate* shape diameter function (mSDF), which is a follow-up of the SDF shape descriptor used in mesh skeletonization and segmentation in the field of computer graphics [SSCO08]. Here, mSDF is used to find protein cavities; that is, to find a protein segmentation into cavities and their complement.

**Chapter 5**: This chapter introduces a grid-based method, called CavSeeker. This method builds upon the eigenvalue analysis to determine the minima and saddle points of the Gaussian scalar field of the protein. Such critical points indicate where cavities are located outside the protein. In a way, we can say that this method takes advantage of the Morse theory in differential geometry.

**Chapter 6**: The final chapter of this doctoral thesis reinforces the main contributions of this thesis, draws the most relevant conclusions, and points out some hints for future work.

## 1.8 References

[FW08] A. Feldman-Salit and R. C. Wade. Molecular recognition: computational analysis and modelling. *Wiley Encyclopedia of Chemical Biology*, pages 1–10, 2008. xiii, 1

[KKL$^+$16] M. Krone, B. Kozlíková, N. Lindow, M. Baaden, D. Baum, J. Parulek, H.-C. Hege, and I. Viola. Visual analysis of biomolecular cavities: state of the art. In *Computer Graphics Forum*, volume 35, pages 527–551. Wiley Online Library, 2016. xiv, 2

[KMLT07] A. Kahraman, R. J. Morris, R. A. Laskowski, and J. M. Thornton. Shape variation in protein binding pockets and their ligands. *Journal of Molecular Biology*, 368(1):283–301, 2007. xiii, 1

[NSG12] B. Nisius, F. Sha, and H. Gohlke. Structure-based computational analysis of protein binding sites for function and druggability prediction. *Journal of Biotechnology*, 159(3):123–134, 2012. xiv, 2

[SLD$^+$17] T. Simões, D. Lopes, S. Dias, F. Fernandes, J. Pereira, J. Jorge, C. Bajaj, and A. Gomes. Geometric detection algorithms for cavities on protein surfaces in molecular graphics: a survey. In *Computer Graphics Forum*, volume 36, pages 643–683. Wiley Online Library, 2017. xiv, 2

[SSCO08] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249, 2008. xvi, xx, 4, 7

[SSE$^+$10] P. Schmidtke, C. Souaille, F. Estienne, N. Baurin, and R. T. Kroemer. Large-scale comparison of four binding site detection algorithms. *Journal of Chemical Information and Modeling*, 50(12):2191–2200, 2010. xiv, 2

# Chapter 2

## Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey

This chapter concerns the following article:

Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey
Tiago Simões, Daniel Lopes, Sérgio Dias, Francisco Fernandes, João Pereira, Joaquim Jorge, Chandrajit Bajaj, and Abel Gomes.
*Computer Graphics Forum*, Vol. 36, No. 8, pp. 643-683, June 2017.

### Overview

This chapter surveys methods for the detection of cavities in proteins, in particular those that use geometric properties of the protein to identify cavities or putative binding sites. A taxonomy for geometric methods is also presented. Additionally, some issues are discussed for each family of methods, namely: grid-spacing sensitivity (GSS); mouth-opening ambiguity (MOA); and protein-orientation sensitivity (POS). Finally, some challenges in cavity detection are put forward for future research. These include: new and more advanced protein surface formulations; new geometric segmentation techniques; and new techniques in the field of dynamic or time-varying methods.

The issues mentioned above are part of the problem related to low accuracy of geometric methods. Indeed, increasing the accuracy is the main objective of the algorithms described in this thesis, in particular those described in Chapters 3, 4, and 5. For that purpose, this thesis introduces three algorithms based on computer graphics concepts and techniques. In a way, this thesis shows how to apply computer graphics algorithms to computational biology and chemistry. Therefore, this thesis fits in the field of molecular graphics.

DOI: 10.1111/cgf.13158

COMPUTER GRAPHICS *forum*
*Volume 36 (2017), number 8 pp. 643–683*

# Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey

Tiago Simões[1,2], Daniel Lopes[3], Sérgio Dias[1,2], Francisco Fernandes[3], João Pereira[3,4], Joaquim Jorge[3,4], Chandrajit Bajaj[5] and Abel Gomes[1,2]

[1]Instituto de Telecomunicações, Portugal
tiagomiguelcs@gmail.com, sergioduartedias@sapo.pt, agomes@di.ubi.pt
[2]Universidade da Beira Interior, Portugal
[3]INESC-ID Lisboa, Portugal
dsl.7125@gmail.com, francisco.fernandes@ist.utl.pt, {jap, jaj}@inesc-id.pt
[4]Instituto Superior Técnico, Universidade de Lisboa, Portugal
[5]The University of Texas at Austin, Texas, USA
bajaj@cs.utexas.edu

**Abstract**
*Detecting and analysing protein cavities provides significant information about active sites for biological processes (e.g. protein–protein or protein–ligand binding) in molecular graphics and modelling. Using the three-dimensional (3D) structure of a given protein (i.e. atom types and their locations in 3D) as retrieved from a PDB (Protein Data Bank) file, it is now computationally viable to determine a description of these cavities. Such cavities correspond to pockets, clefts, invaginations, voids, tunnels, channels and grooves on the surface of a given protein. In this work, we survey the literature on protein cavity computation and classify algorithmic approaches into three categories: evolution-based, energy-based and geometry-based. Our survey focuses on geometric algorithms, whose taxonomy is extended to include not only sphere-, grid- and tessellation-based methods, but also surface-based, hybrid geometric, consensus and time-varying methods. Finally, we detail those techniques that have been customized for GPU (graphics processing unit) computing.*

**Keywords:** biological modelling, modelling, geometric modelling, computational geometry

**ACM CCS:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.3.8 [Computer Graphics]: Applications – Molecular Graphics; J.3 [Life and Medical Sciences]: Biology and Genetics – Computational Biology

## 1. Introduction

In 1894, Fischer conducted pioneer studies on detection of protein cavities [LS94]. From these studies, he concluded that the binding of a molecule to another is similar to the paradigm of inserting a key into a lock. In other words, this means that the affinity between two molecules exists if the shape of one molecule matches the shape of the other. However, this model was considered to be overly simplistic, because shape cannot be the only factor that influences the detection of protein cavities since proteins are highly flexible and change shape over time. Generally speaking, protein binding sites are specific, large and deep clefts [LLST96]. However, protein shape can vary considerably, depending on the protein we have at hand. For example, the protein binding site of a ribonuclease is an extended rut or groove, while the protein binding site of an endonuclease is a spherical cavity, and for an enzyme, it is usually the largest cavity [LWE98].

In fact, a protein can bind many types of molecules, largely because of its non-negligible number of cavities. Indeed, many properties can be inferred from these molecular regions, furthering our understanding of molecular interfaces and interaction regions. This, in turn, provides valuable information for the design of complementary compounds, that may act as active protein inhibitors or disruptors of protein–protein interactions. In general, those binding site regions have large surface areas and correspond to concave, cleft or hole-shaped regions on a protein surface [KG07]. For all these reasons and more, it becomes necessary to develop accurate tools to characterize protein cavities. Cavity properties of interest include its geometry such as shape, size and depth, and also its associated

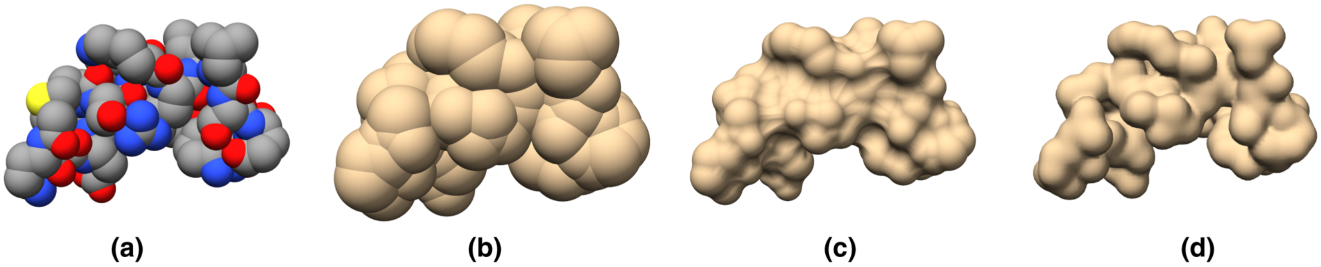


**Figure 1:** *(a) Van der Waals surface; (b) SAS surface; (c) SES surface; (d) Gaussian surface. Images generated with UCSF Chimera [PGH*04] for protein 1wbr.*

biochemical and biophysical properties, such as pH, electrostatics, hydrogen-bonding propensity, etc. It is the conjugation of all of these factors that enable a ligand (small molecule) or another protein to recognize the correct place to bind a given target protein [Kub06].

To make laboratory experiences easier, it would be helpful to have computational methods capable of simulating biochemical processes underlying protein–ligand interactions. However, these biochemical processes are tough to recreate *in silico*. These difficulties are related to the variety of suitable ligands, the variety of protein cavities, the protein shape variations themselves and the physicochemical factors that act on cavity regions.

Such regions usually correspond to pockets, clefts, inner cavities and grooves on protein surfaces. Therefore, a better understanding of the process entangled in binding proteins requires the detection of cavities on the molecular surfaces. A computational estimate of the location of such protein regions may be instrumental in improving the design of new drugs, before initiating any experimental laboratory work in the drug discovery process. For that purpose, many algorithms for predicting and identifying protein cavities have been developed so far. Such algorithms divide into three broad categories:

- *Evolutionary algorithms*: They rely on multiple sequence alignments to find the location of cavities on a given protein surface (e.g. ConSurf [AGBT01], Rate4Site [PBM*02] and GarLig [PPG10]).
- *Energy-based algorithms*: In this case, cavities are detected by computing the interaction energies between protein atoms and a small-molecule probe (e.g. Grid [Goo85], QSiteFinder [LJ05] and AutoLigand [HOG08]).
- *Geometric algorithms*: These algorithms analyse the geometric properties of a molecular surface to detect cavities (e.g. SURFNET [Las95], LIGSITE [HRB97] and PocketDepth [KC08]).

Each approach has its drawbacks. For example, geometric methods relying on a grid are sensitive both to protein orientation and grid spacing. Energy-based methods depend on their filtering procedures, force field parameterizations and scoring functions. In turn, evolutionary-based methods depend on the quality of the alignment tool, and also on the number of available sequences. These problems show us that there is still a long way to go in this field so that there is a need for further analysis of all the processes involved in detecting binding sites of proteins [KG07]. This explains why detecting molecular cavities still is a very active research area [HSAH*09].

Although several authors have surveyed cavity detection algorithms [GS11, ZGWW12, BCG*13, Duk13, KSL*15], these surveys only present brief citations backed by summary descriptions, that is, they do not provide enough detail on the algorithms. Furthermore, these surveys agree on a simplified classification of cavity detection algorithms into the following classes: sphere-based, grid-based and Voronoi-based. More importantly, such surveys lack a critical comparison between algorithms. As an exception, a more detailed survey focusing on the visual analysis of biomolecular cavities was recently published [KKL*16], that is, with a flavour in molecular visualization. On the contrary, our survey adopts a more geometry-based approach to protein cavity detection.

This survey falls in the scope of molecular graphics and modelling, that is, a research area at the intersection of computational biology, bioinformatics, computational geometry and computer graphics. More specifically, this paper approaches the computer graphics and computational geometry side of cavity detection methods, that is, the geometry of proteins; hence, the focus is on geometry-based algorithms for identifying cavities on protein surfaces such as those depicted in Figure 1. As mentioned above, geometric methods for detecting cavities on proteins fall into three main categories: grid-based, sphere-based and Voronoi-based. We extend this classification of geometric methods as a tool to organize the survey itself, as illustrated in Figure 2.

## 2. Background

There has been considerable work on cavity detection for molecules. This is especially relevant for molecular docking and related problems. A molecule is considered to be an orderly grouping of atoms bound by favourable chemical connections [JKSS96, WM97]. In particular, the family of biomolecules spans the building blocks of living organisms. This family includes large macromolecules, namely, *proteins*, polysaccharides, lipids, nucleic acids and small molecules (e.g. primary metabolites and secondary metabolites). In this paper, we are interested in proteins and their cavities, where their interactions with ligands usually take place.

### 2.1. Proteins

Proteins constitute about 20% of the human body, and play a crucial role in most biological processes. Amino acids are the building

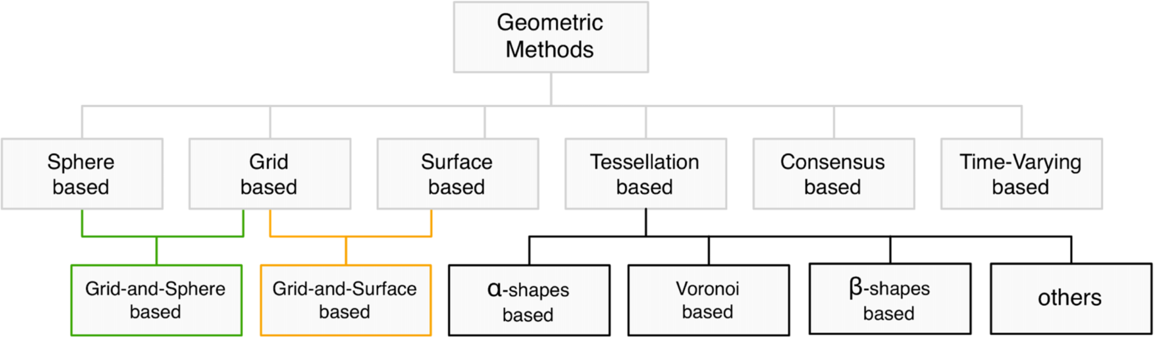


**Figure 2:** *Taxonomy of geometry-based methods.*

blocks that make up proteins [Whi05]. In summary terms, a protein can be understood at four distinct structural levels [AJL*07]. The *primary structure* of a protein is given by its sequence (or chain) of amino acids. The *secondary structure* of a protein comprises amino acid subsequences that exhibit a specific structural regularity. These secondary regular structures are known as alpha-helices (alpha-helixes) and beta-pleated sheets (beta-sheets). Alternatively, the secondary structure can be defined using the regularity of backbone dihedral angles of amino acid residues. The *tertiary structure* denotes the geometric shape of a given protein, that is, it refers to the folding of the whole protein chain (including the secondary structures) into its final three-dimensional (3D) shape. Recall that it is the protein folding that makes the protein acquire its functional shape or conformation. Also, many proteins have two or more polypeptide chains or tertiary structures that are held together by the same non-covalent forces as those of tertiary structures, that is, many proteins can fold into a *quaternary structure*, resulting in a protein complex.

### 2.2. Protein surfaces

Taking into consideration that proteins fold in an aqueous medium, that is, soluble biomolecules adopt their stable conformation in water (hydrophobic effect) [Sim03], we can think of protein cavities as recesses on a protein surface where water can enter and stay for some time. Therefore, detecting protein cavities depends on features found on the protein surface.

In the literature, we find many mathematical formulations of protein surfaces, namely: van der Waals surface (vdW), solvent-excluded surface (SES), solvent-accessible surface (SAS) and Gaussian surface, as illustrated in Figure 1. For modelling purposes, atoms are often conceptualized as hard spheres, but that is not true because their electronic fields partially overlap within a molecule (e.g. a protein). The *van der Waals surface* is given by the surface of the union of such atomic spheres [LR71] [Whi97], as shown in Figure 1(a).

Initially proposed by Lee and Richards [LR71], SAS was introduced to model the molecular hydrophobic effect using the vdW surface plus a probe sphere of radius 1.4 Å featuring the water molecule. SAS is the surface generated by tracing the centre of the water probe sphere rolling on the vdW surface. In mathematical terms, SAS is also defined as the surface of the union of atomic spheres, but with their radii increased by 1.4 Å. Obviously, SAS is bulkier than van der Waals surface, because the water is taken into account on the molecule, as shown in Figure 1(b).

SES was introduced by Richards [Ric77] (see Figure 1c), and also uses the rolling probe sphere as SAS, that is, the probe sphere featuring the water molecule rolls on the vdW surface. SES consists of two parts, the contact surface and the reentrant surface. The contact surface comprises disconnected patches of the vdW surface that enters in contact with the probe, while the reentrant surface is made up of disconnected patches resulting from the interior-facing part of the probe when it enters simultaneously in contact with two or more atoms. SES is the union of these contact and reentrant patches, resulting in a connected surface.

*Gaussian surface* is an analytical formulation for molecular surfaces that results from summing up Gaussian functions representing the electronic density fields of atoms that form a molecule [Bli82] (see Figure 1d). The Gaussian surface is smooth because the subsidiary functions decay smoothly to zero with the distance to each atom centre.

It is clear that cavity detection algorithms usually start with the reading of the set of atoms in memory, that is, they inherently use the vdW surface. But, as explained throughout the paper, there is a trend to use analytical surfaces like SES and Gaussian surfaces to detect cavities using geometric properties as of differential geometry, as is usual in segmentation techniques studied in computer graphics [PTRV12] [DG17].

### 2.3. Protein cavities

Seemingly, there is no consensus about the definition of cavity, neither about the classification of cavities. Terms such as cavity, pocket, channel, tunnel, void and cleft are often used in a slightly different way, or even not being defined at all. Some authors describe a pocket as a non-flat and concave molecular surface feature [LJ06, HSAH*09, PSM*10, CS10, GS13], so that pockets and cavities are used interchangeably. Other authors define a cavity as an inner region inside the molecular surface [HRB97, BHH*10, VG10,

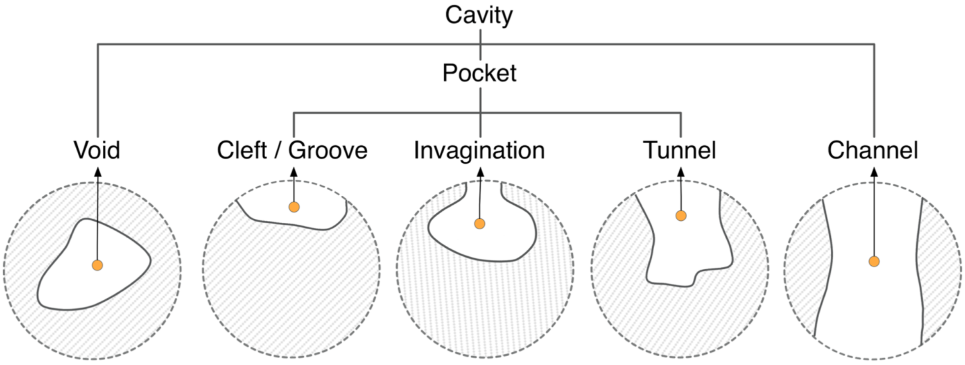


**Figure 3:** *Types of cavities.*

OMV11, KLKK16], which may lead to the idea that a cavity is a void. It is also observed in the literature that there is an unclear distinction between tunnels and channels [POB*06, OFH*14, PEG*14]. In fact, a formal mathematical definition of protein cavity remains absent in the literature [OFH*14].

How can we define a protein cavity? Informally, we can say that a cavity is a concavity on the protein surface. This leads us to put the theory of convexity in the context of geometric cavity detection methods. Apart its generality, the advantage of using the mathematical theory of convexity [Lay82] is that it provides a formal definition of protein cavity, as follows:

> *A **cavity** is a connected component of the complement space of the protein inside its convex hull.*

Note that the concept of connected component is topological, and has to do with the first Betti number $\beta_0$ (the number of connected components) of such complement space [Hat02]. Looking at the protein itself as a shape in 3D, we know that its connected components, channels and voids correspond to Betti numbers $\beta_i$ ($i = 0, 1, 2$) in 3D. It is clear that these channels and voids belong the complement space. The remaining connected components of the complement space are pockets. In short, we can breakdown cavities into three classes: *pockets*, *channels* and *voids* (see Figure 3).

The detection of cavities is mostly based on the hydrophobic effect of water on the protein surface; it is assumed that the water molecule is approximately a ball of 1.4 Å of radius. Nevertheless, some channels do not control the flow of water molecules; for example, ion channels control the flow of ions. But, in general, cavities are assumed to be located where the water molecule gets in without slipping on the surface. A major problem with detection of protein cavities has to do with delineating the boundary of each cavity on the protein surface, which consists of zero or more surface contours, called mouth openings. In this sense, a cavity refers to a $m$-ary cavity, with $m \in \mathbb{N}$ standing for the number of mouth openings to the outside; for example, a void is a 0-ary cavity, that is, a cavity without mouth openings, a pocket is a 1-ary cavity with a single mouth, while a channel is 2-ary cavity. Note that, from topology's point of view, a $m$-ary cavity ($m \geq 3$) is a set of $m - 1$ 2-ary cavities of the protein in 3D, which is nothing more than the first Betti number, that is, $\beta_1 = m - 1$. Some of these cavities play an important role in the function of proteins because they are the suitable sites for binding of ligands [GS13].

Summing up, a protein in 3D may only possess three types of cavities: pockets (0-ary cavities), channels (1-ary cavities) and voids (2-ary cavities). Pockets include clefts, grooves, invaginations and tunnels. A pocket may have zero or more chambers without direct contact with the outside, though they are reachable from outside through a tunnel. Clefts and grooves have no chambers nor tunnels. An invagination is a pocket with a single chamber and no tunnel. A tunnel is a pocket without chambers. As shown in Figure 4, a pocket can be made of recesses, tunnels and chambers. Similarly, channels and voids may also possess recesses, tunnels and chambers.

## 3. Sphere-Based Algorithms

Sphere-based algorithms are based on the concept of probe sphere.

### 3.1. Kuntz *et al.*'s method

The first sphere-based method was proposed by Kuntz *et al.* [KBO*82], though in the context of geometric docking between a macromolecule and ligands. In fact, the receptor and the ligand are both represented as SES. The cavities of the receptor are filled with probe spheres, and the ligand itself is filled by probe spheres in both cases tangent to the surface points. Then, shape matching operations between the ligand and the receptor probe spheres are approximated under rigid transformations of the ligand. Furthermore, the overlap is evaluated to detect cavities that fit with the ligand. Note that the SES is given as a set of surface points with normals. For further details about probes and receptor–ligand matching, the reader is referred to [KBO*82]. Indeed, the most important aspect of this method is that it is the first method based on the geometry of the ligand.

### 3.2. HOLE

This method is specialized in tracking channels or holes through proteins [SGW93] [SNW*96]. It requires that the user indicates the seed point inside the channel and vector that represents the direction of the channel approximately. A probe sphere is then centred at the seed point without overlapping the atoms bordering the channel.

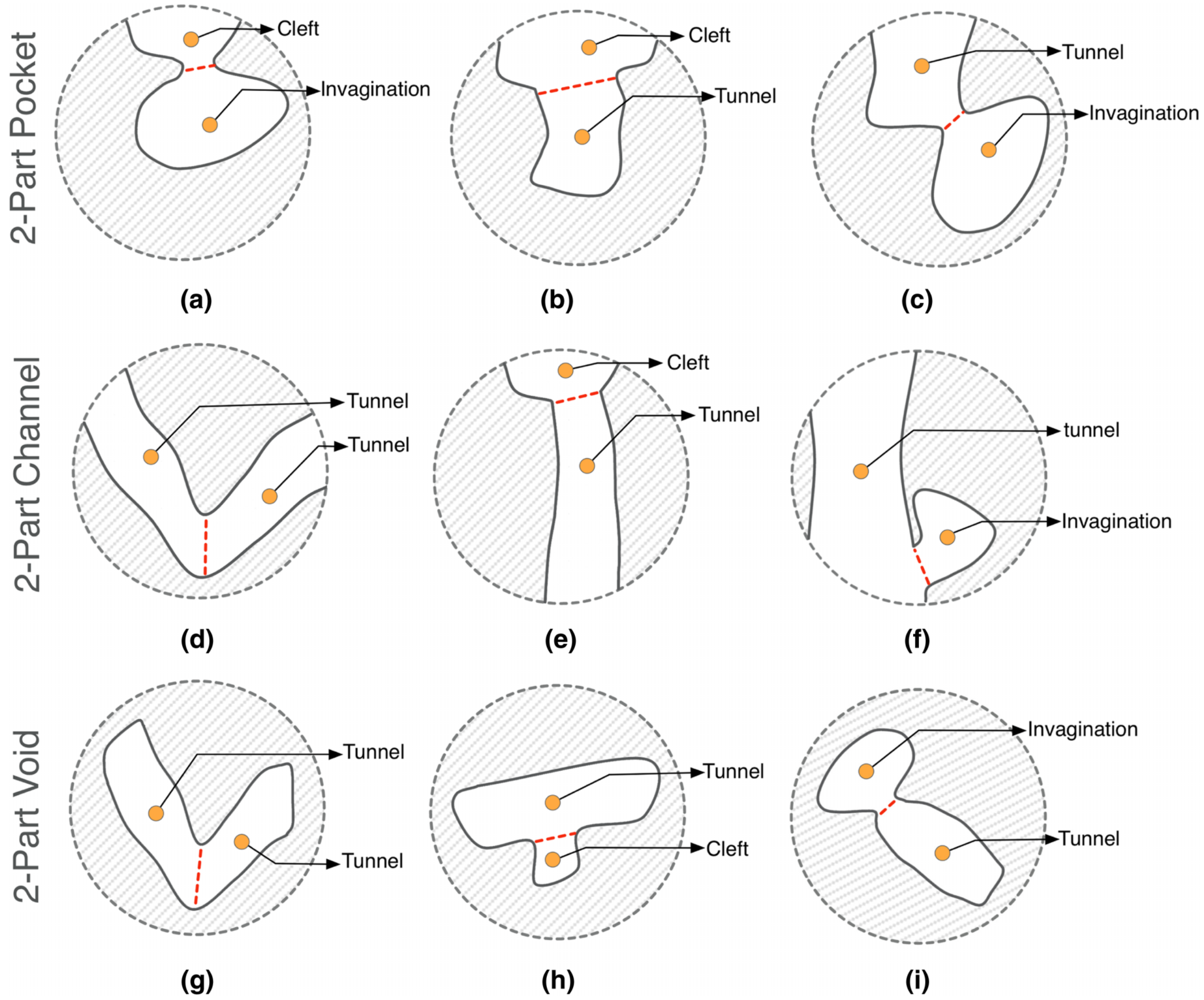


**Figure 4:** *Hierarchical 2-part pocket, channel and void examples. (a) A pocket composed by a cleft and a invagination; (b) A pocket composed by a cleft and a tunnel; (c) A pocket composed by a tunnel and a invagination; (d) A channel composed by two tunnels; (e) A channel composed by a cleft and a tunnel; (f) A channel composed by a tunnel and a invagination; (g) A void composed by two tunnels; (h) A void composed by a tunnel and a cleft; (i) A void composed by a invagination and a tunnel.*

Then, the probe sphere is moved along the channel, with its radius being adjusted using the Monte Carlo simulated annealing procedure [MRR*53] [KJV83]. Similar to Kuntz *et al.*'s method, HOLE utilizes large probe spheres of 5 Å radius as stopgap or delimiter of channels.

### 3.3. SURFNET

**SURFNET** proposed by Laskowski [Las95] is similar to the method proposed by Kuntz *et al.* [KBO*82]. Therefore, its leading idea is also to fill in cavities with probe spheres of varying sizes. However, it differs from Kuntz *et al.*'s method in the computation of probe spheres. Basically, for every pair of relevant atoms, we place a probe sphere centred at the midpoint of their atomic centres. Then the radius of the probe sphere is adjusted to guarantee that it does not overlap with any neighbouring atoms, as illustrated in Figure 5.

### 3.4. PASS

**PASS** (Putative Active Sites with Spheres) is another sphere-based algorithm [BS00]. Cavity filling with probe spheres is carried out in layers, based on three-point Connolly-like sphere geometry [Con83]. That is, the placement of probe spheres of the first layer is performed by looping over triplets of overlapping protein atoms, computing then the three locations at which a probe sphere is tangential to such atoms, as shown in Figure 6(a). The first layer on the surface consists of probes with radius of 1.8 Å for protein without hydrogen atoms; this radius is 1.5 Å if the hydrogen atoms are taken into account. The subsequent layers accrete probes with 0.7 Å of radius.

The retained probes must satisfy three conditions: (i) they cannot overlap any atom (see counterexample red probes Figure 6c); (ii) they cannot overlap with one another (see some counterexample

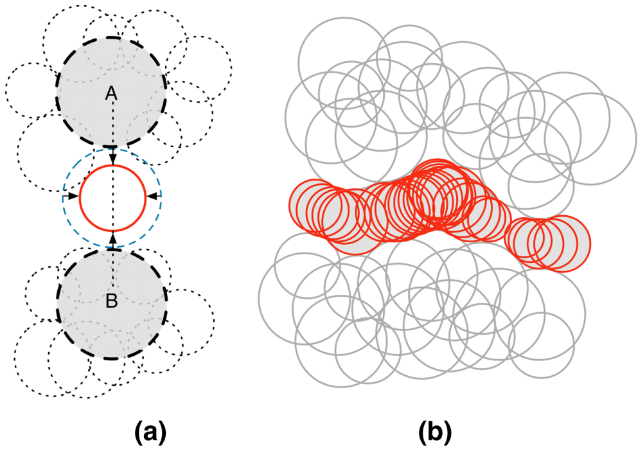


**Figure 5:** *Detecting cavities through SURFNET: (a) Each probe sphere is placed at the midpoint of a pair of atoms ($A$, $B$) but, if such probe sphere overlaps at least an atom (dashed spheres), its radius has to be reduced until it just has a tangential contact with the overlapped atom; (b) all probe spheres placed into cavity after considering all pairs of atoms and the surface enclosing of the cavity (pictures taken and modified from [Las95]).*

red probes Figure 6c); (iii) the burial threshold of each probe must be greater than 55 atoms for hydrogen-free proteins and 75 for proteins with hydrogen atoms; these threshold values were obtained empirically. The buriedness of a probe is determined by the number of protein atoms that lie within an empirical radius of 8 Å, that is, each probe is given a burial count.

After the accretion and filtering steps (see Figure 6), it remains to determine the active site points (ASPs) of pockets, a single ASP per pocket. So, an ASP represents a potential binding site for a ligand. The ASP of each pocket is determined by identifying the central probe of the corresponding cluster of probe spheres with higher weight (also called probe weight), which depends on the burial count. See Brady and Stouten [BS00] for further details.

### 3.5. PHECOM

**PHECOM** (Probe-based HECOMi finder) is yet another sphere-based algorithm and was developed by Kawabata and Go [KG07]. Similar to PASS, it also uses the three-point Connolly-like sphere geometry (i.e. placing a sphere tangential to three atoms of the protein, see Connolly [Con83] for more details) to coat the protein with a set of small probe spheres; the radius of each small probe sphere was set to 1.87 Å, which corresponds to the size of a single methyl group ($-CH_3$), as illustrated in Figure 7(a). Additionally, PHECOM also produces a coating of the protein with large probe spheres, so that one removes small probe spheres that overlap with the large probe spheres, as shown in Figure 7(b). Doing so, one considers that a cavity is an empty space into which a small probe sphere gets in, but not a large probe sphere; for example, this is shown in Figure 7(c), where small probe spheres (in grey) overlap, indicating the location of a cavity. Note that the probe spheres are allowed to overlap with each other, but not with protein atoms.

### 3.6. dPredgeo

**dPredgeo** was developed by Schneider and Zacharias [SZ12]. It is similar to PHECOM because it also uses rolling probes. More specifically, it uses two types of probes with fixed radii. The first probe is 1.4 Å radius and approximates the water molecule, which rolls on the vdW surface of the protein. This rolling procedure of probes reduces itself to the placement of probes on the protein sur-

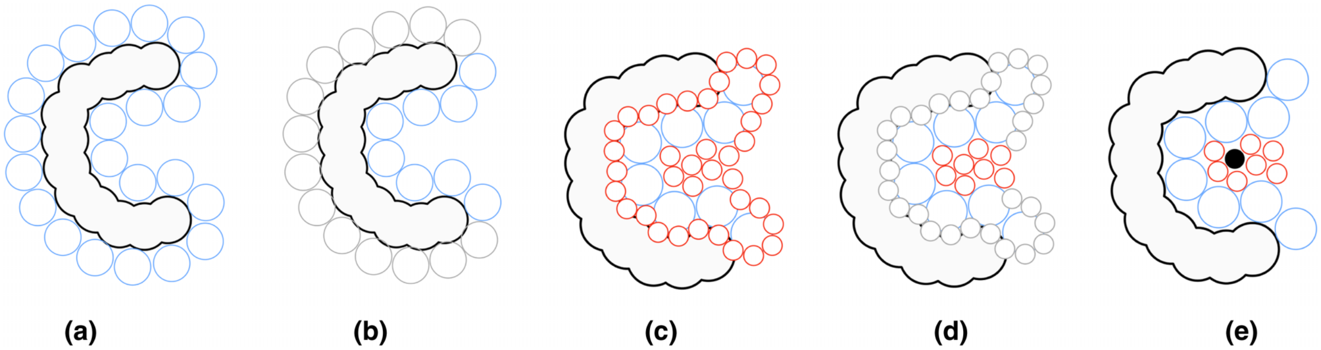


**Figure 6:** *Detecting cavities through PASS: (a) coating the molecular surface with the initial layer of probe spheres (blue spheres)—Probe spheres are tangentially placed to three atoms of the molecular surface; (b) probes of the initial layer (blue spheres) are filtered; they are removed from the initial layer if (i) overlap with any atom belonging to the protein surface, (ii) are in contact with any previous placed probes, and (iii) is at some extend less buried than other probes. In (b) a set of blue spheres, now represented as larger grey spheres, were removed because of (i); (c) more layers are added to the previous layer (red spheres); (d) spheres, as in (b), are filtered until we find an accretion layer that does not contain new probes (i.e. all probes were removed by the set of filters); In (d) a set of red spheres, now represented as smaller grey spheres, were removed because of (i) and (ii). The only remaining set of red spheres are those considered to be more buried on the molecular surface; (e) for each probe, its weight (PW) is computed and the ASP (black sphere) is identified in the cluster (pictures inspired in [WPS07] and [BS00]).*

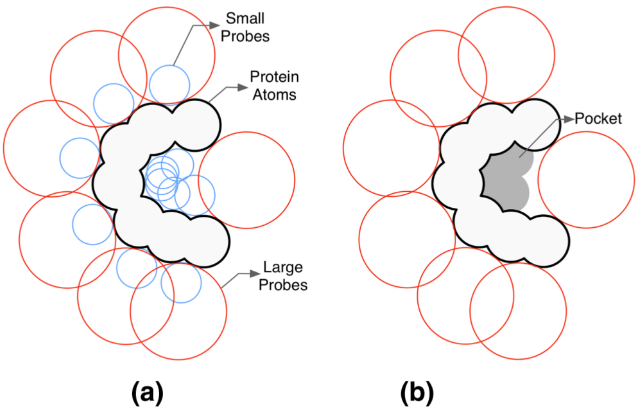


**Figure 7:** *Detecting cavities through PHECOM: (a) small and large probes are placed on the van der Waals surface; (b) small probes that overlap with the large ones are removed—The remaining set of small probes forms the pocket (taken and modified from [KG07]).*

face according to the principle of three-point geometry mentioned above. The same rolling procedure applies to set of larger probes with 4.5 Å of radius. As for PHECOM, these large probes solve the ambiguity problem that stems from the lack of a cavity stopgap. Then, one discards the small probes overlapping with large probes. Cavities are identified by clusters of the remaining small probes on the protein surface.

### 3.7. Sphere-based methods: Discussion

Table 1 summarizes the characteristics of sphere-based methods in the detection of cavities on protein surfaces. In this regard, we note the following:

- *Molecular Surfaces*. Sphere-based methods use the *set of atoms* (SA)—and thus the van der Waals surface indirectly—of each protein as the basis to identify cavities on the protein surface. The first three methods (Kuntz *et al.*, HOLE and SURFNET) use two-point sphere geometry, while the last three methods (PASS, PHECOM and dPredgeo) use tree-point Connolly-like sphere geometry [Con83].
- *Limitations*. One of the main problems of cavity detection methods has to do with automatically finding and delineating cavity boundaries, also called mouth openings, without ambiguity. But, unlike most sphere-based methods, HOLE requires the user provides a seed point inside each channel to start filling it with probe spheres. This means that, unlike most sphere-based methods, HOLE is not capable of determining cavities in an automated manner, that is, it uses *user-assisted cavity localization* (UACL). Note that HOLE has been designed only to identify channels.
  In general, sphere-based methods do not suffer from the problem of *mouth-opening ambiguity* (MOA). Kuntz *et al.*, HOLE and SURFNET use varying-radius probes (1.4 Å minimum) to fill cavities, though HOLE has been designed only to identify channels. This filling process stops when the probe sphere radius exceeds 5 Å, which works as the stopgap of the cavity; consequently, we can then delineate the corresponding mouth opening. Nevertheless, SURFNET does not utilize large probe spheres as stopgaps of cavities, because the placement of probe spheres in the empty space between pairs of atoms makes such large probes unnecessary.
  The remaining three methods (PASS, PHECOM and dPredgeo) use two constant-radius probes, a small probe (about 1.4 Å radius) and a large probe (with a radius greater than or equal 4 Å). These methods follow the principle that a cavity is a site where the small probe gets in, but the large probe does not. As noted above, large probe spheres can work as stopgaps (or delimiters) of cavities, so eliminating the MOA. However, these large probes are unnecessary for voids because every single void has no mouth opening.
- *Cavities*. In general, sphere-based methods are capable of detecting any cavity (see Table 1). This is so because these methods are capable of not only filling cavities with probe spheres but also to stop such a filling process. SURFNET utilizes a technique for bracketing probes in the empty space between every atom–atom pair, while PASS takes advantage of the concept of burial threshold; the remaining four methods use large probe spheres as stopgaps of cavities.

In the future, one might exploit the concept of mutual visibility for surface atoms as a way to further speed up sphere-based methods, making redundant the usage of empirically large probe spheres as

**Table 1:** *Sphere-based methods.*

| | | | | Cavities | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | Pockets | | | | |
| Methods | Reference | Molecular surfaces<br>SA/vdW | Limitations<br>UACL | Clefts/Grooves | Invaginations | Tunnels | Channels | Voids |
| Kuntz *et al.* | [KBO*82] | • | | • | • | • | • | • |
| HOLE | [SGW93] | • | • | | | • | • | |
| SURFNET | [Las95] | • | | • | • | • | • | • |
| PASS | [BS00] | • | | • | • | • | • | • |
| PHECOM | [KG07] | • | | • | • | • | • | • |
| dPredgeo | [SZ12] | • | | • | • | • | • | • |

Abbreviations: **SA/vdW**: set of atoms/van der Waals surface; **UACL**: user-assisted cavity location.

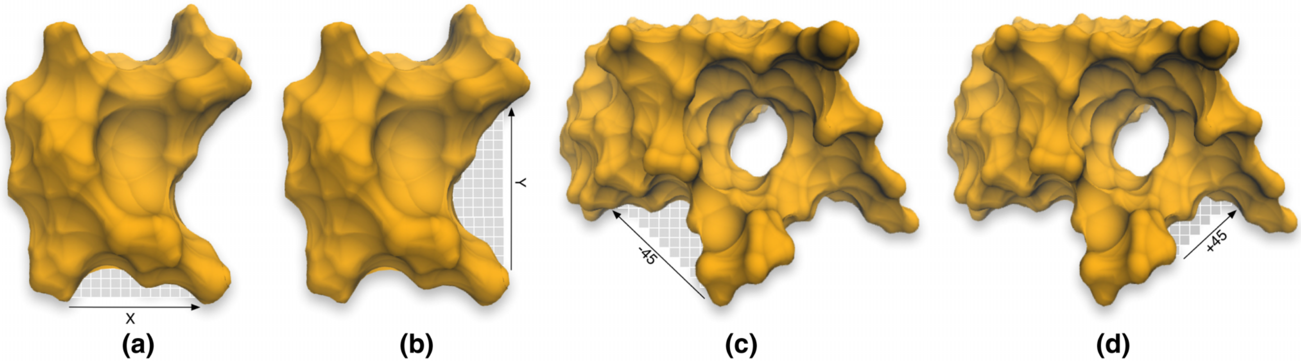


**Figure 8:** *Detecting cavities through POCKET (see [LB92]): (a) in the x-direction; (b) in the y-direction. Detecting cavities through LIGSITE (see [HRB97]): (c) in the −45°-direction; (d) in the +45°-direction.*

stopgaps of cavities. Note that these large empirical probes work well for small and medium size cavities, but not for shallow cavities, that is, sphere-based methods have problems with the identification of shallow cavities. In a way, such mutual visibility technique may be seen as a faster follow-up of SURFNET. Another way of improving the identification of cavities would be to consider the detection of $n$-part cavities.

## 4. Grid-Based Algorithms

Grid-based methods are characterized by the following: (i) they use an axis-aligned 3D dimensional lattice; (ii) they use a density map (i.e. a scalar field) so that each grid node is usually assigned an integer value, which gives rise to an integer grid map. Then, one uses some voxel clustering to collect relevant empty voxels into cavities.

### 4.1. CAVITY SEARCH

This method was introduced by Ho and Marshall [HM90]. It uses a slice-to-slice filling procedure for each cavity in a single direction, which is perpendicular to slices, to isolate and delineate the boundary of such cavity, thereby producing a cast (i.e. a cumulative set of slices of grid nodes or voxels) of the cavity. After filling in a slice of a given cavity using a 2D flood fill algorithm, we have to step forward to the next slice, repeating the filling procedure. However, this procedure suffers from two shortcomings. First, it requires a starting seed node for each cavity, which is supposedly supplied by the user. Second, the filling of a cavity may go wrong if the slice of voxels extends out of the cavity, as a consequence of the non-closedness of the boundary of the cavity.

Summing up, the detection of a cavity is done per slice of the grid, but one only considers slices that are transverse to cavities, that is, the cavity inside a slice is delimited all around. The main drawback of this method is that it fails to detect clefts/grooves when the slices do not meet the incomplete boundary of the cavity, although voids are always identified correctly. Invaginations, tunnels and channels may also not be correctly identified for the same reason.

### 4.2. POCKET

**POCKET** was proposed by Levitt and Banaszak [LB92]. Its leading idea is to search for cavities along one or more directions. As a grid-based algorithm, it firstly maps the molecule onto an axis-aligned grid of equally spaced points. The detection of cavities is carried out by scanning them along with $x$, $y$ and $z$ axes. The $x$-axis scan is repeatedly done for all $y$ and $z$ values, starting on those grid points belonging to the leftmost plane of the 3D grid where $x$ is minimum, that is, $x = x_{min}$, as illustrated in Figures 8(a) and (b); analogous procedure applies to $y$-axis scans and $z$-axis scans.

A grid is used to calculate a density map for a given protein. Initially, all grid points are set to a density value of 0. Then, for each voxel on the vdW surface of the protein, one has to check whether there is or not another boundary voxel along the $x$, $y$ and $z$ directions outwards the surface. If so, all the voxels between those two boundary voxels are set to a density value 1. In this way, we end up having voxels with density value 1 that are gathered into separate clusters of value-1 voxels, a cluster per cavity.

Unlike CAVITY SEARCH, this method works in an automated manner, that is, it does not require the user assistance to indicate the seed node of each cavity. However, the identification of cavities still depends on the alignment of the protein about the coordinate system of the grid [LJ06]. For example, a counterclockwise rotation of the molecule shown in Figure 8(a) by 45°, makes its bottom cavity undetectable along the $x$ direction. That is, POCKET is protein-orientation sensitive (POS), and this is particularly noticeable for clefts/grooves.

### 4.3. LIGSITE

To mitigate this ambiguity problem that results from aligning a protein in grid coordinate system, Hendlich *et al.* [HRB97] developed a more sophisticated scanning method, which was implemented in **LIGSITE**. In addition to the three scans along $x$, $y$ and $z$, they used four more scans along the Cartesian cubic diagonals [LJ06], in a total of seven directions, in the attempt of making the identification of cavities less dependent on the orientation of the protein embedded in the 3D grid, as illustrated in Figures 8(c) and (d). These seven directions correspond to 14 oriented directions; for example,

$x$ direction corresponds to two oriented directions, $x$ and $-x$. In practice, if we think in terms of grid cubes neighbouring a given grid cube, these 14 oriented directions are those defined by 14 out of 26 grid cubes surrounding a given cube. As explained further ahead, Li *et al.* [LTA*08] extended the number of scanning directions to those 26 oriented directions in **VisGrid**.

### 4.4. Exner *et al.*'s method

Exner *et al.* [EKMB98] proposed a grid-based method similar to POCKET [LB92] to predict cavities in molecular structures, in the sense that it also uses negative and positive $x$, $y$ and $z$ directions for scanning cavities of a given protein. The grid spacing is set to 0.5 Å. The grid points inside protein atoms are labelled as 'in' points, while those outside such atoms are labelled as 'out' points.

Exner *et al.*'s method distinguishes itself from other grid-based methods because the bracketing strategy for each 'out' grid point is confined to a distance of 12 Å, that is, to a ball of 12 Å radius centred at each 'out' grid point. That is, an 'out' grid point is defined as a cavity point if it is bracketed by two 'in' grid points along at least two Cartesian axes. This means that grooves are not detected at all.

Then, those 'out' grid points that are cavity points are combined to form clusters or cavities. Exner *et al.*'s method uses two cellular logic operations, known as contraction and expansion, to build up such clusters [Del92].

### 4.5. PocketPicker

An algorithm similar to POCKET and LIGSITE was developed by Weisel *et al.* [WPS07] and is called **PocketPicker**. The main difference between PocketPicker and its predecessors is that the scanning is performed along 30 directions equally distributed on a sphere [SK97]. A scan is performed for a probe sphere centred at each grid point beyond the *protein surface* (i.e. vdW surface) and falling short of an *outer surface* that does not exceed a maximal distance of 4.5 Å relative to the protein surface (Figure 9). Grid points inside the protein surface and outside the outer surface are not considered in the computations.

The solvent accessibility of a grid probe along its 30 directions determines the buriedness of each grid point. Whenever a vector defined by one of these directions hits a protein atom, the buriedness index of the grid point increases by one. After calculating the buriedness index for each grid point between the protein surface and the outer surface, it remains to cluster the grid points into pockets. A pocket consists of connected grid points with a buriedness index greater than 15 (out of 30 directions), what intuitively indicates that the grid points belong to a concavity of the protein surface. A grid point whose buriedness index is less than 15 is one that is above a convex part of the protein surface. Note that the buriedness index is a discrete measure of the solid angle of Connolly [Con86].

### 4.6. PocketDepth

**PocketDepth** is another grid-based algorithm, which was proposed by Kalidas *et al.* [KC08]. It is similar to POCKET in the sense that it uses six oriented scanning directions for each voxel, each direction per voxel face. Thus, it is also protein-orientation sensitive (POS). Also, it resembles the Travel Depth method (see Section 7.3), provided that its scalar field is set by calculating the depth of each cube's centre of putative cavities within an axis-aligned grid. But, unlike Travel Depth, the depth is counted in an incremental manner, rather than measured (see Equation 3), from a grid cube to its neighbours.

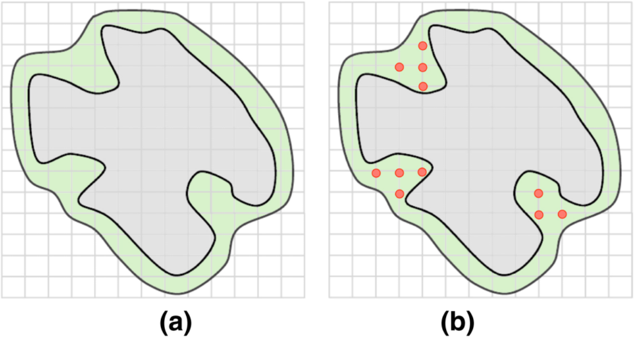


**Figure 9:** *Detecting cavities using PocketPicker [WPS07]: (a) Group of grid points in the outer surface (green squares) inside the protein surface (grey squares) and outside of the outer surface (white squares); (b) Cluster of grid points that represent cavity regions (pictures taken and modified from [WPS07]).*

The algorithm is as follows. First, all grid cubes are assigned the zero depth and labelled as internal, external or surface. Note that each surface cube defines six axis-aligned vectors. Second, considering only the axis-aligned vectors that go out the surface, and that are blocked by any surface cube on the other side of the surface, the depth of each cube located between two opposite blocking cubes on the surface is incremented by 1. Third, grid cubes with a depth greater than zero are then clustered into cavities regarding their cumulative depth and spatial proximity. The cube clustering is based on the DBSCAN, which is a density-based clustering scheme due to Ester *et al.* [EKSX96].

### 4.7. VisGrid

With **VisGrid**, Li *et al.* [LTA*08] extended LIGSITE in the sense that it uses the 26 oriented directions defined by the 26 voxels of the first layer around a given voxel, and 98 when the second layer is taken into account. Therefore, the problem of orientation-sensitivity inherent to grid-based methods is rather mitigated. The grid voxel length is set to 0.9 Å. The scalar field associated with the grid considers three integer values for voxels: $-1$ for voxels inside the protein atoms augmented by 1.4 Å concerning the water molecule radius, 0 for voxels transverse to SAS, and 1 for empty voxels outside SAS, although the SAS does not need to be evaluated. Note that the negative scalar value ascribed to interior voxels allows us to find also protrusions as cavities inside SAS.

### 4.8. PoreWalker

**PoreWalker** [PCMT09] is a method specifically designed to identify and describe channels (or pores) in transmembrane proteins. A channel is used as a path for ions or other molecules to cross the membrane. Its centre and axis can be defined by the pore-lining residues in the protein structure that the algorithm calculates by taking into account the special geometry of transmembrane proteins, as their structures run approximately perpendicular to the membrane plane, crossing the membrane from one side to another.

First, an initial approximation of the main axis of the channel is obtained by taking the $C_\alpha$ coordinates of the residues and calculating the average vectors of the secondary structure (helices and strands). The protein is then re-oriented so that these secondary structures are mainly perpendicular to the membrane and their averaged centre of gravity lies at the reference frame's origin. Next, the centre of the pore is identified by iteratively maximizing the number of detected pore-lining residues, that is, water-accessible amino acids whose $C_\alpha{-}C_\beta$ vector points towards the current pore axis, with the preliminary centre and axis of the pore being redefined in each step. The final pore axis is obtained by using an iterative slice-based approach to refine it. The protein structure is mapped onto a 3D grid and then divided into slices of height 1Å, perpendicular to the current pore axis. For each slice, located at different pore heights, a local pore centre is identified by the centre of the sphere with the maximum radius that the slice can accommodate. These spheres then define a new vector used to align and re-orient the structure.

Finally, the algorithm calculates several pore features and quantitative descriptors, such as the diameter profiles and position of pore centres at different heights along the channel, the atoms and corresponding residues lining the channel walls, and the size, shape and regularity of the channel cavity.

### 4.9. DoGSite

**DoGSite** was introduced by Volkamer *et al.* [VGGR10], and is based on the concept of DoG (Difference of Gaussian) [GW07], borrowed from image processing and analysis. The difference is that now we apply a DoG to a 3D grid instead of a 2D image. Grid points are ascribed either the value 0 for points outside the vdW surface or the value 1 for points inside or on the surface. Unlike most cavity detection methods, DoGSite is capable of structuring cavities into subcavities, resulting in a more detailed shape description of putative binding sites.

DoGSite was developed from the leading idea that active sites quite often possess invaginations as large as that they are capable of accommodating one or more heavy atoms. When a 3D DoG filter is applied to a grid representation of the protein, such invaginations can be identified because it determines where are spherically shaped structures in the grid, known as DoG cores. These cores correspond to sub-cavities that are then gathered into cavities.

### 4.10. VICE

**VICE** (Vectorial Identification of Cavity Extents) is another grid-based method, which was developed by Tripathi and Kellogg [TK10]. Similar to other grid-based methods, VICE discards grid points that fall inside the protein surface (e.g. vdW surface). Only the grid points that fall outside the protein surface are assigned a score according to a buriedness-like metric.

Similar to POCKET, VICE uses an integer (Boolean) grid, but the values 0 and 1 assigned to grid points have a distinct meaning. The value 0 is assigned to every single grid point inside an atom; otherwise, its value is set to 1. VICE uses an integer density map to define the scan directions through integer arithmetic vectors as a way of speeding up the computations associated with the grid. It is clear that the grid points outside the protein potentially are cavity points, and this leads us to the ambiguity problem of cavity bounds. Each outside grid point is subject to a search procedure to determine whether it belongs to a cavity or not.

The decision is based on a discrete variant of the Connolly function, in a way similar to that one of PocketPicker. Basically, one considers a set of eight 2D vectors $(1, 0)$, $(1, 1)$, $(0, 1)$, $(-1, 1)$, $(-1, 0)$, $(-1, -1)$, $(0, -1)$ and $(1, -1)$ from each grid point to its eight neighbouring points in the same axis-aligned plane (e.g. parallel to the XY plane), and calculate the rate of blocked vectors to the total number of vectors starting at such grid point. A blocked vector is defined as any vector that hits the molecular surface (or atom); otherwise, it is a clear vector. Such a rate has a nominal cutoff value given by 0.5, which sets the line between the convexity and the concavity. A rate clearly above 0.5 denotes the presence of a putative cavity, while a rate noticeably under 0.5 means that the grid point is close a convex region of protein surface or it is far away from the protein surface. It happens that a few grid points, mostly those close to the cavity mouth, remain ambiguous because the rate varies in the range $[0.5 - 0.05, 0.5 + 0.05]$; in this case, one uses a supplementary set of 2D vectors given by $(2, 1)$, $(1, 2)$, $(-1, 2)$, $(-2, 1)$, $(-2, -1)$, $(-1, -2)$, $(1, -2)$ and $(2, -1)$ for disambiguation purposes.

### 4.11. Phillips *et al.* method

This method was proposed by Phillips *et al.* [PGD*10]. It is based on ray casting, as known from computer graphics, with the difference that rays are parallel to each other in $z$ direction. This technique utilizes a ray passing through the centres of voxels of an axis-aligned 3D grid hosting the protein. As usual in ray casting, rays are not blocked by the protein, so that we end up having door-in and door-out points on the molecular surface (e.g. vdW surface) for each ray. These intersection points between rays and the molecular surface are carried out as usual in computer graphics. In the end, we have only to collect those voxels outside the surface that are traversed by door-out-door-in ray segments. Unfortunately, and similar to CAVITY SEARCH and other methods with a small number of scanning directions, this technique may miss cavities other than voids.

### 4.12. Grid-based methods: Discussion

Grid-based methods are built upon three entities: the set of atoms (SA) of a given protein, an axis-aligned grid, and a scalar field. The scalar field is either boolean or integer. The key idea of these methods is the one of blocking oriented directions or visibility vectors from

**Table 2:** *Grid-based methods.*

| | | | | | | | | | Cavities | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Molecular surfaces | | | Limitations | | | | Pockets | | | | |
| Methods | Reference | SA/vdW | SES | SAS | GSS | POS | MOA | UACL | Clefts/Grooves | Invaginations | Tunnels | Channels | Voids |
| CavitySearch | [HM90] | • | | | • | • | • | • | | | | | • |
| POCKET | [LB92] | • | | | • | • | • | | | • | • | • | • |
| LIGSITE | [HRB97] | • | | | • | | | | • | • | • | • | • |
| Exner *et al.* | [EKMB98] | • | | | • | | | | | • | • | • | • |
| PocketPicker | [WPS07] | • | | | • | | | | • | • | • | • | • |
| PocketDepth | [KC08] | • | | | • | • | • | | | • | • | • | • |
| VisGrid | [LTA*08] | • | | • | • | | | | • | • | • | • | • |
| PoreWalker | [PCMT09] | • | | | • | | | | | | • | • | |
| VICE | [TK10] | • | | | • | | | | • | • | • | • | • |
| DoGSite | [VGGR10] | • | | | | | | | • | • | • | • | • |
| Phillips *et al.* | [PGD*10] | • | • | | • | • | • | | | | | | • |

Abbreviations: SA/vdW, set of atoms/van der Waals surface; SES, solvent-excluded surface; SAS, solvent-accessible surface; GSS, grid-spacing sensitivity; POS, protein-orientation sensitivity; MOA, mouth-opening ambiguity; UACL, user-assisted cavity location.

each voxel. A brief glance at Table 2 shows us grid-based methods enjoy the following characteristics:

- *Molecular Surfaces*. As shown in Table 2, and similar to sphere-based methods, grid-based methods mostly rely on the *set of atoms* (SA). Atoms allow us to distinguish the grid nodes inside the protein (or inside of atoms) from those lying outside it.
- *Limitations*. We have identified two main limitations with grid-based methods. The first has to do with *grid-spacing sensitivity* (GSS). A distinct grid voxel length may result in finding distinct cavities for the same protein [OFH*14], as well as a different number of cavities. Clearly, this has not only a significant impact on the accuracy of a given grid-based method but also on its performance regarding memory space and time complexity. In fact, a grid with smaller voxels implies more memory space consumption and poorer time performance, in particular for voxel length less than 1.0 Å. To mitigate the problem of GSS, one has to find a way of automatically adjusting and calculating the appropriate voxel length. With the exception of DoGSite, no other method can automatically adjust the voxel length to the size of a given protein regarding the number and density of atoms. Larger proteins should lead to longer voxel length [VGGR10], and thus a less number of voxels, as well as an increasing of time performance. Recall that the time complexity of any algorithm based on a 3D grid is cubic unless one uses parallel computing [DG17].
  The second limitation concerns *protein-orientation sensitivity* (POS). This means that a distinct orientation of the protein within the grid may result in finding a distinct set of cavities on the same protein surface [BAM*14]. That is, grid-based methods are not rotation-invariant; their accuracy depends on rotations of a given protein in 3D space. Using multiple scanning directions is a way of mitigating this problem.
  Note that the problem concerning protein orientation can be solved since we can determine the boundary of each cavity, that is, the problem of delineating the cavity ceiling [OFH*14]. As shown in Table 2, most grid-based methods have no difficulties in finding cavity mouth openings from the blocking technique of scanning directions.
- *Cavities*. With the exception of CAVITY SEARCH, grid-based methods identify cavities in an automated manner. Besides, only POCKET and its follow-up method called PocketDepth may miss clefts/grooves because of the small number of scanning directions they use in the detection of cavities.

At last, with the exception of DoGSite, these methods were not designed to identify $n$-part cavities in a structured manner, that is, each $n$-part cavity is identified as a whole, not in parts or subcavities.

## 5. Grid-and-Sphere-Based Methods

Grid-and-sphere-based methods combine the advantages of both grid- and sphere-based methods. Similar to grid-based methods, they also sustain themselves on a scalar field defined at every single grid point. Additionally, they mostly use large probe spheres rolling on the vdW surface, which have the function of delimiting cavities between the probe-generated surface and the molecular surface. This solves the problem of ambiguity that stems from the necessity of identifying cavities and their stopgaps (or mouth openings). The identification of a cavity's stopgap is known as the cavity ceiling problem. As noted by Oliveira *et al.* [OFH*14], the cavity ceiling problem can be controlled using customizable probe sizes. Consequently, grid-and-sphere based methods are not orientation-sensitive.

### 5.1. VOIDOO

**VOIDOO** is a grid-and-sphere based method proposed by Kleywegt and Jones [KJ94]. It was thought of to only identify voids and invaginations using a process named atom fattening. Unlike a void, an invagination is exposed to the outside of the protein, but it can be closed off by increasing the atomic radii, that is, an invagination

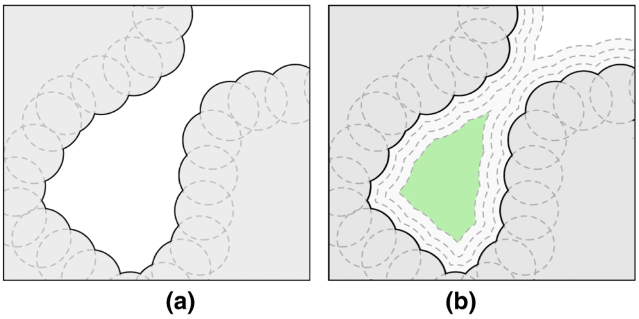


**Figure 10:** *Detecting cavities using VOIDOO [KJ94]: (a) Region of the protein with atoms having the normal van der Waals radii; (b) The increase of the atomic radii of the atoms encloses a cavity (green zone). This process of atom fattening allows a well delineation of the void (pictures taken and modified from [KJ94]).*

becomes a void using such process of atom fattening. Additionally, an invagination may possess one or more mouth openings, so that channels are also identified using VOIDOO. Unfortunately, wide and shallow clefts/grooves cannot be detected in this manner. Note that the atoms and probes of gradually increasing radii are concentric. This process is shown in Figure 10.

This method starts by mapping the protein onto a 3D grid with the following characteristics: (i) grid spacing of 0.5 to 1.0 Å; (ii) grid nodes ascribed with the value 0. The second step consists in labelling all grid points inside protein's atoms (i.e. vdW surface) as 1. The third step carries out the labelling of those grid points that gradually are caught between the vdW surface and the SAS-like outer surface under the process of atom fattening. This process stops as soon as the invagination turns into a void.

### 5.2. HOLLOW

**HOLLOW** is a grid-and-sphere method proposed by Ho and Gruswitz [HG08]. HOLLOW uses a grid with a spacing of 0.5 Å, and probe spheres (called dummy atoms) of 1.4 Å radius. Unlike sphere-based methods, which place probe spheres tangential to three atoms of the protein, here each probe is centred at a grid point.

Then, those dummy atoms overlapping atoms of the molecule are thrown away from the grid. Also, dummy atoms located outside the envelope of the protein are removed. In the same manner, the remaining dummy atoms within each cavity are eliminated under the condition that the total volume of each cavity remains the same. The envelope of the molecule is defined by the process of rolling a large probe sphere of 8.0 Å on the surface atoms. Consequently, all cavities of the molecule are identified by HOLLOW, but this evidently depends on the grid spacing.

### 5.3. POCASA

**POCASA** (Pocket-Cavity Search Application) includes a sphere-based grid algorithm, called Roll, which was designed and developed by Yu *et al.* [YZTY10]. The scalar field is boolean, so that grid points inside the protein are assigned the value of 1 (i.e. occupied grid points), while grid points outside the protein take on the value 0 (i.e. free grid points).

Roll makes use of a large probe sphere of a varying radius much greater than 1.4 Å, which rolls on the protein surface, being the rolling direction controlled by the inner border tracing algorithm borrowed from image analysis and processing [SHB16]. Nevertheless, the size of the probe sphere may vary to identify cavities of distinct sizes. The crust-like surface generated by the probe works as a second envelope of the protein and is called *probe surface*. The leading idea is to identify cavities as the loci consisting of free grid points or voxels between the protein's vdW surface and the probe surface. In practice, the probe surface is not generated, being enough to consider as cavity voxels the free voxels outside the protein that are not touched by the probe. Obviously, the voxels beyond the probe are discarded straight away.

### 5.4. McVol

**McVol** method was proposed by Till and Ullmann [TU10] to calculate the volume of molecular structures through a Monte Carlo integration. The molecular volume is used to identify surface clefts and voids. This method takes advantage of four main tools: (i) an axis-aligned grid enclosing the molecule; (ii) the SAS; (iii) spherical probe rolling on the set of atoms of the molecule, whose radius is desirably equal to the atomic radius of the solvent; (iv) the random placement of points in the grid-discretized domain (i.e. bounding box) enclosing the molecule.

The random placement of points in the domain serves two purposes: the computation of the molecular volume and the identification of voids. In fact, the molecular volume consists of the volume enclosed by the outer surface minus the volumes (voids) enclosed by the inner surfaces. Therefore, the computation of the molecular volume requires identifying the molecular voids. Note that grid-based methods are suited to compute volumes through integration using Monte Carlo techniques. See Till and Ullmann [TU10] for further details. A point that belongs to a void satisfies the following two conditions: (i) its distance to any atom centre is less than the vdW radius of such atom plus the rolling probe sphere radius; (ii) its distance to SAS' closest point is greater than the rolling probe sphere radius.

Identifying surface clefts is inspired by the technique used to identify voids. We define a 3D local box centred at each solvent grid point (i.e. grid point outside the molecule) to determine the percentage of cleft grid points in the local box. If such a percentage is greater than a given threshold, the solvent grid point is marked as cleft, what is equivalent to use a discrete Connolly function. The clustering of solvent grid point into clefts is performed using a breadth-first search (BFS) over the grid.

### 5.5. GHECOM

**GHECOM** (grid-based HECOMi finder) is a grid-and-sphere based method due to Kawabata [Kaw10]. It is a follow-up of the sphere-based method, called PHECOM, proposed before by Kawabata and Go [KG07]. Following the principle that probes with different radii capture distinct protein cavities, PHECOM uses the smallest probe whose radius is 1.87 Å, which corresponds to the size of a single methyl group ($-CH_3$), and a variable size for the large probe that defines not only the cavity ceiling but also the shallowness of the cavity. Besides, this idea is taken to a limit in methods based on $\alpha$-shapes (see Section 8), where radius-zero probes outputs the van der Waals surface and radius-$\infty$ probes gives rise to the convex hull of a set of atoms.

As Kawabata noted, placing probes on the protein atoms in conformity with the principle of three contacts (i.e. three-point geometry) might fail for proteins with irregular shapes. Besides, computing the minimum inaccessible radius for a set of large probes is very time-consuming. This amounts to computing the optimal $\alpha$-sphere that defines the ceiling (i.e. stopgaps) for all relevant cavities of a protein (see Section 8).

GHECOM solves these problems by combining spheres with voxels of a 3D grid, together with the theory of mathematical morphology [Mat75] [Ser84]. This theory is used in digital analysis of geometric features in imaging, although it had also been used in the structural analysis of proteins before by the hand of Delaney [Del92] and Masuya and Doi [MD95]. According to Masuya and Doi, given the set $X$ of the union of the atoms of a given protein, pockets can be defined as the result of closing of $X$ by a large probe and opening of $X$ by a small probe; note that closing ($\bullet$) and opening ($\circ$) are two morphological operations. Masuya and Doi also put forward that the SAS and SES can also be defined through morphological operations.

Kawabata's solution for identifying cavities also uses those morphological operators, which reflect the PHECOM definition of a pocket: 'a small probe can enter but a large probe cannot' [Kaw10]. In fact, GHECOM uses the same two operations to define a pocket of $X$ as follows:

$$P_X(L, S) = ((X \bullet L) \cap X^C) \circ S, \tag{1}$$

where $L$ and $S$ stand for the large and small probes, and $X^C$ is the set complement of $X$. As shown in Figure 11, the operation (1) produces a pocket as the space outside the protein $X$ ($X^C$), where the large probe $L$ cannot enter (closing of $X$ by $L$), but the small probe $S$ can (opening of $(X \bullet L) \cap X^C$ by $S$). So, it was made possible to efficiently calculate multi-scale pockets (i.e. deep to shallow pockets), simultaneously, from multi-scale spherical probes (i.e. small to large probes). It is noteworthy that the expression (1) simplifies analogous expression advanced by Masuya and Doi, and is valid for both continuous and discrete point sets, that is, it applies to sets defined in the 3D grid of the domain where the protein resides.

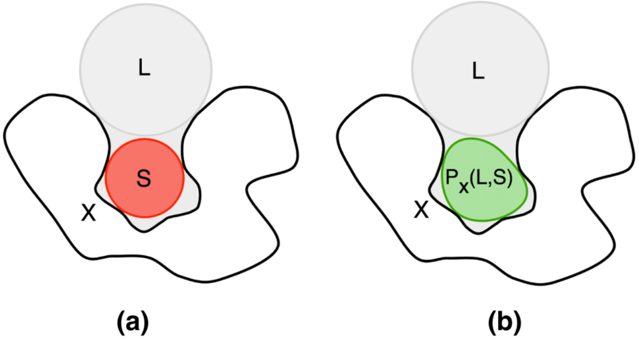


**Figure 11:** *Detecting cavities using GHECOM [Kaw10]: (a) representation of the molecular surface ($X$), a small probe ($S$) in a cavity and a large probe ($L$) on the protein surface; (b) cavity as given by $P_X(L, S)$.*

### 5.6. 3V

Voss and Gerstein [VG10] introduced the **3V** (Voss Volume Voxelator) method. It also uses two probes that roll on the set of atoms of the protein, whose radii can be adjusted relative to their 1.5 and 6 Å default values. These probes define two SESs, but these surfaces are not analytically built or triangulated.

The leading idea is to determine grid points inside the outer surface not accessible to a large probe, as well as grid points inside the inner surface not accessible to a small probe, so cavities result from the difference between the previous two grid point sets. That is, the empty space between the two surfaces is calculated in a discrete manner using a 3D grid of points or voxels. Thus, there is no room for mouth-opening ambiguity (MOA).

### 5.7. VolArea

**VolArea** was introduced by Ribeiro *et al.* [RTC*13]. It also follows the leading idea of mapping a protein onto a 3D grid of voxels, where the cavities are 3D sites that consist of empty voxels located outside the protein. VolArea utilizes the concept of *cavity probe sphere* that is concentric with every single atom, but whose radius is greater than the vdW radius of its concentric atom. Therefore, similar to VOIDOO (see Section 5.1) and PocketPicket (see Section 4.5), we end up having two surfaces: a vdW surface and an SAS-like surface.

The question is then how to collect the relevant empty voxels of a cavity among all those lying between those surfaces. This is accomplished with the user assistance, who has to first choose the region where to search for a cavity. The user must also set the radius of the cavity probe, which depends on the size and shape of the pocket, cleft or cavity under study.

Then, the cavity is identified from the cluster of empty voxels located inside 3D regions that result from intersecting cavity sphere probes. This means that the voxel length must be much smaller than the radius of any atom. With Volarea, very small cavities are discarded, in particular, those smaller than a hydrogen atom regarding occupied volume.

### 5.8. KVFinder

More recently, Oliveira *et al.* [OFH*14] introduced **KVFinder**, which is another grid-and-sphere based algorithm similar to the one proposed by Voss and Gerstein [VG10]. The scalar field associated with the grid is boolean. This allows them to define every single geometric cavity regarding theory of mathematical morphology [Ser84], as explained below.

The MOA problem is approached using two probe spheres: probe-in sphere and probe-out sphere. Only grid points outside the protein are taken into account in the process of detection of cavities. The first sphere is small to guarantee that it fits in most cavities, while the second is larger to guarantee that it does not fit in those cavities. It is clear that we are assuming that these spheres do not overlap the protein surface.

By centring the probe-in sphere at each outer grid point, we easily see that most outer grid points end up being caught by the probe-in sphere; only those grid points of tiny cavities whose size is less than the size of the probe-in sphere are discarded. This concludes the first step of the algorithm. The second step is identical to the first step, with the difference that now one uses the probe-out sphere, instead of the probe-in sphere.

A cavity point is thus every single grid point overlapped by the probe-in sphere which is not overlapped by the probe-out sphere. Note that the probe-in sphere rolling on the protein surface defines a surface that is the SES approximately, while probe-out sphere rolling on the protein surface gives rise to another surface that tends to make a shortcut on the surface, more specifically where the cavities are located. However, these surfaces are not evaluated or determined analytically. In short, the probe-out sphere solves the MOA problem that is typical in grid-based algorithms. But, finding a suitable radius for the probe-out is a difficult—not to say impossible—task because the radius depends on the size and shape of each cavity.

### 5.9. PrinCCes

Recently, a method designated as **PrinCCes** (Protein internal Channel and Cavity estimation) was proposed by Czirják [Czi15]. The method relies on a 3D grid, whose grid spacing is user-defined and varies between 0.1 and 2.4 Å. Two probe spheres are also employed in the process. A larger probe (with a radius of 1.0 to 10.0 Å of radius) aims at identifying the shell volume (i.e. protein volume plus its cavity volumes), while a smaller probe (with a radius of 0.6 to 5.0 Å of radius) aims at detecting cavity volumes.

This method is quite different from those that place probe spheres in contact with protein's surface atoms (see, e.g. 3V [VG10]). Instead of rolling probe spheres on protein's surface atoms, both larger and smaller probes are placed at the centre of each (surface) atom to collect cavity grid points. In fact, this method relies on a novel algorithm called Find Continuous SubSpace algorithm (FCSS), which decomposes the space between the larger and the smaller probe into distinct cavities.

More specifically, each cavity is delineated by moving a controllable-size probe sphere along the 26 possible directions defined by each cavity grid point and their neighbours, but without colliding with the molecular surface. These movable probes are located in the space between surface atoms and their larger probes.

According to its authors, this method is more faithful to represent the geometric structure of tunnels. That is, it avoids the representation of tunnels as a group of different sized spheres (as seen in CAVER [POB*06] and MolAxis [YFW*08]). Furthermore, the user does not need to provide seed points indicating the direction or location of cavities to detect and delineate cavity zones.

### 5.10. Grid-and-sphere-based methods: Discussion

Using probes in grid-based methods follows three different techniques. The first is based on atom fattening (originating SAS or SAS-like surfaces), as it the case of VOIDOO, McVol, VolArea and PrinCCes (see column 'SAS' on Table 2). The second takes advantage of the concept of rolling probes of unequal radii on the vdW surface, as in POCASA, McVol, GHECOM and 3V. The third was only incorporated in KVFinder and consists in placing concentric probes of unequal radii at grid points so that the small probe gets in cavities, but not large probes.

As shown in Table 2, grid-and-sphere-based methods can be characterized as follows:

- *Molecular surfaces*. As usual, these methods directly use the *set of atoms* (SA) of a given protein to identify its cavities (see Table 3). Also, and given the hybrid flavour of grid-and-sphere-based methods, they take advantage of three tools: an axis-aligned grid, a scalar field and probe spheres.
- *Limitations*. The issue concerning GSS can be solved since the voxel length is at most $(1/2\,R)$, with $R$ the radius of the water probe sphere, in conformity with Nyquist theorem [DG15]; otherwise, cavities cannot be properly sampled by empty voxels. *Protein-orientation sensitivity* (POS) is a typical problem in grid-based methods. But, using large probe spheres (approx. 5 Å), we can block cavity entries/exits or mouth openings, solving the POS problem in this manner.
  MOA is another issue of grid-based methods, simply because mouth-openings do not block scanning directions. As said above, this problem can be solved using large blocking spheres on the protein surface. With the exception of VolArea, the methods listed in Table 2 resolve the MOA problem, that is, they are capable of delineating the mouth openings of cavities. This is accomplished at the cost of using probe spheres that isolate cavities from the empty outer space. Let us also mention that only POCASA, GHECOM and PrinCCes support multiscale probes.
  Therefore, these methods determine protein cavities in an automated manner without the user intervention; the exception is VolArea, which requires the user-assisted cavity localization (UACL).
- *Cavities*. In general, grid-and-sphere based methods are capable of automatically identifying cavities. Only VolArea needs user's interactive assistance to identify such cavities (see column 'UACL' in Table 3). Nevertheless, VOIDOO may miss shallow cleft/grooves, whereas McVol was designed only for detecting cleft/grooves and voids. At last, among all methods listed in Table 3, only PrinCCes can organize a cavity from its sub-cavities or parts.

To summarize, using probe spheres together with grids solves two typical problems of grid-based methods, namely, MOA and POS. In fact, the use of multi-scale probes allows us to define suitable stopgaps for each cavity.

**Table 3:** *Grid-and-sphere-based methods.*

| | | | | | | Cavities | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Molecular surfaces | | Limitations | | Pockets | | | | |
| Methods | Reference | SA/vdW | SAS | GSS | UACL | Clefts/Grooves | Invaginations | Tunnels | Channels | Voids |
| VOIDOO | [KJ94] | • | • | • | | | • | • | • | • |
| HOLLOW | [HG08] | • | | • | | • | • | • | • | • |
| POCASA | [YZTY10] | • | | • | | • | • | • | • | • |
| McVol | [TU10] | • | • | • | | • | | | | • |
| GHECOM | [Kaw10] | • | | • | | • | • | • | • | • |
| 3V | [VG10] | • | | • | | • | • | • | • | • |
| Volarea | [RTC*13] | • | • | • | • | • | • | • | • | • |
| KVFinder | [OFH*14] | • | | • | | • | • | • | • | • |
| PrinCCes | [Czi15] | • | • | • | | • | • | • | • | • |

Abbreviations: SA/vdW, set of atoms/van der Waals surface; SES, solvent-excluded surface; SAS, solvent-accessible surface; GSS, grid-spacing sensitivity; UACL, user-assisted cavity location.

## 6. Surface-Based Methods

These methods are based on analysis of geometric properties of the molecular surface [NH06]. Examples of such geometric properties are solid angles [Con86], the surface fractal dimension [PB99] and curvature [NWB*06], so surface cavities look like valleys in the middle of mountain ranges.

### 6.1. NSA

**NSA** (Nearest Surface Atom) method was introduced by Del Carpio *et al.* [DCTS93]. This method starts by sampling the surface of each atom as proposed by Lee and Richards [LR71] for computing the SAS. Then, one removes the occluded points, that is, those points inside other atoms (see Lee and Richards [LR71]), so that we end up obtaining a set of points, called free points, which sample the van der Waals surface of the protein. After discarding those occluded points, one calculates the distance between each atom and the centre of gravity of the protein. A smaller distance from an atom to protein's gravity centre means that the atom is located at a deeper site in the protein. After finding the NSA from the centre of gravity, a cavity is formed by clustering of the nearby surface atoms that are visible to NSA's free points on the vdW surface (see Figure 12(a)). The process is repeated while there exists some concavity to detect on the molecular surface (see Figure 12(b)). The concave regions are the places where the protein cavities are located [DCTS93, LJ06]. However, this method has difficulties in dealing with $n$-part cavities because it is based on a visibility criterion from the free points of the NSA, that is, there is space for ambiguity in the identification of cavity mouth openings.

### 6.2. SCREEN

Nayal and Honig [NH06] proposed a surface-based method, called **SCREEN** (Surface Cavity REcognition and EvaluatioN). This method generates two molecular surfaces through GRASP [NSH91]. The first surface is the standard SES generated from rolling a solvent probe sphere with 1.4 Å of radius on the van der Waals surface (or set of atoms), here called inner surface. The second surface, called surface envelope, is generated in the same manner, but with a probe sphere of 5 Å of radius.

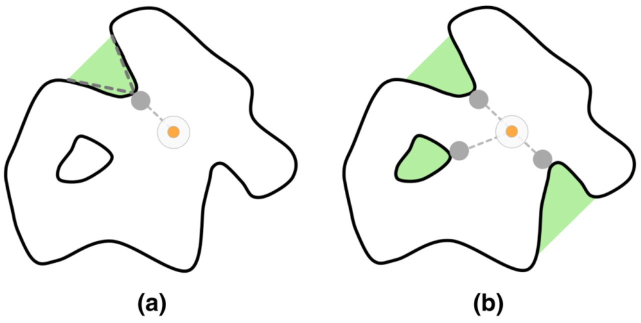


**Figure 12:** *NSA: (a) the gravity centre (in orange) of the protein is displayed together with its nearest surface atom (NSA), from which the cavity (in green) is formed by the clustering of nearby surface atoms that are visible from NSA; (b) the process is repeated while there is some cavity to form on the protein surface (pictures inspired in [LJ06]).*

Cavities boil down to the space between the two surfaces. The SES patch of the inner surface on the cavity floor represents the cavity, while the homologous patch of the surface envelope represents the cavity ceiling. As such, cavity envelope is well defined, as well as its mouth openings, volume and surface area, which can be then analytically computed in a precise way. No grid is used here for any purpose.

### 6.3. CHUNNEL

**CHUNNEL** was introduced by Coleman and Sharp [CS09]. This method is based on the SES, which is determined using the GRASP algorithm [NSH91]. CHUNNEL was specifically developed to automatically find, characterize, and display channels (or pores) of a given protein, particularly for large and very large proteins.

By relying on the triangulation of the SES of the protein to determine the number of channels in conformity with the Euler-Poincaré formula, CHUNNEL automatically finds the location of the channel mouth without any user's guidance or clues, as well as multiple channels throughout the entire protein surface. For that purpose, one uses the convex hull of the SES triangulation to locate each channel's entrance and exit (i.e. opening mouths).

### 6.4. MSPocket

**MSPocket** (Molecular Surface Pocket) was introduced by Zhu and Pisabarro [ZP11]. It directly identifies pockets on the SES of a given protein, without resorting to any regular grid as usual in grid-based methods. Therefore, unlike grid-based methods, MSPocket is not dependent on protein orientations. In fact, MSPocket utilizes an analytical formulation of SES as given by MSMS software package, which is due to Sanner *et al.* [SOS96]. MSMS produces a set of sample points on SES, called surface vertices, each one of which is associated with a protein atom. These vertices allow us not only to build an SES triangulation but also to determine their normal vectors by averaging normals of neighbour triangles.

Such normal vectors play an instrumental role in locating the concavities on the SES. First, for each vertex, one calculates the angle between its normal and the normal at each one of its adjacent vertices. Then, one calculates the average angle of these angles, assigning it to the central vertex if it is less than 90 degrees. A vertex of this sort is here called concave vertex, and a triangle delimited by three concave vertices is said to be concave. Likewise, a subset of connected concave triangles is a cavity (i.e. either a pocket or a void). This induces a mesh segmentation of SES into cavities (i.e. concave triangles) together with the remaining non-concave triangles belonging to SES. It is clear that this requires the clustering of concave triangles into cavities, so that we end up getting their boundaries or mouth openings. However, similar to NSA method, the lack of an outer surface of the protein may make such mouth openings uncertain, what leads to some degree of ambiguity in their computation; as a consequence, the computation of each cavity's volume and area is not correct either. The reader is referred to [ZP11] for further details.

### 6.5. Giard *et al.*'s method

**Giard** *et al.*'s method [GAGM11] was designed as a follow-up of Travel Depth due to Coleman and Sharp [CS06] (see Section 7.3 for further details). It aims to reduce the (time and memory) complexity of Travel Depth by confining the geometric processing to the SES, and thus eliminating the unnecessary processing of samples (i.e. grid nodes) lying outside and inside the protein. In other words, as a surface-based method, it does not use any grid to help in identifying protein cavities.

Its leading idea is to utilize the triangulated molecular surface and its convex hull to determine the cavities that stand in the middle. The molecular surface is an SES approximation generated by summing Gaussian functions centred on atoms, that is, a molecular Gaussian surface (GS). The distance of each vertex of the GS mesh to its nearest vertex of the convex hull works as a depth metric, which determines whether a GS vertex belongs to a cavity or not.

The main advantage of this method is its reduced complexity regarding consumption of memory space (i.e. no grid is used at all) and time performance (i.e. only unpaired vertices of the GS mesh and its convex hull are processed after all). The main drawback is that it is necessary to use some visibility criterion to ensure the correct measure of depth for GS vertices buried in $n$-part cavities, which are not in the line of sight of any convex hull vertex.

### 6.6. Surface-based methods: Discussion

As shown in Table 4, surface-based methods can be characterized as follows:

- *Molecular Surfaces*. These methods distinguish themselves from others in that they use an analytical molecular surface to directly find the protein cavities. SES is dominant in these methods, but eventually other analytical formulations of molecular surfaces may be used in the future (e.g. surfaces defined by bounded kernel functions) [GVJ*09].
- *Limitations*. These methods operate in an automated manner, so user's assistance is not necessary. SCREEN uses two analytical SES generated from two probes with different radii so that the outer surface works as the ceiling for cavities. This outer surface plays the same role as that one of large probes in sphere-based methods. The difference here is that the surface ends up being generated. Therefore, SCREEN does not suffer from MOA. Similarly, CHUNNEL and Giard *et al.*'s methods do not suffer from MOA because it takes advantage of the convex hull of SES triangulation to locate each channel's mouth opening.
  But, unlike SCREEN, CHUNNEL and Giard *et al.*, NSA and MSPocket methods suffer from ambiguity in delineating each cavity's mouth opening. This is so because they are based on a visibility criterion (e.g. the line of sight from free points, and normal vectors as a measure of curvature), without resorting to a supplementary outer surface (e.g. convex hull) enveloping the protein's atoms.
- *Cavities*. Among those methods listed in Table 4, only NSA and MSPocket are capable of identifying all sorts of cavities. Nevertheless, it is not certain that NSA and MSPocket are capable of correctly determine the entire extent of a cavity structured into parts, largely because of the lack of a supplementary outer surface enclosing the protein. On the other hand, CHUNNEL is focused on identifying channels (and tunnels). Note that CHUNNEL and Giard *et al.*'s methods have difficulties in detecting voids, largely because the surface mesh bounding each void does not meet any convex hull. This problem is mitigated using two SES, but, in this case, it may happen that both triangulations coincide if the void is convex or, alternatively, small depressions arise if the void is not convex, tricking us about the number of cavities where such void is located.

In short, using the analytical, geometric properties of molecular surfaces to identify protein cavities can be seen an emerging trend in molecular graphics and modelling, in particular for those interested in applications of geometric modelling and computational

**Table 4:** *Surface-based methods.*

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | Cavities | | | | |
| | | Molecular surfaces | | | | Limitations | Pockets | | | | |
| Methods | Reference | SA/vdW | SES | GS | CH | MOA | Clefts/Grooves | Invaginations | Tunnels | Channels | Voids |
| NSA | [DCTS93] | ● | | | | ● | ● | ● | ● | ● | ● |
| SCREEN | [NH06] | ● | ● | | | | ● | ● | ● | ● | ● |
| CHUNNEL | [CS09] | ● | ● | | ● | | | | ● | ● | |
| MSPocket | [ZP11] | ● | ● | | | ● | ● | ● | ● | ● | ● |
| Giard *et al.* | [GAGM11] | ● | | ● | ● | | ● | ● | ● | ● | |

Abbreviations: SA/vdW, set of atoms/van der Waals surface; SES, solvent-excluded surface; GS, Gaussian surface; CH, convex hull; MOA, mouth-opening ambiguity.

geometry to biology and chemistry. This leads us to the origins of this research field in the sense that we have to ask ourselves which is the best mathematical formulation to represent and model not only the surface of a molecule (e.g. a protein) but also the surface shape descriptors of their cavities.

## 7. Grid-and-Surface-Based Methods

These methods combine the advantages of the grid- and surface-based algorithms. Analogously to probe spheres, surfaces eliminate the ambiguity problem of grid-based methods, particularly in defining the stopgaps (and, consequently, mouth openings) of cavities. They use the concept of scalar field in conjunction with a 3D grid. The scalar field may be defined by a distance function, a depth function, an electron density field or any other function.

### 7.1. FRODO

**FRODO**, which is due to Voorintholt *et al.* [VKV*89], is considered by many as the first grid-based cavity detection algorithm. This algorithm assigns a real value to every single grid point, which depends on whether such point is inside the molecule, between the van der Waals (vdW) surface and SAS, or beyond SAS. Such real value assigned to each grid point is produced by a real function, which depends on the distance of such grid point to the NSA, and is as follows:

$$F(\mathbf{x}) = \begin{cases} C & \text{if} \quad d < R_w \\ C \cdot \frac{(R_w+R_p)^2-d^2}{(R_w+R_p)^2-R_w^2} & \text{if} \quad R_w < d < R_w + R_p \\ 0 & \text{if} \quad d > R_w + R_p, \end{cases} \tag{2}$$

where $d$ is the distance of the grid node $\mathbf{x}$ to its nearest atom, $R_w$ represents the van der Waals radius of such nearest atom, $R_p$ denotes the maximal radius of the probe that delineates the SAS and $C$ is the maximal value (= 100) assigned to a grid node.

So, we end up having a distance map associated to the grid. It is clear that cavities are located between the vdW surface and SAS, but truly speaking FRODO does not detect cavities [KJ94], having it been designed only for the visualization of SAS. In fact, as noted by Ho and Marshall [HM90], although FRODO is effective in finding regions where cavities are located, it is not that easy to isolate and define the extent of each specific cavity, including their mouth openings (i.e. cavity entrances and exits). That is, FRODO suffers from the MOA problem.

In fact, voids and invaginations can be identified by a cluster null-valued grid nodes enclosed by a shell of non-null-valued grid nodes. However, the detection of some invaginations may fail if its mouth radius is greater than the radius of the water molecule (i.e. 1.4 Å); the same applies to tunnels and channels. Recall that this process on the interaction between the water probe sphere and vdW atoms is equivalent to consider SAS (with atom radii increased by 1.4 Å). This means that invaginations with large mouths and channels with large tunnels, as well as large clefts, cannot be detected using FRODO because augmented atoms facing other augmented atoms of the SAS on the opposite side of the cavity do not touch or intersect.

### 7.2. CAVER

This method was primarily designed to identify pathways from buried active sites (i.e. clefts, pockets and cavities) to the solvent outside the protein [POB*06], though it was also designed to be applied to molecular dynamic trajectories. **CAVER** utilizes two geometric tools to determine pathways and, as a consequence, the protein cavities themselves: (i) an axis-aligned grid embedding the protein and (ii) the convex hull of the protein's body. Grid nodes are then categorized as outer and inner nodes in relation to the protein body (i.e. set of vdW atoms). Outer nodes that fall inside the convex hull identify where cavities are.

Such outer nodes allow us to construct a positively node-weighted graph (with one or more components), from which one uses a modified form of Dijkstra's algorithm to identify the shortest low-cost path from each point located in a protein cavity to the bulk solvent outside the convex hull. This requires the preliminary identification of the outer nodes lying on the boundary of the convex hull. It is clear that each possible path from the active site to the convex hull is evaluated using a cost function that depends on the number of nodes and the amount of free space around each node. Consequently short and direct paths are 'cheaper' than long and complicated ones. Also, nodes that are surrounded by sufficient empty space are preferred, since they allow for a hypothetical substrate to pass through the channel without the risk of collision.

Subsequent upgrades of the CAVER software were introduced in CAVER 2.0 [MBS08], in which the axis-aligned grid was replaced by the Voronoi diagram to describe the skeleton of tunnels within the structure. Later on, CAVER 3.0 [CPB*12] (see Section 11.8) implemented new algorithms for the accurate calculation and clustering of pathways, improving the effective analysis of the time evolution of pathways in molecular dynamics simulations.

### 7.3. Travel Depth

In computational biology and chemistry, the depth is a measure of the buriedness of a protein atom, so that it is often defined as the distance from the atom centre to the nearest water molecule on the protein surface [PSA*91]. Coleman and Sharp [CS06, CS10] introduced a grid-and-surface based method, called **Travel Depth**. This method takes advantage of two distinct surfaces, the triangulated surface (e.g. triangulated SAS or SES) and its convex hull, which is determined using any 3D convex hull algorithm (e.g. Quickhull [BDH96]). The convex hull works as a delimiter of cavities on the protein surface, that is, the cavities are located between the triangulated molecular surface and its convex hull (see Figure 13).

After determining the convex hull (i.e. a convex set of triangles enclosing the triangulated surface), we have to collect the grid cubes whose centres lie outside the triangulated surface and inside the convex hull into a set of eligible cubes for cavities. The cubes whose centres are outside the convex hull are assigned the depth 0. Starting from the shell of outside cubes lining the convex hull, one calculates the depth of each of their $i$th neighbouring cubes in the set of eligible cubes as follows:

$$d_i = \min_j (d_j + |\mathbf{x}_i - \mathbf{x}_j|), \tag{3}$$

where $j$ denotes every neighbour node of $i$; equivalently, $\mathbf{x}_j$ denotes the centre of each cube neighbouring $\mathbf{x}_i$ for which we are calculating the depth $d_i$. Therefore, the depth increases from the convex hull down towards the triangulated molecular surface. The depth value $d_i$ corresponds to the minimum path length needed to travel towards convex hull boundary, in a way similar to what one does to calculate the shortest path in pathfinding (e.g. Dijkstra algorithm). Such concept of depth allows us to organize cavities into sub-cavities in a hierarchical manner [CS10], which agrees with our shape hierarchy proposal in Section 2.3.

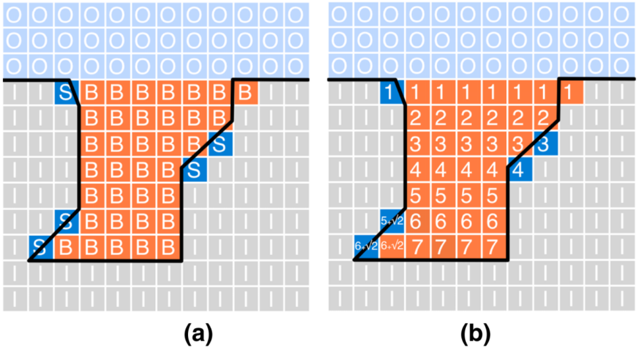


**Figure 13:** *Detecting cavities using Travel Depth [CS06]: (a) each voxel is classified as (i) outside the convex hull (O), (ii) inside the protein surface and intersecting at least one surface atom (S), (iii) inside the molecular surface (I) and (iv) between the convex hull and the protein surface (B); (b) the depth is computed for each voxel in conformity with Equation (3) (pictures taken and modified from [CS06]).*

### 7.4. Zhang and Bajaj's method

Zhang and Bajaj [ZB07] introduced a new cavity detection method based on a signed distance function in relation to the molecular surface. That is, the distance function is induced by the molecular surface. The extraction of pockets can be performed in relation to any closed molecular surface (e.g. van der Waals surface, Gaussian isosurface, SES and SAS) embedded in a regular grid.

This two-step marching algorithm is oriented to pockets (i.e. surface cavities). The first step involves the outward propagation of the surface $S$ to an outer shell surface $O$ that is topologically equivalent to a ball. The second step consists in the backward propagation of $O$ to an inner shell surface $I$, also enclosing $S$. Therefore, the pockets are the empty regions between $S$ and $I$.

More specifically, the cavities correspond to grid points outside the molecular surface where the following signed distance function—called pocket function—is positive:

$$\phi(\mathbf{x}) = \min(d_S(\mathbf{x}), d_O(\mathbf{x}) - t), \tag{4}$$

where $t$ denotes the varying parameter of the level set $d_S(\mathbf{x}) = t$, $d_S(\mathbf{x})$ is the signed distance function relative to the surface $S$, which is positive/negative if $\mathbf{x}$ is outside/inside $S$, while $d_O(\mathbf{x})$ stands for the signed distance function relative to the surface $O$, but, unlike $d_S(\mathbf{x})$, $d_O(\mathbf{x})$ is positive/negative when $\mathbf{x}$ is inside/outside $O$; also, $d_O(\mathbf{x})$ changes from negative to positive at $d_O(\mathbf{x}) = t$. For further details, the reader is referred to [ZB07].

### 7.5. Grid-and-surface-based methods: Discussion

After a brief glance at Table 5, we observe the following:

- *Molecular Surfaces*. Surfaces (e.g. convex hull, SES and SAS) play the role of cavity delimiters, and thus they solve the ambiguity problem inherent to grid-based methods so that cavities are determined by clustering voxels (or their centres) between the inner and outer surfaces that enclose the set of atoms of a given protein. The only remaining problem has to do with the eventual need of better delineating each cavity' mouth openings and discarding voxels that do not belong to any cavity, which may incorrectly connect two separate two cavities. This issue may eventually arise from the use of convex hull as the outer surface, but it is rather difficult to happen when one uses two SES because their triangulations partially overlap on the convex regions of both SES.
- *Limitations*. As a consequence of disambiguation of each cavity's mouth openings, yet in an approximate manner, there is no need for the user assistance in detecting cavities, that is, cavities are

**Table 5:** *Grid-and-surface-based methods.*

| | | | | | | | | | Cavities | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Molecular surfaces | | | | | Limitations | | Pockets | | | | |
| Methods | Reference | SA/vdW | SES | SAS | GS | CH | GSS | MOA | Clefts/Grooves | Invaginations | Tunnels | Channels | Voids |
| FRODO | [VKV*89] | ● | | ● | | | ● | ● | | | | | ● |
| CAVER | [POB*06] | ● | | | | ● | ● | | ● | ● | ● | ● | ● |
| Travel Depth | [CS06] | ● | ● | ● | | ● | ● | | ● | ● | ● | ● | ● |
| Zhang and Bajaj | [ZB07] | ● | ● | ● | ● | | ● | | ● | ● | ● | | |

Abbreviations: SA/vdW, set of atoms/van der Waals surfaces; SES, solvent-excluded surface; SAS, solvent-accessible surface; GS, Gaussian surface; CH, convex hull; MOA, mouth-opening ambiguity.

determined in an automated manner. Besides, the usage of two surfaces makes easier the *voxel clustering* into cavities. Also, the disambiguation of cavity boundaries rids off the typical problem of grid-based methods, which has to do with protein orientation dependence, that is, these methods are not POS. However, they still are *grid-spacing sensitive* (GSS). In fact, as noted in Section 5.10, the grid spacing cannot go over $1/2R$, where $R$ is the radius of the water molecule; otherwise, we risk missing some cavities of the protein.

- *Cavities*. These two-surface grid-based methods have the advantage of determining the extent of pockets and channels in terms of voxels. However, those methods using the convex hull as outer surface are inadequate to find voids; none of these methods mentions how voids are identified from the convex hull of the protein. Nevertheless, such voids can be easily determined as components (or clusters) of outer grid nodes inside the convex hull.

In summary, using surfaces in conjunction with grids allows us to overcome the most common problems associated with grid-based methods, namely: POS and MOA. However, the problem of GSS remains, unless we use a grid spacing of $1/2R$ maximum, but this significantly increases the memory space consumption.

## 8. Tessellation-Based Methods

The foundations of the tessellation-based methods lie in the field of computational geometry, in largely after the introduction of alpha shapes in the plane by Edelsbrunner, Kirkpatrick and Seidel [EKS83], which were later generalized in 3D by Edelsbrunner and Mucke [EM94]. Edelsbrunner himself and colleagues [EFFL95] end up publishing a work on measuring pockets and voids in proteins. There are three main sub-families of tessellation-based methods: (i) $\alpha$-shape methods; (ii) Voronoi-based methods and (iii) $\beta$-shape methods. However, all these methods result somehow from the theory of $\alpha$-shapes.

### 8.1. Theory of $\alpha$-shapes

Given a set of points in 3D, it is well-known that Delaunay triangulation of such points satisfies the circumsphere rule, which states that no point is inside of the circumsphere of any of its tetrahedra. This is illustrated in Figure 14(a), where we see the Delaunay triangulation of a set of points (in yellow) on the plane, with the corresponding circumcircles drawn in grey.

By construction, the $\alpha$-complex is a simplicial subcomplex of the Delaunay triangulation, where $\alpha$ determines the maximum admissible value of the radius of any circumsphere; the 0-complex ($\alpha = 0$) reduces to the initial set of points, while $\infty$-complex ($\alpha = \infty$) is the convex hull of the initial set of points. Therefore, the tetrahedra inscribed in circumspheres of radius greater than $\alpha$ are discarded from the $\alpha$-complex, as illustrated in Figure 14. By varying the value of $\alpha \in \mathbb{R}^+$, we obtain a filtration of sub-complexes of the Delaunay triangulation. The $\alpha$-shape is defined as the union of all simplices (i.e. vertices, edges, triangles and tetrahedra) belonging to the $\alpha$-complex.

In summary, $\alpha$-shape methods build upon the Delaunay triangulation of atomic centres of a given protein. The parameter $\alpha$ is the key idea behind a geometric carving process of generating a subcomplex of the Delaunay triangulation. The question is whether such a carving process helps anyhow in the delineation of the cavities of the molecular structure. More specifically, is there an optimal value of $\alpha \in ]0, \infty[$ to detect voids? Similarly, are there values of $\alpha$ that separate pockets from clefts?

### 8.2. APROPOS

**APROPOS** (Automatic PROtein Pocket Search) was introduced by Peters *et al.* [PFF96]. It is based on the theory of 3D $\alpha$-shapes due to Edelsbrunner and Mücke [EM94]. This pocket detection method is based on the SAS, but, in practice, one uses a subset of atoms augmented with the radius of 1.4 Å of solvent water probe. This is important to delimit the carving process that is typical in $\alpha$-shape methods.

APROPOS builds up two envelope $\alpha$-shape triangulations for a given protein, each one of which corresponds to a distinct value of $\alpha$. The first (outer) envelope is coarser than the second (inner) envelope; the outer envelope is constructed with $\alpha = 20$ Å, while the inner envelope is generated for a value of $\alpha$ in the range [3.5, 4.5] Å. These values of $\alpha$ are empirical and were obtained from experimental testing. In this manner, one ends up having an outer envelope and an inner envelope of the protein, and so cavities are in the space between these inner and outer envelopes, or $\alpha$-shapes.

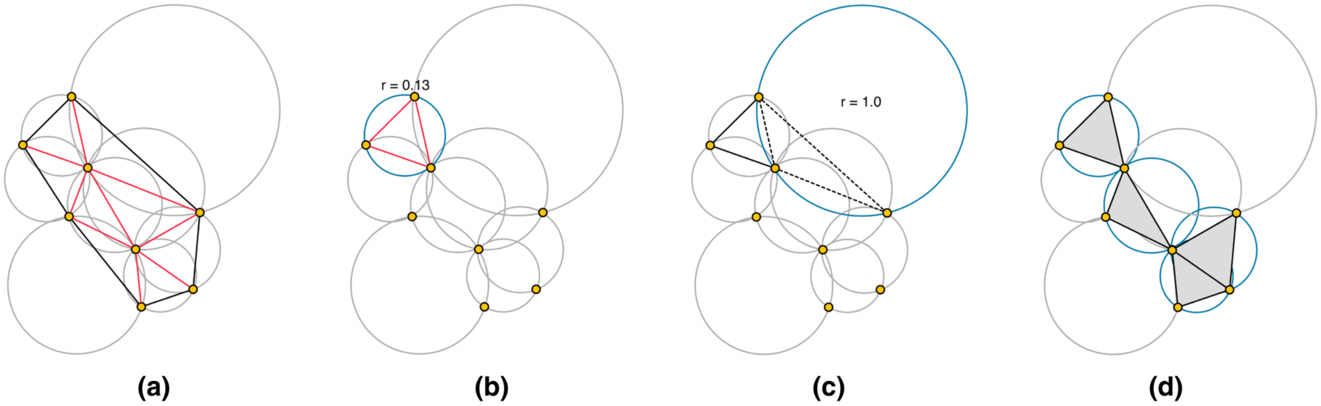


**Figure 14:** *Alpha-shape example where $\alpha = 0.15$: (a) convex hull (in black), Delaunay triangulation (in red), and atom centres (in yellow); (b) the k-simplex (in red) is part of the $\alpha$-shape because the current circumsphere has a radius smaller than $\alpha$; (c) the k-simplex (in black and dotted) is not part of the $\alpha$-shape because the current circumsphere has a radius greater than $\alpha$; (d) after testing each circumsphere, as seen in (b) and (c), we get the final $\alpha$-shape.*

### 8.3. CAST

The main drawback of APROPOS stems from the need of tuning the value of $\alpha$ for both outer and inner $\alpha$-shapes, in particular the one concerning the inner $\alpha$-shape, which is more sensitive to the surface shape variations of the protein itself. To overcome this problem, Edelsbrunner *et al.* [Ede98] introduced the dual sub-complex of the union of balls featuring van der Waals atoms, which amounts to the $\alpha$-shape that is entirely inside such union of balls. In essence, they proposed the convex hull as the outer envelope, and the dual sub-complex as the inner envelope of atomic coordinate centres (see Figure 15).

**CAST** was introduced by Liang *et al.* [LWE98] as an implementation of the method detailed by Edelsbrunner *et al.* [Ede98], and consists of the following steps:

- *Voronoi diagram*. Firstly, one creates a Voronoi space decomposition from the atoms (atomic coordinates) of the molecule, as shown in Figure 15(a).
- *Convex hull*. Secondly, one calculates the corresponding convex hull (i.e. Delaunay triangulation), as illustrated in Figure 15(b).
- *Dual sub-complex*. Then, one removes the simplexes (e.g. triangles) that are not completely inside the molecule, resulting so in an $\alpha$-shape of the original molecule, as depicted in Figure 15(c).

The leading idea here is to get a triangulation with the same topological type as the original set of atoms that comprise the molecule so that we can extract the cavities in a straightforward manner. Note that we have assumed that all atoms possess the same radius (see Figure 15). In case of using the actual van der Waals atoms,

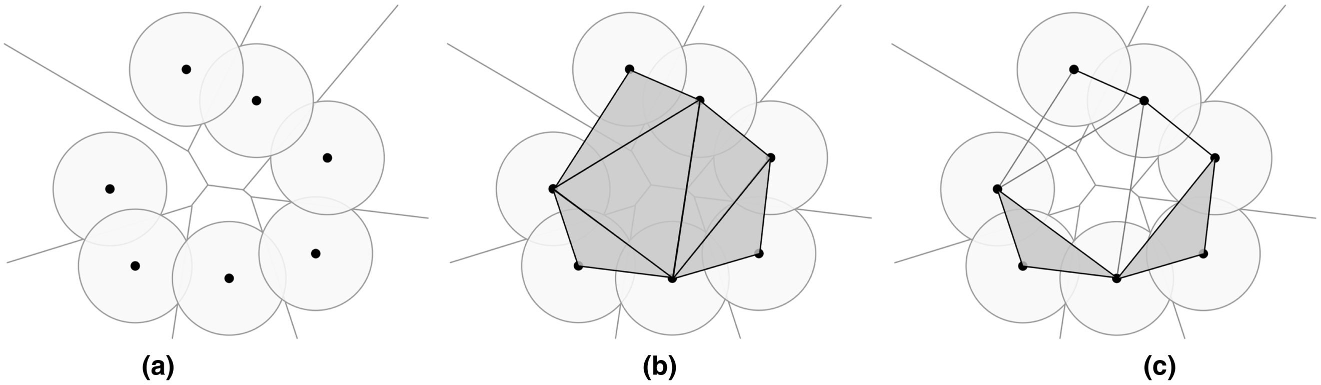


**Figure 15:** *Detecting cavities through CAST: (a) Voronoi diagram of a molecule (i.e. set of spherical atoms); (b) convex hull of the atomic centres, together with Delaunay triangulation; (c) $\alpha$-shape with triangles, edges and vertices in black, where the empty triangles denote the existence of a cavity (taken and modified from [LWE98] [WPS07]).*

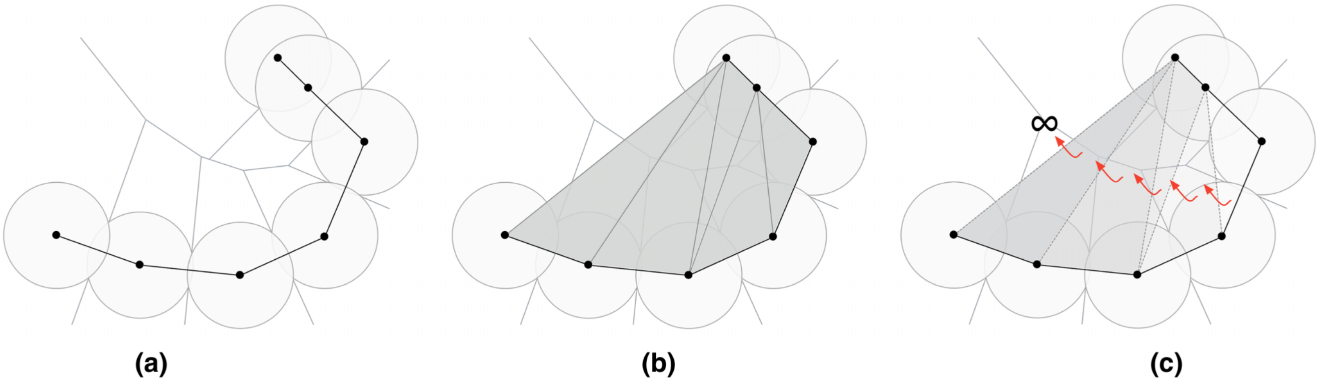


**Figure 16:** *Discrete-flow method at work: (a) Voronoi space decomposition of a molecule; (b) Flow of obtuse triangles from the initial space decomposition; (c) example that shows a cavity that cannot be properly identified by the method, because the group of obtuse triangles are flowing to infinity (taken and modified from [LWE98]).*

one has to use, instead, the weighted Delaunay triangulation, being the weighted Voronoi cells necessarily different.

Additionally, Edelsbrunner *et al.* [Ede98] introduced a discrete-flow method to decide on the existence of cavities or pockets in the complement of the dual sub-complex within the convex hull here called complement sub-complex (i.e. the sub-complex of empty or partially empty triangles). The eligibility of a cavity as part of the complement sub-complex is determined in conformity with the principle of a fluid flowing into a sink. Let us imagine the water flow field generated by filling each triangle with water, so that the water of each *obtuse triangle* flows to the next one until it reaches a pocket or sink represented by an *acute triangle*, as illustrated in Figure 16. This means that every single pocket is formed by growing from an acute triangle.

But, as Edelsbrunner *et al.* [Ede98] noted, some cavities cannot be identified using this discrete flow process, simply because the Delaunay triangulation can lead to the flow of obtuse triangles to the infinity, that is, some cavities do not match acute triangles or tetrahedra. In fact, Edelsbrunner and co-authors formally defined cavities as 3D regions in the complement space of the protein that possess limited accessibility from the complement space itself. Cavities were deliberately defined in this manner to exclude shallow valleys or depressions, like the one shown in Figure 16(b), although some shallow valleys match well-known binding sites.

Summing up, CAST does not solve the fundamental problem of the stopgaps (or delimiters) of some cavities (in particular, wide clefts/grooves), that is, it is not always possible to know where the cavity begins and the outside space occupied by the solvent ends. Liang *et al.* [LWE98] identified this as a difficult problem to overcome; hence, the ‘can of worms’ problem that they mention in their paper. In fact, in CAST, the discrete flow condition (or acute triangle condition) is not satisfied for all types of cavities; it is only valid for the types of cavities considered by Edelsbrunner *et al.* [Ede98] and Liang *et al.* [LWE98], say pockets with $i$ mouth openings, with $i = 0, 1, \ldots, n$, and $n \in \mathbb{N}$.

CAST is the basis of other methods and systems, namely: CASTp web server [BNL03], SplitPocket web server [TDCL09] and RobustVoids [SDP*13], just to mention a few of them. CASTp is also based on the theory of $\alpha$-shapes, and arguably can detect all pockets and voids of a given protein, as well as the surface atoms participating at each cavity. SplitPocket also uses the weighted Delaunay triangulation and the discrete flow procedure to predict each pocket of a given protein. But, unlike CAST, it utilizes not only geometric information but also physicochemical and evolutionary information (e.g. conservation index) for putative binding cavities. RobustVoids builds on the weighted Delaunay triangulation to construct a filtration of $\alpha$-shapes to extract pockets and voids in a robust manner with the user assistance. The accuracy of this system comes from the fact that cavities are correctly determined independently of the small inaccuracies resulting from crystallographic measurements (X-ray crystallography) or the perturbation of atomic radii, which, as widely known, are determined empirically.

### 8.4. GP method

The geometric potential (**GP**) method is due to Xie and Bourne [XB07]. It is similar to CAST in the sense that the carving process has the effect of peeling empty triangles and tetrahedra off the convex hull (i.e. the Delaunay triangulation). However, the peeling-off of simplices is based on empirical parameters like the maximum size of 30.0 Å for a ligand binding pocket.

The steps of the GP method are the following:

- *$C_\alpha$ atom-based structure*. Firstly, one constructs the protein structure from its $C_\alpha$ atoms (or alpha carbon atoms), as shown in Figure 17(a). An amino acid (or, amino acid residue, to be more precise) consists of an amino group ($NH_2$), a hydrogen atom (H), a carboxyl group (COOH) and a side chain (R) bound to a $C_\alpha$ atom [Pro14]. $C_\alpha$ atoms are the central atoms of amino acids that form a protein.
- *Convex hull*. Secondly, one constructs the convex hull (i.e. Delaunay triangulation), as illustrated in Figure 17(b).

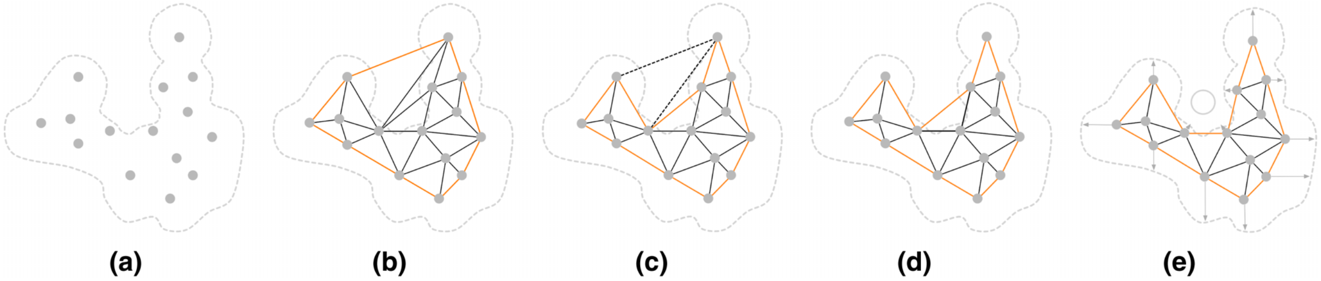


**Figure 17:** *GP method [XB07]: (a) $C_\alpha$ atom-based structure (grey points); (b) convex hull (in orange) and Delaunay triangulation (in dark grey); (c) first carving procedure that removes simplexes whose edges are longer than 30.0 Å (black dashed line segments); the resulting environmental boundary (i.e. outer envelope of the protein) is represented by orange solid line segments; (d) second carving procedure removes k-simplexes circumscribed by spheres with radius larger than 7.5 Å (in orange); this results in the inner envelope of the protein (i.e. protein boundary); (e) geometric potential (GP) and residue surface direction are used to predict binding cavities (taken and modified from Xie and Bourne [XB07]).*

- *First carving step*. Thirdly, one proceeds to the peeling of the tetrahedra from the convex hull; this carving procedure is limited to simplexes whose edges are longer than 30.0 Å (black dashed lines), as depicted in Figure 17(c). The resulting triangulation is bounded by the so-called environmental boundary, which functions as the outer envelope of the protein.
- *Second carving step*. Then, one proceeds to the further peeling of the tetrahedra circumscribed by spheres with a radius larger than 7.5 Å. This results in the inner envelope of the protein, also called protein boundary, which mostly overlaps the outer envelope. See Figure 17(d).
- *Prediction of binding cavities*. Finally, it comes the time of predicting where binding cavities are, as illustrated in Figure 17(e). For that purpose, one uses shape descriptors such as the geometric potential and residue surface direction for each $C_\alpha$ atom.

The novelty of the GP method is twofold:

- The use of $C_\alpha$ atoms of a given protein instead of its entire set of atoms. This speeds up the algorithm because we are considering one atom per amino acid instead of its nine atoms (excluding the side chain), but it produces a very rough approximation that leads to significant geometric inaccuracies. Indeed, the GP method uses a coarser atomic structure, where each $C_\alpha$ atom features an amino acid.
- The use of GP parameter as a new shape descriptor capable of distinguishing cavities that bind from those that do not bind ligands.

Xie and Bourne [XB07] used the following formula:

$$P = d + \sum_i \frac{d_i}{D_i + 1.0} \frac{\cos(\alpha_i) + 1.0}{2.0} \tag{5}$$

to calculate the value of the geometric potential $P$ at each $C_\alpha$, where $d$ stands for the distance of the $C_\alpha$ atom to the environmental boundary, $d_i$ is the distance of its $i$th neighbouring $C_\alpha$ atom to the environmental boundary, while $D_i$ and $\alpha_i$ denote the distance and direction to its $i$th neighbouring $C_\alpha$ atom; note that we only consider the $i$th neighbouring $C_\alpha$ atoms belonging to the protein boundary, with the further condition that they are not obstructed by other residues within the protein boundary.

Then, it remains to calculate the geometric potential for each putative binding cavity, which is given by the average of the geometric potentials for all $C_\alpha$ atoms within the cavity. A cavity is considered as a ligand binding site if its geometric potential is around 50 (on the scale of 0–100); otherwise, the cavity does not qualify as ligand binding site, being its geometric potential usually close to zero.

### 8.5. MOLE

**MOLE** [PKKO07] is a follow-up of CAVER [POB*06] (see Section 7.2), both developed by Petřek and colleagues. CAVER is a grid-and-surface method, while MOLE is a Voronoi tessellation-based method, though CAVER has later evolved to incorporate Voronoi tessellations in its direct follow-ups, CAVER 2.0 [MBS08] and CAVER 3.0 [CPB*12].

As argued by Petřek *et al.* [PKKO07], CAVER suffers from two drawbacks: (i) the use of grid makes it very memory space and time-consuming in exploring large ramified channels; (ii) the introduction of unavoidable grid approximation errors. On the contrary, MOLE takes advantage of the Voronoi tessellation to find pathways defined by Voronoi vertices in the empty space corresponding to channels, tunnels and pores (Figure 18). Such pathways defined by the Voronoi tessellation's edges are found using Dijkstra's pathfinder so that such cavities are found with greater accuracy, in less time and is fully automated when compared to CAVER. Superficial cavities like clefts/grooves are determined with the help of the convex hull that encloses the molecule.

See [SSVB*13] for further details about a more recent follow-up of MOLE, called **MOLE 2.0**, which also estimates physicochemical properties of the identified channels, such as, hydropathy, hydrophobicity, polarity, charge and mutability.

### 8.6. Medek *et al.*'s method

This method is focused on the computation of channels, as proposed by **Medek** *et al.* [MBS07]. It is based on the Delaunay triangulation

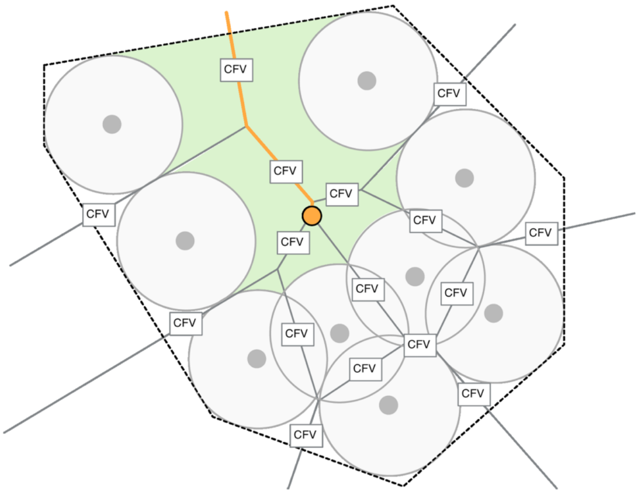


**Figure 18:** *Detecting cavities using MOLE [PKKO07]: Two-dimensional example of the Voronoi diagram of a molecule comprised by a set of atoms (grey spheres). The convex hull is represented as dotted black lines and each Voronoi edge is label with a cost function value (CFV). The Dijkstra's algorithm is accomplished using each CFN from a user-given start point (orange small sphere). The path delineated by the previous algorithm (orange line) is identified as a cavity (pictures taken and modified from [PKKO07]).*

(the dual of Voronoi diagram), which has the advantage of functioning also like the envelope of the molecule, in a way similar to convex hull. Indeed, the convex hull is easily found from Delaunay triangulation. However, for performance purposes, Medek *et al.* do not use the exact formulation of the Delaunay triangulation of a set of points, but instead a Delaunay triangulation of a set of spheres representing atoms. See [KCK04] for further details about the Voronoi diagram of a set of spheres, also referred as the additively weighted Voronoi diagram or Euclidean Voronoi diagram of spheres.

Such a Delaunay triangulation of a set of spheres can be then interpreted as a weighted graph. Two simplifications, conservative and approximate, were introduced to give different weights to the graph. The conservative simplification sets the radii of all atoms to the biggest atom's radius, whereas the approximate simplification assumes that all atoms have identical radii. The authors show that the ideal tunnel is obtained from the graph using a modified Dijkstra algorithm, in the sense that Dijkstra's pathfinder is optimal and complete, it finds the lowest cost path (if it exists) along the interior of a channel. Note that Dijkstra's pathfinder is limited by the convex hull, which significantly shortens its computation time. Both approaches provide a good trade-off between tunnel quality (without noticeable loss of accuracy) and computational time. Although the conservative simplification gives less accurate results, it is faster than its approximate counterpart due to its greater simplicity. Both simplifications show a much better ratio of speed to accuracy when compared to CAVER, although the tests only considered two molecules with little less than 2500 atoms.

### 8.7. Kim *et al.*'s method (KCC*)

One of the main limitations of $\alpha$-shapes stems from the assumption that, in a set of spheres, all spheres are of the same size [KKS01a] [KKS01b] [KSK*06]. Edelsbrunner tried to solve this problem through the generalization of $\alpha$-shapes to weighted $\alpha$-shapes [Ede95], but, even so, they did not take into consideration the variations in size of input spheres, in the sense that the proximity among spheres is not fully described in relation to Euclidean metric [KSK*06].

With this in mind, Kim *et al.* [KCKC06] proposed a method based on $\beta$-shapes, which take into account distinct van der Waals (vdW) radii for atoms (Figure 19). In this sense, beta shapes can be seen as a generalization of alpha shapes. Essentially, they proposed an algorithm that first determines the Voronoi diagram of vdW atoms of a given protein. Note that the Voronoi diagram of atoms is not the same as the ordinary Voronoi diagram for points (centres of atoms) since the Euclidean distance is measured not relative to the centres of the atoms, but relative to the surface of the atoms. After determining such extraordinary Voronoi diagram, one constructs the corresponding beta shape using a spherical probe.

Following the same line of research, **Kim** *et al.* [KCC*08] built up a blending mesh of triangles derived from a surface generated from blending atoms, as illustrated in Figure 19. Then, they construct the

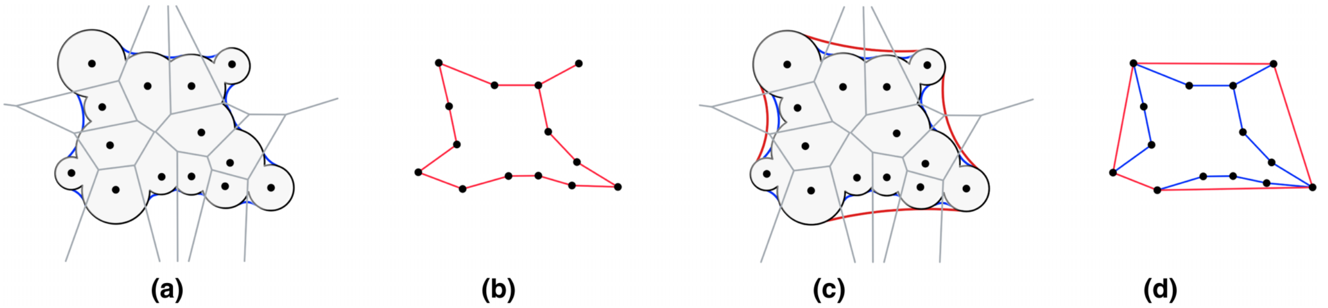


**Figure 19:** *(a) van der Waals surface in black, and inner blending surface as a connected arrangement of blue and black spherical patches; (b) inner blending mesh constructed from the atomic centres and blending surface; (c) outer blending surface as a connected arrangement of red and black spherical patches; (d) outer blending mesh as the convex hull of atomic centres (taken and modified from Kim et al. [KCC*08]).*

convex hull from such a blending mesh. Cavities are found in places of the convex hull that are not occupied by the blending mesh. Also, Kim *et al.* [KCC*08] use the Voronoi diagram of atoms, not the Voronoi diagram of atom centres, to easily calculate the molecular surface.

### 8.8. MolAxis

This method was developed by Yaffe *et al.* [YFW*08]. **MolAxis** relies on two geometric concepts: $\alpha$-shapes and medial axis. The use of $\alpha$-shapes means that the molecule is seen as a set of 3D balls featuring constant-radius atoms. In respect to the medial axis of a geometric object, it can be defined as the set of points that possess one or more closest points on the boundary of such object [Blu67]; for example, the midpoint of a straight line segment, the centre of a sphere or the axis of a cylinder. In the case of MolAxis, the geometric object at hand is the vdW surface of a molecule. Taking into account that the surface is closed in 3D, we end up having two medial axes: inner medial axis and outer medial axis. The inner medial axis can be understood as the skeleton of the molecule, while the outer medial axis is the skeleton of the complement of the molecule in 3D, that is, the space outside the molecule.

MolAxis is focused on the computation of outer medial axis because it indicates where channels and tunnels of a molecule are. The outer medial axis is similar to Voronoi pathways of MOLE (see Section 8.5) in the complement space because MolAxis takes advantage of the inner and outer medial axes of the protein built upon the Voronoi diagram. The main novelty of MolAxis is how it approximates the outer medial axis of the complement space of the molecule to construct channels. That is, it approximates the additively weighted Voronoi diagram, as already used by Medek *et al.* [MBS07]. This approximation is the result of approximating each vdW atom by one or more unit balls, that is, the weighted Voronoi diagram is approximated by the Voronoi diagram of atomic centres. But, the outer medial axis can be calculated in an exact manner using the weighted Voronoi diagram, also called Apollonius diagram [BD05].

### 8.9. Fpocket

Guilloux *et al.* [LGST09] introduced **Fpocket**, which primarily builds upon the Voronoi diagram of the set of centres of the atoms of a given protein (see Figure 20). For that purpose, one computes the Voronoi tessellation of the atomic centres, what is performed using the publicly available qvoronoi's source code at http://www.qhull.org , a well-known package that firstly calculates the convex hull of a set of points through the Quickhull algorithm. However, Fpocket does not use any triangulation.

Instead, Fpocket uses the Voronoi tessellation and alpha spheres. Every single alpha sphere is centred at a distinct Voronoi vertex, although an alpha sphere is smaller than its homologous Voronoi ball. Its radius is given by the distance from its Voronoi vertex to the closest atom centre minus the radius of such atom. Thus, alpha spheres in the complement space of a protein are tangential spheres in contact with surface atoms.

Recall that a Voronoi vertex is the centre of an empty circumsphere, called Voronoi ball, through four points, which coincides with an empty circumsphere of the Delaunay triangulation; this is so because the Voronoi tessellation and Delaunay triangulation are dual structures. So, in conformity with the empty circumsphere rule of the Delaunay triangulation, an alpha sphere has always four points of contact with surface atoms, featuring thus the local curvature of the molecular surface. That is, cavities are located where we find alpha spheres; this thus requires the use of some clustering of alpha spheres to form such cavities. In other words, locating alpha spheres is equivalent to detect cavities on protein surfaces.

The main steps of the method are as follows:

- *Voronoi tessellation*. Firstly, one constructs the Voronoi diagram of the atomic centres, as illustrated in Figure 20(a).
- *Computation of alpha spheres*. Secondly, one determines the contact alpha spheres centred at the Voronoi vertices in the complement space of the molecule. The minimum size of an alpha sphere is naturally solvent (water) probe sphere, which has 1.4 Å of radius, but bigger radii may be used. This allows us to immediately discard solvent inaccessible alpha spheres. Nevertheless, we have to define a maximum size for alpha spheres to also discard rather exposed alpha spheres. This *a priori* pruning of too small and big alpha spheres significantly reduces the number of false positives and false negatives for cavities. This is illustrated in Figure 20(b).
- *Clustering of alpha spheres*. Thirdly, one proceeds to the clustering of alpha spheres, as shown in Figure 20(c). The clustering procedure uses the proximity and neighbourhood relationships of Voronoi vertices to aggregate their alpha spheres into separate clustered pockets within the empty complement space.
- *Pocket ranking*. Finally, the ranking of cavities takes place to check their ability to bind ligands. For that purpose, one uses a straightforward scoring scheme that is based on the partial least squares (PLS) regression, which is somehow related to the principal components regression. This has the effect of further reducing the number of false positives and false negatives for cavities.

### 8.10. CAVE

**CAVE** was introduced by Busa *et al.* [BHH*10] to solely identify voids in proteins. Its leading idea is to construct an enveloping triangulation enclosing each void. That is, it does not make usage of $\alpha$-shapes, Voronoi diagram, $\beta$-shaped, or Apollonius diagram. The enveloping triangulation is a tetrahedralization whose vertices are the atomic centres, so it is a Delaunay-like triangulation in 3D.

As its authors noted, van der Waals radii of atoms are augmented by the (water) probe sphere radius. That is, the number, sizes and shapes of cavities are strongly dependent on the probe radius. The goal is to construct a minimal closed 2-cycle (envelope) of triangles enclosing each void. Any tetrahedron' triangle intersecting the void is not considered as being part of the minimal closed 2-cycle of a void. Let us mention that CAVE also allows for detecting voids, as well as for studying properties of each void, namely its location, boundary atoms, volume and surface area.

### 8.11. VoroProt

**VoroProt** was proposed by Olechnovic *et al.* [OMV11]. It resembles MOLE and MolAxis because they are all based on the additively weighted Voronoi diagram of a set of atoms, which is also known as Apollonius diagram [EM94, EK06]. Therefore, a molecule is a set of atoms represented as vdW spheres. Then, one constructs the Apollonius diagram, which can be seen as the Voronoi diagram of the set of vdW spheres. At last, it takes place the construction of the Apollonius graph (i.e. the dual of the Apollonius diagram), which works as the delimiter of the molecule. Apollonius graph unequivocally defines the set of atoms neighbouring each atom. This construction is similar to the Delaunay triangulation, with the difference that one uses spheres instead of points, and tangent spheres instead of circumspheres (circumsphere rule). As for MOLE and MolAxis, cavities in the complement space are detected using skeletal pathways (for invaginations, tunnels and channels) in the Apollonius diagram together with the boundaries of the Apollonius graph (surface grooves and voids). Thus, there is no room for ambiguity in locating entries and exits of cavities of the molecule.

### 8.12. Lindow *et al.*'s method (LBH)

Similar to Voroprot, **Lindow** *et al.*'s method [LBH11] also relies on the Apollonius diagram (i.e. the Voronoi diagram of spheres). It aims at identifying transport pathways in molecules. Such pathways are determined using depth-first search in the graph built from the edges and nodes of the Apollonius diagram.

Unlike Voroprot, Lindow *et al.* did not use the Apollonius graph as a delimiter of the molecule. Instead, they used omnidirectional casting of rays from every single Apollonius vertex to determine whether it lies in a cavity of not; more specifically, if more than 50% of rays hit the molecular surface, one concludes that the vertex belongs to a cavity. This threshold of 0.5 is a value that leads to approximately discard the vertices outside the convex hull of the molecule. Therefore, there is no ambiguity in identifying cavity entries and exits.

### 8.13. BetaVoid

**BetaVoid** was introduced by Kim *et al.* [KCL*14] to identify voids exclusively, that is, the method was not designed to identify cavities in general. It is a freeware solution for molecular void recognition and accurate computation of void volume, area and topology. It relies on a geometric formalization of molecular voids allied with an analytic approach that uses the Voronoi diagram of spherical atoms and the $\beta$-complex. The proposed algorithms identify both van der Waals and Lee-Richards solvent-accessible voids, along with the residues that belong to each void atom. Also, BetaVoid allows users to vary atom radii from the default values of the Bondi radii. One of its main contributions is a general and unified geometric framework that allows us to analyse molecular voids in an efficient and mathematically correct manner.

### 8.14. CCCPP

More recently, Benkaidali *et al.* [BAM*14] introduced an alpha-shape variant, called **CCCPP**, which supposedly takes advantage of the size and the shape of the ligand. This method essentially finds the empty space where channels, pockets and cavities are in the complement of the alpha shape of the protein to its convex hull. That is, the convex hull works here as the outer envelope of the protein.

Therefore, the convex hull works as the ceiling for each concavity of the protein. As their authors argued, the focus of the method is on the shape of the channels (i.e. empty space inside the convex hull), not the shape of the protein. To find those channels, one uses a door-in-door-out principle for empty (or partially) tetrahedra. This principle is similar to the discrete flow principle, with the difference

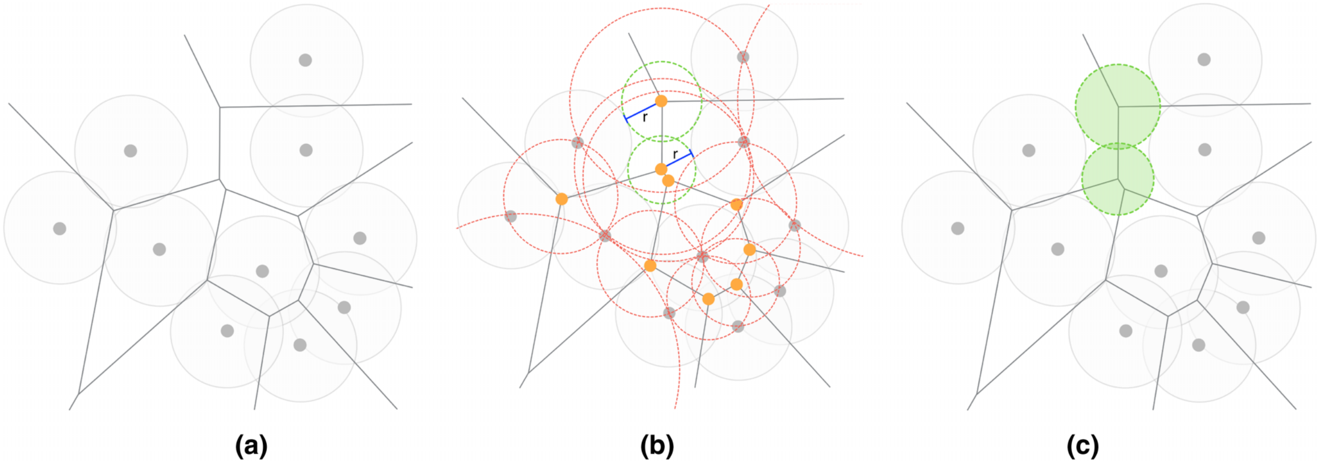


**Figure 20:** *Detecting cavities through Fpocket [LGST09]: (a) Voronoi diagram of the atomic centres; (b) similar to a Voronoi ball (dotted red circles), each $\alpha$-sphere (dotted green circle) is also centred at a Voronoi vertex (orange points), but it is a contact sphere that is tangential to surface atoms (solid grey circles); (c) cluster of $\alpha$-spheres (solid green circles) that fill a cavity.*

that now the convex hull is functioning as the outer boundary for all channels.

Channels are found as follows. Starting from the Delaunay triangulation, one constructs a graph for empty (or partially empty) tetrahedra inside the convex hull. This is a graph whose nodes represent empty (or partially empty) tetrahedra, and edges represent triangles bounding those tetrahedra, much like we do in the construction of the Voronoi diagram from the Delaunay triangulation, with the difference that we are not imposing any geometric constraints to such graph, here called facial graph. A channel is a connected subgraph of the facial graph.

Note that Benkaidali *et al.* argue that the spherical model for ligands, usually given by a probe sphere featuring a water molecule is often not adequate for the detection of channels, so they ended up by applying the cylindrical model instead. The adoption of the cylindrical model is seen by the authors as a step forward in the conventional alpha-shape approaches.

### 8.15. Tessellation-based methods: Discussion

Looking at Table 6, we observe the following:

- *Molecular Surfaces*. These methods do not use any molecular surface. To identify molecular cavities, they only use vdW atoms or their centres; at most, we can say that they indirectly use the vdW surface. More specifically, $\alpha$-shape and Voronoi-based methods use atomic centres and, implicitly constant-radius spheres to represent atoms, while $\beta$-shape methods and Apollonius-based methods take advantage of varying-radius spheres to represent those atoms.
- *Limitations*. As shown in Table 6, tessellation-based methods do not suffer from significant limitations indeed. In a way, these limitations are all related to accuracy in identifying not only the correct location of each binding cavity of a given protein, but also its number of surface atoms and its boundary—and, subsequently, its area and volume—in the complement space. In respect to *$\alpha$-shape methods*, they are focused on the occupied space by a protein so that any tiny empty space less than a water molecule inside a tetrahedron originates a false positive. Also, two buried chambers interconnected via a small channel with a radius less than the water molecule is reported as a single cavity, when it consists of two distinct cavities or a cavity with two sub-cavities. This shows that alpha shapes are sensitive to false negatives. In fact, $\alpha$-shape methods tend to fail to detect wide surface pockets and shallow valleys. On the other hand, *$\beta$-shape methods* produce more accurate results than $\alpha$-based methods because they are based on vdW atom-featuring spheres instead of atomic centres.
  On the contrary, *Voronoi-based methods* put their focus on the empty complement space, filling it with contact spheres, called alpha spheres, centred at Voronoi vertices. By using the least radius of 1.4 Å for alpha spheres, one guarantees the number of false positives and false negatives is reduced to a minimum. The cavities are where there is a higher density of contact spheres. Furthermore, they provide a skeleton per channel in a way similar to medial axis. Finally, *Apollonius-based methods* produce more accurate results than Voronoi-based methods, because they are based on vdW atom-featuring spheres instead of atomic centres. For example, the skeletal pathways of channels approximate the medial axis of the complement space.
- *Cavities*. With the exception of a few cavity detection methods, we can say that tessellation-based methods are accurate in identifying cavities of proteins. In general, Voronoi- and Apollonius-based methods are adequate to identify any cavity, in particular channels; surface pockets are also easily identified because of the use of the convex hull, Delaunay triangulation or Apollonius graph, which work as delimiters of the protein.

**Table 6:** *Tessellation-based methods.*

| | | | | | | | Cavities | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Molecular surfaces | | Limitations | | Pockets | | | | |
| Methods | Reference | Tessellation | SA/vdW | CH | EAT | MOA | Clefts/Grooves | Invaginations | Tunnels | Channels | Voids |
| APROPOS | [PFF96] | $\alpha$ | ● | | ● | ● | ● | ● | ● | ● | ● |
| CAST | [LWE98] | $\alpha$ | ● | ● | | ● | ● | ● | ● | ● | ● |
| GP | [XB07] | $\alpha$ | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| MOLE | [PKKO07] | AD | ● | ● | | | ● | ● | ● | ● | ● |
| Medek *et al.* | [MBS07] | DT | ● | ● | | | | | ● | ● | |
| KCC* | [KCC*08] | $\beta$ | ● | ● | | ● | ● | | | | |
| MolAxis | [YFW*08] | AD, MA | ● | | | | ● | ● | ● | ● | ● |
| Fpocket | [LGST09] | $\alpha$, Voronoi | ● | | | | ● | ● | | | |
| CAVE | [BHH*10] | ET | ● | | | ● | | | | | ● |
| VoroProt | [OMV11] | AD | ● | | | | ● | ● | ● | ● | ● |
| LBH | [LBH11] | AD | ● | ● | | | ● | ● | ● | ● | ● |
| BetaVoid | [KCL*14] | $\alpha$ | ● | | | | | | | | ● |
| CCCPP | [BAM*14] | $\alpha$ | ● | ● | | | ● | ● | ● | ● | ● |

Abbreviations: AD, Apollonius diagram [EK06]; DT, Delaunay triangulation; MA, medial axis [Blu67]; ET, enveloping triangulation [BHH*09]; SA/vdW, set of atoms/van der Waals surface; CH, convex hull; EAT, empirical alpha tuning; MOA: mouth-opening ambiguity.

As far as we know, there is not any tessellation-based method in the literature to identify cavities into sub-cavities. Nevertheless, it is straightforward to accomplish that with Voronoi- and Apollonius-based methods because they produce skeletal pathways and their branches.

## 9. Consensus Methods

To the best of our knowledge, **Metapocket** is the only consensus method found in the literature, which was proposed by Huang [Hua09]. Consensus methods are approaches that combine the results produced by two or more cavity detection techniques. More specifically, this method combines the predictions of four methods to improve the success rate in predicting the location of binding cavities; three of these methods are purely geometric (LIGSITE$^{cs}$, PASS and SURFNET), while the fourth is an energy-based method (Q-SiteFinder [LJ05]).

Since these four methods have different ranking scoring functions, it is hard to compare and evaluate the predictions directly. Therefore, a z-score is calculated separately for each cavity using different methods, to make the ranking scores comparable. Probes within a given distance threshold are grouped together as a cluster, and each cluster is ranked by a scoring function consisting of the sum of the z-scores of the cavities in that cluster. For the dataset of proteins referred by Huang [Hua09], MetaPocket improved the success rate up to 90% over individual methods.

Later, Zhang *et al.* [ZLL*11] continued this work by adding more four methods (GHECOM, ConCavity, POCASA and Fpocket) to further improve the prediction success rate. This resulted in the development of **MetaPocket 2.0**, a consensus method which combines the predicted cavity sites of a total of eight methods.

## 10. GPU-Based Methods

In the last decade, we have noted an increasing use of GPU computing in molecular modelling, rendering and visualization [KBE09, SSE*10, DBG10, LBH11, KKC*11, CVT*11, PTRV12, TPS12, PRV13, DG13, PTRV13, LLNW14, DCD*14, DG15, HGVV16]. However cavity detection methods taking advantage of GPU processing power are not so commonly found in the literature; the exception lies in the methods we describe below.

### 10.1. Parulek *et al.*'s method

After introducing an implicitly defined formulation for SES [PTRV12], **Parulek** *et al.* [PTRV13] proposed a cavity detection algorithm solely for molecular visualization purposes, that is, they were not concerned about benchmarking the accuracy of their algorithm against a ground-truth of already known binding cavities. Arguably, most computations were performed on GPU using CUDA and GLSL, but no details about the implementation of their method were published.

Therefore, this is a surface-based method, which has the particularity of using a random sampling of the domain, much like in McVol [TU10] (see Section 5.4). More specifically, they generate point samples inside balls centred at atomic centres, but with a radius that is twice the vdW radius of each atom. The samples inside SES are dropped straight away. The remaining samples outside SES are used to determine the cavities on SES.

Parulek *et al.* take advantage of an implicit formulation of SES to determine the direction of the gradient at each point sample outside SES. Similar to grid-based methods with scanning directions, this gradient vector and its symmetric vector determine the existence of a cavity if they hit two opposite boundary walls of SES. As the last step, they use mutual visibility test between pairs of points satisfying the scanning direction condition between walls of SES with the goal of clustering the sampled points into distinct cavities. However, as the authors mentioned in their paper, their method may not identify all and especially shallow cavities [PRV13].

### 10.2. Krone *et al.*'s method

Similar to Parulek *et al.*'s method, **Krone** *et al.* [KRS*13] used an implicit formulation for molecular surfaces, not only for representing and modelling molecular surfaces but also to help in extracting the molecular cavities for visualization purposes. More specifically, they use a Gaussian surface that better adjusts to SES, in conformity with the parameter set in [GP95] and [Ric77]. This work is a follow-up of their previous work detailed in [KFR*11], which arguably was the first method to extract cavities in real-time extraction. Cavities are detected using an ambient occlusion-based visibility criterion due to Borland [Bor11], who used an ambient occlusion-based approach to get an adequate visualization of the internal structure of proteins. Once again, this method focuses on molecular rendering and visualization of cavities, and not on the accuracy of the method in detecting and locating cavities, even with respect to benchmark results.

Thus, the leading idea of the method was to obtain a surface segmentation with noticeable cavities. For that purpose, the molecular surface is triangulated beforehand using the marching tetrahedra algorithm due to Doi and Koide [DK91]. The resulting triangles are then tagged as either shadowed or unshadowed, what depends on their computed ambient occlusion (AO) factors in relation to an user-defined threshold. It is clear that shadowed triangles are those that belong to eventual cavities so that they are clustered into cavities using the principle of connectedness, that is, two adjacent shadowed triangles in the molecular surface belong to the same cavity. This clustering-based segmentation is based on the labelling technique due to Hawick *et al.* [HLP10], which was specially designed for GPU computing. However, because it mostly aims at molecular visualization, its authors did not embark in any benchmarking with other cavity detection method regarding accuracy (e.g. the number of cavities and their locations).

### 10.3. PLB-SAVE

To the best of our knowledge, the first cavity detection method to run *entirely* on GPU (via CUDA) is due to Lo *et al.* [LWP*13], and is called **PLB-SAVE**. Furthermore, it uses the LigASite dataset of binding sites for benchmarking comparisons [DLW08].

The leading idea of this method is to take advantage of the Connolly function for segmentation of the molecular surface into cavities and its complement. While using the Connolly function to divide the molecular surface into convex, concave and saddle patches is not a novelty [CCL03], their segmentation produces numerous fine patches to be useful in cavity detection. One ideally requires a coarser surface segmentation, and especially with larger binding cavities. Natarajan *et al.* [NWB*06] introduced a Morse theory-based segmentation of molecular surfaces to solve this problem. Instead of using the Connolly function, Natarajan *et al.* used the Mitchell-Kerr-Eyck function [MKE01] as a way to merge neighbour segments into larger segments by simplifying the atomic density function.

PLB-SAVE essentially is a grid-based method applied to the set of atoms of a given molecule. This method maps each atom over their occupied voxels in the 3D space. This way, one can identify the protein surface, from which one calculates the solid angles associated with each atom. In practice, PLB-SAVE thus uses a discrete version of the Connolly function. Instead of measuring the solid angle Ω associated to each surface point, one measures the solid angle of each surface voxel, which is given by the following expression:

$$\Omega = \frac{n}{N} \cdot 4\pi, \tag{6}$$

where $N$ is the whole number of voxels occupied by a probe sphere of 6 Å centred at each surface voxel, and $n$ denotes the number of those voxels overlapping the protein. This means that $\Omega \in [0, 4\pi]$; if $\Omega \in [0, 2\pi[$, the corresponding surface voxel lies in a convex region of the surface; if $\Omega \in ]2\pi, 4\pi]$, the corresponding surface voxel is located in a concave region of the surface; if $\Omega \approx 2\pi$, the corresponding voxel belongs to an approximately flat region of the surface.

Next, one proceeds to the clustering of connected surface voxels around those with similar, highest solid angles, that is, only the concave regions concerning cavities are taken into account. Note, that the Connolly function is translation- and rotation-invariant because it is defined over the molecular surface. However, clustering voxels with similar solid angle levels can often lead to misleading results (i.e. unreliable cavity locations). This is so because a binding cavity may include concave and approximately flat regions, as a result of the fine-grain segmentation that results from the Connolly function. To overcome this problem of significant variations in the solid angle of a cluster of voxels, Lo *et al.* introduced the concept of average depth for a cavity [LWP*13].

### 10.4. CAVE-CL

**CAVE-CL** is an OpenCL implementation of the CAVE method that is authored by Bus̆a *et al.* [BHH*09, BHH*10, BHHW15]. This method was designed to detect voids solely, also called internal cavities. CAVE-CL operates on a set of balls featuring the atoms of a molecule, but the size of each atom is increased with the radius of the probe sphere, as is usual for the SAS.

The atom centres are vertices of the so-called envelope triangulation (ET), which can be seen as a sub-complex (or subset) of the nerve of an alpha shape triangulation. After building up this envelope triangulation, we are ready to detect where the voids of the protein are. Each void is commonly encountered inside a closed polyhedron that makes part of the envelope triangulation.

### 10.5. Kim *et al.*'s method (KLKK)

**KLKK** is a hybrid method due to Kim *et al.* [KLKK16], which is capable of detecting voids, chambers, tunnels and channels. It operates simultaneously on two GPU data structures (via CUDA): a sphere tree and a grid of voxels. The sphere tree is a novel representation of a given protein (i.e. the set of atoms). In fact, one generates a sphere tree for each peptide chain of a given protein; a sphere tree is held in GPU memory as a 1D array. This new representation of a protein allows us to accelerate the proximity search queries on GPUs.

After forming the sphere tree, one constructs an approximate convex hull that encloses the protein with the help of such proximity queries on the GPU. The voxels inside an approximate convex hull of the molecule are then classified as follows: occupied, empty and empty-boundary. The voxels occupied by a given protein denote the absence of cavities; the empty voxels—in particular those containing Voronoi edges—identify the location of cavities on or inside the protein; the empty-boundary voxels are those that identify exit/entrance doors for channels and tunnels. Furthermore, this method uses the Dijkstra algorithm to determine the shortest path from a chamber to an exit mouth of a tunnel or channel. Note that KLKK takes advantage of an approximate convex hull of the molecule to distinguish the empty voxels of cavities inside the convex hull from those empty voxels outside the convex hull.

### 10.6. CriticalFinder

**CriticalFinder** is a grid-and-surface method proposed by Dias *et al.* [DNJG17], whose program entirely runs on GPU via CUDA. This method builds upon the theory of critical points (also known as Morse theory), and relies on the assumption that each cavity can be identified by a cluster of *approximate* critical points of the same sort. These approximate critical points are corners of voxels intersecting the Gaussian surface that encloses the protein. The result is a *meaningful* segmentation of the protein surface into cavities and saliences.

CriticalFinder calculates the approximate critical points of the Gaussian scalar field (or function) that describes the molecular surface through the eigenvalues of its Hessian matrix, that is, it takes advantage of curvature analysis. Other research works have already used curvature information (e.g. Natarajan *et al.* [NWB*06]) to segment molecular surfaces. However, there is no evidence that the resulting segmentation is a meaningful segmentation in terms of cavities, because no comparison was carried out relative to any ground-truth dataset of known binding sites (e.g. LigASite at http://ligasite.org/).

### 10.7. GPU-based methods: Discussion

Given the new advances in parallel computing (e.g. GPU-based applications) in last decade, we decided to define a category specifically dedicated to GPU-based methods, although they are clearly framed in geometric categories, as indicated in Table 7. Let us then discuss the characteristics of these methods:

**Table 7:** *GPU-based methods.*

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | Cavities | | | | | |
| | | Molecular surfaces | | | | GPU | Limitations | | Pockets | | | | | |
| Methods | Reference | SA/vdW | SES | GS | CH | Computing | GSS | MOA | Clefts/Grooves | Invaginations | Tunnels | Channels | Voids | Category |
| Parulek *et al.* | [PTRV13] | ● | ● | | | CUDA/GLSL | | ● | | ● | ● | ● | ● | Surface-based |
| Krone *et al.* | [KRS*13] | ● | | ● | | CUDA | | | ● | ● | ● | ● | ● | Surface-based |
| PLB-SAVE | [LWP*13] | ● | | | | CUDA | ● | ● | | ● | ● | ● | ● | Grid-based |
| CAVE-CL | [BHHW15] | ● | | | | OpenCL | | | | | | | ● | Tessellation-based |
| KLKK | [KLKK16] | ● | | | ● | CUDA | ● | | ● | ● | ● | ● | ● | Grid-based; Voronoi |
| CriticalFinder | [DNJG17] | ● | | ● | | CUDA | | ● | ● | ● | ● | ● | ● | Grid-and-surface-based |

Abbreviations: SA/vdW, set of atoms/van der Waals surface; SES, solvent-excluded surface; GS, Gaussian surface; CH, convex hull; GSS, grid-spacing sensitivity; MOA, mouth-opening ambiguity.

- *Molecular Surfaces*. Despite the fact that these six methods belong to three different geometric categories, they all rely on the *set of atoms* (SA) or van der Waals surface of the protein. Nevertheless, Parulek *et al.* [PTRV13], Krone *et al.* [KRS*13] and KLKK [KLKK16] also take advantage of surface formulations as the SES, *Gaussian surface* (GS) and *convex hull* (CH), respectively, to represent the molecular surface somehow.
- *Limitations*. Taking into consideration that every single GPU-based method belongs to some geometric category of methods, each one of them suffers from the limitations inherent to its category. For example, PLB-SAVE and KLKK are grid-based methods, so they are sensitive to grid spacing (GSS), but because KLKK uses the convex hull as the outer envelope of the molecule, it does suffer from MOA. PLB-SAVE is partially ambiguous because of the strict threshold used to classify the convex and concave surface regions through a discrete variant of the Connolly function; as a consequence, it tends to miss shallow grooves. On the other hand, Parulek *et al.* and Krone *et al.* are surface-based methods, but Parulek *et al.*'s method may miss identifying shallow grooves on the molecular surface because the random domain sampling may not sample such cavities in a proper way.
- *Cavities*. As expected, these methods are capable of correctly identifying most cavities of proteins. Nevertheless, as explained above, both Parulek *et al.*'s and PLB-SAVE may miss shallow grooves because of their criteria to identify cavities. As an exception, CAVE-CL, a parallel variant of CAVE (see Section 8.10), was designed to detect voids solely.

As seen from Table 7, the grid-based methods are still in their infancy, so a long way has to be traced in relation to $n$-part cavity detection (i.e. sub-cavities). Furthermore, there is not yet a benchmarking tool to compare different methods regarding performance and accuracy.

Finally, in terms of performance, most GPU implementations of the methods described above were compared with their CPU counterparts [KRS*13, LWP*13, BHHW15]. In contrast, Kim *et al.* [KLKK16] only benchmarked the GPU implementation of their algorithm using an increasing number of GPUs, while Dias *et al.* [DNJG17] adopted both strategies, CPU-GPU and multiple GPU-GPU. As expected, the use of a GPU setup speeds up the execution of the programs relative to CPU setup, and the performance boost is also noticeable when the number of GPUs increases, particularly for proteins with a large number of atoms (approx. 100,000 atoms or more).

## 11. Time-Varying Methods

The cavity detection methods discussed above apply to a single protein conformation at a given time, that is, to a static structure. However, protein molecules, along with their cavities, are dynamically changing their conformation and shape over time. In fact, a major problem with static cavity detection methods is that they miss cavities that become only accessible in dynamic molecular conformations that are different to the crystal conformation [EH07]. However, in the last decade, a few works have addressed tracking the geometric evolution of molecular cavities throughout the course of molecular dynamics (MD) trajectories, as the protein molecule switches between a sequence of stable conformation states. In a more general setting, the reader is referred to Al-Bluwin *et al.* [ABSC12] for more details. A summary of these methods can be found in Table 8.

### 11.1. EPOS$^{BP}$

Seemingly, **EPOS$^{BP}$** was the former method to detect and track transient protein cavities across a sequence of MD snapshots (or time steps) [EH07]. EPOS$^{BP}$ aims at protein-protein interactions. It is based on PASS (see Section 3.4), which is a sphere-based method. Essentially, for a significant number of MD snapshots, the PASS method is used to identify the protein cavities in each snapshot. Each cavity is given an ID to track it during MD trajectories. Surprisingly, Eyrisch and Helms [EH07] noted that all cavities change over the time window of 10 ns, that is, they vanished and reappeared several times over time. Note that to simulate a narrow lapse of 10 ns of biological time may require computing resources of CPU-weeks.

**Table 8:** *Time-varying methods.*

| | | | | | Cavities | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | Pockets | | | | |
| Methods | Reference | Core method | Category | Dynamic trajectories | Clefts/Grooves | Invaginations | Tunnels | Channels | Voids |
| EPOS$^{BP}$ | [EH07] | PASS | Sphere-based | MD | ● | ● | ● | ● | ● |
| TexMol | [BGG*10] | TexMol | Surface-based | NMA | ● | ● | ● | ● | ● |
| dxTuber | [RK11] | dxTuber | Grid-and-sphere-based | MD | ● | | ● | | ● |
| MDpocket | [SBCLB11] | Fpocket | Voronoi, Grid-based | MD | ● | ● | | | |
| PocketAnalyzer$^{PCA}$ | [CPG*11] | LIGSITE | Grid-based | MD, PCA | ● | ● | ● | ● | ● |
| Provar | [AMA*12] | PASS, Fpocket, LIGSITE | Sphere, Grid-based, Voronoi | MD, ED, NMA, CBM | ● | ● | ● | ● | ● |
| PPIAnalyzer | [MPK*12] | LIGSITE | Grid-based | MD, FRODA | ● | ● | ● | ● | |
| CAVER 3.0 | [CPB*12] | CAVER 2.0 | Voronoi | MD | ● | ● | ● | ● | ● |
| TRAPP | [KRH*13] | TRAPP | Grid-based | MD, PCA | ● | | | | |
| LBBH | [LBBH13] | LBH | Tessellation-based | MD | ● | ● | ● | ● | ● |
| trj_cavity | [PEG*14] | trj_cavity | Grid-based | MD | ● | ● | ● | ● | ● |
| Epock | [LCC*15] | POVME | Grid-based | MD | | | | ● | ● |
| Desdouits *et al.* | [DNB15] | GHECOM | Grid-and-sphere-based | MD, PCA | ● | ● | ● | ● | ● |

Abbreviations: NMA, normal mode analysis; MD, molecular dynamics; PCA, principal component analysis; ED, essential dynamics; CBM, constraint-based methods; FRODA, constrained geometric.

Another surprising result was the fact that transient protein cavities are one order of magnitude more than the number of cavities identified for the crystal structures of the apo proteins. This shows that time-varying cavity detection methods are particularly useful in elucidating unknown binding sites, that is, the protein crystal structure lacks information about those missed binding cavities.

### 11.2. TexMol

**TexMol** (Texture Molecular Viewer) is a molecular visualization client software that provides a user interface to a set of software packages, including the one concerning detection and tracking methods for pockets and tunnels [BDST04].

Unlike EPOS$^{BP}$, which is based on MD trajectories, TexMol uses normal mode analysis (NMA) for the computation of small and large time-scale molecular trajectories [BGG*10]. Note that MD runs on the scale of nanoseconds to microseconds, and needs Brownian motion trajectory filtering to tune simulation results. On the other hand, NMA yields longer range molecular trajectories (on the scale of milliseconds to seconds), and trajectory filtering is obtained by selecting a subset $k$ of the eigenmodes of the eigenmode expansion (EME).

Based on the techniques described in Section 6, Bajaj and coauthors [SB06, BGG09] take advantage of time-varying contour trees to track birth, growth, and dissolution of cavities (topology), as well as to compute stable manifolds on these NMA molecular trajectories to track the change of the mouths of cavities (geometry).

### 11.3. dxTuber

In the line of the time-varying methods, which usually detect cavities for an ensemble of conformations, Raunest and Kandt [RK11] developed a mixed grid-and-sphere based technique, called **dxTuber**, which does not neglect alternate protein forms and relies on cavity dynamics. Their technique is capable of detecting all three main types of cavities (voids, channels, and pockets) by making use of protein flexibility and solvent residence probabilities, which are derived from molecular dynamics simulations, using solvent molecules to probe for cavities. Therefore, dxTuber allows studying cavities from a molecular dynamics perspective.

Solvent and protein trajectories computed via molecular dynamics (MD) simulations are converted to a voxel representation of mass-weighed spatial density maps using VMD [HDS96], which outputs protein-internal and protein-external solvent regions. dxTuber then separates both voxel regions and classifies a cavity as a contiguous voxel set of protein-internal regions of high solvent residence probability.

For each type of cavity, a different search algorithm was implemented. Also, dxTuber was compared with SURFNET, CAVER and PyMol to evaluate its computational performance. Only six proteins that contain the most representative protein cavities (voids, channels and pockets) were tested. Since dxTuber relies on molecular dynamics to probe protein cavities, this technique requires a large amount of computational power to perform cavity analysis. Therefore, simulation length, molecular size and voxel resolution directly determine dxTuber's performance.

### 11.4. MDpocket

**MDpocket** relies on Fpocket (see Section 8.9) [SBCLB11]. Recall that Fpocket builds upon the Voronoi tessellation to detect protein cavities. It returns such cavities as clusters of $\alpha$-spheres that are tangential to surface atoms of a given protein.

Tracking of transient cavities is performed using an axis-aligned grid of nodes equally spaced, with each voxel having the size (volume) of 1.0 Å$^3$. Firstly, Fpocket is run in each snapshot, that is, Fpocket is executed as many times as the number $n$ of pre-defined snapshots. Secondly, for each snapshot $l$, each $\alpha$-sphere is assigned to the closest grid node $(i, j, k)$, being then the number $\alpha_l$ of counted $\alpha$-spheres normalized by the number of snapshots as follows:

$$\rho_{(i,j,k)} = \frac{1}{n}\sum_{l=1}^{n}\alpha_l, \tag{7}$$

where $n$ stands for the number of snapshots. It is clear that this originates a cavity density map $\rho$ within the grid. This cavity density map indicates how many $\alpha$-spheres are packed within cavities of the complement space.

Thirdly, for each snapshot $l$, each grid node $(i, j, k)$ is given a binary occupancy parameter $\delta_l$, that is, $\delta_l = 1$ if the node has been assigned at least an $\alpha$-sphere; otherwise, $\delta_l = 0$. It follows a cavity frequency map $\Phi$ over the grid, which is generated through the normalization of the binary occupancy parameter $\delta_l$ associated to each node $(i, j, k)$ across the sequence of snapshots as follows:

$$\Phi_{(i,j,k)} = \frac{1}{n}\sum_{l=1}^{n}\delta_l. \tag{8}$$

This means that the grid node $(i, j, k)$ is persistently accessible to the solvent if $\Phi_{(i,j,k)} = 1$, blocked if $\Phi_{(i,j,k)} = 0$, and transiently accessible if $0 < \Phi_{(i,j,k)} < 1$. Summing up, encoding cavities in a grid over time allows us to track cavities during MD trajectories. Thus, MDpocket renders a more generic and less error-prone identifying and tracking technique for cavities than EPOS$^{\mathrm{BP}}$, provided that ID labelling is unnecessary.

### 11.5. PocketAnalyzer$^{\mathrm{PCA}}$

**PocketAnalyzer$^{\mathrm{PCA}}$** is another method to detect and track dynamic cavities along MD trajectories [CPG*11]. It was developed aiming at the characterization of protein–ligand interactions. It implements a variant of the grid-based cavity detection algorithm LIGSITE (see Section 4.3) to identify cavities—as connected aggregates of grid nodes—in each snapshot (or time step), as well as principal component analysis (PCA) to track the shape evolution of cavities.

More specifically, this method applies PCA directly on the grid nodes of each cavity, with the purpose of unveiling the dominant deformation of the cavity over time. Note that the PCA might also be applied to the atomic centres, in which case we would have to guarantee that PCA would be applied to *all* the atoms bordering the cavity; otherwise, some cavities may not be identified. As an example, only using C$_\alpha$ atoms in the computation of MD trajectories makes some cavities undetected.

This method involves two major steps: PCA and clustering. The PCA step provides the following: (i) principal component (PC) eigenvectors, which unveil the dominant deformation modes of the cavity, and (ii) PC projections (called 'scores') that characterize the cavity conformational distribution (CCD). In the second step, clustering of the CCD results of a given protein that deforms over time allows us to reduce the entire set of its structures to a small subset that holds noticeably different binding pocket conformations.

### 11.6. Provar

**Provar** (Probability of variation) was developed by Ashford *et al.* [AMA*12]. As other cavity tracking methods, the leading idea of Provar is gaining insight into binding cavities through the inspection of time-varying conformations of any protein. As suggested above, the argument is that the cavity prediction based on a single static structure may fail to detect putative binding sites, in particular, transient cavities that change in their shape and size over time; persistent cavities are less prone to be left out.

Provar admits, as input, sequences of conformational variants (or conformations) of a single protein produced from a number of sources, namely: molecular dynamics (MD), essential dynamics (ED), normal mode analysis (NMA) or constraint-based methods (CBM) (e.g. CONCOORD and tCONCOORD), solution-NMR conformational ensembles, multiple protein structures solved in distinct crystal forms, or with distinct ligands or experimental conditions. The detection of cavities for each conformation of the same protein can be performed using PASS, LIGSITE or Fpocket. Provar automatically identifies and scores cavity-lining atoms and residues, that is, those atoms and residues bounding each cavity, after which it undertakes the probabilistic analysis of changes of cavities on protein surface in terms of shape and size.

### 11.7. PPIAnalyzer

**PPIAnalyzer** is due to Metz *et al.* [MPK*12]. It is targeted at protein–protein interactions (PPIs). Metz and co-authors noted two challenges to bear in mind in dealing with PPIs. First, in contrast to protein–ligand bindings, protein–protein interfaces—enabling the interaction between proteins—are rather flat, that is, they lack a noticeable binding cavity. Second, taking into account the commonly large size of protein-protein interfaces—which may vary in the range 1200 to 4660 Å$^2$ approximately—protein–protein binding tends to be broader in terms of occupied area of the interface.

In fact, as noted by Metz *et al.* [MPK*12], the experimental evidence suggests that residues participating in protein–protein interactions tend to be spatially clustered in protein-protein interfaces, resulting in the so-called 'hot spot' regions. Furthermore, it was also observed an opening of transient cavities in protein–protein interfaces. Therefore, one concludes that to determine protein–protein interfaces, one has to look for hot spots and transient cavities.

This method works as follows. First, one uses and compares molecular dynamics (MD) and constrained geometric (FRODA) simulations to generate structural ensembles. Second, PPIAnalyzer proceeds to the analysis of structural properties of protein–protein interfaces in such ensembles, with the goal of identifying transient cavities exclusively using geometric criteria. Third, one identifies hot spots and ranks protein–protein interface modulators (PPIMs) by applying the molecular mechanics Poisson–Boltzmann (generalized Born) surface area (MM-PB(GB)SA) approach.

### 11.8. CAVER 3.0

**CAVER 3.0** was proposed by Chovancova *et al.* [CPB*12] as a time-varying follow-up of the previous CAVER (see Section 7.2) method to predict tunnels and channels, which play an important role as transport pathways of water solvent, ions and small molecules in many proteins. While CAVER applies to static macromolecular structures, CAVER 3.0 was designed to cope with transient tunnels and channels over time. Moreover, CAVER 3.0 puts forward new algorithms capable of identifying and clustering such transport pathways. CAVER 3.0 was also incorporated as part of the CAVER Analyst 1.0 graphic tool [KSS*14].

The method of CAVER 3.0 consists of three steps: (i) identification of pathways for each MD simulation's snapshot; (ii) clustering of such pathways across all snapshots; and (iii) ranking of pathway clusters. Note that the steps concerning the identification and clustering of pathways are independent of each other, so that their calculation within distinct snapshots can be performed in parallel.

The identification of pathways (first step) within each snapshot starts with the construction of a pseudo-Voronoi diagram of a given protein. Let $r$ the vdW radius of the smallest atom of the protein. Every single atom with a radius greater than $r$ is approximated by a user-specified number of balls of radius $r$, that is, each large atom is approximated by a set of smallest pseudo-atoms. The idea here is to approximate the weighted Voronoi diagram (also known as Apollonius diagram) of a set of atoms through the ordinary Voronoi diagram of an augmented set of atomic centres. Then, as usual, pathways are identified as graph paths made up of Voronoi vertices and edges.

After detecting pathways within each snapshot, these are clustered (second step) regarding their geometric similarities (e.g. geometric distance). To identify the same cavity in disparate snapshots, the authors have proposed a modification of the average-link hierarchical clustering algorithm [LPFL08] by computing on-the-fly the distance between pathways.

Each cluster is then ranked by priority $p = k/n$, where $k$ is the sum of throughputs of all pathways in such a cluster, and $n$ is the total number of snapshots of the MD simulation. This means that both the number of pathways and their throughputs in a cluster contribute to its ranking. It is clear that, if the cluster contains two or pathways in the same snapshot, only the highest-throughput pathway is taken into account.

### 11.9. Lindow *et al.*'s method (LBBH)

This method was proposed by Lindow *et al.* [LBBH13], and it is here also named LBBH method after its authors. It extends the LBH method [LBH11] designed for a single conformation of a molecule and its cavities to dynamic cavities in molecular dynamics trajectories. This method consists of two steps: pre-processing step and interactive step.

The pre-processing step consists in computing the Apollonius diagram (i.e. Voronoi diagram of wdW spheres), which represents the skeletal structure of cavities, for each molecular simulation's snapshot. In other words, this step aims at computing the static molecular paths for each snapshot in a separate manner, as in an authors' previous work [LBH11].

In mathematical terms, a static molecular path is nothing more than a subset of the skeleton of the distance function determined by the vdW spheres; specifically, it consists of maxima and index-2 saddles and maxima of such a distance function, together with their interconnecting separatrices.

The interactive step allows the user to identify, choose and visualize the dynamic cavities and their changes over time, that is, users observe how dynamic molecular paths (cavities) evolve over time.

### 11.10. TRAPP

Kokh *et al.* [KRH*13] introduced **TRAPP** (TRAnsient Pockets in Proteins). TRAPP works on ensembles of protein conformations obtained from simulations or from experimental structures, from which it is capable of identifying the stable and transient regions of cavities in an automated manner.

TRAPP uses a grid-based method for cavity detection that determines the shape and physical properties of every single binding site. The detection of transient cavity regions is performed using two distinct techniques. The first takes advantage of PCA to correlate cavity variations, muck like in PocketAnalyzer$^{PCA}$. The second calculates the averaged deviation of the cavity shape in a molecular trajectory (i.e. across an ensemble of structures or conformations of a given protein) relative to a reference (crystal) structure; such a deviation was named the averaged relative deviation from a reference structure (ARDR).

This method distinguishes itself from others in that it only considers binding sites for which there are already known ligands. To validate the ability of TRAPP in detecting stable and transient cavities, their authors used a set of holo-proteins and already known protein motion trajectories, more specifically, trajectories generated by standard MD simulation over 10 ns, which are available from the MoDEL database [MDH*10].

### 11.11. trj_cavity

**trj_cavity** was developed by Paramo *et al.* [PEG*14] within the GROMACS (www.gromacs.org) framework for quickly identifying and characterizing cavities detected along MD trajectories. The method is based on a new grid-based approach to detect cavities on each frame (or snapshot) by efficiently searching neighbour voxels; in fact, its time complexity is linear with respect to the number of voxels. More specifically, trj_cavity searches for each voxel belonging to a cavity along each of six directions defined by the positive and negative $x$, $y$ and $z$ axes. The method can detect cavities along the trajectory by assuming that the next frame has the same cavity on the current frame, and they overlap somehow partially in space.

The performance of trj_cavity is heavily dependent on some parameters, which include the voxel size and the number of cavities that the user aims to detect, for example, cavities with a predefined value volume. Furthermore, although the grid-based method underlying trj_cavity does not require the user to choose a cavity of interest, he/she has to do that in the context of cavity's trajectory analysis.

### 11.12. Epock

Laurent *et al.* [LCC*15] developed **Epock**, a software package used for tracking a protein cavity volume throughout MD trajectories, which is intended not for cavity identification, but instead to follow *a priori* determined cavities over time. It extends the method proposed in the POVME program [DdOM11], and takes as input an MD trajectory and a topology of the cavity under analysis, defined by a maximum encompassing region that provides spatial bounds for each cavity using a combination of simple three-dimensional objects (spheres, cylinders and cuboids). For each cavity, Epock then calculates its free space, composed of the set of all grid points where the distance to the protein exceeds a user-defined probe radius. Finally, it outputs cavity volume variations, residue contributions and the computed trajectory of this free space over time, which can be visualized by VMD [HDS96].

### 11.13. Desdouits *et al.*'s method

Similar to PocketAnalyzer$^{PCA}$, **Desdouits** *et al.***'s** method [DNB15] also uses the PCA technique to track the dynamic geometry of protein cavities over time. Their method builds upon gHECOM (grid-based HECOMi finder) described in Section 5.5. Recall that gHECOM is a grid-and-sphere-based method that uses probe spheres of minimum and maximum sizes to better delineate the cavity bounds (i.e. mouth openings), reducing this way the occurrence of cavity false positives and negatives. In fact, small cavities (with volume less than 12.0 Å$^3$) are thrown away. By definition, a cavity is a concavity accessible to the solvent probe (i.e. the water molecule of 1.4 Å radius).

Unlike PocketAnalyzer$^{PCA}$, cavity trajectories are indirectly determined by identifying the cavities on each conformation of atomic trajectories. As argued by Desdouits *et al.* [DNB15], determining cavity trajectories using the absolute 3D positions of their grid nodes is sensitive to alignment of the protein in space. They also confirmed the dynamic nature of the cavity evolution over time, as advanced by Eyrisch and Helms [EH07], with cavities—no matter their size—appearing and disappearing at several locations of the protein.

### 11.14. Time-varying methods: A discussion

With the advent of GPU computing in the last decade, it became feasible to simulate MD trajectories of atoms and molecules (and, implicitly, their cavities) within a reasonable time window. This, combined with datasets of trajectories (e.g. MoDEL database [MDH*10]), has ushered in time-varying methods to identify dynamic or transient cavities. As a consequence, we now have tools to uncover unknown cavities and putative binding sites that result from protein–ligand and protein–protein interactions.

As shown in Table 8, most time-varying methods are based on existing static methods; for example, EPOS$^{BP}$ is based on PASS, which is a sphere-based method. But, note that most of them belong to the category of grid-based methods. However, as argued above, Voronoi diagram-based methods, in particular, Apollonius diagram-based methods, are more accurate than grid-based methods.

On the other hand, trajectories of atoms and molecules computed by molecular dynamics (MD) simulations are adequate for short time-scales, and are dominant in the current state-of-the-art of time-varying methods, as shown in Table 8. Only a couple of these methods (i.e. TexMol and Provar) take advantage of NMA simulations, which are more suited for large time-scales. Both MD and NMA simulations are computationally rather expensive; in particular, an MD simulation of a few nanoseconds for a large a protein takes a very long time, because solving Newton's equations is computationally expensive. Hence, the increasing use of high-performance computation resources (e.g. GPUs) to speed up these simulations. Recall that the computation an MD simulation is akin to $N$-body simulation, that is, it involves pairwise interactions of $N$ particles.

## 12. Limitations, Challenges, and Future Directions

A more comprehensive characterization of what is a protein cavity in structural and functional terms would allow for a refinement of the current detection algorithms. As noted in [OFH*14], the initial challenge for any cavity detection method lies in the mathematical specification of the cavity. This is noticeable when it comes to identifying the boundary atoms that make up a cavity, that is, its 'walls, floor, and ceiling (mouth)'.

The current cavity specifications of the various methods described above lead to some trade-offs. *Sphere-based algorithms* have difficulties in dealing with cavities of different sizes simultaneously, because that requires using probe spheres of empirically distinct sizes for each protein, resulting in difficulties in detecting and delineating cavity mouth openings on proteins. In fact, a relatively small probe can function as a stopgap for invaginations with small mouth openings, but shallow cavities (i.e. grooves) require large probes as ceiling bounds. Therefore, they are probe-radius sensitive. Besides, as Benkaidali *et al.* [BAM*14] noted, the spherical model of probes is often inadequate for the detection of shallow cavities (e.g. depressions or grooves), and cavities of cylindrical shapes (e.g. tunnels or channels). In the same vein, *grid-based algorithms* suffer from ambiguity issues related with grid-spacing, protein-orientation sensitivity and delineation of mouth openings, in particular, the cavity entry/exit that separates the empty space of a given cavity from the remaining empty space [NH06]. *Tessellation-based algorithms* also suffer from MOA, and this explains why some of them use the convex hull as outer boundary. Besides, they (at least the former methods of this category) may fail in detecting some cavities (i.e. false negatives), and detect cavities that are false positives quite easily.

Summing up, the deficiencies of these methods explain the need for refined techniques to detect cavities on protein surfaces. This is the case of mixed geometric methods, and surface methods, as well as the consensus methods. Mixed methods are an attempt of aggregating the strengths of two distinct techniques and, at the same time, mitigating their weaknesses. Consensus methods act on the results of two or more methods, without re-engineering any of them. Furthermore, surface-based methods seemingly are an alternative to the other more conventional categories of methods. Besides, we need for further developments in *hierarchical segmentation* techniques for protein structures and surfaces.

Thus, we envisage the following challenges in the near future:

- *Sphere-based methods*. To study and apply geometric segmentation techniques, as of computer graphics, to a set of balls featuring atoms, and its complementary space in 3D space. Can we segment such a set of balls in a way to get a meaningful segmentation in terms of cavities as putative binding sites?
- *Grid-based methods*. In the line of a few methods found in the literature, like those due to Delaney [Del92], Masuya and Doi [MD95] and [Kaw10], grid-based methods would benefit in large from a proper generalization of image segmentation techniques from 2D to 3D, as of in image processing and analysis field.
- *Surface-based methods*. We will need more advanced formulations for protein surfaces to take advantage of geometric properties and shape descriptors of smooth surfaces in differential geometry (e.g. gradient, normal vector and so forth) to segment protein surfaces into cavities and protrusions. Surface-based methods do not use space decompositions, grids and probe spheres, and are potentially faster in their computations to find protein cavities. Besides, and following Lindow *et al.* [LBH14], we likely need to explore and design (or reformulate) new algorithms based on new types of molecular surfaces, as it is the case of the ligand-excluded surface (LES), which can be seen as a generalization of SES. Note that there is not an analytical formulation for LES yet.
- *Tessellation-based methods*. In part, the communities of computer graphics and geometric computing (i.e. computational geometry and computer aided geometric design) already brought part of the bulk of knowledge related to combinatorial geometry and numerical geometry into the field of molecular graphics and modelling. Therefore, one expects that this research in cavity detection methods will continue in the future.

Obviously, all these methods are essentially static, that is, they operate on only one protein conformation. If we wish to mimic the dynamic behaviour of proteins and their interactions with other molecules, we need to develop new models, techniques and tools capable of coping with geometry that varies over time. That is, we need to develop an adequate theory of the dynamic geometry of molecules (e.g. via contour trees) based on tracing of singularities of the vector field generated by the electron density map associated with a molecule. In this respect, the search for more robust and efficient time-varying geometry methods will be central to future breakthroughs in the field of molecular graphics and modelling.

## 13. Conclusions

We have reviewed the literature concerning geometric methods to detect cavities on proteins. We have identified four main families of cavity detection algorithms: sphere-based, grid-based, surface-based and tessellation-based. Additionally, we were able to identify three additional families of mixed methods, namely those based on grid-and-sphere (Section 5), on grid-and-surface (Section 7) and also on consensus (Section 9). All these techniques were designed for analysing a single protein conformation so that they identify static cavities.

A current trend in this field is to develop dynamic models for protein surfaces that deform over time and mimic their biophysical behaviour. To this end, we need surface models for proteins that take into account protein–ligand and protein–protein interactions; for example, we need a model that is further capable of representing induced conformations on molecular binding and thereby captures topological transformations of, for example, a void into a pocket, and vice versa. In many ways, this is a challenge for those involved in physically based geometry research, which directly involves Computer Graphics and Geometry Processing. The promise borne by these new approaches is both a more faithful and farther reaching model of protein–ligand interactions that could yield significant gains in molecular simulation and modelling.

## Acknowledgements

This research has been partially supported by the Portuguese Research Council (Fundação para a Ciência e Tecnologia), under the doctoral Grant SFRH/BD/99813/2014, postdoctoral grants SFRH/BPD/97449/2013 and SFRH/BPD/111836/2015, and also by FCT Projects UTAP-EXPL/QEQ-COM/0019/2014, IT-MEDEX PTDC/EEI-SII/6038/2014, UID/EEA/50008/2013 and UID/CEC/50021/2013. We gratefully acknowledge the support of NVIDIA Corporation. At last, we would like to thank the anonymous reviewers for their suggestions that contributed to improving this paper.

## References

[ABSC12] Al-Bluwi I., Siméon T., Cortés J.: Motion planning algorithms for molecular simulations: A survey. *Computer Science Review 6*, 4 (2012), 125–143.

[AGBT01] Armon A., Graur D., Ben-Tal N.: ConSurf: An algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information. *Journal of Molecular Biology 307*, 1 (2001), 447–463.

[AJL*07] Alberts B., Johnson A., Lewis J., Raff M., Roberts K., Walter P.: *Molecular Biology of the Cell*. Garland Science, New York, USA, 2007.

[AMA*12] Ashford P., Moss D. S., Alex A., Yeap S. K., Povia A., Nobeli I., Williams M. A.: Visualisation of variable binding pockets on protein surfaces by probabilistic analysis of related structure sets. *BMC Bioinformatics 13*, 1 (2012), 1–16.

[BAM*14] Benkaidali L., André F., Maouche B., Siregar P., Benyettou M., Maurel F., Petitjean M.: Computing cavities, channels, pores and pockets in proteins from non spherical ligands models. *Bioinformatics 30*, 6 (2014), 792–800.

[BCG*13] Brezovsky J., Chovancova E., Gora A., Pavelka A., Biedermannova L., Damborsky J.: Software tools for identification, visualization and analysis of protein tunnels and channels. *Biotechnology Advances 31*, 1 (2013), 38–49.

[BD05] Boissonnat J., Delage C.: Convex hull and Voronoi diagram of additively weighted points. In *Proceedings of the 13th Annual European Conference on Algorithms*, Brodal G., Leonardi S., (Eds.), vol. 3669 of *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, 2005, pp. 367–378.

[BDH96] Barber C. B., Dobkin D. P., Huhdanpaa H.: The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software 22*, 4 (1996), 469–483.

[BDST04] Bajaj C., Djeu P., Siddavanahalli V., Thane A.: Texmol: Interactive visual exploration of large flexible multi-component molecular complexes. In *Proceedings of the IEEE Conference on Visualization* (IEEEVis'04), Austin, Texas, USA, October 10-15 (2004), IEEE Press, pp. 243–250.

[BGG09] Bajaj C., Gillette A., Goswami S.: Topology based selection and curation of level sets. In *Topology-Based Methods in Visualization II*, Hege H.-C., Polthier K., Scheuermann G., (Eds.), Mathematics and Visualization. Springer-Verlag, Berlin Heidelberg, 2009, pp. 45–58.

[BGG*10] Bajaj C., Gillette A., Goswami S., Kwon B. J., Rivera J.: Complementary space for enhanced uncertainty and dynamics visualization. In *Topological Methods in Data Analysis and Visualization*, Pascucci V., Tricoche X., Hagen H., Tierny J., (Eds.), Mathematics and Visualization. Springer-Verlag, Berlin Heidelberg, 2010, pp. 217–228.

[BHH*09] Buša J., Hayryan S., Hu C.-K., Skřivánek J., Wu M.-C.: Enveloping triangulation method for detecting internal cavities in proteins and algorithm for computing their surface areas and volumes. *Journal of Computational Chemistry 30*, 3 (2009), 346–357.

[BHH*10] Buša J., Hayryan S., Hu C.-K., Skřivánek J., Wu M.-C.: CAVE: A package for detection and quantitative analysis of internal cavities in a system of overlapping balls: Application to proteins. *Computer Physics Communications 181*, 12 (2010), 2116–2125.

[BHHW15] Buša J., Hayryan S., Hu C.-K., Wu M.-C.: CAVE-CL: An OpenCL version of the package for detection and quantitative analysis of internal cavities in a system of overlapping balls: Application to proteins. *Computer Physics Communications 190* (2015), 224–227.

[Bli82] Blinn J. F.: A generalization of algebraic surface drawing. *ACM Transactions on Graphics 1*, 3 (July 1982), 235–256.

[Blu67] Blum H.: A transformation for extracting new descriptors of shape. In *Proceedings of the Symposium on Models for the Perception of Speech and Visual Form*, Boston, Massachusetts, November 11–14, 1964 (1967), Wathen-Dunn W., (Ed.), MIT Press, pp. 362–380.

[BNL03] Binkowski T. A., Naghibzadeh S., Liang J.: CASTp: computed atlas of surface topography of proteins. *Nucleic Acids Research 31*, 13 (2003), 3352–3355.

[Bor11] Borland D.: Ambient occlusion opacity mapping for visualization of internal molecular structure. *Journal of WSCG 19*, 1–3 (2011), 17–24.

[BS00] Brady G. P., Stouten P. F. W.: Fast prediction and visualization of protein binding pockets with PASS. *Journal of Computer-Aided Molecular Design 14*, 4 (May 2000), 383–401.

[CCL03] Cazals F., Chazal F., Lewiner T.: Molecular shape analysis based upon the Morse-Smale complex and the Connolly function. In *Proceedings of the 19th Annual Symposium on Computational Geometry* (SCG'03), San Diego, California, USA, June 8-10 (2003), ACM Press, pp. 351–360.

[Con83] Connolly M.: Analytical molecular surface calculation. *Journal of Applied Crystallography 16*, 5 (October 1983), 548–558.

[Con86] Connolly M.: Measurement of protein surface shape by solid angles. *Journal of Molecular Graphics 4*, 1 (1986), 3–6.

[CPB*12] Chovancova E., Pavelka A., Benes P., Strnad O., Brezovsky J., Kozlikova B., Gora A., Sustr V., Klvana M., Medek P., Biedermannova L., Sochor J., Damborsky J.: CAVER 3.0: A tool for the analysis of transport pathways in dynamic protein structures. *PLoS Computational Biology 8*, 10 (2012), e1002708:1–12.

[CPG*11] Craig I. R., Pfleger C., Gohlke H., Essex J. W., Spiegel K.: Pocket-space maps to identify novel binding-site conformations in proteins. *Journal of Chemical Information and Modeling 51*, 10 (2011), 2666–2679.

[CS06] Coleman R. G., Sharp K. A.: Travel depth, a new shape descriptor for macromolecules: Application to ligand binding. *Journal of Molecular Biology 362*, 3 (2006), 441–458.

[CS09] Coleman R. G., Sharp K. A.: Finding and characterizing tunnels in macromolecules with application to ion channels and pores. *Biophysical Journal 96*, 2 (2009), 632–645.

[CS10] Coleman R., Sharp K.: Protein pockets: Inventory, shape, and comparison. *Journal of Chemical Information and Modeling 50*, 4 (2010), 589–603.

[CVT*11] Chavent M., Vanel A., Tek A., Levy B., Robert S., Raffin B., Baaden M.: GPU-accelerated atom and dynamic bond visualization using hyperballs: A unified algorithm for balls, sticks, and hyperboloids. *Journal of Computational Chemistry 32*, 13 (2011), 2924–2935.

[Czi15] Czirják G.: PrinCCes: Continuity-based geometric decomposition and systematic visualization of the void repertoire of proteins. *Journal of Molecular Graphics and Modelling 62* (November 2015), 118–127.

[DBG10] Dias S., Bora K., Gomes A.: CUDA-based triangulations of convolution molecular surfaces. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing* (HPDC'10), Chicago, Illinois, USA, June 21-25 (2010), ACM Press, pp. 531–540.

[DCD*14] D'Agostino D., Clematis A., Decherchi S., Rocchia W., Milanesi L., Merelli I.: CUDA accelerated molecular surface generation. *Concurrency and Computation: Practice and Experience 26*, 10 (2014), 1819–1831.

[DCTS93] Del Carpio C., Takahashi Y., Sasaki S.: A new approach to the automatic identification of candidates for ligand receptor sites in proteins: (I). Search for pocket regions. *Journal of Molecular Graphics 11*, 1 (1993), 23–29.

[DdOM11] Durrant J., de Oliveira C., McCammon J. A.: POVME: an algorithm for measuring binding-pocket volumes. *Journal of Molecular Graphics and Modelling 29*, 5 (2011), 773–776.

[Del92] Delaney J. S.: Finding and filling protein cavities using cellular logic operations. *Journal of Molecular Graphics 10*, 3 (1992), 174–177.

[DG13] Dias S., Gomes A.: Triangulating molecular surfaces on multiple GPUs. In *Proceedings of the 20th European MPI Users' Group Meeting* (EuroMPI'13), Madrid, Spain, September 15–18 (2013), ACM Press, pp. 181–186.

[DG15] Dias S., Gomes A. J.: Triangulating Gaussian-like surfaces of molecules with millions of atoms. In *Computational Electrostatics for Biological Applications*, Rocchia W., Spagnuolo M., (Eds.). Springer International Publishing, Cham, Switzerland, 2015, pp. 177–198.

[DG17] Dias S., Gomes A.: GPU-based detection of protein cavities using Gaussian implicit surfaces *(submitted for publication)* (2017).

[DK91] Doi A., Koide A.: An efficient method of triangulating equivalued surfaces by using tetrahedral cells. *IEICE Transactions on Information Systems E74-D, 1* (1991), 214–224.

[DLW08] Dessailly B. H., Lensink M. F., Wodak S. J.: LigASite: A database of biologically relevant binding sites in proteins with known apo-structures. *Nucleic Acids Research 36* (2008), D667–673.

[DNB15] Desdouits N., Nilges M., Blondel A.: Principal component analysis reveals correlation of cavities evolution and functional motions in proteins. *Journal of Molecular Graphics and Modelling 55* (February 2015), 13–24.

[DNJG17] Dias S. E., Nguyen Q. T., Jorge J. A., Gomes A. J.: Multi-GPU-based detection of protein cavities using critical points. *Future Generation Computer Systems 67* (February 2017), 430–440.

[Duk13] Dukka B.: Structure-based methods for computational protein functional site prediction. *Computational and Structural Biotechnology Journal 8*, 11 (2013), 1–8.

[Ede95] Edelsbrunner H.: The union of balls and its dual shape. *Discrete & Computational Geometry 13*, 3 (1995), 415–440.

[Ede98] Edelsbrunner H.: On the definition and the construction of pockets in macromolecules. *Discrete Applied Mathematics 88*, 1–3 (November 1998), 83–102.

[EFFL95] Edelsbrunner H., Facello M., Fu P., Liang J.: Measuring proteins and voids in proteins. In *Proceedings of the 28th Hawaii International Conference on System Sciences* (HICSS'95), Maui, Hawaii, January 3-6 (1995), IEEE Press, pp. 256–264.

[EH07] Eyrisch S., Helms V.: Transient pockets on protein surfaces involved in protein-protein interaction. *Journal of Medicinal Chemistry 50*, 15 (2007), 3457–3464.

[EK06] Emiris I. Z., Karavelas M. I.: The predicates of the apollonius diagram: Algorithmic analysis and implementation. *Computational Geometry 33*, 1 (2006), 18–57.

[EKMB98] Exner T., Keil M., Moeckel G., Brickmann J.: Identification of substrate channels and protein cavities. *Journal of Molecular Modeling 4*, 10 (1998), 340–343.

[EKS83] Edelsbrunner H., Kirkpatrick D. G., Seidel R.: On the shape of a set of points in the plane. *IEEE Transactions on Information Theory 29*, 4 (1983), 551–559.

[EKSX96] Ester M., Kriegel H., Sander J., Xu X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (KDD'96), Portland, Oregon, USA, August 2–4 (1996), AAAI Press, pp. 226–231.

[EM94] Edelsbrunner H., Mucke E. P.: Three-dimensional alpha shapes. *ACM Transactions on Graphics 13* (1994), 43–72.

[GAGM11] Giard J., Alface P. R., Gala J.-L., Macq B.: Fast surface-based travel depth estimation algorithm for macromolecule surface shape description. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 8*, 1 (2011), 59–68.

[Goo85] Goodford P. J.: A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *Journal of Medicinal Chemistry 28*, 7 (1985), 849–857.

[GP95] Grant J. A., Pickup B. T.: A Gaussian description of molecular shape. *Journal of Physical Chemistry 99*, 11 (1995), 3503–3510.

[GS11] Ghersi D., Sanchez R.: Beyond structural genomics: Computational approaches for the identification of ligand binding sites in protein structures. *Journal of Structural and Functional Genomics 12*, 2 (2011), 109–117.

[GS13] Gao M., Skolnick J.: A comprehensive survey of small-molecule binding pockets in proteins. *PLOS Computational Biology 9*, 10 (2013), 1–12.

[GVJ*09] Gomes A., Voiculescu I., Jorge J., Wyvill B., Galbraith C.: *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*. Springer-Verlag, London, 2009.

[GW07] Gonzalez R. C., Woods R. E.: *Digital Image Processing*. Pearson Prentice Hall, Upper Saddle River, New Jersey, USA, 2007.

[Hat02] Hatcher A.: *Algebraic Topology*. Cambridge University Press, Cambridge, United Kingdom, 2002.

[HDS96] Humphrey W., Dalke A., Schulten K.: VMD: Visual molecular dynamics. *Journal of Molecular Graphics 14*, 1 (1996), 33–38.

[HG08] Ho B., Gruswitz F.: HOLLOW: Generating accurate representations of channel and interior surfaces in molecular structures. *BMC Structural Biology 8* (2008), 49:1–49:6.

[HGVV16] Hermosilla P., Guallar V., Vinacua A., Vázquez P.: High quality illustrative effects for molecular rendering. *Computers & Graphics 54* (2016), 113–120.

[HLP10] Hawick K., Leist A., Playne D.: Parallel graph component labelling with GPUs and CUDA. *Parallel Computing 36*, 12 (2010), 655–678.

[HM90] Ho C., Marshall G.: Cavity search: An algorithm for the isolation and display of cavity-like binding regions. *Journal of Computer-Aided Molecular Design 4*, 4 (1990), 337–354.

[HOG08] Harris R., Olson A. J., Goodsell D. S.: Automated prediction of ligand-binding sites in proteins. *Proteins: Structure, Function, and Bioinformatics 70*, 4 (2008), 1506–1517.

[HRB97] Hendlich M., Rippmann F., Barnickel G.: LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling 15*, 6 (1997), 359–363.

[HSAH*09] Henrich S., Salo-Ahen O. M. H., Huang B., Rippmann F. F., Cruciani G., Wade R. C.: Computational approaches to identifying and characterizing protein binding sites for ligand design. *Journal of Molecular Recognition 23*, 2 (2009), 209–219.

[Hua09] Huang B.: Metapocket: A meta-approach to improve protein ligand binding site prediction. *OMICS 13*, 4 (2009), 325–330.

[JKSS96] Jenkins A., Kratochvil P., Stepto R., Suter U.: Glossary of basic terms in polymer science (IUPAC Recommendations 1996). *Pure and Applied Chemistry 68*, 12 (1996), 2287–2311.

[Kaw10] Kawabata T.: Detection of multiscale pockets on protein surfaces using mathematical morphology. *Proteins: Structure, Function, and Bioinformatics 78*, 5 (2010), 1195–1211.

[KBE09] Krone M., Bidmon K., Ertl T.: Interactive visualization of molecular surface dynamics. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1391–1398.

[KBO*82] Kuntz I. D., Blaney J. M., Oatley S. J., Langridge R., Ferrin T. E.: A geometric approach to macromolecule-ligand interactions. *Journal of Molecular Biology 161*, 2 (October 1982), 269–288.

[KC08] Kalidas Y., Chandra N.: PocketDepth: A new depth based algorithm for identification of ligand binding sites in proteins. *Journal of Structural Biology 161*, 1 (2008), 31–42.

[KCC*08] Kim D., Cho C., Cho Y., Ryu J., Bhak J., Kim D.: Pocket extraction on proteins via the Voronoi diagram of spheres. *Journal of Molecular Graphics and Modelling 26*, 7 (April 2008), 1104–1112.

[KCK04] Kim D.-S., Cho Y., Kim D.: Edge-tracing algorithm for Euclidean Voronoi diagram of 3D Spheres. In *Proceedings of the 16th Canadian Conference on Computational Geometry*(CCCG'04), Montréal, Quebec, Canada, August 9–11 (2004), pp. 176–179.

[KCKC06] Kim D.-S., Cho C.-H., Kim D., Cho Y.: Recognition of docking sites on a protein using $\beta$-shape based on Voronoi diagram of atoms. *Computer-Aided Design 38*, 5 (2006), 431–443.

[KCL*14] Kim J.-K., Cho Y., Laskowski R. A., Ryu S. E., Sugihara K., Kim D.-S.: BetaVoid: Molecular voids via beta-complexes and Voronoi diagrams. *Proteins: Structure, Function, and Bioinformatics 82*, 9 (2014), 1829–1849.

[KFR*11] Krone M., Falk M., Rehm S., Pleiss J., Ertl T.: Interactive exploration of protein cavities. *Computer Graphics Forum 30*, 3 (2011), 673–682.

[KG07] Kawabata T., Go N.: Detection of pockets on protein surfaces using small and large probe spheres to find putative ligand binding sites. *Proteins: Structure, Function, and Bioinformatics 68*, 2 (2007), 516–529.

[KJ94] Kleywegt G., Jones T.: Detection, delineation, measurement and display of cavities in macromolecular structures. *Acta Crystallographica 50, Part 2* (1994), 178–185.

[KJV83] Kirkpatrick S., Jr. C. D. G., Vecchi M. P.: Optimization by simulated annealing. *Science 220*, 4598 (1983), 671–680.

[KKC*11] Kim B., Kim K.-J., Choi J.-H., Baek N., Seong J.-K., Choi Y.-J.: Finding surface atoms of a protein molecule on a GPU. In *SIGGRAPH Asia 2011 Posters* (2011), ACM Press, New York, NY, USA, pp. 32:1–32:1.

[KKL*16] Krone M., Kozlikova B., Lindow N., Baaden M., Baum D., Parulek J., Hege H.-C., Viola I.: Visual analysis of biomolecular cavities: State of the art. *Computer Graphics Forum 35*, 3 (2016), 527–551.

[KKS01a] Kim D.-S., Kim D., Sugihara K.: Voronoi diagram of a circle set from Voronoi diagram of a point set: I. Topology. *Computer Aided Geometric Design 18*, 6 (2001), 541–562.

[KKS01b] Kim D.-S., Kim D., Sugihara K.: Voronoi diagram of a circle set from Voronoi diagram of a point set: II. Geometry. *Computer Aided Geometric Design 18*, 6 (2001), 563–585.

[KLKK16] Kim B., Lee J. E., Kim Y. J., Kim K.-J.: GPU accelerated finding of channels and tunnels for a protein molecule. *International Journal of Parallel Programming 44*, 1 (2016), 87–108.

[KRH*13] Kokh D. B., Richter S., Henrich S., Czodrowski P., Rippmann F., Wade R. C.: TRAPP: A tool for analysis of transient binding pockets in proteins. *Journal of Chemical Information and Modeling 53*, 5 (2013), 1235–1252.

[KRS*13] Krone M., Reina G., Schulz C., Kulschewski T., Pleiss J., Ertl T.: Interactive extraction and tracking of biomolecular surface features. *Computer Graphics Forum 32*, 3 (2013), 331–340.

[KSK*06] Kim D.-S., Seo J., Kim D., Ryu J., Cho C.-H.: Three-dimensional beta-shapes. *Computer-Aided Design 38*, 11 (2006), 1179–1191.

[KSL*15] Kuenemann M. A., Sperandio O., Labbé C. M., Lagorce D., Miteva M. A., Villoutreix B. O.: In silico design of low molecular weight protein-protein interaction inhibitors: Overall concept and recent advances. *Progress in Biophysics and Molecular Biology 119*, 1 (2015), 20–32.

[KSS*14] Kozlikova B., Sebestova E., Sustr V., Brezovsky J., Strnad O., Daniel L., Bednar D., Pavelka A., Manak M., Bezdeka M., et al.: CAVER Analyst 1.0: Graphic tool for interactive visualization and analysis of tunnels and channels in protein structures. *Bioinformatics 30*, 18 (2014), 2684–2685.

[Kub06] Kubinyi H.: Chemogenomics in drug discovery. In *Chemical Genomics: Small Molecule Probes to Study Cellular Function*, Jaroch S., Weinmann H., (Eds.), vol. 58 of *Ernst Schering Research Foundation Workshop Series*. Springer-Verlag, Berlin Heidelberg, 2006, pp. 1–19.

[Las95] Laskowski R. A.: SURFNET: A program for visualizing molecular surfaces, cavities, and intermolecular interactions. *Journal of Molecular Graphics 13*, 5 (1995), 323–330.

[Lay82] Lay S. R.: *Convex Sets and Their Applications*. Dover Publications, Inc., Mineola, New York, USA, 1982.

[LB92] Levitt D. G., Banaszak L. J.: POCKET: A computer graphic method for identifying and displaying protein cavities and their surrounding amino acids. *Journal of Molecular Graphics 10*, 4 (1992), 229–234.

[LBBH13] Lindow N., Baum D., Bondar A.-N., Hege H.-C.: Exploring cavity dynamics in biomolecular systems. *BMC Bioinformatics 14*, Suppl. 19 (2013), S5:1–12.

[LBH11] Lindow N., Baum D., Hege H.-C.: Voronoi-based extraction and visualization of molecular paths. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2025–2034.

[LBH14] Lindow N., Baum D., Hege H.-C.: Ligand excluded surface: A new type of molecular surface. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2486–2495.

[LCC*15] Laurent B., Chavent M., Cragnolini T., Dahl A. C. E., Pasquali S., Derreumaux P., Sansom M. S., Baaden M.: Epock: Rapid analysis of protein pocket dynamics. *Bioinformatics 31*, 9 (2015), 1478–1480.

[LGST09] Le Guilloux V., Schmidtke P., Tuffery P.: Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics 10*, 1 (December 2009), 1–11.

[LJ05] Laurie A. T. R., Jackson R. M.: Q-SiteFinder: An energy-based method for the prediction of protein-ligand binding sites. *Bioinformatics 21*, 9 (2005), 1908–1916.

[LJ06] Laurie A. T., Jackson R. M.: Methods for the prediction of protein-ligand binding sites for structure-based drug design and virtual ligand screening. *Current Protein and Peptide Science 7*, 5 (October 2006), 395–406.

[LLNW14] Li H., Leung K.-S., Nakane T., Wong M.-H.: iview: an interactive WebGL visualizer for protein-ligand complex. *BMC Bioinformatics 15* (2014), 56:1–56:6.

[LLST96] Laskowski R. A., Luscombe N., Swindells M. B., Thornton J. M.: Protein clefts in molecular recognition and function. *Protein Science 5*, 12 (1996), 2438–2452.

[LPFL08] Loewenstein Y., Portugaly E., Fromer M., Linial M.: Efficient algorithms for accurate hierarchical clustering of huge datasets: Tackling the entire protein space. *Bioinformatics 24*, 13 (2008), i41–i49.

[LR71] Lee B., Richards F.: The interpretation of protein structures: Estimation of static accessibility. *Journal of Molecular Biology 55*, 3 (February 1971), 379–380.

[LS94] Lemieux R. U., Spohr U.: How Emil Fischer was led to the lock and key concept for enzyme specificity. *Advances in Carbohydrate Chemistry and Biochemistry 50* (1994), 1–20.

[LTA*08] Li B., Turuvekere S., Agrawal M., La D., Ramani K., Kihara D.: Characterization of local geometry of protein surfaces with the visibility criterion. *Proteins: Structure, Function, and Bioinformatics 71*, 2 (2008), 670–683.

[LWE98] Liang J., Woodward C., Edelsbrunner H.: Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design. *Protein Science 7*, 9 (1998), 1884–1897.

[LWP*13] Lo Y.-T., Wang H.-W., Pai T.-W., Tzou W.-S., Hsu H.-H., Chang H.-T.: Protein-ligand binding region prediction (PLB-SAVE) based on geometric features and CUDA acceleration. *BMC Bioinformatics 14*, Suppl 4 (2013), S4:1–S4:11.

[Mat75] Matheron G.: *Random Sets and Integral Geometry*. John Wiley & Sons, Inc., New York, USA, 1975.

[MBS07] Medek P., Beneš P., Sochor J.: Computation of tunnels in protein molecules using Delaunay triangulation. *Journal of WSCG 15*, 1–3 (2007), 107–114.

[MBS08] Medek P., Beneš P., Sochor J.: Multicriteria tunnel computation. In *Proceedings of the Tenth IASTED International Conference on Computer Graphics and Imaging*, ACTA Press (2008), pp. 160–164.

[MD95] Masuya M., Doi J.: Detection and geometric modeling of molecular surfaces and cavities using digital mathematical morphology operations. *Journal of Molecular Graphics and Modelling 13*, 6 (1995), 331–336.

[MDH*10] Meyer T., D'Abramo M., Hospital A., Rueda M., Ferrer-Costa C., Pérez A., Carrillo O., Camps J., Fenollosa C., Repchevsky D., Gelpí J. L., Orozco M.: MoDEL (Molecular Dynamics Extended Library): A database of atomistic molecular dynamics trajectories. *Structure 18*, 11 (2010), 1399–1409.

[MKE01] Mitchell J. C., Kerr R., Eyck L. F. T.: Rapid atomic density methods for molecular shape characterization. *Journal of Molecular Graphics and Modelling 19*, 3-4 (2001), 325–330.

[MPK*12] Metz A., Pfleger C., Kopitz H., Pfeiffer-Marek S., Baringhaus K.-H., Gohlke H.: Hot spots and transient pockets: Predicting the determinants of small-molecule binding to a protein-protein interface. *Journal of Chemical Information and Modeling 52*, 1 (2012), 120–133.

[MRR*53] Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H., Teller E.: Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*, 6 (1953), 1087–1092.

[NH06] Nayal M., Honig B.: On the nature of cavities on protein surfaces: Application to the identification of drug-binding sites. *Proteins 63*, 4 (February 2006), 892–906.

[NSH91] Nicholls A., Sharp K., Honig B.: Protein folding and association: Insights from the interfacial and thermodynamic properties of hydrocarbons. *Proteins 11*, 4 (1991), 281–296.

[NWB*06] Natarajan V., Wang Y., Bremer P.-T., Pascucci V., Hamann B.: Segmenting molecular surfaces. *Computer Aided Geometric Design 23*, 6 (2006), 495–509.

[OFH*14] Oliveira S. H., Ferraz F. A., Honorato R. V., Xavier-Neto J., Sobreira T. J., de Oliveira P. S.: KVFinder: Steered identification of protein cavities as a PyMOL plugin. *BMC Bioinformatics 15*, 197 (2014), 1–8.

[OMV11] Olechnovič K., Margelevičius M., Venclovas Č.: Voroprot: An interactive tool for the analysis and visualization of complex geometric features of protein structure. *Bioinformatics 27*, 5 (2011), 723–724.

[PB99] Pettit F. K., Bowie J. U.: Protein surface roughness and small molecular binding sites. *Journal of Molecular Biology 285*, 4 (1999), 1377–1382.

[PBM*02] Pupko T., Bell R. E., Mayrose I., Glaser F., Ben-Tal N.: Rate4site: An algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues. *Bioinformatics 18* (2002), S71–S77.

[PCMT09] Pellegrini-Calace M., Maiwald T., Thornton J. M.: Porewalker: A novel tool for the identification and characterization of channels in transmembrane proteins from their three-dimensional structure. *PLoS Comput Biol 5*, 7 (2009), e1000440.

[PEG*14] Paramo T., East A., Garzón D., Ulmschneider M. B., Bond P. J.: Efficient characterization of protein cavities within molecular simulation trajectories: Trj_cavity. *Journal of Chemical Theory and Computation 10*, 5 (2014), 2151–2164.

[PFF96] Peters K. P., Fauck J., Frommel C.: The automatic search for ligand binding sites in proteins of known three-dimensional structure using only geometric criteria. *Journal of Molecular Biology 256*, 1 (1996), 201–213.

[PGD*10] Phillips M., Georgiev I., Dehof A. K., Nickels S., Marsalek L., Lenhof H.-P., Hildebrandt A., Slusallek P.: Measuring properties of molecular surfaces using ray casting. In *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on* (2010), IEEE, pp. 1–7.

[PGH*04] Pettersen E. F., Goddard T. D., Huang C. C., Couch G. S., Greenblatt D. M., Meng E. C., Ferrin T. E.: Ucsf chimeraâĂŤa visualization system for exploratory research and analysis. *Journal of Computational Chemistry 25*, 13 (2004), 1605–1612.

[PKKO07] Petřek M., Košinová P., Koča J., Otyepka M.: Mole: A Voronoi diagram-based explorer of molecular channels, pores, and tunnels. *Structure 15*, 11 (2007), 1357–1363.

[POB*06] Petřek M., Otyepka M., Banáš P., Košinová P., Koča J., Damborský J.: CAVER: A new tool to explore routes from protein clefts, pockets and cavities. *BMC Bioinformatics 7*, 316 (2006), 1–9.

[PPG10] Pfeffer, P., Fober, T., Hüllermeier, E., Klebe, G.: Garlig: a fully automated tool for subset selection of large fragment spaces via a self-adaptive genetic algorithm. *Journal of Chemical Information and Modeling 50*, 9 (2010), 1644–1659.

[Pro14] Prošková J.: Description of protein secondary structure using dual quaternions. *Journal of Molecular Structure 1076* (November 2014), 89–93.

[PRV13] Parulek J., Ropinski T., Viola I.: Seamless visual abstraction of molecular surfaces. In *Proceedings of The 29th Spring Conference on Computer Graphics* (2013), ACM, pp. 107–114.

[PSA*91] Pedersen T., Sigurskjold B., Andersen K., Kjaer M., Poulsen F., Dobson C., Redfield C.: A nuclear magnetic resonance study of the hydrogen-exchange behaviour of lysozyme in crystals and solution. *Journal of Molecular Biology 218*, 2 (1991), 413–426.

[PSM*10] Pérot S., Sperandio O., Miteva M. A., Camproux A.-C., Villoutreix B. O., et al.: Druggable pockets and binding site centric chemical space: A paradigm shift in drug discovery. *Drug Discovery Today 15*, 15-16 (2010), 656–667.

[PTRV12] Parulek J., Turkay C., Reuter N., Viola I.: Implicit surfaces for interactive graph based cavity analysis of molecular simulations. In *Proceedings of the 2012 IEEE Symposium on Biological Data Visualization* (BioVis'2012) (October 2012), IEEE Press, pp. 115–122.

[PTRV13] Parulek J., Turkay C., Reuter N., Viola I.: Visual cavity analysis in molecular simulations. *BMC Bioinformatics 14*, Suppl 19 (2013), S4:1–S4:15.

[Ric77] Richards F.: Areas, volumes, packing, and protein structure. *Annual Review of Biophysics and Bioengineering 6*, 3 (February 1977), 151–176.

[RK11] Raunest M., Kandt C.: dxtuber: Detecting protein cavities, tunnels and clefts based on protein and solvent dynamics. *Journal of Molecular Graphics and Modelling 29*, 7 (2011), 895–905.

[RTC*13] Ribeiro J. V., Tamames J. A., Cerqueira N. M., Fernandes P. A., Ramos M. J.: Volarea: A bioinformatics tool to calculate the surface area and the volume of molecular systems. *Chemical Biology and Drug Design 82*, 6 (2013), 743–755.

[SB06] Sohn B., Bajaj C.: Time-varying contour topology. *IEEE Transactions on Visualization and Computer Graphics 12*, 1 (2006), 14–25.

[SBCLB11] Schmidtke P., Bidon-Chanal A., Luque F. J., Barril X.: MDpocket: Open-source cavity detection and characterization on molecular dynamics trajectories. *Bioinformatics 27*, 23 (2011), 3276–3285.

[SDP*13] Sridharamurthy R., Doraiswamy H., Patel S., Varadarajan R., Natarajan V.: Extraction of robust voids and pockets in proteins. In *Proceedings of the EuroVis: Short Papers* (Leipzig, Germany, 2013), Eurographics Association, pp. 67–71.

[Ser84] Serra J.: *Image Analysis and Mathematical Morphology*. Academic Press, Orlando, Florida, USA, 1984.

[SGW93] Smart O. S., Goodfellow J. M., Wallace B.: The pore dimensions of gramicidin A. *Biophysical Journal 65*, 6 (1993), 2455–2460.

[SHB16] Sonka M., Hlavac V., Boyle R.: *Image Processing, Analysis, and Machine Vision*, 4th ed. Cengage Learning, Stamford, Connecticut, USA, 2016.

[Sim03] Simonson T.: Electrostatics and dynamics of proteins. *Reports of Progress in Physics 66* (2003), 737–787.

[SK97] Saff E., Kuijlaars A.: Distributing many points on a sphere. *The Mathematical Intelligencer 19*, 1 (1997), 5–11.

[SNW*96] Smart O. S., Neduvelil J. G., Wang X., Wallace B., Sansom M. S.: HOLE: A program for the analysis of the pore dimensions of ion channel structural models. *Journal of Molecular Graphics 14*, 6 (1996), 354–360.

[SOS96] Sanner M., Olson A., Spehner J.: Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers 38*, 3 (1996), 305–320.

[SSE*10] Schmidtke P., Souaille C., Estienne F., Baurin N., Kroemer R. T.: Large-scale comparison of four binding site detection algorithms. *Journal of Chemical Information and Modeling 50*, 12 (2010), 2191–2200.

[SSVB*13] Sehnal D., Svobodová Vařeková R., Berka K., Pravda L., Navrátilová V., Banáš P., Ionescu C.-M., Otyepka M., Koča J.: MOLE 2.0: Advanced approach for analysis of biomacromolecular channels. *Journal of Cheminformatics 5*, 1 (2013), 39.

[SZ12] Schneider S., Zacharias M.: Combining geometric pocket detection and desolvation properties to detect putative ligand binding sites on proteins. *Journal of Structural Biology 180*, 3 (2012), 546–550.

[TDCL09] Tseng Y. Y., Dupree C., Chen Z. J., Li W.-H.: Splitpocket: Identification of protein functional surfaces and characterization of their spatial patterns. *Nucleic Acids Research 37* (2009), W384–W389.

[TK10] Tripathi A., Kellogg G. E.: A novel and efficient tool for locating and characterizing protein cavities and binding sites. *Proteins: Structure, Function, and Bioinformatics 78*, 4 (2010), 825–842.

[TPS12] Tanner D. E., Phillips J. C., Schulten K.: GPU/CPU algorithm for generalized born/solvent-accessible surface area implicit solvent calculations. *Journal of Chemical Theory and Computation 8*, 7 (2012), 2521–2530.

[TU10] Till M. S., Ullmann G. M.: Mcvol-a program for calculating protein volumes and identifying cavities by a monte carlo algorithm. *Journal of Molecular Modeling 16*, 3 (2010), 419–429.

[VG10] Voss N. R., Gerstein M.: 3V: Cavity, channel and cleft volume calculator and extractor. *Nucleic Acids Research 38* (2010), W555–W562.

[VGGR10] Volkamer A., Griewel A., Grombacher T., Rarey M.: Analyzing the topology of active sites: On the prediction of pockets and subpockets. *Journal of Chemical Information and Modeling 50*, 11 (November 2010), 2041–2052.

[VKV*89] Voorintholt R., Kosters M., Vegter G., Vriend G., Hol W.: A very fast program for visualizing protein surfaces, channels and cavities. *Journal of Molecular Graphics 7*, 4 (1989), 243–245.

[Whi97] Whitley D. C.: Van der waals surface graphs and molecular shape. *Journal of Mathematical Chemistry 23*, 3-4 (1997), 377–397.

[Whi05] Whitford D.: *Proteins: Structure and Function*, 1st ed. John Wiley & Sons, Ltd., Chichester, West Sussex, England, 2005.

[WM97] Wilkinson A., McNaught A.: *IUPAC Compendium of Chemical Terminology, (the "Gold Book")*. International Union of Pure and Applied Chemistry, Zurich, Switzerland, 1997.

[WPS07] Weisel M., Proschak E., Schneider G.: PocketPicker: Analysis of ligand binding-sites with shape descriptors. *Chemistry Central Journal 1*, 1 (2007), 1–17.

[XB07] Xie L., Bourne P.: A robust and efficient algorithm for the shape description of protein structures and its application in predicting ligand binding sites. *BMC Bioinformatics 8*, Suppl 4 (2007), S9:1–S9:13.

[YFW*08] Yaffe E., Fishelovitch D., Wolfson H. J., Halperin D., Nussinov R.: MolAxis: Efficient and accurate identification of channels in macromolecules. *Proteins 73*, 1 (2008), 72–86.

[YZTY10] Yu J., Zhou Y., Tanaka I., Yao M.: Roll: A new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. *Bioinformatics 26*, 1 (2010), 46–52.

[ZB07] Zhang X., Bajaj C.: Extraction, quantification and visualization of protein pockets. *Computer Systems Bioinformatics Conference 6* (2007), 275–286.

[ZGWW12] Zheng X., Gan L., Wang E., Wang J.: Pocket-based drug design: Exploring pocket space. *The AAPS Journal 15*, 1 (2012), 228–241.

[ZLL*11] Zhang Z., Li Y., Lin B., Schroeder M., Huang B.: Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. *Bioinformatics 27*, 15 (2011), 2083–2088.

[ZP11] Zhu H., Pisabarro M. T.: MSPocket: An orientation-independent algorithm for the detection of ligand binding pockets. *Bioinformatics 27*, 3 (2011), 351–358.

# Chapter 3

## CavVis — A Field-of-View Geometric Algorithm for Protein Cavity Detection.

This chapter concerns the following article:

CavVis — A Field-of-View Geometric Algorithm for Protein Cavity Detection.
Tiago Simões, and Abel Gomes.
*Journal of Chemical Information and Modeling*, Vol. 59, No. 2, pp. 786-796, January 2019.

### Overview

This chapter proposes a new geometric algorithm called CavVis that identifies cavities through the concept of visibility of points of the molecular surface. CavVis takes advantage of computer graphics concepts such as the field-of-view (FoV), voxel ray casting, and back-face culling, in addition to the analytic formulation of the Gaussian surface. This method belongs to the category of grid-and-surface methods because it is based on the two mathematical concepts: Gaussian surface and the 3D grid that embeds it.

Taking into account the findings of Chapter 2, CavVis not only increases the accuracy in the detection of cavities on protein surfaces, but also solves the main issues of grid-and-surface methods, namely: grid-spacing sensitivity, protein-orientation sensitivity, and mouth-opening ambiguity. The first two issues are mitigated using the field-of-view (FoV), voxel ray casting, and back-face culling. The third issue is solved using the convex hull of cavity vertices.

# CavVis — A field-of-view geometric algorithm for protein cavity detection

Tiago M.C. Simões*,†,‡ and Abel J.P. Gomes*,†,‡

†*Instituto de Telecomunicações, Covilhã, Portugal*
‡*Universidade da Beira Interior, Covilhã, Portugal*

E-mail: tiago.simoes@it.ubi.pt; agomes@di.ubi.pt

## Abstract

Several geometric-based methods have been developed for the last two to three decades to detect and identify cavities (i.e., putative binding sites) on proteins, as needed to study protein-ligand interactions and protein docking. This paper introduces a new protein cavity method, called CavVis, which combines voxelization (i.e., a grid of voxels) and an analytic formulation of Gaussian surfaces that approximates the solvent-excluded surface (SES). This method builds upon visibility of points on protein surface to find its cavities. Specifically, the visibility criterion combines three concepts we borrow from computer graphics, the field-of-view (FoV) of each surface point, voxel ray casting, and back-face culling.

## Introduction

Macromolecular structures like proteins play a fundamental role in cellular processes. The interactions between proteins and nucleotides, peptides, catalytic substrates, or man-made chemicals allow the execution of their biological functions,[1] namely catalysing metabolic reactions, DNA replication, cell signalling, or intracellular transport. There are several interaction types: protein-ligand, protein-protein, protein-DNA, and so forth. These interactions are accomplished under conditions of chemical complementarity and shape complementarity.

In this paper, we are mainly interested in shape complementarity of protein-ligand interactions, in particular in the detection and identification of protein cavities (i.e., putative binding sites). This cavity detection and identification process is often seen as a first step for molecular docking.[2–4] We follow the definition of protein cavity put forward by Simões et al.[6]

There are three main families of cavity detection methods: evolution-based, energy-based, and geometry-based. Specifically, there are various types of geometric methods to detect and identify cavities on the protein surface,[5–7] namely: sphere-based (e.g., PHECOM[8]), grid-based (e.g., ConCavity[9]), surface-based (e.g., MSPocket[10]), tessellation-based (e.g., Fpocket[11]), as well as mixed geometric methods that combine some of those. CavVis, the computational cavity detection method here proposed, belongs to the class of grid-and-surface geometric methods.

Grid-based methods (or voxelization-based methods) rely on the set of protein atoms embedded into an axis-aligned grid; representative methods of this category are, for example, POCKET,[12] LIGSITE,[13] PocketPicker,[14] PocketDepth,[15] and VICE.[16] Atom centers and their radii allow to distinguish the grid nodes lying inside from those on or outside the protein (i.e., set of atoms); a voxel lying on the protein surface (i.e., van der Waals surface) owns at least one inside grid node and one outside grid node, and is called surface voxel. This labelling of grid nodes amounts to the generation of a discrete scalar field over the grid. The leading idea of these methods is then finding linear paths of outside grid nodes bracketed between surface voxels. However, grid-based methods suffer from two main deficiencies, grid-spacing sensitivity and protein-orientation sensitivity, i.e., changing the grid spacing or the protein orientation may result in detecting a different number and different locations of cavities.[6] These problems can be mitigated by increasing the number of paths than come out from each surface voxel. We solve these problems in CavVis by using a field-of-view (FoV) cone to capture the FoV-visible triangle regions of the molecular in front of cone's apex.

SCREEN,[17] CHUNNEL,[18] Giard et al.,[19] and MSPocket[10] are surface-based methods for the detection of cavities on proteins. SCREEN generates two solvent-excluded-surfaces (SES) for the protein using probe spheres with distinct radii for the inner (1.4 Å) and outer (5 Å) surfaces. In this case, a cavity is identified by the empty space (or gap) between the inner and outer surface where at least one water molecule fits in. CHUNNEL was specifically designed to detect protein channels (or pores). It takes advantage of the Euler-Poincaré formula to extract the number and location of such channels from the 2-dimensional simplicial triangulation of the protein's SES. In turn, Giard et al.'s method builds upon both the triangulation of the molecular surface and its convex hull (another triangulation) to find the cavities that boil down to the empty space (or gap) between the two triangulations; they use the Gaussian molecular surface that supposedly approximates the SES. Unlike the previous methods, MSPocket only uses one surface triangulation, specifically a SES triangulation. The key idea of this method is to identify concave vertices of the triangulation so that a triangle is said to be concave if its vertices are all concave. A cavity is then seen as a connected set of concave triangles. To identify a concave vertex, MSPocket first calculates the angle between the vertex normal and the average normal of its neighbor vertices (i..e, vertices of its star of incident triangles). If the angle is less than 90°, the vertex is classified as concave. This geometric criterion amounts to evaluate the *local* curvature in the neighborhood of each vertex, and thus suffices to determine a significant number of protein cavities; but, MSPocket tends to miss larger cavities that have some convex regions. In contrast, CavVis overcomes this problem using a *zonal* visibility criterion based on three computer graphics tools: the FoV cone, voxel ray casting, and back-face culling[20].[21] The first two tools operate similarly to blocking scanning directions of grid-based methods, with the difference we only use a single scanning direction per surface vertex, while the third tool discards triangles that are not FoV-visible to the viewer virtually located at each surface vertex.

In turn, Travel Depth,[22,23] CriticalFinder[4] and GaussianFinder[24] are representatives of grid-and-surface methods. The leading idea behind grid-and-surface methods is to take advantage of the virtues of both grid- and surface-based methods, and simultaneously to rid off their drawbacks. In fact, most grid-and-surface-based methods build upon two surfaces to capture the grid nodes of cavities between them, so solving the problem of protein-orientation sensitivity of grid-based methods, as well as the mouth-opening ambiguity (i.e., uncertainity about the location of mouth openings or stopgaps of cavities) of single surface-based methods.[6] This mixture of both grid- and surface-based techniques also solves the problem of grid-spacing sensitivity since the grid spacing is less or equal to $1/2r$, where $r$ is the radius of the water molecule.[25]

The protein cavity detection method here proposed, called CavVis, is another grid-and-surface method. But, unlike other grid-and-surface methods, it only uses a single surface enclosing the protein. Therefore, as shown throughout the manuscript, the typical problems of grid-based methods are solved in a different manner. In fact, CavVis performs the voxelization of the space that embeds the protein in order to construct a triangulation of the molecular surface, in particular the one outputted by the marching cubes (MC) algorithm.[26] After triangulating the molecular surface, one uses the field-of-view of each vertex to compute its visible vertices that are disposed in front of it. These visible vertices are then

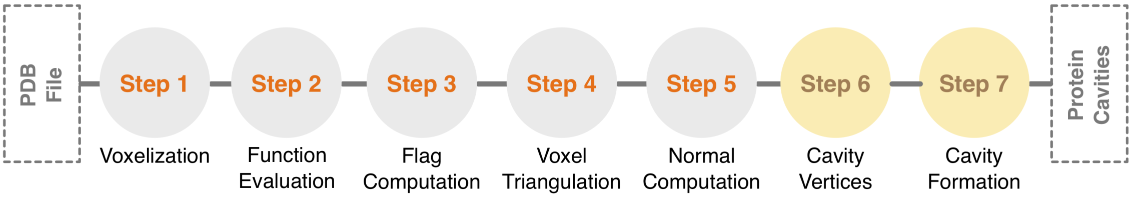


Figure 1: CavVis' steps.

grouped together into clusters. Each cluster originates a cavity after filling its convex hull with balls. Computing the convex hull for each cluster, rather than for the entire molecular surface, resolves the protein-orientation sensitivity problem enunciated above for grid-based methods, and also the mouth-opening ambiguity problem of single surface-based methods.

# Methods

CavVis builds upon the MC algorithm.[26] The MC algorithm is a well-known triangulation algorithm in computer graphics and volume rendering[27].[28] It is here used to triangulate the Gaussian molecular surface generated from $N$ atoms as follows:

$$f(\mathbf{p}) = \sum_{i=1}^{N} f_i(\mathbf{p}) \tag{1}$$

which represents the overall electron density at the point $\mathbf{p}$ as the summation of subsidiary electron densities $f_i(\mathbf{p}) = e^{-d(\frac{||\mathbf{p}-\mathbf{p}_i||^2}{r_i^2}-1)}$ of all atoms $i$ at $\mathbf{p}$; here, $||\mathbf{p}-\mathbf{p}_i||$ denotes the distance of the arbitrary point $\mathbf{p}$ to the center $\mathbf{p}_i$ of the $i$-th atom, $r_i$ stands for the radius of atom $i$, and $d$ represents the decay rate of the Gaussian kernel associated to each atom. The Gaussian surface has been used in several research works as, for example, Zhang and Bajaj,[29] Giard et al.,[19] Krone et al.,[30] and Dias et al.[4] In particular, we use the Gaussian formulation that most approximates the SES (solvent-excluded surface);[31–34] that is, the Gaussian surface defined by $f(\mathbf{p}) = T$ (with the threshold $T = 1.0$) and decay rate $d = 2.35$.

## CavVis Overview

Briefly, as shown in Fig. 1, CavVis consists of the following steps:

1. Voxelize the bounding box enclosing the protein.

2. Evaluate $f$ at each node of the voxelized box or grid.

3. Find the 8-bit flag for each voxel, which determines if the voxel is inside, outside or crossed by the surface, which are known as interior, exterior, or surface voxels, respectively.

4. Triangulate the surface voxels. The triangulation inside each surface voxel is performed in conformity with its 8-bit flag.

5. Compute normals at triangle vertices. These normals are used not only for correctly rendering the surface, but also work as the support for our cavity detection algorithm.

6. For each surface vertex, find other vertices inside its FoV cone, which is defined by the triple (vertex, normal, angle). Any vertex whose cone does not span triangle vertices is immediately discarded; it does not belong to any cavity.

7. Construct cavities using the forward visibility condition of each surface vertex; that is, if the cone of the vertex $A$ includes $B$ (i.e., if $B$ is visible to $A$), and the cone of $B$ includes $C$, then $A$, $B$, and $C$ belong to the same cavity.

The first five steps concern the MC algorithm, while the last two take advantage of the former steps to detect cavities on protein surface. Let us then briefly describe each step of the former five; we describe the last two steps in more detail because they are specific to our cavity detection method.

## Voxelization

First, one embeds the molecule into its axis-aligned bounding box. For that purpose, one determines the minimum and maximum coordinates of the atomic centers, that is, $(x_{min}, y_{min}, z_{min})$ and $(x_{max}, y_{max}, z_{max})$, which define the opposite corners of the bounding box. To guarantee that all atoms get inside the bounding box, one uses a padding of 2.0Å. Then, the bounding box is voxelized with a grid spacing $\Delta = 0.6$Å in order to guarantee the correct sampling of each cavity with at least a voxel; that is, we assume that a single voxel is enough to detect a protein cavity. More details about parameter tuning, including the grid spacing are discussed further ahead.

## Function Evaluation

Second, one computes the value of $f$ (cf. Eq. 1) at the 0-th corner of each voxel; note that the computation of $f$ on other corners of the same voxel will be carried when one iterates on neighboring voxels. In order to speed up computations, the value of $f$ is calculated locally per atom, rather than globally per voxel.[35] In other words, for each atom $i$, we calculate the value of $f_i$ only for voxels of a $n \times n \times n$ sub-box centered at each atom, with $n = \frac{2 \,.\, r_{max}}{\Delta}$, where $r_{max}$ is the radius of the biggest atom of the input protein; for example, considering $r_{max} = 1.8$Å and $\Delta = 0.6$Å, we obtain $n = 6$ for every single atom. Therefore, the final value of $f$ at the 0-th corner of each voxel is the result of summing up the values $f_i$ of its neighboring atoms, specifically atoms whose atom sub-boxes contain such a voxel. Each atom sub-box avoids to compute each $f_i$ for all voxels of the bounding box, making the complexity of CavVis dependent on the number of atoms rather than the cubic resolution of the grid. In other words, this procedure translates into a process of the linearization of the algorithmic complexity of CavVis.

## Flag Computation

The value of $f$ at each corner of a voxel determines the value of its 8-bit flag. According to Lorensen and Cline,[26] this means that a voxel has 256 possible configurations. If the voxel flag is either 00000000 or 11111111, the voxel is not crossed by the molecular surface; otherwise, the surface intersects the voxel. For example, if the flag is 10010001, with the 0-th, 4-th, and 7-th bits (or corners) set to 1, then we know we are in presence of a surface voxel, that is, a voxel traversed by the molecular surface. The flag 00000000 denotes a voxel outside the molecular surface, whereas 11111111 a voxel inside the molecular surface. The value of either 0 or 1 held by a flag bit depends on the value of $f$ at the corresponding voxel corner; if $f > T$ (with the isovalue $T = 1.0$), the flag bit is set to 1; otherwise, it remains unchanged to its initial value 0. Therefore, the Gaussian molecular surface is an isosurface.

## Voxel Triangulation

The 8-bit flag of each voxel determines the triangle configuration inside such voxel, that is, how the surface is triangulated inside such voxel. For further details about the MC triangulation, the reader is referred to.[26,28] However, as explained further ahead, only surface voxels are of interest here for the purpose of the cavity detection on molecular surfaces, since they are the only ones that contain surface triangles (and their vertices).

## Normal Computation

Computing normals at surface vertices is a straightforward task because the molecular surface is analytical (cf. Eq. (1)). Specifically, a normal vector at a point of the Gaussian molecular is given usually by the gradient vector. However, we know that gradient vector points into the direction of the steepest ascent[36],[28] so it points inwards the Gaussian surface. It happens that we want the normal vector pointing outwards the surface for cone visibility and rendering purposes, so that the normal vector $\vec{n}$ is the negative gradient vector $\nabla$, that is

$$\vec{n} = -\nabla f = \big(-\frac{\partial f}{\partial x}, -\frac{\partial f}{\partial y}, -\frac{\partial f}{\partial z}\big) \tag{2}$$

with

$$\begin{aligned} \frac{\partial f}{\partial x}(\mathbf{p}) &= -\sum_{i=1}^{N} \frac{2d(x-x_i)f_i(\mathbf{p})}{r_i^2} \\ \frac{\partial f}{\partial y}(\mathbf{p}) &= -\sum_{i=1}^{N} \frac{2d(y-y_i)f_i(\mathbf{p})}{r_i^2} \\ \frac{\partial f}{\partial z}(\mathbf{p}) &= -\sum_{i=1}^{N} \frac{2d(z-z_i)f_i(\mathbf{p})}{r_i^2} \end{aligned} \tag{3}$$

where $\mathbf{p} = (x, y, z)$ is an arbitrary point, $\mathbf{p}_i = (x_i, y_i, z_i)$ the center of atom $i$, $r_i$ the radius of atom $i$, and $d$ the decay rate of electron density of each atom.

However, the computation of normal vectors using Eq. (2) becomes very expensive when the number of atoms increases in a significant manner. In fact, for a protein of 25,000 atoms, each partial derivative of the gradient at a single surface point requires the computation of 3×25,000= 75,000 subsidiary derivatives. To overcome this problem, we use the mean vector of the normal vectors of triangles around each vertex of the triangulation.

## Cavity Vertices

For each surface vertex, we find the nearest surface voxel within its FoV cone. This is so because a viewer positioned at such vertex and looking in the direction of the vertex normal must see the opposite side of the cavity where the closest surface voxel lies in. The FoV cone is defined by the vertex normal and the aperture angle $\phi$ (or angle of view).

So, given a surface vertex, collecting its FoV-visible vertices and triangles is as follows:

1. Find the nearest surface voxel along its FoV cone axis.
2. Find surface vertices and triangles simultaneously inside the FoV cone and 1.4 Å-radius sphere centered at the previous surface voxel.
3. Discard non-visible surface vertices and triangles through back-face culling.

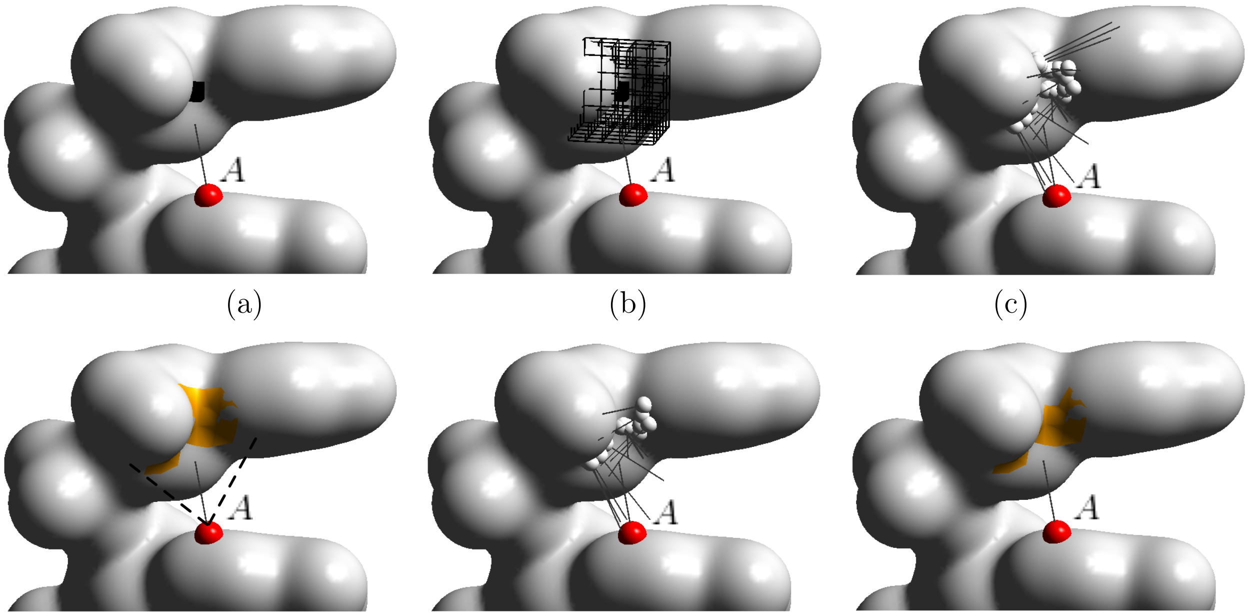
Let us now describe these three steps in more detail.

*Nearest surface voxel.* Here, we assume that we already have the array of all surface voxels (i.e., with triangles therein). These voxels are those traversed by the molecular surface, and are outputted by the MC algorithm (first five steps of CavVis), so that surface triangles, vertices, and their normals can directly retrieved for each surface voxel.

Computing the nearest surface voxel is performed using the voxel ray casting algorithm due to Amanatides and Woo.[21] This is accomplished by considering only the first surface voxel crossed by the FoV axis, which is defined by the normal vector $\vec{v}$. This is illustrated in Figure 2(a).

*Finding surface vertices inside FoV cone.* After finding the surface voxel, we collect all the vertices and triangles inside the $5 \times 5 \times 5$ sub-box centered at the surface voxel (Figure 2(b)). This sub-box is $5\Delta = 3.0$ long, which almost minimally contains a sphere featuring the water molecule. This constant-sized box or sphere forces the FoV angle –associated to each surface vertex– to vary, depending on the distance between such surface vertex and the corresponding surface voxel in front of it. In practice, we do not need to compute the FoV angle ($\phi$) explicitly. The vertices collected inside FoV cone are represented by small white spheres in Figure 2(c), while their corresponding triangles (in orange) are shown Figure 2(d).

*Back-face culling of vertices and triangles.* Among the previous tentative cavity vertices and triangles, we need to discard those not visible to the viewer placed at the FoV cone apex, who is looking in the direction of the normal vector. This is performed using the well-known back-face culling technique in computer graphics (see, for example, Hughes et al.[20]). The back-face culling discards all triangles (and their vertices) that satisfy the condition $\vec{n} \cdot \vec{v} \geq 0$, where $\vec{n}$ stands for the triangle



(a) (b) (c)

(d) (e) (f)

Figure 2: Finding the vertices and triangles visible to a surface vertex $A$: (a) a surface voxel (in black) was found using voxel ray casting along the cone axis defined by the normal vector at $A$; (b) the $5 \times 5 \times 5$ sub-box inscribed inside a sphere centered at surface voxel; this radius and the vertex-voxel distance implicitly define the FoV cone; (c) the triangulation vertices (in white) and their normal spikes inside the sub-box (or sphere); (d) the triangles (in orange) inside the sub-box (or sphere); (e) the triangulation vertices (in white) after back-face culling; (f) the triangles (in orange) after back-face culling.

normal vector and $\vec{v}$ the viewer's vector; that is, all triangles that are not visible to the cone apex (here considered as viewer's location) are discarded. The back-face culling of vertices and triangles is illustrated in Figure 2(e)-(f) relative to Figure 2(c)-(d), respectively.

Summing up, for each surface vertex, we determine its visible vertices and triangles. The identifiers (or labels) of these vertices and triangles are hold in separate arrays associated to each surface vertex. If the FoV semi-axis does not intersect the molecular surface, it is not necessary thus to label any vertex or triangle. Let us now see how these vertices and triangles are aggregated into cavities on the molecular surface.

## Cavity Formation

Before proceeding any further, recall that all non-cavity vertices have been already discarded before. Therefore, the formation of cavities builds upon clustering of cavity vertices on the molecular surface. This procedure involves three main steps:

1. Clustering of vertices of each putative cavity using a breadth-first strategy (see Fig. 3);
2. Merge putative surface cavities if they are close enough to each other (see Fig. 4);
3. Find protein cavities by filling then with dummy atoms (i.e., pseudo-atoms) (see Fig. 5).

*Vertex clustering.* Grouping surface vertices into distinct clusters takes advantage of the labelling mentioned above. Each vertex collects the labels (or ids) of its FoV-visible vertices. As shown in Fig. 3, we can use a breadth-first strategy to collect all vertices of each cluster. For example, let us assume that the vertex 1

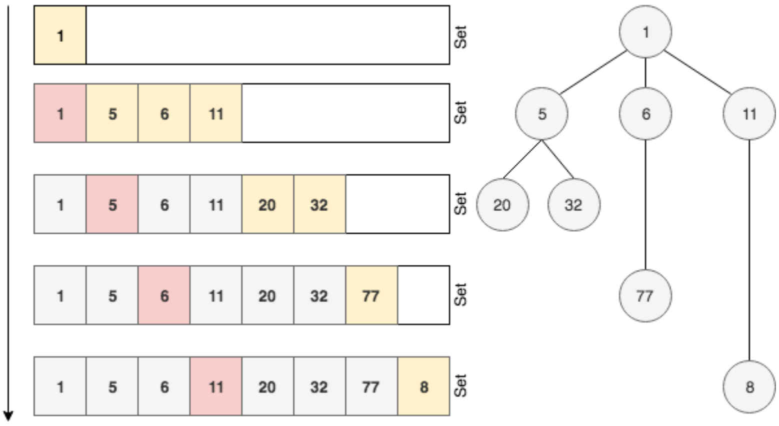


Figure 3: Vertex clustering using the concept of set (to avoid repetitions) and a breadth-first strategy. On the left-hand side, we show how a cluster is formed in an array set, while on the right-hand side we show the clustering process using a breadth-first strategy on a tree.

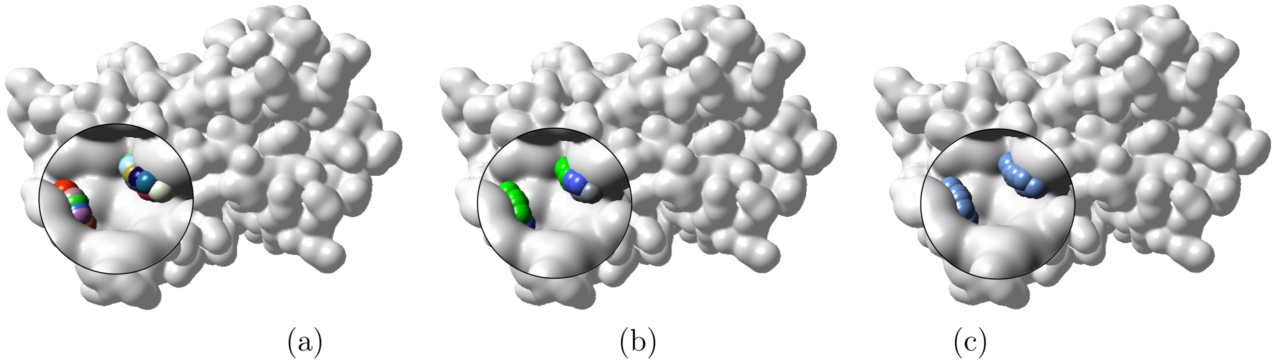


Figure 4: Building up of a cavity of the protein 2VQ4: (a) the cavity vertices featured by multi-colored balls are not grouped together yet; (b) three clusters (in green, blue, and gray) were formed after vertex clustering; (c) merging the previous three clusters originates a single cluster (cavity).

(the tree root) holds the set of labels $\{5, 6, 11\}$. This means that all vertices with at least one of those labels belong to the same cluster, and corresponds to second level of the tree, that is, the array set is $\{1, 5, 6, 11\}$. Then, we sequentially reunite all labels of each vertex of the second level into the set, say $\{1, 5, 6, 11, 20, 32, 77, 8\}$ to form the third level; that is, we add 20 and 32 for vertex 5, we add 77 for vertex 6, and we add 8 for vertex 11. This process repeats again and again while there is at least a label to add to array set; otherwise, a new cluster is complete. Note that we avoid repetitions because, by definition, a set has no repetitions. To form a new cluster, we pick a vertex that does not belong to a cluster yet, and repeat the clustering process above. This vertex clustering process is illustrated in Figure 4(a)-(b), where the multicolor small balls representing surface vertices are grouped together into three clusters of the same cavity; these three clusters are merged into a single one to form a cavity, as shown in Figure 4(c).

*Cluster merging.* As illustrated in Figure 4(b), merging clusters of vertices is necessary when two or more clusters have been identified for a cavity. For that purpose, we need to compute the distance between every two clusters, which is given by the distance between two closest vertices belonging to distinct clusters. Merging takes place if the minimum distance between two clusters is less than 1.4Å (i.e., radius of

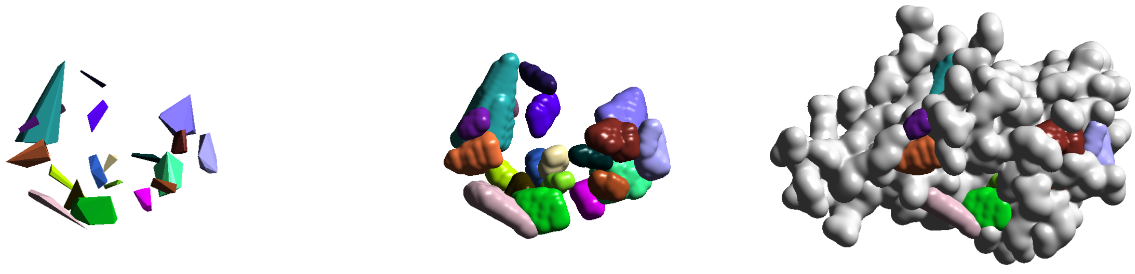


Figure 5: Cavity filling for the protein 3CAF: (a) convex hulls of all clusters of vertices; (b) each cavity is depicted as a volume generated by a Gaussian surface whose kernel functions are centered at each vertex of cavity's convex hull or at each grid node inside the convex hull; (c) the protein depicted together with its cavities.

water molecule). But, we end up selecting only the first ten bigger clusters as cavities for the sake of performance evaluation, because bigger cavities has a bigger probability of being binding sites. Such bigger clusters are those with greater number of vertices.

*Cavity filling.* Terminated the cluster merging, we end up obtaining a single cluster per cavity. Then, one computes the convex hull of all vertices of the cavity cluster. Filling a cavity with spheres of radius 1.4Å (water molecules) must satisfy the following conditions: (i) its center must be a grid node inside the convex hull; (ii) its center must be a convex hull vertex; (iii) its center must be outside the molecular surface; This filling procedure is illustrated in Figure 5. A few more examples of cavities as detected by CavVis are shown in Figure 6.

## Molecular Visualization

CavVis software also includes a visualization module that uses the MC triangulation algorithm in conjunction with the OpenGL 2.0 graphics system in order to visualize not only the molecular surface, but also protein cavities and their surfaces. As mentioned above, the visualization of protein surfaces and their cavities build upon the MC triangulation algorithm. In fact, all pictures displaying molecules and cavities in this paper were generated using such OpenGL visualization module. Also, in addition to OpenGL output, CavVis also generates .xyz files describing cavities, a single .xyz per cavity. These files allow us to visualize CavVis cavities using well-known molecular visualization tools such as, for example, VMD[37] and Chimera.[38]

# Results and Discussion

## Hardware/Software Setup

Testing was performed on an Apple iMac desktop, equipped with a 3.2 GHz Intel Core i5, 16 GB RAM, and a NVIDIA GeForce GT 755M, running OS X Yosemite operating system. Let us also mention that we run all benchmarking methods, including CavVis, on this desktop computer. CavVis was implemented in C++ and is publicly available at `https://github.com/MediaLabProjects/CavVis/`.

## Ground Truth

In testing, we used PDBSum as ground-truth dataset of binding sites.[44] More specifically, we used a subset of 1239 proteins (335 apo proteins and 904 holo proteins) of PDBSum. This subset corresponds to the intersection set between PDBsum[44] and LigASite[39] datasets.

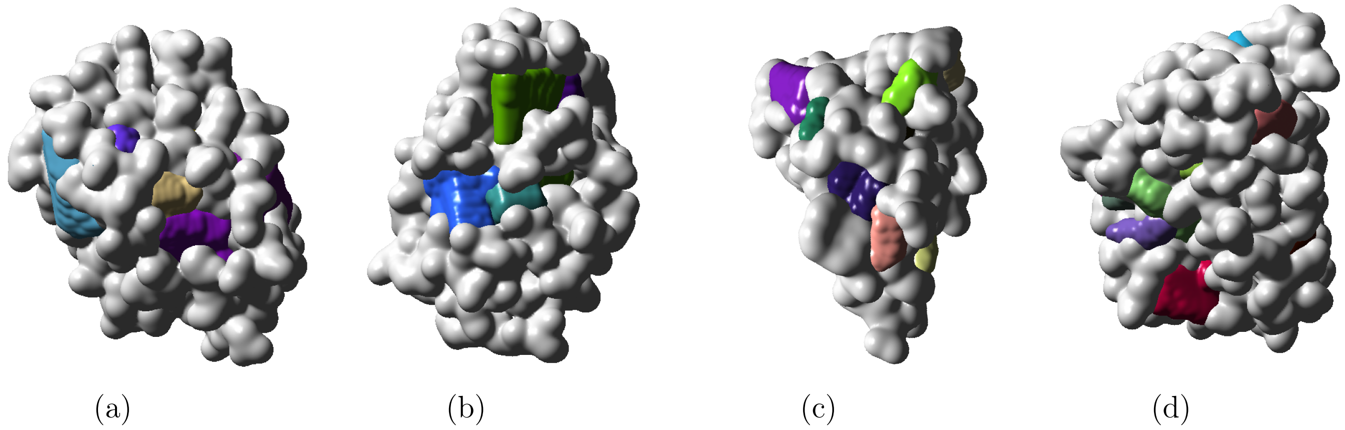


Figure 6: CavVis' cavity detection for four proteins: (a) 1MZL; (b) 1FK3; (c) 2RK2; and (d) 2PCY.

## Benchmarking Methods

CavVis was compared with the following cavity detection methods:

- Fpocket: This tessellation-based algorithm builds upon the Voronoi diagram (of the atom centers) and theory of alpha spheres to detect cavities on proteins (see Guilloux et al.[11] for more details).

- MSPocket: This surface-based algorithm takes advantage of surface sampling and point normals to detect protein cavities (see Zhu and Pisabarro[10]). Basically, one looks at the normals in the neighborhood of each sampled point to decide whether the point neighborhood is concave or not; concave neighborhoods usually correspond to protein cavities.

- GHECOM: This grid-and-sphere method employs multi-size spherical probes in conjunction with a 3D grid and the theory of mathematical morphology to detect cavity regions (see Kawabata[8]). The leading idea is that a cavity is the locus outside the protein where a small enough probe gets in, but not a large probe.

- CriticalFinder: It is a grid-and-surface method that finds cavities through the detection of critical points on voxels intersected by the Gaussian surface (see Dias et al.[40] for more details).

- POCASA: This grid-and-sphere algorithm uses a boolean scalar field classifier of the voxelized domain to distinguish the inner and outer grid nodes of the protein surface (see Yu et al.[41] for more details). Then, one uses a large probe sphere rolling the molecular surface to identify cavities; a cavity is a cluster of outer grid nodes not caught by the probe sphere.

- ConCavity: A grid-based algorithm that depends on algorithms based on sequence and structure to detect cavity regions (see Capra et al.[9] for more details).

- PASS: This sphere-based method fills cavities with probe spheres in conformity with the three-point Connolly-like sphere geometry.[42] See Brady Jr. and Stouten[43] for further details.

We chose these methods because their source codes are publicly available to the scientific community, with the advantage of being accompanied by a supporting scientific article published elsewhere. Also, these methods cover the most significant families of methods.[5–7]

## Parameter Tuning

As seen above, we use a number of parameters in CavVis, namely: Gaussian decay rate

Table 1: CavVis' time performance (total time and time per step) for the six biggest proteins of the dataset.

| Proteins | # Atoms | Voxelization | | Function Evaluation | | Flag Computation | | Voxel Triangulation | | Normal Computation | | Cavity Vertices | | Cavity Formation | | Total Time (s) | Time per Atom (ms) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Time (s) | % | Time (s) | % | Time (s) | % | Time (s) | % | Time (s) | % | Time (s) | % | Time (s) | % | | |
| 1VPN | 13230 | $3 \times 10^{-6}$ | $10^{-4}$% | 1,110 | 34,3% | 0,440 | 13,6% | 0,420 | 13,0% | 0,179 | 5,5% | 0,708 | 21,9% | 0,376 | 11,6% | 3,232 | 0,244 |
| 2YWB | 15065 | $3 \times 10^{-6}$ | $10^{-4}$% | 1,434 | 31,5% | 0,560 | 12,3% | 0,580 | 12,8% | 0,211 | 4,6% | 1,031 | 22,7% | 0,730 | 16,1% | 4,547 | 0,302 |
| 3M4D | 16762 | $3 \times 10^{-6}$ | $10^{-4}$% | 1,553 | 27,2% | 0,610 | 10,7% | 0,620 | 10,9% | 0,310 | 5,4% | 1,129 | 19,8% | 1,483 | 26,0% | 5,706 | 0,340 |
| 2AHU | 16967 | $3 \times 10^{-6}$ | $10^{-4}$% | 1,434 | 28,1% | 0,590 | 11,5% | 0,620 | 12,1% | 0,229 | 4,5% | 1,071 | 21,0% | 1,166 | 22,8% | 5,110 | 0,301 |
| 1TYF | 21308 | $3 \times 10^{-6}$ | $10^{-4}$% | 1,854 | 27,4% | 0,740 | 10,9% | 0,790 | 11,7% | 0,345 | 5,1% | 1,446 | 21,4% | 1,591 | 23,5% | 6,767 | 0,318 |
| 1XO6 | 25641 | $3 \times 10^{-6}$ | $10^{-4}$% | 2,185 | 27,5% | 0,870 | 10,9% | 0,920 | 11,6% | 0,357 | 4,5% | 1,835 | 23,1% | 1,782 | 22,4% | 7,949 | 0,310 |

($d = 2.35$), isovalue ($T = 1.0$), grid spacing ($\Delta = 0.6$), local sub-box resolution ($n \times n \times n$) of each atom, field-of-view angle ($\phi$), and maximum cluster-cluster distance (1.4Å). The values of $d$ and $T$ remain unchanged because they both guarantee that the respective Gaussian molecular surface is the isosurface that better approximates the solvent-excluded surface (SES).

The grid spacing $\Delta$ is very important because it determines the number of cavities identified by CavVis; that is, the number (and, subsequently, the location) of cavities varies by changing the grid spacing. As shown by Dias and Gomes,[25] the correct sampling of cavities is regulated by the Theorem of Nyquist. Here, we relax the Nyquist condition, since we ensure that at least a single voxel fits in such a cavity. In fact, assuming that a cavity at least hosts a water molecule (with radius of 1.4 Å), we come to the conclusion that the maximal cube inscribed in a water molecule sphere is about 2.0 Å long approximately (using Pythagoras theorem). This means that it is impossible to detect many cavities when one uses a grid spacing greater than 2.0 Å; hence, we use the default grid spacing $\Delta = 0.6$, which allows to come up with a speed-accuracy tradeoff in the detection of protein cavities.

Also, we use a sub-box of voxels around and centered at each atom to avoid to evaluate $f_i$ everywhere. Basically, we evaluate $f_i$ at every single voxel of the sub-box centered at its corresponding $i$-th atom. Note that this procedure also guarantees that Gaussian functions associated to overlapping atoms mix properly without gaps. The resolution of such sub-box is given by $n \times n \times n$, with $n = \frac{2 \,.\, r_{MAX}}{\Delta}$, where $r_{MAX} = 1.8$Å is the radius of the largest atom of the input protein and $\Delta = 0.6$Å. Therefore, the resolution of each atom's local sub-box is $6 \times 6 \times 6$. However, this resolution automatically changes if the grid spacing parameter were user-redefined.

Each atom's sub-box allows us to calculate the scalar field $f$ —associated to a protein— in *a per* atom basis; that is, one calculates $f_i$ only for the voxels of the sub-box centered at $i$-th atom, which is then added to the current value of $f$ at each voxel using an accumulation strategy. This procedure works as a sort of linearization of the entire electron density field computation of the protein, because the scalar field computations are performed per atom of a 1-dimensional array, rather than per voxel of a 3-dimensional grid.

Concerning the FoV angle ($\phi$), we can say that it is dynamically variable because it depends on the distance between each triangulation vertex and its corresponding surface voxel in front of it on the opposite side of a given cavity. In practice, the value of $\phi$ is never calculated because we use a fixed $5 \times 5 \times 5$ sub-box centered at each surface voxel found using the voxel ray tracing (see Fig. 2). This sub-box is the one that fits the water molecule sphere of radius 1.4 Å. Therefore, the value of $\phi$ also depends on this radius, but we do not need to calculate $\phi$ explicitly.

Finally, after constructing the clusters of triangulation vertices, one assumes that the maximum distance between any two clusters is by default 1.4 Å maximum. Otherwise, they are merged into a single cluster. The cluster-cluster distance is given by the minimum vertex-vertex distance. It is clear that the cluster-cluster distance parameter maybe user-redefined, but this procedure has implications in the final number of detected cavities. Increasing the distance results in a less number of cavities because of the effect of merging more clusters; on the contrary, the number of cavities increases with decreasing of the cluster-cluster distance.

## Time Performance

Before proceeding any further, let us mention that CavVis is a single-threading program that runs on CPU (Central Processing Unit). To have a glimpse of the time complexity of CavVis, let us have a look at Table 1. The last column of Table 1 ("Time per Atom") indicates that the experimental time complexity seems to be linear. The time spent per atom is about 0.3 milliseconds, no matter the number of atoms of the protein.

This linear complexity may be surprising because one would expect cubic time complexity given the 3D grid. It happens that the first step (voxelization) is very fast, while the remaining ones are relatively fast and of the same order of magnitude. Initially, the second step (function evaluation) and the fifth step (normal computation) were the most expensive computationally.

In fact, the function evaluation initially had cubic complexity because it was performed on a voxel basis. Besides, we had to compute $N$ functions $f_i$ for each voxel, where $N$ stands for the number of atoms. As shown above, this procedure was linearized evaluating the scalar field on an atom basis (i.e., over a 1D array of atoms, rather than in a per voxel basis over the entire 3D grid of voxels).

In turn, the computation of normal vectors was also very time-consuming because we had to compute three derivatives per vertex, with each derivative involving the computation of $N$ functions $f_i$ (cf. Eq. (2). We solved this problem through the computation of the outwards vector of each triangle. Each triangle vector is determined using the cross product of two consecutive triangle edges.

## Accuracy Metrics

As suggested above, to evaluate the accuracy of CavVis, we compared it with state-of-the-art methods relative to a dataset of known binding sites, which works here as ground-truth. Such evaluation was carried out using the well-known metrics of precision and recall, which are defined as

$$P = \frac{TP}{TP + FP} \tag{4}$$

and

$$R = \frac{TP}{TP + FN} \tag{5}$$

respectively, where $TP$ denotes the number of true positives, $FP$ the number of false positives, and $FN$ the number of false negatives. A true positive is a hit, that is, when a ground-truth cavity and a method-specific cavity overlap in some extent. A false positive is a method-specific cavity that does not meet any ground-truth cavity. A false negative is a ground-truth cavity that was not detected by a specific method.

As usual in statistical analysis, we employ the F-score metric ($F$) to rank the benchmarking methods. The F-score is defined as the harmonic mean of precision and recall, whose expression is as follows:

$$F = 2\frac{P.R}{P + R} = \frac{2TP}{2TP + FP + FN} \tag{6}$$

Thus, the precision and recall should have high values to ensure that F-score also gets a high value. The accuracy results are shown in Table 2 and Figure 7.

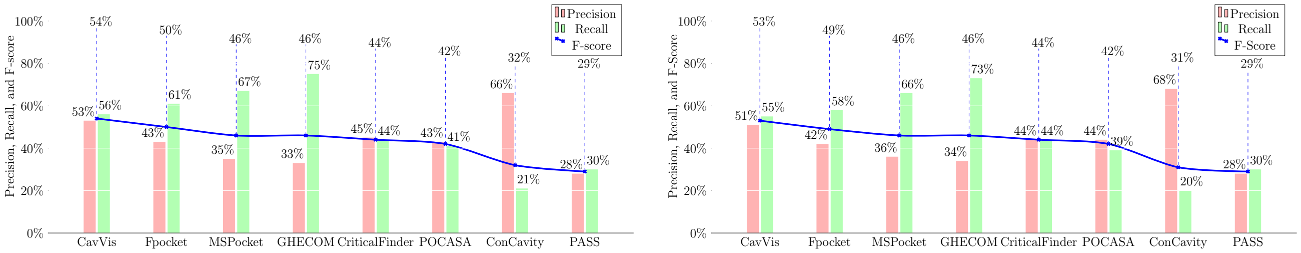
## Discussion

After a brief glance at the results shown in Table 2 and Figure 7, we observe that CavVis is not top-ranked in terms of precision and recall separately. ConCavity ranks first in terms of precision, while GHECOM ranks on the top concerning recall, and this applies to both unbounded proteins (or apo proteins) and bounded proteins (or holo proteins).

However, ConCavity performs badly in terms of recall because it produces too many false neg-

Table 2: Accuracy of the benchmark methods for the apo and holo proteins of the ground-truth dataset of binding sites.

| **Method** | **apo proteins** | | | | | | **holo proteins** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TP | FP | FN | ≈ P | ≈ R | ≈ F | TP | FP | FN | ≈ P | ≈ R | ≈ F |
| CavVis | 1872 | 1693 | 1478 | 53% | 56% | 54% | 4931 | 4675 | 4109 | 51% | 55% | 53% |
| Fpocket | 2040 | 2754 | 1310 | 43% | 61% | 50% | 5259 | 7161 | 3781 | 42% | 58% | 49% |
| MSPocket | 2239 | 4096 | 1111 | 35% | 67% | 46% | 5929 | 10558 | 3101 | 36% | 66% | 46% |
| GHECOM | 2509 | 5087 | 841 | 33% | 75% | 46% | 6632 | 12883 | 2408 | 34% | 73% | 46% |
| CriticalFinder | 1487 | 1850 | 1863 | 45% | 44% | 44% | 3939 | 4992 | 5101 | 44% | 44% | 44% |
| POCASA | 1358 | 1804 | 1992 | 43% | 41% | 42% | 3568 | 4571 | 5472 | 44% | 39% | 42% |
| ConCavity | 695 | 356 | 2655 | 66% | 21% | 32% | 1786 | 833 | 7254 | 68% | 20% | 31% |
| PASS | 1018 | 2669 | 2332 | 28% | 30% | 29% | 2616 | 6759 | 6250 | 28% | 30% | 29% |



(a) (b)

Figure 7: Accuracy study of valid predictions across the set of methods benchmarked. a) Precision, recall, and F-score percentages for apo structures. b) Precision, recall, and F-score percentages for holo structures.

atives (i.e., it misses ground-truth cavities quite often). This significant number of false negatives likely has to do with too restrictive cavity filtering criteria. In fact, ConCavity discards small and large cavities; specifically, it discards cavities with radius less than 1 Å and greater than 5 Å. It is clear this also results in a small number of true positives. This prevents the usage of the top-10 filtering criterion (from the biggest to the smallest cavity in terms of volume) as in PDBsum.

On the contrary, GHECOM is the method that most hits the binding sites of the ground-truth dataset (i.e., with the highest number of true positives), but it has a poor performance in terms of precision because it detects too many false positives (i.e. fake cavities). GHECOM's performance can be explained by the fact that it does not use the top-10 filtering criterion either. Obviously, the unbalanced values of precision and recall impairs the $F$-score values of ConCavity and GHECOM.

Similar to GHECOM, MSPocket also produces many true and false positives because it generates many clusters of concave triangles, being the triangle-triangle connectivity the only criterion to consider triangles are constituents of a cavity. Eventually, an additional criterion to merge more clusters might improve the accuracy of MSPocket, even considering MSPocket already uses the top-10 filtering criterion.

POCASA is similar to GHECOM, yet the minimum probe sphere has radius greater than 2.0Å, which works as a filtering criterion for cavities. The difference is that POCASA delivers the top-ranked cavities in terms of decreasing volumes; consequently, the number of false positives decreases in relation to GHECOM.

CriticalFinder detects the surface voxels belonging to cavities, called critical voxels. These critical voxels are then grouped together into separate clusters since they are within a specific distance. In a way, grouping such surface voxels together is much like to group the surface triangles in MSPocket. The difference is that CriticalFinder uses a more effective clustering algorithm in the formation of cavities. To this it is not strange the fact that CriticalFinder and POCASA have similar figures in terms of TP, FP, and FN (cf. Table 2), although CriticalFinder is slightly more accurate than POCASA.

Regarding PASS, it is one of the methods that less hits the protein cavities of the ground-truth dataset; that is, its number of true positives is relatively small and its number of false negatives is noticeable high when compared to other methods. Its number of fake cavities (FP) is also high. This can be explained by the fact that it is difficult to correctly evaluate the buriedness of cavities by accretion of probe spheres. Worse, it is the fact that PASS is another method that does not use the top-10 filtering criterion.

As far as Fpocket is concerned, it is one of the methods that most hits and misses less the protein cavities of the ground-truth dataset. However, it produces many fake cavities (FP), even considering that it uses a minimum alpha sphere of radius of 3.0Å, which works as a filtering criterion for cavities. Nevertheless, Fpocket does not use top-ranking filtering criterion. Looking at Table 2, we see that Fpocket would excel in detecting protein cavities if we were able to reduce the number of false positives.

Finally, CavVis ends up being the method with the best accuracy among those of the benchmark because it makes a more balanced mixture of precision and recall, resulting in the highest $F$-score of 54% for apo proteins and 53% for holo proteins. CavVis ranks top largely due to the following: (i) its ability to overcome the issues related to grid-spacing sensitivity, protein-orientation sensitivity, and mouth-opening ambiguity; (ii) the clustering procedure that leads to the formation of cavities; (iii) and the rank-based filtering criterion for cavities.

# Conclusions

We have here proposed a new grid-and-surface method, called CavVis, to detect cavities on proteins. CavVis builds upon the concept of the field-of-view defined by each Gaussian surface vertex and its normal vector. Essentially, the field-of-view works as a criterion of visibility for each triangle vertex of the triangulated surface. Additionally, the whole process is supported and speeded up using voxel ray casting and back-face culling. That is, culling is used to exclude triangles that are not visible to the surface vertex, yet they are inside its field-of-view.

We also used the clustering of visible surface vertices according to a breadth-first strategy to form vertex clusters and create their convex hulls as tentative cavities, solving the problems of protein-orientation sensitivity, grid-spacing sensitivity, and mouth-opening ambiguity, so common in grid-based methods. Furthermore, we used a statistical analysis methodology (say, precision, recall, and F-score) to compare CavVis to other well-known cavity detection methods, which has shown that it outperforms other benchmarking methods considered in our work.

Summing up, the novelty of CavVis comes from the fact that it employs computer graphics tools to solve the geometric problem of detecting and delineating cavities of proteins. We also show that clustering (of cavity constituents) and filtering (biggest cavities have a higher probability of being binding sites) have a significant impact on the accuracy of cavity detection methods. In fact, accuracy and time performance still are the two main issues in cavity detection methods, particularly in those methods that dynamically detect protein cavities over time (or trajectories).

**Acknowledgement** This research has been partially supported by the Portuguese Research Council (Fundação para a Ciência e Tecnologia), under the doctoral Grant SFRH/BD/99813/2014 and also by FCT Project UID/EEA/50008/2013. Lastly, thank you to reviewers for their reviews and criticism that allowed us to significantly improve this paper.

## References

(1) Du, X.; Li, Y.; Xia, Y.-L.; Ai, S.-M.; Liang, J.; Sang, P.; Ji, X.-L.; Liu, S.-Q. Insights into Protein–Ligand Interactions: Mechanisms, Models, and Methods. *Int. J. Mol. Sci.* **2016**, *17*, 144.

(2) Kuntz, I. D.; Blaney, J. M.; Oatley, S. J.; Langridge, R.; Ferrin, T. E. A Geometric Approach to Macromolecule-ligand Interactions. *J. Mol. Biol.* **1982**, *161*, 269–288.

(3) Shoichet, B. K.; Kuntz, I. D.; Bodian, D. L. Molecular Docking Using Shape Descriptors. *J. Comput. Chem.* **1992**, *13*, 380–397.

(4) Dias, S.; Nguyen, Q. T.; Jorge, J.; Gomes, A. Multi-GPU-Based Detection of Protein Cavities using Critical Points. *Future Gener. Comput. Syst.* **2017**, *67*, 430–440.

(5) Krone, M.; Kozlikova, B.; Lindow, N.; Baaden, M.; Baum, D.; Parulek, J.; Hege, H.-C.; Viola, I. Visual Analysis of Biomolecular Cavities: State of the Art. *Comput. Graph. Forum* **2016**, *35*, 527–551.

(6) Simões, T.; Lopes, D.; Dias, S.; Fernandes, F.; Pereira, J.; Jorge, J.; Bajaj, C.; Gomes, A. Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey. *Comput. Graph. Forum* **2017**, *36*, 643–683.

(7) Kozlíková, B.; Krone, M.; Falk, M.; Lindow, N.; Baaden, M.; Baum, D.; Viola, I.; Parulek, J.; Hege, H.-C. Visualization of Biomolecular Structures: State of the Art Revisited. **2017**, *36*, 178–204.

(8) Kawabata, T.; Go, N. Detection of Pockets on Protein Surfaces Using Small and Large Probe Spheres to Find Putative Ligand Binding Sites. *Proteins: Struct., Funct., Bioinf.* **2007**, *68*, 516–529.

(9) Capra, J. A.; Laskowski, R. A.; Thornton, J. M.; Singh, M.; Funkhouser, T. A. Predicting Protein Ligand Binding Sites by Combining Evolutionary Sequence Conservation and 3D Structure. *PLoS Comput. Biol.* **2009**, *5*, e1000585.

(10) Zhu, H.; Pisabarro, M. T. MSPocket: An Orientation-independent Algorithm for the Detection of Ligand Binding Pockets. *Bioinformatics* **2011**, *27*, 351–358.

(11) Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An Open Source Platform for Ligand Pocket Detection. *BMC Bioinf.* **2009**, *10*, 1–11.

(12) Levitt, D. G.; Banaszak, L. J. POCKET: A Computer Graphic Method for Identifying and Displaying Protein Cavities and Their Surrounding Amino Acids. *J. Mol. Graphics* **1992**, *10*, 229–234.

(13) Hendlich, M.; Rippmann, F.; Barnickel, G. LIGSITE: Automatic and Efficient Detection of Potential Small Molecule-binding Sites in Proteins. *J. Mol. Graphics Modell.* **1997**, *15*, 359–363.

(14) Weisel, M.; Proschak, E.; Schneider, G. PocketPicker: Analysis of Ligand Binding-sites with Shape Descriptors. *Chem. Cent. J.* **2007**, *1*, 1–17.

(15) Kalidas, Y.; Chandra, N. PocketDepth: A New Depth Based Algorithm for Identification of Ligand Binding Sites in Proteins. *J. Struct. Biol.* **2008**, *161*, 31–42.

(16) Tripathi, A.; Kellogg, G. E. A Novel and Efficient Tool for Locating and Characterizing Protein Cavities and Binding Sites. *Proteins: Struct., Funct., Bioinf.* **2010**, *78*, 825–842.

(17) Nayal, M.; Honig, B. On the Nature of Cavities on Protein Surfaces: Application to the Identification of Drug-binding Sites. *Proteins: Struct., Funct., Bioinf.* **2006**, *63*, 892–906.

(18) Coleman, R. G.; Sharp, K. A. Finding and Characterizing Tunnels in Macromolecules with Application to Ion Channels and Pores. *Biophys. J.* **2009**, *96*, 632–645.

(19) Giard, J.; Alface, P. R.; Gala, J.-L.; Macq, B. Fast Surface-based Travel Depth Estimation Algorithm for Macromolecule Surface Shape Description. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2011**, *8*, 59–68.

(20) Hughes, J.; van Dam, A.; McGuire, M.; Sklar, D.; Foley, J.; Feiner, S.; Akeley, K. *Computer Graphics: Principles and Practice*; Addison-Wesley Professional, Upper Saddle River, NJ, USA, 2013.

(21) Amanatides, J.; Woo, A. A Fast Voxel Traversal Algorithm for Ray Tracing. In Eurographics'87. 1987; pp 3–10.

(22) Coleman, R. G.; Sharp, K. A. Travel Depth, a New Shape Descriptor for Macromolecules: Application to Ligand Binding. *J. Mol. Biol.* **2006**, *362*, 441–458.

(23) Coleman, R.; Sharp, K. Protein Pockets: Inventory, Shape, and Comparison. *J. Chem. Inf. Model.* **2010**, *50*, 589–603.

(24) Dias, S. E. D.; Martins, A. M.; Nguyen, Q. T.; Gomes, A. J. GPU-Based Detection of Protein Cavities using Gaussian Surfaces. *BMC Bioinf.* **2017**, *18*, 493:1–493:10.

(25) Dias, S.; Gomes, A. J. In *Computational Electrostatics for Biological Applications*; Rocchia, W., Spagnuolo, M., Eds.; Springer International Publishing, 2015; pp 177–198.

(26) Lorensen, W. E.; Cline, H. E. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Comput. Graph. (ACM)* **1987**, *21*, 163–169.

(27) Bloomenthal, J., Bajaj, C., Binn, J., Cani-Gascuel, M.-P., Rockwood, A., Wyvill, B., Wyvill, G., Eds. *Introduction to Implicit Surfaces*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1997.

(28) Gomes, A.; Voiculescu, I.; Jorge, J.; Wyvill, B.; Galbraith, C. *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*; Springer-Verlag, London, 2009.

(29) Zhang, X.; Bajaj, C. Extraction, Quantification and Visualization of Protein Pockets. *Comput. Syst. Bioinf.* **2007**, *6*, 275–286.

(30) Krone, M.; Reina, G.; Schulz, C.; Kulschewski, T.; Pleiss, J.; Ertl, T. Interactive Extraction and Tracking of Biomolecular Surface Features. *Comput. Graph. Forum* **2013**, *32*, 331–340.

(31) Blinn, J. F. A Generalization of Algebraic Surface Drawing. *ACM Trans. Graph.* **1982**, *1*, 235–256.

(32) Grant, J. A.; Pickup, B. T. A Gaussian Description of Molecular Shape. *J. Phys. Chem.* **1995**, *99*, 3503–3510.

(33) Gabdoulline, R.; Wade, R. Analytically Defined Surfaces to Analyze Molecular Interaction Properties. *J. Mol. Graphics* **1996**, *14*, 341 – 353.

(34) Zhang, Y.; Xu, G.; Bajaj, C. Quality Meshing of Implicit Solvation Models of Biomolecular Structures. *Comput. Aided Geom. Des.* **2006**, *23*, 510 – 530.

(35) Dias, S. E.; Gomes, A. J. Graphics Processing Unit-based Triangulations of Blinn Molecular Surfaces. *Concurr. Comput.* **2011**, *23*, 2280–2291.

(36) Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press, 2004.

(37) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual Molecular Dynamics. *J. Mol. Graphics* **1996**, *14*, 33–38.

(38) Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E. UCSF Chimera – A Visualization System for Exploratory Research and Analysis. *J. Comput. Chem.* **2004**, *25*, 1605–1612.

(39) Dessailly, B. H.; Lensink, M. F.; Orengo, C. A.; Wodak, S. J. LigASite: A Database of Biologically Relevant Binding Sites in Proteins with Known Apo-structures. *Nucleic Acids Res.* **2007**, *36*, D667–D673.

(40) Dias, S. E.; Nguyen, Q. T.; Jorge, J. A.; Gomes, A. J. Multi-GPU-based Detection of Protein Cavities Using Critical Points. *Future Gener. Comput. Syst.* **2017**, *67*, 430–440.

(41) Yu, J.; Zhou, Y.; Tanaka, I.; Yao, M. Roll: A New Algorithm for the Detection of Protein Pockets and Cavities with a Rolling Probe Sphere. *Bioinformatics* **2010**, *26*, 46–52.

(42) Connolly, M. Analytical Molecular Surface Calculation. *J. Appl. Crystallogr.* **1983**, *16*, 548–558.

(43) Brady, G. P.; Stouten, P. F. W. Fast prediction and Visualization of Protein Binding Pockets with PASS. *J. Comput.-Aided Mol. Des.* **2000**, *14*, 383–401.

(44) Laskowski, R. A.; Hutchinson, E. G.; Michie, A. D.; Wallace, A. C.; Jones, M. L.; Thornton, J. M. PDBsum: A Web-based Database of Summaries and Analyses of All PDB Structures. *Trends Biochem. Sci.* **1997**, *22*, 488–490.

# Chapter 4

## CavShape — A Cavity Detection Algorithm Through the Multivariate Shape Diameter Function

This chapter concerns the following article:

CavShape — A Cavity Detection Algorithm Through the Multivariate Shape Diameter Function.
Tiago Simões, and Abel Gomes.

*Article submitted to PLOS Computational Biology.*

### Overview

This chapter proposes a new geometric algorithm called CavShape, which builds upon a mesh segmentation technique introduced in computer graphics, the shape diameter function (SDF). SDF is a shape descriptor to segment a mesh into saliences. This shape descriptor has been modified to detect protein cavities, rather than saliences. The modified SDF is here called *multivariate* shape diameter function (mSDF). CavShape belongs to the category of surface-based methods because it only relies on surface information; that is, no grid information is used to detect cavities on protein surfaces.

It is worth noting that CavShape does not suffer from the problems of grid-spacing ambiguity and protein-orientation sensitivity because it is not based on grids. Furthermore, it solves the issue of mouth-opening ambiguity inherent to surface-based methods because CavShape fills the cavities with spheres that in turn ends up delineating the region of each cavity of the protein surface.

# CavShape – A Cavity Detection Algorithm Through the Multivariate Shape Diameter Function

Tiago Simões[1,2,◑, *], Abel Gomes[1,2,◑, **]

**1,2** Instituto de Telecomunicações, Universidade Beira Interior, Covilhã, Portugal

◑These authors contributed equally to this work.
¤Current Address: Department of Informatics, Universidade da Beira Interior, Covilhã, Portugal
* tiago.simoes@it.ubi.pt, ** agomes@di.ubi.pt

## Abstract

Finding cavities or putative binding sites on protein surfaces can be understood as a mesh segmentation problem as usual in computer graphics and computer vision. Bearing in mind this segmentation idea, we propose an algorithm to segment protein surface based on the concept of shape diameter function (SDF). SDF is a scalar function that measures the diameter of the interior volume of a closed surface in the neighborhood of each one of its points. Interestingly, when applied to Gaussian surfaces, which are modeled by the sum of Gaussian functions that are positive everywhere, SDF ends up measuring the diameter of the exterior volume of the protein exterior, from which we can infer the location and extent of protein cavities. Moreover, unlike other cavity detection methods, SDF is largely independent of pose changes of the protein and holds similar values in separate cavities, allowing us to identify such cavities using clustering of points with similar SDF values.

## Author summary

Tiago M.C. Simões is a PhD student on Informatics Engineering at the University of Beira Interior, Portugal. He received his BSc and MSc degrees in Informatics Engineering from the University of Beira Interior in 2010 and 2012, respectively. He is also a researcher at the Instituto de Telecomunicações, Portugal (as a PhD student) and the University of Beira Interior, Portugal. His current research interests focus on Computer Graphics, Molecular Graphics, and BioInformatics.

## Introduction

Proteins play a role of paramount importance in the living cell. They perform a number of different biological functions as, for example, catalytic reactions, DNA replication, cell signalling, and intracellular transport, as a result of their interactions with nucleotides, other peptides, catalytic substrates, or even man-made chemicals [1]. Such interactions can be classified into protein-ligand, protein-protein, protein-DNA interactions, just to mention a few. It is clear that such interactions take place under complementarity conditions at both chemical and shape levels. However, this paper only addresses the problem of detecting protein cavities (i.e., putative binding sites), which can be understood as a requirement for shape complementarity analysis and molecular docking [2–4].

In the literature, we find three main categories of cavity detection methods: evolution-based, energy-based, and geometry-based. Taking into consideration that we here proposing a geometric method (called CavShape) to detect cavities on protein surfaces, let us briefly review such geometry-based methods. For more thorough reviews, the reader is referred to [5, 6]. Geometric methods divide into the following major classes: grid-based (e.g., ConCavity [7]), sphere-based (e.g., PHECOM [8]), tessellation-based (e.g., Fpocket [9]), surface-based (e.g., MSPocket [10]), as well as hybrid geometric methods that mix two or more of those geometric methods. Let us mention that CavShape is a surface-based method.

Grid-based methods (also called voxelization-based methods) build upon rendering the protein atoms onto an axis-aligned grid, so that we can label grid nodes as 'inside', 'outside', and 'on' the molecular surface, here called interior, exterior, and surface nodes, respectively. The leading idea is then to find linear paths of exterior nodes bracketed by surface nodes to identify and form cavities. Examples of grid-based methods are POCKET [11], LIGSITE [12], PocketPicker [13], PocketDepth [14], and VICE [15]. However, the performance of grid-based methods depends on grid spacing and protein orientation; in other words, the number and location of cavities may vary with changes in grid spacing changes or protein orientation. CavShape overcomes these problems using a pose-independent function on the protein surface, which is called shape diameter function (SDF) [16].

Sphere-based methods build upon the idea of filling in cavities with probe spheres of varying radii. They differ from each other in the way they fill in cavities with probe spheres. Most sphere-based methods take advantage of the principle of rolling probe spheres on surface atoms of the protein according to the three-point Connolly-like sphere geometry [17]. They do not suffer from the drawbacks of grid-based methods mentioned above, with the further advantage of riding off the mouth-opening ambiguity. In fact, in addition to small cavity-filling probe spheres, using a large probe it is feasible to know a cavity where a cavity starts and eventually ends. The principle behind this procedure is that a cavity is a locus where small probes get in, but the large probe does not. However, it is difficult to know in advance which are the adequate radii of the small filling probes and large cavity-blocking probe, because they vary from protein to protein. Examples of sphere-based methods are PASS [18], PHECOM [8] and dPredgeo [19].

Tessellation-based methods are based on computational geometry techniques. We divide them into $\alpha$-shape methods, Voronoi-based methods, and $\beta$-shape methods. Concerning $\alpha$-shape methods, they rely on the Delaunay triangulation of atomic centers. The parameter $\alpha$ determines the carving process of the Delaunay triangulation that leads to the identification of cavities. The main issue is to find the optimal value of $\alpha$, since it varies from protein to protein, not to say from cavity to cavity; hence $\alpha$-shape methods suffer from the same issue as sphere-based methods in respect to the radii of small and large probe spheres. In respect to Voronoi-based methods, they focus on the complement space of the protein, so no carving procedure is not necessary at all. We fill in such complementary space with alpha spheres centered at Voronoi vertices. Most of these methods use the convex hull as the tool to resolve the mouth-opening ambiguity; that is, the convex hull works as a stopgap for pockets. Concerning $\beta$-shape methods, they improve on $\alpha$-shape methods by directly considering van der Waals spheres featuring atoms rather than atomic centers. Also, Apollonius-based methods improve on Voronoi-based methods because they directly use van der Waals spheres instead of atomic centers. Overall, these methods are particularly adequate to identify and retrieve channels and tunnels. Examples of tessellation-based methods are Fpocket [9], VoroProt [20], and BetaVoid [21].

Surface-based methods form the smallest category of cavity detection methods. They rely on the mathematical formulation of molecular surface and its shape invariants. Some of these methods take advantage of two surfaces (an inner and an outer one) to enclose the protein. The inner surface is the natural molecular surface that fits tight to the set of atoms, while the outer surface is fatter to give room for cavities. Cavities are between the inner and outer surfaces; see, for example, SCREEN [22] and Giard et al. [23] for more details. Thus, similar to convex hull, the outer surface works as the stopgap for cavities. Other surface-based methods only use a single surface. For example, MSPocket [10] relies on the solvent-excluded surface (SES) and its triangulation to identify concave vertices that are part of each cavity, as a cavity is then understood as a group of concave triangles. The issue here is that cavities are not only sets of strictly concave triangles because the surface shape of a cavity may oscillate locally between convex and concave triangles. In fact, the main problem of surface-based methods is the lack of region shape invariants to identify cavities of varying sizes, yet there are many local shape invariants like, for example, Gaussian curvature, eigenvalues, and so forth.

Here we propose a new surface-based method, called CavShape, which builds upon a shape invariant known as shape diameter function (SDF). This invariant was introduced by Shapira et al. [16] in the field of computer graphics to segment triangle meshes, and is not sensitive to pose or mouth-opening ambiguity. In practice, we intend to apply such an invariant to protein segmentation into cavities. At our best knowledge, it is the first time that SDF is applied to solve the problem of detecting and identifying cavities on protein surfaces. Traditionally, the SDF algorithm maps inner region volumes by casting a cone of rays from each vertex inwards the surface. Then, we calculate intersection distances for each vertex and their weighted average within a standard deviation of the median of all distances. These weighted average is the SDF value of such vertex. Clustering similar SDF values finally achieve the segmentation of the object. Additionally, CavShape implements changes to the original algorithm to map outer void volumes (i.e., cavities) rather than inner volumes of the protein. Those are regions devoided of molecular mass and highly prone to be known binding sites. Finally, we need to distinguish and delineate cavity regions with relevant SDF values. Specifically, a cavity filling algorithm with spheres and sphere clustering algorithm end up to form cavities on the protein surface.

The remainder of the paper is organized as follows. Section *Background* overviews the background that underpins CavShape. Section *Method* details the steps of CavShape. Section *Testing and Results* presents a benchmark study of CavShape against other cavity detection methods relative to a ground-truth dataset of well-known binding sites. Finally, Section *Conclusion and Future Work* presents the conclusions and future work.

# Background

## Protein Surface

In the literature, we find mainly three mathematical mathematical formulations, namely: van der Walls surface [24, 25], solvent-accessible surface (SAS) [26], and solvent-excluded surface (SES) [27]. As known, SES is the one that better represents the molecular surface. In the present article, we use the Gaussian surface (also called sum of Gaussians) that more accurately approximates the SES. It is defined by the implicit function $f : \mathbb{R}^3 \to \mathbb{R}$ as follows:

$$f(\mathbf{p}) = \sum_{i=1}^{n} e^{D_i(\frac{\|\mathbf{p}-\mathbf{p}_i\|^2}{r_i^2}-1)} \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^3$, $\|\mathbf{p} - \mathbf{p}_i\|$ the distance from the point $\mathbf{p}$ to the center $\mathbf{p}_i$ of the $i$-th atom, $r_i$ the radius of the atom $i$, and finally $D_i$ is the decay rate of the Gaussian kernel tied to atom $i$. We use the value $D_i = 2.35$ that guarantees the better approximation to SES [28–31]. Note that $f$ represents the global electron density at an arbitrary point $\mathbf{p}$ as the sum of local electron densities associated to atoms $i$ at $\mathbf{p}$. Note that this Gaussian surface that approximates SES was also adopted by other authors, namely Zhang and Bajaj [32], Giard et al. [23], Krone et al. [33], and Dias et al. [4].

## Surface Triangulation

There are many algorithms to triangulate a molecular surface, depending on the mathematical formulation of the surface itself. That is, the triangulation depends of whether the surface is defined in the parametric form, implicit form, or else (see [34] for further details). The Gaussian surface defined above is in the implicit form, so that any triangulation algorithm for implicit surfaces applies, namely continuation methods (see, for example, [35–37]) and space partitioning methods (see, for example, [38–41]). In the present article, we use the modified marching cubes algorithm due to Dias and Gomes [42, 43], which belongs to the category of space partitioning methods. Recall that the marching cubes algorithm was originally proposed by Lorensen and Cline [38].

## Shape Diameter Function

The shape diameter function (SDF) was originally introduced by Shapira et al. [16] as a shape descriptor for mesh skeletonization and segmentation in computer graphics. Let $M$ be a mesh surface (manifold) enclosing a protein. We define a scalar function $d : M \to \mathbb{R}$, here called the shape diameter function (SDF), as the neighborhood diameter of the protein at the arbitrary point $\mathbf{p} \in M$. This diameter at $\mathbf{p}$ is given by the distance to its antipodal surface point in the inward-normal direction, as illustrated in Fig.1(a). Note that it is assumed that the normal vector at $\mathbf{p}$ is pointing outwards the surface.

However, taking into consideration that we are using a piecewise linear mesh, we cannot define exactly the antipodal point. To overcome this problem, we use a cone at $\mathbf{p}$ with its axis aligned with the inward-normal direction (the symmetric normal vector), triggering several cone rays inwards (Fig.1(a)). The shape diameter function at $\mathbf{p}$ is then given by the weighted average of all rays lengths falling within the standard deviation relative to the length median. But, considering that there are more rays with large angles than rays with small angles, it is necessary to use weights to equalize the rays within the cone; specifically, the weight of a ray is the inverse of the angle between the ray and cone axis. Thus, the SDF generates a scalar field on the surface. This scalar field allows for the segmentation of the mesh surface $M$ into surface protrusions $\{P_i\}$ and their complement $\overline{M}$ in $M$, that is $M = \{P_i\} \cup \overline{M}$.

Therefore, the SDF-based segmentation mainly finds protrusive regions, not cavities. In fact, SDF was designed to segment articulated and tubular models, which are common place in computer graphics. Recall that the diameter of a surface point measures the weighted average of ray lengths starting at a point to several anti-podal surface points across the interior of the model. It is also assumed that the normal vector at the centroid of each triangle is pointing to outside of the surface.

However, in the particular case of the Gaussian surface here used to model protein surfaces, the normal vectors point to inside the surface in the direction of the steepest ascent of the gradient because the Gaussian function is positive everywhere taking a local maximum at each atom center. Consequently, and taking into consideration that the SDF is computed in the opposite direction to gradient normal at each triangle centroid (see Fig.1(b)), we conclude that SDF induces a segmentation of the mesh surface $M$ into surface cavities $\{C_i\}$ and their complement $\overline{M}$ in $M$, that is $M = \{C_i\} \cup \overline{M}$. This illustrated in Fig. 2, where we observe that cavities of three proteins can be identified by yellow-to-red regions. However, their histograms are bimodal (one mode for protrusive

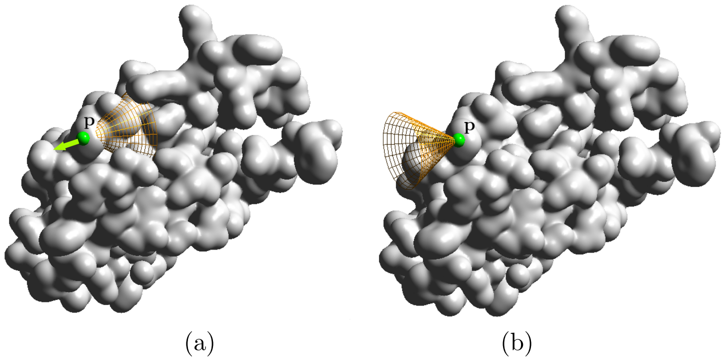


**Fig 1.** Ray casting direction is opposite to the cone: (a) the normal vector (gradient vector) is pointing outwards as usual for most surfaces; (b) the normal vector (gradient vector) is pointing inwards in the case of Gaussian surfaces.

regions and another mode for depressive regions), so we cannot distinguish cavities from one another. This means that we cannot use a Gaussian mixture model (GMM) to fit $k$ Gaussians to the histogram of SDF values of the mesh triangles to find cavities of the protein surface, as proposed by Shapira et al. [16]. Such fitting could be achieved using, for example, the expectation-maximization (EM) algorithm [44], but, as shown in Fig. 2, the SDF-based segmentation is useless for molecular surface segmentation. To overcome this problem, we modified the original SDF algorithm into two main aspects:

- We changed the univariate shape diameter function (SDF) to a *multivariate shape diameter function* (mSDF). This means that we no longer use one weighted diameter per triangle centroid, but a chunk of diameters per triangle centroid, each concerning a distinct ray of the cone. Each raw diameter forms a feature in the space of parameters of each triangle centroid.
- We use the DBSCAN clustering algorithm to identify the cavities in the space of parameters.

## Method

The main steps of CavShape are the following:

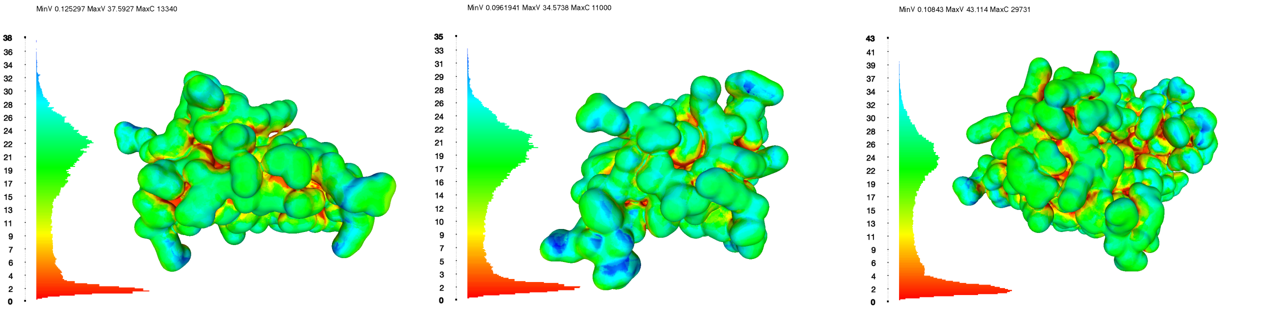


**Fig 2.** Gaussian molecular surfaces for proteins 4PTI, 2RK2, and 1FKD represented with their SDF scalar fields and their histograms. Images generated by MeshLab [45].

1. *Surface triangulation.* This step can be done using marching cubes algorithm [38] or any other triangulation algorithm (e.g., [37]).

2. *Multivariate shape diameter function.* Second, one computes the multivariate shape diameter function at each triangle centroid.

3. *Diameter clustering.* Third, one constructs the clusters of diameters per triangle centroid. If only one cluster is formed, then the centroid belongs to a cavity; otherwise, the centroid does not belong to any cavity. The admissible maximum diameter difference (or threshold) in a cluster is 2.8 Å, which is the diameter of water molecule approximately. The diameter clustering is performed using a multi-core DBSCAN [46].

4. *Filling cavities with spheres.* Fourth, one places spheres along rays connecting each point to its antipodal points of each cluster. This allows us to populate cavities with spheres over the molecular surface, though cavities are not apart yet.

5. *Sphere clustering.* The filling spheres are grouped into different clusters. The sphere clustering is performed using a multi-core DBSCAN mentioned above. The diameter of each filling sphere is $d$ =0.5 Å, while the distance threshold ($\tau > d$) is 1.25 Å.

These sphere clusters constitute the cavities of the protein. However, only the ten larger cavities are considered for benchmarking (see Section *Testing and Results*). The flow diagram of the CavShape is shown in Fig. 3. The next subsections detail the steps of CavShape.

## Surface triangulation

Surface triangulation is the first step of CavShape, which outputs a triangle mesh for the molecular surface. In the present implementation, we use a variant of the marching cubes algorithm for surface triangulation [38], with the grid spacing of $\Delta = 0.6$ Å. However, it is feasible to use other triangulation algorithms [37, 39–41]. This means that CavShape is not a grid-based method. In fact, CavShape is a surface-based method that exclusively depends on the surface properties of the protein.

## Multivariate SDF

Let us refer that the computation of the multivariate shape diameter function (mSDF) at each triangle centroid is independent of the protein pose. That is, and unlike other methods, CavShape's output does not depend on the protein pose in 3D Euclidean space. Basically, several rays (say, $k$ rays) are casted from each triangle centroid to intersect

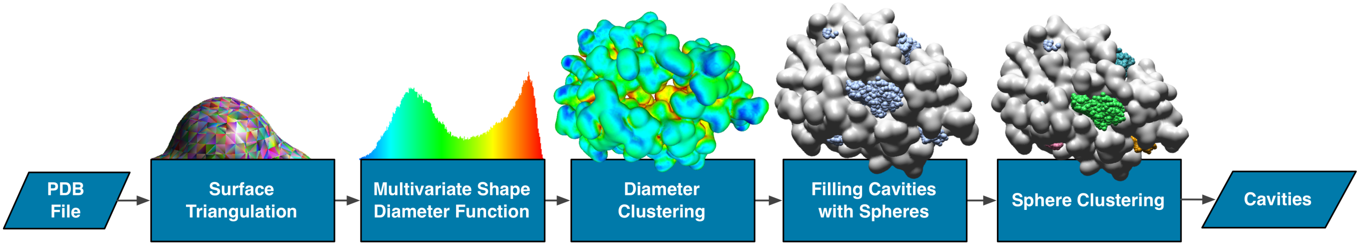
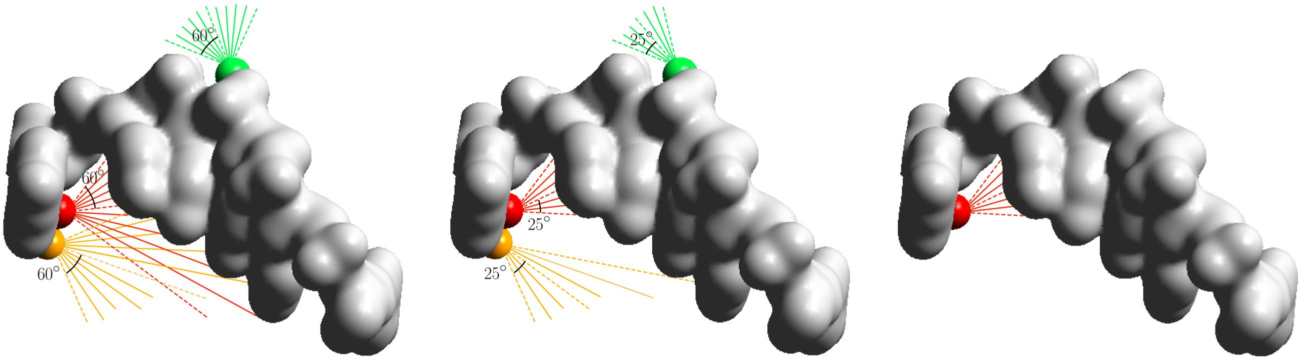


**Fig 3.** CavShape's flow diagram that illustrates the main steps of the algorithm.



(a) (b) (c)

**Fig 4.** Casting rays casted from three triangle centroids represented as red, green and yellow spheres: (a) using a cone with an angle of view of 60° (as in the traditional SDF); (b) using a cone with an angle of view of 25° (as in our multivariate SDF); (c) the resulting cone of rays for a cavity using our multivariate SDF after discarding the yellow and green cones because some their rays do not intersect the surface mesh (i.e. infinite diameters).

other mesh triangles, as illustrated in Fig. 1(a). Each ray is assigned a SDF value, which is the distance from its centroid to the intersection point. In practice, we end up having $k$ SDF values per centroid; hence, the *multivariate* SDF. The rays casted from each centroid are distributed uniformly within a cone whose angle of view $\alpha$ was empirically determined as 25°(Fig. 4(b)). The cone axis is aligned with the surface normal at the centroid. Such empirical angle of view is justified by the fact that smaller angle values do not allow for an unambiguous discrimination between different mesh regions, because the sensitivity to local mesh features is more noticeable. Conversely, using larger angle values (i.e., close to a maximum of 90°) may lead to erroneous results because rays may overflow to other regions of the mesh [16], as illustrated in Fig. 4(a).

The ray-triangle intersection is central to this procedure of computing the multivariate SDF at each centroid. Similar to Shapira et al. [16], such ray-triangle intersections are speeded up using an accelerated spatial data structure called octree. To this end, we first determine the octant per ray that satisfies the following conditions: (i) the ray intersects the octant; (ii) the octant at least encloses a triangle; (iii) the octant is the closest one to ray's centroid. This procedure is similar to voxel ray tracing optimized by Amanatides and Woo [47]. Rays that do not intersect the mesh have infinite diameter.

## Diameter Clustering

Unlike Shapira et. al [16], we do not use a single SDF values at each centroid to cluster mesh vertices. Instead, we use the clustering of ray diameters per triangle centroid. This clustering task is performed using a CPU multi-threading implementation of the DBSCAN algorithm proposed by Patwary et. al [46]. Two diameters belong to the same cluster if their difference $\epsilon$ is less or equal to 2.8 Å (i.e. diameter of the water molecule). If all $k$ diameters associated to each triangle centroid form a single cluster of bounded diameters (i.e., finite extension diameters) (red centroid in Fig. 4(c)), the triangle is labelled as belonging to a cavity; otherwise, it is not. However, a triangle with a single cluster of unbounded diameters (i.e., infinite extension diameters) is thrown away because the rays do not intersect the mesh (green centroid in Fig. 4(b)). Also, every triangle with two or more diameter clusters is discarded (yellow centroid in Fig. 4(b)). This is so because at least one ray of the cluster overflows to remote regions of the mesh.

### Filling cavities with spheres

After identifying single-cluster centroids of bounded diameters, each ray segment is populated with a sequence of tangential spheres of radius 0.5 Å. As a result, we end up filling protein cavities with spheres. This radius of 0.5 Å is much less that water molecule radius (i.e., 1.4 Å) to guarantee that a suited packing of spheres inside each cavity; that is, the volume occupied by the spheres converges to the cavity volume when the sphere radius tends to zero.

### Sphere clustering

With the spheres in place, it remains to cluster them into cavities. For that purpose, we use the DBSCAN algorithm [46], which was also used for diameter clustering (see Section *Diameter Clustering*). However, now the distance threshold is 1.25 Å, so that we have a maximum space clearance of 0.25 Å between spheres belonging to the same cluster because each filling sphere radius is equal to 0.5 Å. Finally, only the top-10 cavities in terms of volume are outputted by the algorithm as putative binding sites. Fig. 5 shows four proteins and their cavities as generated by CavShape.

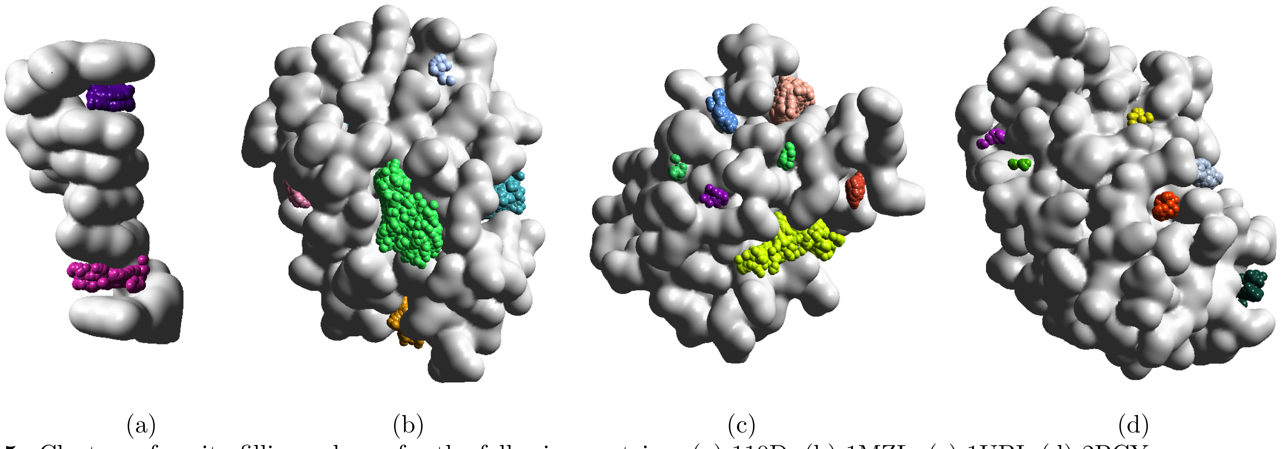
## Testing and Results

For testing and comparison purposes, we used an Apple iMac desktop computer manufactured with a 3.2 GHz Intel Core i5, 16 GB RAM, and a NVIDIA GeForce GT 755M. This computer runs the OS X Yosemite operating system. CavShape is bundled with a molecular visualization module, which is built upon the OpenGL 2.0 graphics system. It is also worth noting that CavShape and its competing benchmark methods were all run on the desktop computer mentioned above. The C++ version of CavShape is publicly available at `https://github.com/MediaLabProjects/CavShape/`.

### Benchmark methods

We compared CavShape with the following open-source cavity detection methods:

- *Fpocket.* This Voronoi tessellation-based method was introduced Guilloux et al. [9], and makes also use of the theory of $\alpha$-spheres.



(a) (b) (c) (d)

**Fig 5.** Clusters of cavity-filling spheres for the following proteins: (a) 110D; (b) 1MZL; (c) 1UBI; (d) 2PCY.

- *MSPocket.* Zhu and Pisabarro [10] proposed this surface-based method. It uses normal-based curvature information around each surface point to classify and merge a set of concave connected regions into a protein cavity.

- *GHECOM.* Kawabata [8] introduced this grid-and-sphere method. As such, it employs a 3D grid that embeds the protein, taking advantage of mathematical morphology theory and spherical probes with multi sizes to detect cavities on the protein surface. In this method, a cavity is defined as the space that is empty on the outside of the protein. Specifically, where a small probe can enter but a larger one can't.

- *POCASA.* Yu et al. [48] put forward this grid-and-sphere algorithm. Firstly, one adopts a boolean scalar field applied to grid nodes to put apart the interior and exterior grid nodes that are protein's surface relative. Secondly, one identifies a cavity as the empty outer space whose grid nodes are not absorbed by a rolling large probe sphere.

- *ConCavity.* Capra et al. [7] introduced this grid-based method that combines the advantages of structure-based and sequence-based methods. This method scores the grid nodes of cavities produced by any grid-based cavity method as, for example, PocketFinder [49]; the scoring procedure assigns sequence conservation values of residues to nearby grid nodes that are associated to cavities.

- *PASS.* Brady Jr. and Stouten [18] proposed this sphere-based method. Filling cavities with probe spheres is performed according to the three-point Connolly-like sphere geometry [17], though only low-solvent exposure probes are taken into account in the formation of cavities.

## Ground Truth

For testing and benchmarking purposes, we used a ground-truth dataset of binding sites which is a subset of both PDBSum (`www.ebi.ac.uk/pdbsum`) and LigASite database (`ligasite.org/`); see Laskowski et al. [50] and Dessailly et al. [51] for further details about PDBsum and LigASite, respectively. Concretely, our ground truth comprises 1239 proteins (335 apo proteins and 904 holo proteins). Similar to PDBsum, CavShape only outputs the top-10 largest cavities. The ten cavities of each protein are sorted in terms of volume, using an order of decreasing volume. Therefore, CavShape only considers 12390 cavities concerning 1239 proteins.

## Performance metrics

The comparative performance analysis of CavShape with the methods mentioned above based itself on the concept of *overlapping* between each method-specific cavity $c_i$ and each ground-truth cavity $C_j$. Note that each ground-truth cavity is a known binding site, that is, a locus on the protein surface that binds to an already known ligand. The performance analysis is based on the matching between binding sites (or ground-truth cavities) and cavities found by different methods (i.e., method-specific cavities). Thus, we consider that a cavity $c_i$ is a hit if $c_i \cap C_j \neq \varnothing$. This allows us to come up with the number of true positives (TP), false positives (FP), and false negatives (FN).

In fact, a hit is nothing more than a true positive. In turn, a false positive represents cavities outputted by a method that does not intersect any ground truth cavity. Conversely, a false negative represents a ground-truth cavity that does not overlap any method-specific cavity. Therefore, the performance analysis can be performed using the well-known metrics like precision

$$P = \frac{TP}{TP + FP} \tag{2}$$

and recall

$$R = \frac{TP}{TP + FN}. \tag{3}$$

Also, according to statistical analysis, we can use the F-score metric (the harmonic mean of precision and recall)

$$F = 2.\frac{P.R}{P + R} \tag{4}$$

to rank the cavity detection methods above, since high values of $P$ and $R$ lead to a high value of F-score. The performance analysis results are shown in Table 1 and Figure 6. We observe that CavShape performs better than any other method considered in our comparative study because its F-score stands over any other.

**Table 1.** Benchmark study for ground truth apo and holo protein structures.

| METHOD | APOS | | | | | | HOLOS | | | | | | APOS and HOLOS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $TP$ | $FP$ | $FN$ | $\approx P$ | $\approx R$ | $\approx F$ | $TP$ | $FP$ | $FN$ | $\approx P$ | $\approx R$ | $\approx F$ | $\approx P$ | $\approx R$ | $\approx F$ |
| CavShape | 1827 | 1376 | 1523 | 57% | 55% | 56% | 5073 | 3556 | 3967 | 59% | 56% | 57% | 58% | 56% | 57% |
| Fpocket | 2040 | 2754 | 1310 | 43% | 61% | 50% | 5259 | 7161 | 3781 | 42% | 58% | 49% | 42% | 59% | 49% |
| MSPocket | 2239 | 4096 | 1111 | 35% | 67% | 46% | 5929 | 10558 | 3101 | 36% | 66% | 46% | 36% | 66% | 46% |
| GHECOM | 2509 | 5087 | 841 | 33% | 75% | 46% | 6632 | 12883 | 2408 | 34% | 73% | 46% | 34% | 74% | 46% |
| POCASA | 1358 | 1804 | 1992 | 43% | 41% | 42% | 3568 | 4571 | 5472 | 44% | 39% | 42% | 44% | 40% | 42% |
| Concavity | 695 | 356 | 2655 | 66% | 21% | 32% | 1786 | 833 | 7254 | 68% | 20% | 31% | 68% | 20% | 31% |
| PASS | 1018 | 2669 | 2332 | 28% | 30% | 29% | 2616 | 6759 | 6250 | 28% | 30% | 29% | 28% | 30% | 29% |

## Discussion

Looking at the results listed in Table 1 and the chart in Fig. 6, we note that CavShape ranks first with a F-score value of 56% and 57% for apo and holo proteins, resulting

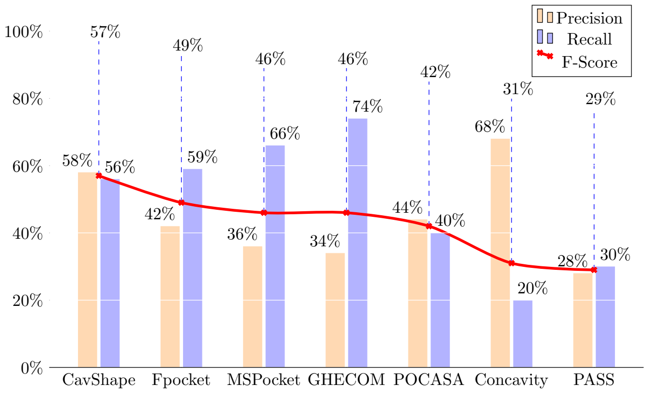


**Fig 6.** Precision, recall, and F-Score percentages for 1239 proteins.

in a combined F-score value of 57%. Interestingly, this is so even considering that CavShape does not score first in terms of precision and recall separately. In fact, ConCavity ranks first concerning precision (68%), while GHECOM ranks first concerning recall (74%). Nevertheless, ConCavity's recall is too low (20%) as a consequence of its exaggerated number of false negatives; that is, ConCavity often misses ground-truth cavities. Conversely, GHECOM's precision because it produces many false positives (or fake cavities). In other words, if the precision and recall are not balanced, the corresponding the F-score decreases too much because it is given by the harmonic mean of precision and recall.

## Conclusion and Future Work

We have introduced a new surface-based method for the detection of protein cavities, called CavShape. It builds upon the definition of shape diameter function (SDF), which was previously introduced by Shapira et al. [16] in the field of computer graphics. No grid or rolling sphere probes are used at all. Recall that CavShape requires the preliminary triangulation of the molecular surface, here defined as the Gaussian surface that better approximates the solvent-excluded surface (SES), yet other sort of molecular surface might be used. Cavities are filled by tangential spheres placed along each diameter, which are then grouped together into clusters according to a proximity criterion. In addition, we used the performance metrics of precision, recall, and F-score to benchmark CavShape against state-of-the-art cavity detection algorithms. The results obtained lead us to the fact that CavShape outperforms such methods.

## Supporting information

**Software S1 CavShape Source Code.** CavShape was encoded in C++ and is publicly available at `https://github.com/MediaLabProjects/CavShape/`.

## Acknowledgments

This research has been partially supported by the Portuguese Research Council (Fundação para a Ciência e Tecnologia), under the doctoral Grant SFRH/BD/99813/2014. Last but not least, we would like to thank to all reviewers that contributed to improving this paper.

## References

1. Du X, Li Y, Xia YL, Ai SM, Liang J, Sang P, et al. Insights into protein–ligand interactions: mechanisms, models, and methods. International Journal of Molecular Sciences. 2016;17(2):144.

2. Kuntz ID, Blaney JM, Oatley SJ, Langridge R, Ferrin TE. A geometric approach to macromolecule-ligand interactions. Journal of Molecular Biology. 1982;161(2):269–288.

3. Shoichet BK, Kuntz ID, Bodian DL. Molecular docking using shape descriptors. Journal of Computational Chemistry. 1992;13(3):380–397.

4. Dias S, Nguyen QT, Jorge J, Gomes A. Multi-GPU-Based Detection of Protein Cavities using Critical Points. Future Generation Computer Systems. 2017;67:430–440.

5. Krone M, Kozlikova B, Lindow N, Baaden M, Baum D, Parulek J, et al. Visual analysis of biomolecular cavities: state of the art. Computer Graphics Forum. 2016;35(3):527–551.

6. Simões T, Lopes D, Dias S, Fernandes F, Pereira Ja, Jorge J, et al. Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: a Survey. Computer Graphics Forum. 2017;36(8):643–683.

7. Capra J, Laskowski R, Thornton J, Singh M, Funkhouser T. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. PLoS Computational Biology. 2009;5(12):e1000585:1–18.

8. Kawabata T, Go N. Detection of pockets on protein surfaces using small and large probe spheres to find putative ligand binding sites. Proteins: Structure, Function, and Bioinformatics. 2007;68(2):516–529.

9. Le Guilloux V, Schmidtke P, Tuffery P. Fpocket: an open source platform for ligand pocket detection. BMC Bioinformatics. 2009;10(1):1–11.

10. Zhu H, Pisabarro MT. MSPocket: an orientation-independent algorithm for the detection of ligand binding pockets. Bioinformatics. 2011;27(3):351–358.

11. Levitt DG, Banaszak LJ. POCKET: A computer graphic method for identifying and displaying protein cavities and their surrounding amino acids. Journal of Molecular Graphics. 1992;10(4):229–234.

12. Hendlich M, Rippmann F, Barnickel G. LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. Journal of Molecular Graphics and Modelling. 1997;15(6):359–363.

13. Weisel M, Proschak E, Schneider G. PocketPicker: analysis of ligand binding-sites with shape descriptors. Chemistry Central Journal. 2007;1(1):1–17.

14. Kalidas Y, Chandra N. PocketDepth: A new depth based algorithm for identification of ligand binding sites in proteins. Journal of Structural Biology. 2008;161(1):31–42.

15. Tripathi A, Kellogg GE. A novel and efficient tool for locating and characterizing protein cavities and binding sites. Proteins: Structure, Function, and Bioinformatics. 2010;78(4):825–842.

16. Shapira L, Shamir A, Cohen-Or D. Consistent mesh partitioning and skeletonisation using the shape diameter function. The Visual Computer. 2008;24(4):249.

17. Connolly M. Analytical Molecular Surface Calculation. Journal of Applied Crystallography. 1983;16(5):548–558.

18. Brady GP, Stouten PFW. Fast prediction and visualization of protein binding pockets with PASS. Journal of Computer-Aided Molecular Design. 2000;14(4):383–401.

19. Schneider S, Zacharias M. Combining geometric pocket detection and desolvation properties to detect putative ligand binding sites on proteins. Journal of Structural Biology. 2012;180(3):546–550.

20. Olechnovič K, Margelevičius M, Venclovas Č. Voroprot: an interactive tool for the analysis and visualization of complex geometric features of protein structure. Bioinformatics. 2011;27(5):723–724.

21. Kim JK, Cho Y, Laskowski RA, Ryu SE, Sugihara K, Kim DS. BetaVoid: Molecular voids via beta-complexes and Voronoi diagrams. Proteins: Structure, Function, and Bioinformatics. 2014;82(9):1829–1849.

22. Nayal M, Honig B. On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. Proteins: Structure, Function, and Bioinformatics. 2006;63(4):892–906.

23. Giard J, Alface PR, Gala JL, Macq B. Fast surface-based travel depth estimation algorithm for macromolecule surface shape description. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2011;8(1):59–68.

24. French AS. Computer simulation of space-filling molecular models. IEEE Transactions on Computers. 1977;(10):1026–1028.

25. Porter TK. The shaded surface display of large molecules. ACM SIGGRAPH Computer Graphics. 1979;13(2):234–236.

26. Lee B, Richards FM. The interpretation of protein structures: estimation of static accessibility. Journal of Molecular Biology. 1971;55(3):379–400.

27. Richards FM. Areas, volumes, packing, and protein structure. Annual Review of Biophysics and Bioengineering. 1977;6(1):151–176.

28. Blinn JF. A generalization of algebraic surface drawing. ACM Transactions on Graphics. 1982;1(3):235–256.

29. Grant JA, Pickup B. A Gaussian description of molecular shape. The Journal of Physical Chemistry. 1995;99(11):3503–3510.

30. Gabdoulline RR, Wade RC. Analytically defined surfaces to analyze molecular interaction properties. Journal of Molecular Graphics. 1996;14(6):341–353.

31. Zhang Y, Xu G, Bajaj C. Quality meshing of implicit solvation models of biomolecular structures. Computer Aided Geometric Design. 2006;23(6):510–530.

32. Zhang X, Bajaj C. Extraction, quantification and visualization of protein pockets. In: Proceedings of the LSS Computer Systems Bioinformatics Conference *(CSB'07), University of California at San Diego, August 13-17.* vol. 6. Life Sciences Society; 2007. p. 275–286.

33. Krone M, Reina G, Schulz C, Kulschewski T, Pleiss J, Ertl T. Interactive Extraction and Tracking of Biomolecular Surface Features. Computer Graphics Forum. 2013;32(3):331–340.

34. Gomes AJP, Voiculescu I, Jorge J, Wyvill B, Galbraith C. Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms. Springer-Verlag, London; 2009.

35. Rheinboldt W. On a Moving Frame Algorithm and the Triangulation of Equilibirum Manifolds. In: Kuper T, Seydel R, Troger H, editors. Bifurcation: Analysis, Algorithms, Applications. vol. 79 of International Series of Numerical Mathematics. Boston: Birkhauser; 1987. p. 256–267.

36. Hartmann E. A Marching Method for the Triangulation of Surfaces. The Visual Computer. 1998;14(3):95–108.

37. Raposo A, Gomes A. Polygonization of Multi-Component Non-Manifold Implicit Surfaces through a Symbolic-Numerical Continuation Algorithm. In: Lee YT, Shamsuddin SM, editors. Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and South-East Asia *(GRAPHITE'06)*. ACM Press; 2006. p. 399–406.

38. Lorensen WE, Cline HE. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. ACM SIGGRAPH Computer Graphics. 1987;21(4):163–169.

39. Bloomenthal J. Polygonization of implicit surfaces. Computer-Aided Geometric Design. 1988;5(4):341–355.

40. Suffern K. An octree algorithm for displaying implicitly defined mathematical functions. The Australian Computer Journal. 1990;22(1):2–10.

41. Suffern K, Balsys R. Rendering the intersections of implicit surfaces. IEEE Computer Graphics & Applications. 2003;23(5):70–77.

42. Dias SED, Gomes AJP. Graphics processing unit-based triangulations of Blinn molecular surfaces. Concurrency and Computation: Practice and Experience. 2011;23(17):2280–2291.

43. Dias S, Gomes AJP. Triangulating Gaussian-like Surfaces of Molecules with Millions of Atoms. In: Rocchia W, Spagnuolo M, editors. Computational Electrostatics for Biological Applications. Springer International Publishing; 2015. p. 177–198.

44. Dempster AP, Laird NM, Rubin DB. Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B. 1977;39(1):1–38.

45. Cignoni P, Callieri M, Corsini M, Dellepiane M, Ganovelli F, Ranzuglia G. Meshlab: an open-source mesh processing tool. In: Proceedings of the Eurographics Italian Chapter Conference. The Eurographics Association; 2008. p. 129–136.

46. Patwary MA, Palsetia D, Agrawal A, Liao Wk, Manne F, Choudhary A. A new scalable parallel DBSCAN algorithm using the disjoint-set data structure. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis *(SC'12)*. IEEE Computer Society Press; 2012. p. 62:1–62:11.

47. Amanatides J, Woo A. A fast voxel traversal algorithm for ray tracing. In: Maréchal G, editor. Proceedings of the European Computer Graphics Conference and Exhibition *(EG'87)*. North-Holland; 1987. p. 3–10.

48. Yu J, Zhou Y, Tanaka I, Yao M. Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. Bioinformatics. 2010;26(1):46–52.

49. An J, Totrov M, Abagyan R. Pocketome via comprehensive identification and classification of ligand binding envelopes. Molecular & Cellular Proteomics. 2005;4(6):752–761.

50. Laskowski RA, Hutchinson EG, Michie AD, Wallace AC, Jones ML, Thornton JM. PDBsum: a Web-based database of summaries and analyses of all PDB structures. Trends in Biochemical Sciences. 1997;22(12):488–490.

51. Dessailly BH, Lensink MF, Orengo CA, Wodak SJ. LigASite: a database of biologically relevant binding sites in proteins with known apo-structures. Nucleic Acids Research. 2007;36:D667–D673.

# Chapter 5

## CavSeeker — Identifying Protein Cavities by Locating Critical Points of the Scalar Field Generated from Summation of Gaussians

This chapter concerns the following article:

CavSeeker — Identifying Protein Cavities by Locating Critical Points of the Scalar Field Generated from Summation of Gaussians.
Tiago Simões, and Abel Gomes.

*Article submitted to Bioinformatics.*

### Overview

This chapter proposes a new geometric algorithm called CavSeeker. The method builds upon the analytical formulation of the Gaussian scalar field to detect cavities. CavSeeker carries out an eigenvalue analysis to identify minima and saddle points in the range $[0.001, 1.0]$ of the Gaussian scalar field of the protein. These points are of particular interest to distinguish cavity regions such as voids, exposed cavities, or tunnels. Note that CavSeeker does not consider maxima because they are inside the protein. CavSeeker belongs to the family of grid-based methods because it uses a scalar field in conjunction with the grid of voxels.

As in previous methods described before, CavSeeker is capable of solving the issues typically associated with grid-based methods (see Chapter 2 for more details). Specifically, the location of each critical point does not depend on the grid-spacing value or protein-orientation. Finally, the scalar field range solves the problem concerning mouth-opening ambiguity. Indeed, such range acts as a stopgap for cavity mouth openings.

*Bioinformatics*
doi.10.1093/bioinformatics/xxxxxx

Structural Bioinformatics

# CavSeeker - Identifying Protein Cavities by Locating Critical Points of the Scalar Field Generated from Summation of Gaussians

**Tiago Simões [1,2] and Abel Gomes [1,2*]**

[1] Instituto de Telecomunicações, Portugal.
[2] Universidade da Beira Interior, Portugal.

* To whom correspondence should be addressed.

## Abstract

**Motivation:** Many algorithms to find protein cavities (or tentative binding sites) have been proposed in the literature for the last two to three decades. However, only a few methods have taken advantage of the critical points of the electron density field of each protein. CavSeeker, the protein cavity detection method here introduced, develops from the leading idea that such cavities are located around specific critical points of the electron density field of the protein. Basically, CavSeeker finds the voxels that are transverse to two iso-surfaces of the electron density field of the protein, between which one can find the critical points and their corresponding cavities. CavSeeker belongs thus to the category of grid-based methods.
**Results:** The accuracy of CavSeeker in finding protein cavities was evaluated using a benchmark suite that includes other methods and a ground-truth dataset of binding sites. Our experimental analysis has shown that CavSeeker seamlessly outperforms other competitor methods.
**Availability:** CavSeeker is publicly accessible at `https://github.com/MediaLabProjects/CavSeeker`.
**Contact:** agomes@di.ubi.pt
**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 Introduction

Several predictive methods to detect protein cavities have been introduced in the literature for the last three decades. These methods are relevant in molecular bioinformatics because protein cavities are tentative sites to where ligands and other proteins may bind. Therefore, they play an essential role in the preliminary stages of studies addressing protein-ligand interactions, protein-protein interactions, molecular docking and the like.

In general, cavity predictive methods fall into three major categories: evolutionary-based, energy-based, and geometry-based Simões *et al.* (2017). Evolutionary-based algorithms are dependent on a sequence alignment procedure to predict binding sites. Energy-based methods rely on the identification of energies underlying interactions between the protein and the binding molecule. Typically, a energy-based test is performed between the set of atoms of the protein and a small probe. Finally, geometry-based methods take advantage of some mathematical description of the protein shape to detect cavities. The main classes of geometry-based methods are the following: sphere-based, tessellation-based, surface-based, and grid-based. Sphere-based methods essentially search for sites (cavities) where a small probe (with water molecule radius of 1.4Å) can enter, but not a larger probe (e.g., 5Å). Tessellation-based methods are computational geometry methods, and can be thus understood as a mathematical formulation for methods based on spheres, particularly after the emergence of alpha shapes (Edelsbrunner *et al.* (1983)), which have led to the development of Voronoi diagram-based methods to identify cavities and pockets of proteins (e.g., Fpocket Le Guilloux *et al.* (2009)).

In turn, surface-based methods build upon the properties (e.g., curvature) of some mathematical formulation of surface for proteins in order to find their cavities. Cazals *et al.* (2003) used the Connolly function to break down the molecular surface into several patches (i.e., convex, concave, and saddle), much like in part segmentation algorithms used in computer graphics (see Rodrigues *et al.* (2018)). However, the resulting surface segmentation does not match the coarse segmentation of the surface into cavities. Natarajan *et al.* (2006) tried to solve this problem through simplification techniques, trying to merge small segments into larger ones, but it remains unclear whether these larger segments correspond to binding sites. Exner *et al.* (2002) used the concept of global curvature (previously

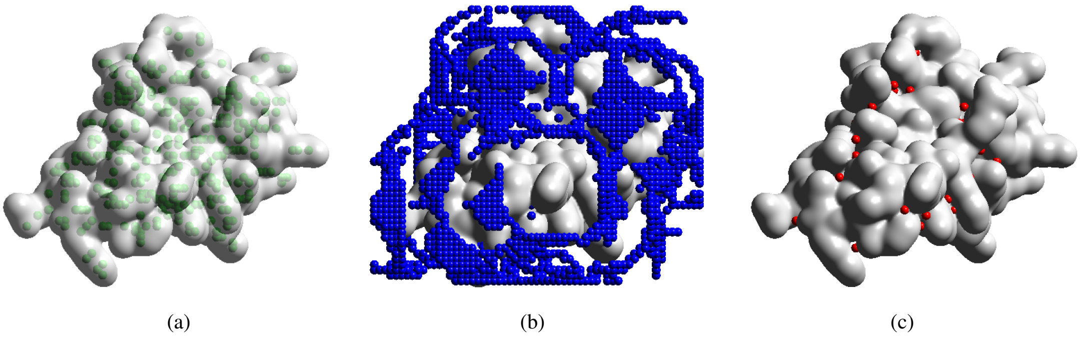


Fig. 1: Critical points for the protein 4PTI: (a) maxima in smud green; (b) minima in blue; (c) saddle points in red.

proposed by Zachmann *et al.* (1992)) to describe such large surface segments, but there is no evidence about the correspondence between such segments and binding sites. Dias *et al.* (2017) proposed a cavity detection method based on the curvature analysis, but such analysis was limited to voxels that intersect the molecular surface. On the contrary, we use spectral analysis of the scalar field on the set of voxels outside the protein, so we get in principle better results in predicting binding sites and their locations.

CavSeeker is a grid-based method to predict binding sites of proteins because it builds upon two main mathematical tools: (i) grid of voxels that embeds the protein; (ii) an electron density scalar field that results from the summation of subsidiary scalar fields of protein atoms. However, unlike other grid-based methods, its scalar field is continuous rather than discrete or boolean. Note that this continuous scalar field allows us to generate a filtration of isosurfaces by continuously varying the isovalue. Additionally, the scalar field allows us to determine if each grid node (or voxel) is transverse to, inside, or outside the protein surface. It is clear that we need some clustering technique to group together specific outside voxels into cavities.

The main deficiencies of the grid-based methods are the following: grid-spacing sensitivity and protein-orientation sensitivity. This means that, by changing the grid-spacing sensitivity or protein orientation, we may obtain a distinct number of cavities, as well as distinct locations for cavities. CavSeeker solves these problems by first locating the critical points of the scalar field generated by the protein. Such critical points are invariant to changes of grid spacing and protein orientation, and can be determined at different levels of detail or refinement. The reader is referred to Delaney (1992), Masuya and Doi (1995), Venkatachalam *et al.* (2003), Laurie and Jackson (2005), Weisel *et al.* (2007), and Oliveira *et al.* (2014) for more details about this family of methods.

The remaining of this paper is organized as follows. Section 2 overviews the theory behind CavSeeker, namely the summation of Gaussians and spectral analysis. Section 3 describes each step of the algorithm that empowers CavSeeker. Section 4 describes the experiments an presents the accuracy results of CavSeeker in comparison with other well-known methods. This benchmark was built upon a subset of the PDBsum dataset of binding sites, that is considered here as the ground truth. Finally, Section 5 concludes the paper.

## 2 Background

CavSeeker is a grid-based method to detect protein cavities. Let us then to introduce its underpinning concepts.

### 2.1 Summation of Gaussians

We use the the summation of Gaussians as the function $f : \mathbb{R}^3 \to \mathbb{R}$ to describe the electron density scalar field of a given protein as follows:

$$f(\mathbf{p}) = \sum_{i=1}^{n} f_i(\mathbf{p}) \tag{1}$$

where

$$f_i(\mathbf{p}) = e^{-d(\frac{||\mathbf{p}-\mathbf{c}_i||^2}{r_i^2}-1)} \tag{2}$$

stands for the $i$-th Gaussian function representing the electron density field of $i$-th atom of the protein, while $r_i$ and $\mathbf{c}_i$ the radius and center of the $i$th atom; in turn, $\mathbf{p}$ denotes an arbitrary point in $\mathbb{R}^3$. This mathematical formulation has the advantage of allowing for the generation of a set of isosurfaces for the same protein. For example, the isosurface $f = 1$ is the one that better approximates the solvent-excluded surface (SES) Blinn (1982), Gabdoulline and Wade (1996), Grant and Pickup (1995), and Zhang *et al.* (2006). Intuitively, we observe this fact from Eq. (2), where $f_i$ is equal to 1 when $||\mathbf{p} - \mathbf{c}_i|| = r_i$.

### 2.2 Critical Points

CavSeeker seeks for critical points of the Gaussian scalar field $f$ (see Eq. (1)) in the domain where the protein is embedded. This domain is an axis-aligned bounding box enclosing the protein. A critical point satisfies the condition $\nabla f = \mathbf{0}$, where $\nabla f$ is the gradient of $f : \mathbb{R}^3 \to \mathbb{R}$. In other words, the partial derivatives vanish at every critical point in the domain.

There are three types of critical points, namely minima, maxima, and saddles, as illustrated in Fig. 1. Taking into consideration that Gaussian functions $f_i$, one per atom, attains a maximum at the center of each atom and is positive everywhere, we observe that maxima of $f$ are all inside the molecular surface (Fig. 1(a)). Also, considering $f$ decreases with the distance to atoms and their surface, it seems obvious that the minima of $f$ are found away from the molecular surface (Fig. 1(b)). Looking at Fig. 1(c), we see that saddles (points where $f$ is maximum and minimum simultaneously) are located in depressions or concavities of the molecular surface. Summing up, maxima are not helpful to determine cavities of the protein. Minima are useful to determine voids. In fact, in a void there is at least a minimum because $f$ increases from such a minimum to any neighboring maximum inside the surface. In turn, saddles determine the existence of exposed cavities and tunnels.

However, the annihilation of the gradient does not allow us to classify the critical points into maxima, minima, and saddles. To make sure about the sort of a critical point, we need to compute the eigenvalues $\lambda_1$, $\lambda_2$, and $\lambda_3$ of the Hessian matrix, which is defined as follows:

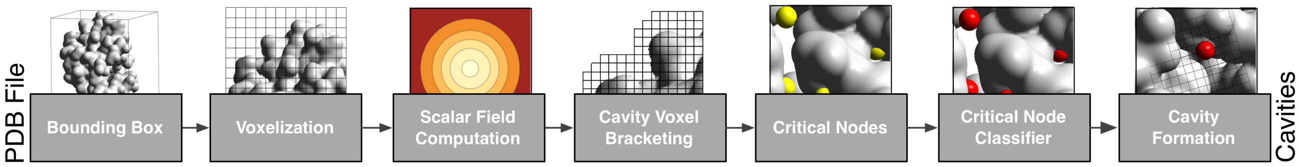


Fig. 2: CavSeeker's steps.

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{bmatrix} \quad (3)$$

Critical points are then classified as follows:

- *Maxima*: A critical point is a maximum if $\lambda_1$, $\lambda_2$, and $\lambda_3$ are all negative.
- *Minima*: A critical point is a minimum if $\lambda_1$, $\lambda_2$, and $\lambda_3$ are all positive or zero.
- *Saddles*: A critical point is a saddle if at least one eigenvalue is positive and another is negative. Specifically, if two eigenvalues are positive and one is negative, we have a type-2 saddle; otherwise, if if two eigenvalues are negative and one is positive, we have a type-1 saddle.

### 2.3 Voxelization of the Domain

Considering that the domain of $f$ will be discretized by a voxelized, axis-aligned bounding box, we classify the domain voxels relative to the isosurface $f = 1$ as follows: inside ($f > 1$), outside ($f < 1$), and surface ($1 + \epsilon < f < 1 - \tau$), where $\epsilon, \tau > 0$. For sake of computational efficiency, computing the location of minima (inside the voids of the protein) and saddles outside the molecular surface is performed in an approximate manner at the grid nodes (i.e., voxel corners). First, we determine the outside nodes where the gradient attains an absolute-valued minimum relative to its neighboring nodes; such nodes are here called critical nodes. Then, such critical nodes are classified accordingly in function of its eigenvalues: minimum nodes (for voids) and saddle nodes (for exposed cavities and tunnels).

## 3 Method

As illustrated in Fig. 2, CavSeeker consists of several steps: (i) computation of the bounding box enclosing the protein; (ii) voxelization (or gridification) of the bounding box; (iii) computation of scalar field at each grid node; (iv) reduction of the domain of outside grid nodes; (v) computation of the outside critical nodes; (vi) classification of outside critical nodes; and (vii) cavity formation.

### 3.1 Bounding Box

Computing the axis-aligned bounding box enclosing the protein is performed calculating the minimum and maximum coordinates of the respective atom centers in the directions $x$, $y$, and $z$; that is, the bounding box is defined by two of its opposite corners, $(x_{min}, y_{min}, z_{min})$ and $(x_{max}, y_{max}, z_{max})$.

### 3.2 Voxelization

The second step concerns the voxelization of the bounding box, that is, its partitioning into equally-sized voxels. This amounts to find a regular grid of nodes (or voxel corners). We used a grid spacing of $\Delta = 0.6$Å. This regular grid is represented by a three-dimensional array, where each component $(i, j, k)$ represents a grid node position $(x, y, z)$.

### 3.3 Scalar Field Computation

Terminated the voxelization of the domain (or bounding box), we need to calculate the value of the scalar field $f$ given by Eq. (1) at the position $(x, y, z)$ of each grid node $(i, j, k)$. The computation of $f$ at a grid node is a relatively slow operation because it depends on the number of atoms of the protein; for example, for a protein with 25,000 atoms, the value of

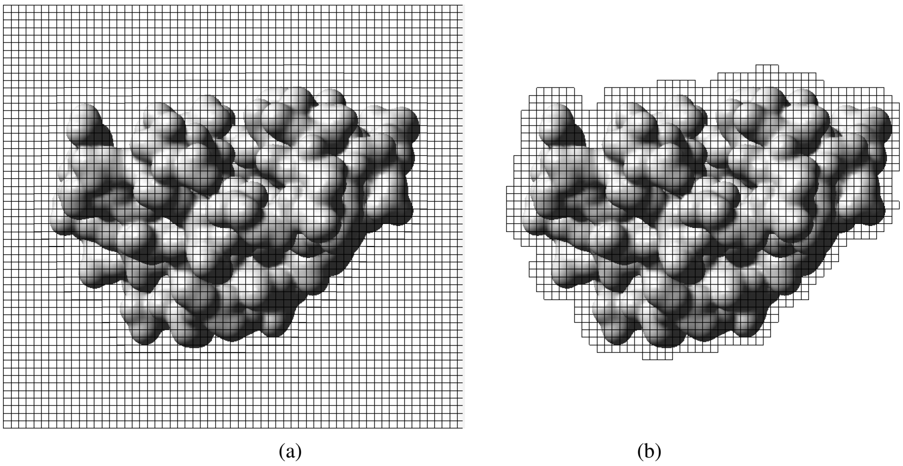


Fig. 3: Cavity voxel bracketing between two surfaces of the protein 1FK3 concerning two distinct isovalues : (a) all set of voxels of the bounding box; (b) grid nodes between the molecular surface defined by $f = 1.0$ and an outer molecular surface defined by $f = 0.001$.

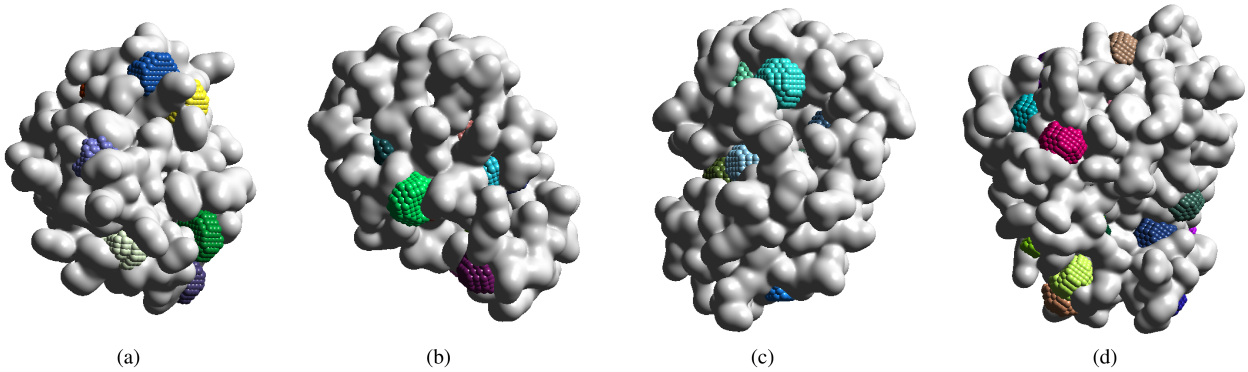

(a) (b) (c) (d)

Fig. 4: Cavities detected by CavSeeker: (a) 1UBI ; (b) 1FK6; (c) 1MZM; (d) 1Y2Q.

$f$ at a single node $(x, y, z)$ requires the computation of the summation of 25,000 Gaussians.

### 3.4 Cavity Voxel Bracketing

Taking into account that the scalar field vanishes with the distance to the protein, we only consider outside grid nodes between the molecular surface (where $f = 1.0$) and the outer molecular surface defined by $f = 0.001$ (Fig. 3(b)), rather than all voxels of the bounding box (Fig. 3(a)). Note that this outer surface is not really sampled and triangulated, and it plays the role of a shell (similar to convex hull) enclosing the protein. This node filtration also means that grid nodes that are inside of the surface and those near the protein's bounding box are discarded (Fig. 3(b)). Additionally, this voxel bracketing procedure solves the problem of mouth opening ambiguity of other grid-based methods.

### 3.5 Critical Nodes

A critical node can be defined as a grid node at which the sum $|\frac{\partial f}{\partial x}|+|\frac{\partial f}{\partial y}|+|\frac{\partial f}{\partial z}|$ attains a minimum relative to its 6 neighboring nodes, two nodes in each axis-aligned direction. Note that we are assuming that a critical node is not necessarily a critical point where the gradient vanishes, but a critical node is supposed to have a critical point nearby. By contraposition, a node that is not a critical node is here called regular node.

### 3.6 Critical Node Classifier

At this stage, we need to classify critical nodes. As shown above (see Section 2.2), outside the protein, we only find two types of critical points, say minima and saddles. A node is a minimum if the eigenvalues $\lambda_1$, $\lambda_2$, and $\lambda_3$ are all positive or zero; otherwise, it is a saddle. Recall that a minimum node identifies the presence of a void, while a saddle denotes the presence of either a tunnel or an exposed cavity.

### 3.7 Cavity Formation

Each cavity is formed from each minimum or saddle node by expanding its neighborhood of voxels. This voxel neighbood is initially $3 \times 3$, and may increases to $5 \times 5$, $7 \times 7$, $\ldots$, $2k + 1$, where $k$ is the level of expansion. The expansion is adaptive because it stops in a given direction when a bound voxel is found. A bound voxel is either a surface voxel or an outer shell voxel. Thus, CavSeeker does not suffer from the mouth opening ambiguity. That is, the expansion of a cavity is blocked by surface voxels and outer shell voxels. Note that the voxel expansion procedure may absorb other critical voxels; this is particularly true for tunnels. In Fig. 4, we can observe the cavities determined by CavSeeker for four proteins, namely 1UBI, 1FK6, 1MZM, and 1Y2Q.

## 4 Results

### 4.1 Experimental setup

All results were obtained using an Apple iMac with a 3.2 GHz Intel Core i5, a NVIDIA GeForce GT755M, and 16 GB RAM, with the OS X Yosemite operating system. Such results concern all benchmarking cavity prediction methods, including CavSeeker. For molecular visualization sake, we used the OpenGL 2.0 graphics system. CavSeeker's source code is publicly available at `https://github.com/MediaLabProjects/CavSeeker`.

### 4.2 Benchmarking methods

To assess the accuracy of CavSeeker in identifying protein cavities, we selected the following competing methods to compare with:

- *Fpocket*. This Voronoi tessellation-based method builds upon the theory of $\alpha$-spheres; see Le Guilloux *et al.* (2009) for further details.
- *MSPocket*. This is a surface-based method Zhu and Pisabarro (2011). It takes advantage of the normal-based curvature data at each surface point to classify and merge nearby concave neighborhoods into a protein cavity.
- *GHECOM*. This grid-and-sphere method is due to Kawabata and Go (2007). It takes advantage of the theory of mathematical morphology and multi-sized spherical probes to identify protein cavities. It adopts the exclusion principle behind sphere-based methods that defines a cavity as the empty space outside the protein where a probe small enough may enter but a smaller one cannot.
- *POCASA*. A grid-and-sphere method was proposed by Yu et al. Yu *et al.* (2010). The method is based on a boolean scalar field on grid nodes to distinguish between inner and outer grid nodes relative to the protein surface. Then, cavities are given by nearby grid nodes that are not absorbed by a large probe sphere rolling outside the surface in the domain.
- *ConCavity*. This grid-based method due to Capra *et al.* (2009) combines the virtues of both structure- and sequence-based methods, trying at the same time to rid off their drawbacks. Essentially, it uses any grid-based method (e.g., PocketFinder due to An *et al.* (2005)) to determine the cavities of a given protein, scoring then the grid nodes of each cavity using sequence conservation values of nearby residues.

- *PASS*. This sphere-based method is due to Brady and Stouten (2000). It builds upon the three-point Connolly-like sphere geometry to fill cavities with probe spheres Connolly (1983).
- *CriticalFinder*: This is a grid-and-surface method that identifies cavities by locating critical points in surface voxels, that is, voxels crossed by the molecular surface (see Dias *et al.* (2017) for more details).

## 4.3 Ground Truth

The cavities found by each method (i.e., the predicted binding sites) were compared with the already known binding sites for proteins as of the ones of PDBSum (`www.ebi.ac.uk/pdbsum`) (see Laskowski *et al.* (1997)). More specifically, we used the cavity dataset concerning 1239 proteins (335 apo proteins and 904 holo proteins) of LigASite (`ligasite.org/`) as ground truth (see Dessailly *et al.* (2007)). Similar to PDBsum, CavSeeker only considers the first ten larger cavities of each protein for benchmarking sake. That is, we measure the accuracy of each method relative to 12390 cavities concerning 1239 proteins.

## 4.4 Scoring Metrics

Traditionally, benchmarking cavity detection methods is accomplished by measuring the distance between the geometric center of the predicted cavity and its corresponding geometric center of the already known binding site. Supposedly, such a distance distance must be less than 4.0 Å to consider a predicted cavity as a hit. On the contrary, we consider that there is a hit if a predicted cavity $c$ overlaps a ground truth binding site $b$, that is $c \cap b \neq \varnothing$. This allows to fairly measure the accuracy of any cavity detection method relative to a ground truth dataset of known binding sites. Specifically, we use F-score as the accuracy metric, which is mathematically given by the following harmonic mean:

$$F = 2\frac{P.R}{P+R} \tag{4}$$

where $P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$ stand for the precision and recall metrics, while $FP$ the number of false positives, $TP$ denote the number of true positives, and $FN$ the number of false negatives. In this context, a true positive represents a predicted cavity that overlaps a ground truth binding site, a false positive represents a predicted cavity that does not overlap any ground truth binding site, and false negative represents a ground truth binding site that is not overlapped by any predicted cavity. Furthermore, *precision* denotes the percentage of ground truth binding sites relative to cavities predicted by a given method. On the other hand, *recall* denotes the percentage of ground truth binding sites that were in fact detected by a given method.

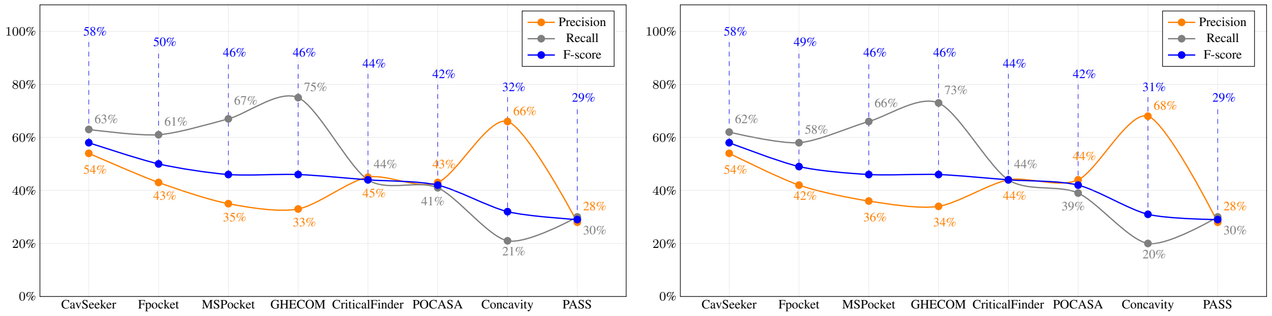
Table 1. Accuracy study for a ground truth of 1239 proteins.

(a) Results for apo proteins

| Method | $TP$ | $FP$ | $FN$ | $\approx P$ | $\approx R$ | $\approx F$ |
|---|---|---|---|---|---|---|
| CavSeeker | 2108 | 1776 | 1242 | 54% | 63% | 58% |
| Fpocket | 2040 | 2754 | 1310 | 43% | 61% | 50% |
| MSPocket | 2239 | 4096 | 1111 | 35% | 67% | 46% |
| GHECOM | 2509 | 5087 | 841 | 33% | 75% | 46% |
| CriticalFinder | 1487 | 1850 | 1863 | 45% | 44% | 44% |
| POCASA | 1358 | 1804 | 1992 | 43% | 41% | 42% |
| Concavity | 695 | 356 | 2655 | 66% | 21% | 32% |
| PASS | 1018 | 2669 | 2332 | 28% | 30% | 29% |

(b) Results for holo proteins

| Method | $TP$ | $FP$ | $FN$ | $\approx P$ | $\approx R$ | $\approx F$ |
|---|---|---|---|---|---|---|
| CavSeeker | 5638 | 4864 | 3402 | 54% | 62% | 58% |
| Fpocket | 5259 | 7161 | 3781 | 42% | 58% | 49% |
| MSPocket | 5929 | 10558 | 3101 | 36% | 66% | 46% |
| GHECOM | 6632 | 12883 | 2408 | 34% | 73% | 46% |
| CriticalFinder | 3939 | 4992 | 5101 | 44% | 44% | 44% |
| POCASA | 3568 | 4571 | 5472 | 44% | 39% | 42% |
| Concavity | 1786 | 833 | 7254 | 68% | 20% | 31% |
| PASS | 2616 | 6759 | 6250 | 28% | 30% | 29% |



(a) (b)

Fig. 5: Accuracy study for ground truth proteins. a) Percentages for apo structures. b) Percentages for holo structures.

### 4.5 Discussion

Looking at Table 1 and Figure 5, we observe that CavSeeker ranks second regarding the number of false positives (FP), just behind Concavity; consequently, Concavity owns the highest value of precision (P). However, Concavity performs very poorly in terms of false negatives (FN), so that its recall (R) is the lowest of all methods. In other words, a high precision value cannot be considered as a good result when the corresponding recall is low. In this case all cavities detected by the method were tagged as being in the ground truth (i.e., high precision). However, not all ground truth binding sites were detected (i.e., low recall).

On the contrary, a high recall value and a low precision is not a good result either, as it is the case of GHECOM. This means that GHECOM is capable of detecting all binding sites of the ground truth (i.e. high recall), but at the cost of a high number of cavity predictions (i.e., low precision). An accurate cavity detection method is characterized by a high trade-off between precision and recall or, equivalently, a high value of the F-score. As observed from Table 1 and Figure 5, CavSeeker outperforms the remaining cavity detection methods because it ranks first regarding its F-score, which is 58% for both apo and holo proteins, well above Fpocket that ranks second.

## 5 Conclusions

We have introduced a new grid-based method, called CavSeeker, for detecting protein cavities. Its foundations lie in the spectral analysis of Gaussian scalar fields in $\mathbb{R}^3$. In other words, we use the gradient of the scalar field of a protein to identify nearly critical points, as well as the eigenvalues of the Hessian matrix to classify the types of cavities. As noted above, the location of critical points outside the protein is invariant to grid spacing and protein orientation, so the main issues inherent to grid-based methods are *a priori* solved. Besides, CavSeeker uses two sets of voxels transverse to two iso-surfaces defined by distinct iso-values to bracket cavities in between, solving the third major problem of grid-based methods known as mouth open ambiguity. For rendering sake, either iso-surface can be triangulated through the marching cubes algorithm or any triangulation algorithm for implicit surfaces. We also used the scoring metrics of precision, recall, and F-score to compare CavSeeker with other cavity detection methods. The benchmarking results show us that CavSeeker outperforms its competitors.

## Acknowledgements

This research has been partially supported by the Portuguese Research Council (Fundação para a Ciência e Tecnologia), under the project UID/EEA/50008/2013 and the doctoral Grant SFRH/BD/99813/2014. Finally, our thanks to all reviewers that contributed to improve this paper.

## References

An,J., Totrov,M. and Abagyan,R. (2005) Pocketome via comprehensive identification and classification of ligand binding envelopes. *Molecular and Cellular Proteomics,* **4** (6), 752–761.

Blinn,J.F. (1982) A generalization of algebraic surface drawing. *ACM Transactions on Graphics,* **1** (3), 235–256.

Brady,G.P. and Stouten,P.F.W. (2000) Fast prediction and visualization of protein binding pockets with PASS. *Journal of Computer-Aided Molecular Design,* **14** (4), 383–401.

Capra,J.A., Laskowski,R.A., Thornton,J.M., Singh,M. and Funkhouser,T.A. (2009) Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. *PLoS Computational Biology,* **5** (12), e1000585.

Cazals,F., Chazal,F. and Lewiner,T. (2003) Molecular shape analysis based upon the Morse-Smale complex and the Connolly function. In *Proceedings of the 19th Annual Symposium on Computational Geometry* (SCG'03), San Diego, California, USA, June 8-10 pp. 351–360 ACM Press.

Connolly,M. (1983) Analytical molecular surface calculation. *Journal of Applied Crystallography,* **16** (5), 548–558.

Delaney,J.S. (1992) Finding and filling protein cavities using cellular logic operations. *Journal Molecular Graphics,* **10** (3), 174–177.

Dessailly,B.H., Lensink,M.F., Orengo,C.A. and Wodak,S.J. (2007) Ligasite: a database of biologically relevant binding sites in proteins with known apo-structures. *Nucleic Acids Research,* **36** (suppl_1), D667–D673.

Dias,S.E., Nguyen,Q.T., Jorge,J.A. and Gomes,A.J. (2017) Multi-GPU-based detection of protein cavities using critical points. *Future Generation Computer Systems,* **67**, 430–440.

Edelsbrunner,H., Kirkpatrick,D.G. and Seidel,R. (1983) On the shape of a set of points in the plane. *IEEE Transactions on Information Theory,* **29** (4), 551–559.

Exner,T., Keil,M. and Brickmann,J. (2002) Pattern recognition strategies for molecular surfaces. I. Pattern generation using fuzzy set theory. *Journal of Computational Chemistry,* **23** (12), 1176–1187.

Gabdoulline,R.R. and Wade,R.C. (1996) Analytically defined surfaces to analyze molecular interaction properties. *Journal of Molecular Graphics,* **14** (6), 341–353.

Grant,J.A. and Pickup,B. (1995) A Gaussian description of molecular shape. *The Journal of Physical Chemistry,* **99** (11), 3503–3510.

Kawabata,T. and Go,N. (2007) Detection of pockets on protein surfaces using small and large probe spheres to find putative ligand binding sites. *Proteins: Structure, Function, and Bioinformatics,* **68** (2), 516–529.

Laskowski,R.A., Hutchinson,E.G., Michie,A.D., Wallace,A.C., Jones,M.L. and Thornton,J.M. (1997) PDBsum: a web-based database of summaries and analyses of all PDB structures. *Trends in Biochemical Sciences,* **22** (12), 488–490.

Laurie,A.T.R. and Jackson,R.M. (2005) Q-SiteFinder: an energy-based method for the prediction of protein-ligand binding sites. *Bioinformatics,* **21** (9), 1908–1916.

Le Guilloux,V., Schmidtke,P. and Tuffery,P. (2009) Fpocket: an open source platform for ligand pocket detection. *BMC Bioinformatics,* **10** (1), 1–11.

Masuya,M. and Doi,J. (1995) Detection and geometric modeling of molecular surfaces and cavities using digital mathematical morphological operations. *Journal of Molecular Graphics,* **13** (6), 331–336.

Natarajan,V., Wang,Y., Bremer,P.T., Pascucci,V. and Hamann,B. (2006) Segmenting molecular surfaces. *Computer Aided Geometric Design,* **23** (6), 495–509.

Oliveira,S.H., Ferraz,F.A., Honorato,R.V., Xavier-Neto,J., Sobreira,T.J. and de Oliveira,P.S. (2014) KVFinder: steered identification of protein cavities as a PyMOL plugin. *BMC Bioinformatics,* **15** (197), 1–8.

Rodrigues,R.S.V., Morgado,J.F.M. and Gomes,A.J.P. (2018) Part-based mesh segmentation: a survey. *Computer Graphics Forum,* **37** (6), 235–274.

Simões,T., Lopes,D., Dias,S., Fernandes,F., Pereira,J., Jorge,J., Bajaj,C. and Gomes,A. (2017) Geometric detection algorithms for cavities on protein surfaces in molecular graphics: a survey. *Computer Graphics Forum,* **36** (8), 235–274.

Venkatachalam,C., Jiang,X., Oldfield,T. and Waldman,M. (2003) LigandFit: a novel method for the shape-directed rapid docking of ligands to protein active sites. *Journal of Molecular Graphics and Modelling,* **21** (4), 289–307.

Weisel,M., Proschak,E. and Schneider,G. (2007) PocketPicker: analysis of ligand binding-sites with shape descriptors. *Chemistry Central Journal,* **1** (1), 1–17.

Yu,J., Zhou,Y., Tanaka,I. and Yao,M. (2010) Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. *Bioinformatics,* **26** (1), 46–52.

Zachmann,C.D., Heiden,W., Schlenkrich,M. and Brickmann,J. (1992) Topological analysis of complex molecular surfaces. *Journal of Computational Chemistry,* **13** (1), 76–84.

Zhang,Y., Xu,G. and Bajaj,C. (2006) Quality meshing of implicit solvation models of biomolecular structures. *Computer Aided Geometric Design,* **23** (6), 510–530.

Zhu,H. and Pisabarro,M.T. (2011) MSPocket: an orientation-independent algorithm for the detection of ligand binding pockets. *Bioinformatics,* **27** (3), 351–358.

# Chapter 6

# Conclusions and Future Work

This thesis builds upon on the research work carried out in protein cavity detection methods, as needed for structure-based drug design and protein docking. Briefly speaking, the main contribution of the research work that has led to the present thesis lies in the usage of shape descriptors borrowed from computer graphics to molecular graphics and modeling, specifically to design and implement protein cavity detection methods.

## 6.1 Research Context

Proteins play an essential role in the functioning of living organisms due to their interactions with other molecules. This role is related with the ”lock-and-key” hypothesis, formulated in 1984 by Hermann Fischer; that is, the shape complementarity between receptor protein and ligand. The research work described in the present thesis has much to do with finding the “locks” of proteins, here called protein cavities (or putative binding sites), and not that much with the “keys” (or ligands).

This thesis focuses on the design and development of new geometric methods. These methods have been designed and implemented considering the following:

- *Shape descriptors*. As usual in mesh segmentation of 3D objects in computer graphics, the protein segmentations proposed in this thesis build upon shape descriptors; for example, CavShape takes advantage of the multivariate shape diameter function (mSDF) to carry out surface segmentations of protein surfaces.
- *Clustering*. As shown in this thesis, clustering of the constituents (e.g., triangles) of a protein cavity is a fundamental task of any cavity detection method. Usually, one uses proximity-base criteria to collect such constituents into cavities.
- *Filtering*. Most cavity detection methods tend to produce an excessive number of cavities. However, it is known that the largest cavities are those with greater probability of being binding sites. Therefore, some methods only consider the largest top-3 cavities in the detection process; that is, the small cavities are commonly discarded in the end of the detection method workflow.

Overall, the research work was planned to come up with more accurate cavity detection methods than the state-of-the-art methods, yet without noticeably losing time performance.

## 6.2 Research Questions

To validate the thesis statement (see Chapter 1), several research questions have been addressed:

- *Is it possible to accurately detect and delineate cavities on proteins using computer graphics concepts (e.g., field-of-view) without using shape descriptors?*
  CavVis does not use shape descriptors to segment a protein into cavities. However, its accuracy is higher or comparable to the benchmarking methods considered in this thesis (see Chapter 3). Recall that CavVis combines three important concepts and tools commonly used in computer graphics, namely: field-of-view (FoV), voxel ray casting, and back-face culling.
- *Is it possible to accurately detect and delineate cavities on protein surfaces using shape descriptors?*
  Both methods described in Chapters 4 (CavShape) and 5 (CavSeeker) use shape descriptors to determine the location of cavities. CavShape uses a shape variant called multivariate shape diameter function (mSDF) to find such cavities, while CavSeeker uses another shape descriptor based on eigenvalue analysis. Specifically, cavities found through CavSeeker correspond to locations of minima and saddles of the electron density scalar field.

Thus, the thesis statement is here positively validated. In particular, we have found very promising to design accurate protein cavity methods using shape descriptors commonly used in computer graphics. Moreover, the main contribution of this thesis likely is the use of shape descriptors to come up with protein segmentations.

## 6.3 Discussion of Results

The thesis statement is also positively supported by the results obtained in Chapters 3, 4, and 5. As shown in Figure 6.1, and considering the ground-truth of 1239 proteins, we see that CavSeeker is ranked first in terms of F-score followed by CavShape and CavVis, while the state-of-the-art methods are left behind with lower F-score values.

These results have much to do with the main issues identified in Chapter 2. In particular, grid-spacing sensitivity, mouth-opening ambiguity, and protein-orientation sensitivity. These issues are thoroughly discussed in the thesis, and are responsible for the low accuracy results of the state-of-the-art methods. In the case of methods that suffer from grid-spacing or protein-orientation sensitivity, the accuracy decreases because by setting an inadequate grid spacing value or by varying the protein orientation one may get a different number of cavities or locations. On the other hand, those that suffer from mouth-opening ambiguity are prone not to be able to define the extent and entrances of each cavity, thus impairing the process of cavity detection.

Briefly, the following can be observed:

*FPocket* and *MSPocket*. These two methods detect a high number of cavities that do not match binding sites of the ground-truth. This fact translates itself into a higher number of false positives, which in turn leads to a lower precision value when compared to the three methods proposed in this thesis. This has much to do with the fact that FPocket does not implement a more involved method of filtering the number of outputted cavities. On the other hand, MSPocket merges cavities only using a criterion of triangle-triangle connectivity; that is, a more robust clustering technique is necessary to obtain better results. The accuracy of MSPocket is also diminished by the fact that it suffers from mouth-opening ambiguity; in fact, it does not use an outer surface to delineate cavity entrances and exits.

*GHECOM*. This method produces a high number of cavities, which results in a higher number of true positives and higher recall. However, and similar to FPocket and MSPocket, most of the predictions of this method do not match ground-truth binding sites, so leading to a higher number of false positives and a lower precision percentage. Although the problem related to grid-spacing sensitivity may be mitigated through an adequate grid spacing value, its F-score is lower when compared to the three methods proposed in this thesis because the method do not use any filtering process to avoid the excessive number of false positives.

*CriticalFinder*. This method forms clusters of critical points of the same type (either saddles or minima) at voxel corners. In practice, each cluster is a set of critical voxels, and thus provides us the corresponding cavity extent, as well as the center of the cavity, which is given by the barycenter of critical points of the cluster. This voxel clustering technique leads to a lower value of precision and recall because only the critical voxels crossed by the molecular surface are taken into account. That is, the cavity extent is not fully formed, which results in mouth-opening ambiguity, and some cavities may be even missed out. In other words, the filtering is excessive.

*POCASA*. This method builds upon a grid embedding the protein and a rolling probe sphere. The probe sphere rolls on the protein surface to generate an outer surface, called probe surface, through the inner border tracing algorithm, which is well-known in the image processing field. Cavities are found between the protein surface and probe surface, but the volume and shape of the cavities depend on the adjustment of probe sphere radius. Therefore, the precision and recall values are low, and consequently the F-score is also low, unless we use a range for probe sphere radius; that is, several probe surfaces. However, this range of radii has a significant impact on the time performance of this method.

*ConCavity* and *PASS*. ConCavity features the lowest recall value of all benchmarked methods, because it outputs a high number of false negatives. This fact results from an inadequate volume-based filtering criterion that wrongly discards both small and large cavities. Regarding PASS, it not only features a high number of false positives, but also a low number of true positives. In other words, PASS detects cavities that do not match ground-truth binding sites. Consequently, the corresponding F-scores are low.

Summing up, the three methods proposed in this thesis attain better results than the state-of-the-art methods because they use better clustering and filtering techniques. For example, in the process of clustering cavities, the diameter of the water molecule is employed for a more suitable clustering of nearby cavities, so reducing the excessive number of detections. Furthermore, a volume-based filtering process is also applied to all cavities detected by CavVis, CavShape, and CavSeeker, so that only the first ten larger cavities are considered; the remaining (smaller) cavities are discarded straight away.

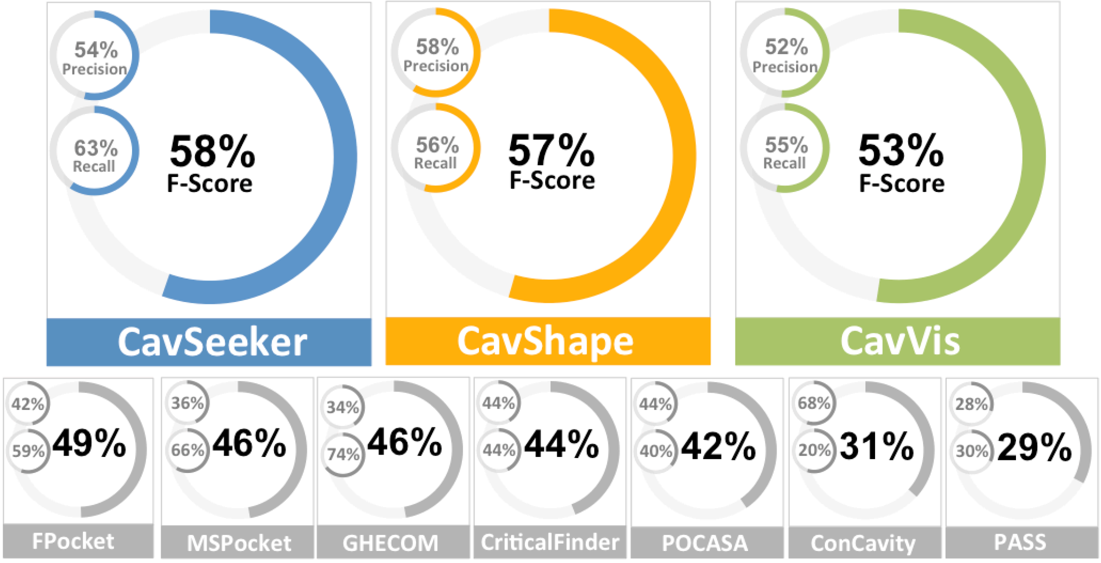


Figure 6.1: Final accuracy results for the methods proposed in this thesis and those of the state-of-the-art against a ground-truth of well-known binding sites.

## 6.4 Research Limitations and Future Work

Looking back to the research work carried out during the PhD programme, there are mainly three research avenues to follow in the future.

First, the accuracy of cavity detection methods is far from being a solved problem because the accuracy still is under 90 percent. Therefore, we need to further investigate for alternative shape descriptors capable of reaching higher accuracy scores.

Second, the speed of cavity detection methods still is relatively slow using single-threaded computations when the number of atoms goes over some dozens of thousands. This phenomenon is particularly noticeable when one uses summation of Gaussian or even other kernel functions. In other words, the speed is highly dependent of the mathematical formulation of protein surface. The question then is whether or not there is an alternative mathematical formulation for molecular surfaces.

Finally, current geometric-based methods are essentially static, as usually only a single conformation of the protein is considered in the detection of cavities. A current trend of molecular graphics and modeling is to design and implement cavity detection methods that take into consideration the dynamic geometry (and topology) of proteins and their cavities over time; e.g., a pocket may evolve to a void and vice-versa.