Universitat Pompeu Fabra Barcelona

PockMan

Theory explanation

Wael Badr, Narine Fischer, Radostina Kislava

Master in Bioinformatics for Health Sciences
Universitat Pompeu Fabra
Structural Bioinformatics and Introduction to Python joint project
April 2025

Introduction

Proteins are the macromolecules in charge of carrying out processes necessary for the cell's survival. These processes are performed thanks to molecular recognition, in which macromolecules interact with each other, or small molecules, to complete their function. In proteins, we talk about protein-ligand interactions [1].

Therefore, protein-ligand interactions are the basis of all processes in living organisms [1]. As a consequence, they can also be the basis of pathological conditions. For that reason, predicting protein-ligand interactions is needed to understand the biological mechanisms underlying organisms' functionality, how their alteration can cause illnesses, and how to develop drugs that address these interactions [2,4].

The interaction between the receptor and the ligand takes place in what is known as the ligand-binding site, normally found in pockets or concavities in the surface of the proteins [2,3,7]. During the years, many computational methodologies have arisen to solve the problem of predicting these sites, increasing their complexity with the advancements in the bioinformatics field. One of the approaches is using geometry (geometry-based approach) or using the energy (energy-based approach) to detect the cavities [3]. More advanced, we find approaches based on Machine Learning (ML), like *P2Rank* [4]. And with the appearance of Deep Learning (DL), we can find programs like *Kalasanty* [5]. Despite this, as the work presented is based on the first approaches, only these will be explained further.

As previously explained, the first approaches could either be geometry-based or energy-based. In geometry-based methods, the surface shape of the protein is analysed to detect cavities. While energy-based uses probes or molecular fragments to record their interaction with the protein and see if there are favourable energetic responses, and in the case that there are, a pocket is detected. Moreover, these two approaches can also be classified into whether they are grid-based (using a Cartesian coordinate grid to represent the protein) or grid-free (no Cartesian grid is used) [3].

An example of a geometry-based and grid-free approach is *SURFNET*. In this program, spheres are fitted between the atoms of the protein's surface, and if the sphere clashes with any atom, its volume is reduced until no clashes are detected. The remaining sphere marks the presence of a pocket [3,6].

For a geometry-based and grid-based approach, we can find *LIGSITE*. The protein is placed in a Cartesian grid, the grid points are then analysed, searching for places that present protein-solvent-protein (PSP). Each grid point is analysed in 7 directions: the x, y, and z axes, and the 4 cubic diagonals. This way, when a PSP is found, the value of the grid-point increases; and when the 7 directions are completed for all grid-points, a cut-off value is used to keep the grid-points with the highest value as reference for the pockets' locations [3,7].

For energy-based and grid-free approaches, an example can be found in molecular docking. In this approach, a known 3D structure is used to see where the interaction may occur. If

many 3D structures bond to the same region, that region is considered a ligand-binding site [3,8].

Finally, for energy-based and grid-based approaches, an example can be *DrugSite*. The protein is placed in a Cartesian grid, each grid point is substituted by a carbon probe. The van der Waals energies between the probe and the protein region within a certain distance are calculated. The energies are analysed to test whether they pass a certain threshold, and the remaining grid points are joined into a pocket [3,9].

PockMan

The program PockMan searches the protein space to detect possible ligand binding sites, following an approach based on using the protein's geometry with the use of grids. PockMan uses the information obtained from a PDB file as input, using the coordinates of the protein's atoms to generate the grid and for the subsequent projection of the protein onto a Cartesian grid. Thanks to this projection, clefts and pockets in the protein space are searched in order to detect possible ligand binding sites, as it has been reported that normally these cavities present the necessary residues for the interaction between a protein and its ligand [2,3,7].

PockMan follows six steps to determine the location of the possible ligand binding sites: loading the protein structure, initialisation of the grids, projection of the protein onto a grid, detection of protein-solvent-protein (PSP) voxels, clustering the detected voxels into pockets, and finally, finding the nearby atoms per pocket to detect which residues are present.

Loading the protein structure

PDB files contain information about different aspects of the structure of the molecule, including the presence of heteroatoms (ligands, solvent molecules, metal ions…), and not only about the crystal structure of the protein. To focus strictly on the protein itself, the first step is to clean the PDB files to keep only the information needed, filtering out the atoms that are not labeled as "ATOM", as these are the ones inherent to the protein.

This step is necessary to avoid increasing computation time and reducing grid precision when trying to detect pockets. If heteroatoms ("HETATM" entries) like water molecules, ligands or ions are included when the grid is generated, the grid's dimensions may be expanded due to the inclusion of unnecessary voxels, which may hinder both the time and quality of the analysis.

After this cleaning step, only the atoms labeled as "ATOM" are loaded to PockMan, making sure that the grid is a more accurate 3D representation of the protein's structure, without any disturbance by non-protein atoms.

Initialization of the grids

After obtaining the structure of the protein, the coordinates of each atom are analyzed. For each spatial axis (x, y and z), the minimum and maximum coordinate values are identified. Based on their values, the grid dimensions (number of voxels) are calculated following the next formula:

$$dimension\ of\ a\ given\ axis\ (n^{o}of\ voxels) = \frac{max.\ coordinate\ of\ the\ axis - min.\ coordinate\ of\ the\ axis}{grid\ spacing} + 1$$

From the difference between the two end coordinates, the distance (in angstroms) between the two points is obtained. This distance is then divided by the desired spacing between grid points to determine the number of voxels needed to cover the same distance. Finally, to the number of voxels, one is added to ensure that the ends are included, obtaining a more accurate representation of the protein's volume.

This is performed independently for the x-, y- and z-axes to determine the number of voxels needed for each dimension. With this information, two 3D grids are initialized:

- Protein grid: grid in which the protein will be projected, therefore, it will be the representation of the physical space occupied by the protein. Each voxel will have a value assigned, a value of 1 for those that contain atoms from the protein, or a value of 0 if the voxel represents a space that would be accessible by solvent.
- Pockets grid: grid in which the voxels will be used to register potential clefts and pockets in the protein, based on geometric scoring.

Both 3D grids will have the same dimensions to facilitate the detection of pockets.

Projection of a protein onto a grid

The projection of the protein onto the Protein grid is necessary to discern which voxels contain part of the protein, and which would be filled with solvent. As PockMan uses a geometrical approach, identifying which voxels contain protein atoms is necessary to determine the presence of pockets in the protein in the next step.

To accurately project the atoms, the van der Waals radii of the different atoms must be taken into account. Depending on the selected spacing between grid points, this distance might be shorter than the radius of a given atom. For example, if the spacing is of 1.0 Å, and we are projecting a carbon atom, the radius of the atom being approximately 1.6 Å, the radius of the atom is bigger than the spacing, and consequently, this atom would occupy the space of more than one voxel. Therefore, the radius of the atoms have to be included in the projection.

As a result, when projecting an atom to the grid, the projection is not done with a single point, but a range of grid coordinates for each axis (x, y and z). This range spans from the center of the atom (obtained from the PDB file) to its surface in both directions, based on its radius. To determine the ranges for each atom, the following formulas are used:

$$Lower\ bound = \frac{atom\ coordinate - atom's\ radius - min.\ coordinate\ of\ the\ axis}{spacing\ grid}$$

$$Upper\ bound = \frac{atom\ coordinate + atom's\ radius - min.\ coordinate\ of\ the\ axis}{spacing\ grid}$$

Thanks to these formulas, the real-space atomic coordinates are converted into grid-space coordinates. For the lower bound, the coordinate of an atom for a given axis is taken from the PDB file. Then, the radius of the atom is subtracted to obtain the coordinate of the atom's surface. Finally a linear transformation is applied, following the same steps as in the grid generation, to obtain the corresponding coordinate that would occupy in the grid-space. For the upper bound, the same process is followed, but with the addition of the atom's radius to obtain the coordinate of the opposite surface point, obtaining this way a range that encapsulates the atom's volume.

The calculation of the limits of the range is done independently for each axis, obtaining this way three ranges that define a box around the atom. Finally, the different voxels' coordinates are evaluated, if a voxel falls inside the three ranges described, and therefore is inside of the box delimitating the atom, it is marked as part of the protein.

At the end of this step, the projection of all protein atoms is done, obtaining a 3D representation of the protein's volume in the grid.

Detection of protein-solvent-protein voxels

Once the the projection is finalized, the next step to detect potential binding sites in the protein is identifying protein-solvent-protein (PSP) regions. In order to accomplish this, PockMan focuses on the voxels with value 0 in the Protein grid, as these are the ones that represent the solvent.

Once PockMan finds a voxel with an assigned value of 0 (solvent voxel), it will check whether it is flanked by two protein voxels (assigned value of 1) along any spatial direction. The idea is that pockets will be enclosed between two protein regions, with solvent between them. To detect this PSP, 13 different axes can be assessed, these axes represent the directions along which the flanking may occur:

- Axial direction:
  - x-axis
  - y-axis
  - z-axis
- Plane diagonals. Each plane can have two possible axis:
  - xy diagonal:
    - x and y both increase and decrease together
    - x increases while y decreases, and reversed
  - xz diagonal
    - x and z both increase and decrease together
    - x increases while z decreases, and reversed

    - yz diagonal
        - y and z both increase and decrease together
        - y increases while z decreases, and reversed
- Cubic diagonals. From which different combinations can be done:
    - x, y and z increase and decrease together.
    - x and y share the same direction, but z being reversed.
    - x and z share the same direction, but y being reversed.
    - y and z share the same direction, but x being reversed.

Each solvent voxel will be assessed for each of this 13 axes. If in one axis, it is seen that the solvent is flanked by protein voxels, the coordinates of the solvent voxel will be used to identify the equivalent voxel in the Pocket grid, incrementing its value by 1.

After iterating over all axes in all solvent voxels, the Pocket grid will be a 3D matrix that represents how enclosed the solvent is in each region. Voxels with a higher value meaning that they are deeper or more surrounded by the protein, and could represent promising ligand-binding sites.

Based on prior studies [7], it had been concluded that the addition of planar diagonals didn't improve the results of similar programs. For this reason, PockMan includes these directions as optional. The user can decide whether or not they want PockMan to perform these tests. It was decided to include them as optional to have the benefits of both options, if the user decides to include them, they will receive a more accurate result; and if they decide not to include them, it will speed up the computation.

Clustering the detected voxels into pockets

With the completed Pocket grid, PockMan searches for voxels that present a score higher than a given cut-off. From the list of filtered voxels, it takes one and starts looking for adjacent voxels that are spatially connected in the same list. It clusters together all voxels that are adjacent, forming a possible pocket. Once a possible pocket has been defined by adjacent voxels, these are removed from the list.

For each possible pocket, a final test must be conducted to determine whether it has access to external solvent or if it is actually an internal cavity inside the protein. By observing the scores of the voxels composing the possible pocket it can be discern if it is a cavity or a pocket; if all scores have the maximum score possible, it means that it is flanked by the protein in all possible directions, and therefore, it is enclosed inside the protein, a cavity. If a cavity is detected, it is discarded from the list of pockets.

This process is repeated until no more voxels are left in the list, and therefore, all of them have been assigned to a pocket, without overlaps between the different clusters detected.

The scores of each voxel are also preserved while clustering when searching for pockets. Once all pockets are detected, they are scored computing the average of scores of its

voxels. Finally, with this score, the pockets are sorted from highest to lowest, prioritizing those pockets that reach deeper inside the protein.

Finding the nearby atoms per pocket

Once the pockets have been detected in the Pocket grid, their location has to be translated to real-space coordinates. To achieve this, the grid indices of each voxel in a pocket are transformed into real-space coordinates using the following formula:

$$Real\ space\ coordinate\ for\ an\ axis = min.\ coordinate + (voxel\ index \times grid\ spacing)$$

With this formula, a linear transformation is done to obtain the coordinate for a given axis. To obtain the coordinates in the 3D space, this formula has to be applied independently to the x-, y- and z-axes.

Once the real-space coordinates are obtained, the distances from that point to the protein's atoms are computed following the next formula:

$$distance = \sqrt{\left(x_{voxel} - x_{atom}\right)^2 + \left(y_{voxel} - y_{atom}\right)^2 + \left(z_{voxel} - z_{atom}\right)^2} - atom's\ radius$$

The van der Waals radius is again taken into account to provide a better interpretation of the space occupied by the atom. As the coordinates in the PDB files are only of the center point of the atom, subtracting its radius allows the distance to more accurately represent the distance between the voxel and the atom's surface.

If the distance is below a user-defined threshold, the atom is assumed to be part of the predicted ligand-binding site. The associated residue and residue ID to this atom are obtained from the PDB to be included in the output files.

This process is repeated for each voxel of a pocket, obtaining at the end a list of residues that constitute the predicted ligand-binding site. Moreover, this analysis is performed independently for each pocket, obtaining multiple candidate binding sites.

Merging predicted binding sites

Between the different predicted binding sites, some might present overlaps, as they share residues and may be located close in space. In the case that two pockets share a high percentage of residues, and they are close to each other, it is assumed that their separation is caused by an artificial fragmentation of a larger, continuous binding site caused by PockMan's algorithm.

To improve the predictions, PockMan performs a post-processing step, trying to identify and merge overlapping pockets. The binding sites detected are compared with each other, searching for significant overlaps between the residues that constitute them. When two predicted ligand-binding sites showed to have a significant number of shared residues, the

center of each site is computed and the distance between them obtained. If the distance between the two centers is close enough, PockMan merges the two binding sites as one.

As a binding site can be overlapping with more than one other predicted site, the merging process is transitive, PockMan searches for these cases and joins them together into a single site. In the end, predictions of ligand-binding sites that may be conformed by several originally predicted sites are obtained, improving accuracy and clarity of the output.

Parameters defined by the user

PockMan allows the user to modify certain parameters of the analysis. This allows the user to better control the prediction of ligand binding sites, may it be based on previous knowledge, or to modify the strictness of the analysis. The modifiable parameters are:

- Grid spacing: space between grid points, the distance in amstrongs that each voxel represents. The default value associated is 1 Å, as it is the value used in previous programs [10]; but PockMan accepts values in a range between 0.5 Å and 2 Å, this maximum being used by other programs such as *POCKET* [10] or *LIGSITE* [7]. The bigger the value, the shorter the computational time, but less accuracy.
- Grid padding: extra space added to the dimensions of the protein when creating the grid. It is added to avoid complications during the protein projection, such as the protein being cut for lack of space. The default value is 5 Å, but PockMan accepts values in a range between 2 Å and 10 Å. The bigger the number, the more computational time.
- Diagonals: as previously stated, PockMan offers the possibility of checking the planar diagonals when searching for PSP areas, instead of doing it automatically. As previous studies have seen little improvement with their addition [7], by default, PockMan does not check planar diagonals. Despite this, in each run, it will ask the user whether they would like diagonals to be taken into account.
- Voxel cut-off: to cluster the voxels into pockets, PockMan first filters those voxels that present a high enough score obtained from the PSP detection, therefore only keeping the voxels that are more enclosed. The default value is 4, but the user can modify it to make it more or less strict; in previous studies, a value of 2 already yielded good results [7], despite this, it was decided to put 4 as default to be more stringent. Depending on whether the planar diagonals have been computed, the maximum value that a voxel may have can change, 13 if having computed them, and 7 if not. This difference should be taken into account when deciding the cut-off.
- Distance voxel-atom: when determining which atoms are part of the pocket, and therefore part of the predicted binding site, it is used by comparing the distance between each atom with the voxel's coordinates, and if they are closer than a threshold, the atom is assumed to be part of the predicted binding site. The default distance threshold is 4 Å, based on typical pocket dimensions and empirical validation in structural biology studies. However, this value can be changed by the user if desired, please, take into account that, depending on the value, the results might lose accuracy, from tests performed by our team, both really low and high numbers lose efficacy, our recommendation is to keep a distance between 3-5 Å.

Output files

The output files can be classified in three different categories:

- Predicted ligand-binding sites text files:
  - Individual ligand-binding sites files: the atoms and residues detected for a specific site are printed in a separate file. In the end, a file per predicted binding site is obtained.
  - General file: a single file containing all the predicted ligand-binding sites detected, ordered by their score obtained during PockMan's run.
- Command files for visualization in Chimera:
  - Individual command files: a command file per site. Each one of them opens the protein and visualizes it with the protein surface, later coloring the specific site according to its PockMan score.
  - General command file: a unique file to visualize all predicted sites at the same time. It opens the protein and visualizes it with the protein surface. Then the sites are colored depending on their PockMan score.
- Visualization files for Pymol:
  - Individual visualization files: a file per site. Can be used to observe each predicted site individually if found to be an interesting binding site.
  - General visualization file: a unique file for visualizing all the predicted sites at the same time.

References

1. Du, X., Li, Y., Xia, Y. L., Ai, S. M., Liang, J., Sang, P., Ji, X. L., & Liu, S. Q. (2016). Insights into Protein-Ligand Interactions: Mechanisms, Models, and Methods. *International journal of molecular sciences*, *17*(2), 144. https://doi.org/10.3390/ijms17020144
2. Dhakal, A., McKay, C., Tanner, J. J., & Cheng, J. (2022). Artificial intelligence in the prediction of protein-ligand interactions: recent advances and future directions. *Briefings in bioinformatics*, *23*(1), bbab476. https://doi.org/10.1093/bib/bbab476
3. T014 · Binding site detection. (n.d.). TeachOpenCADD. https://projects.volkamerlab.org/teachopencadd/talktorials/T014_binding_site_detection.html
4. Krivák, R., & Hoksza, D. (2018). P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of cheminformatics*, *10*(1), 39. https://doi.org/10.1186/s13321-018-0285-8
5. Stepniewska-Dziubinska, M. M., Zielenkiewicz, P., & Siedlecki, P. (2020). Improving detection of protein-ligand binding sites with 3D segmentation. Scientific reports, 10(1), 5035. https://doi.org/10.1038/s41598-020-61860-z

6. Laskowski R. A. (1995). SURFNET: a program for visualizing molecular surfaces, cavities, and intermolecular interactions. *Journal of molecular graphics*, *13*(5), 323–308. https://doi.org/10.1016/0263-7855(95)00073-9
7. Hendlich, M., Rippmann, F., & Barnickel, G. (1997). LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of molecular graphics & modelling*, *15*(6), 359–389. https://doi.org/10.1016/s1093-3263(98)00002-3
8. Li, J., Fu, A., & Zhang, L. (2019). An Overview of Scoring Functions Used for Protein-Ligand Interactions in Molecular Docking. *Interdisciplinary sciences, computational life sciences*, *11*(2), 320–328. https://doi.org/10.1007/s12539-019-00327-w
9. An, J., Totrov, M., & Abagyan, R. (2004). Comprehensive identification of "druggable" protein ligand binding sites. *Genome informatics. International Conference on Genome Informatics*, *15*(2), 31–41.
10. Levitt, D. G., & Banaszak, L. J. (1992). POCKET: a computer graphics method for identifying and displaying protein cavities and their surrounding amino acids. *Journal of molecular graphics*, *10*(4), 229–234. https://doi.org/10.1016/0263-7855(92)80074-n