Protein-Protein Docking Using Multi-Dimensional Spherical Basis Functions on High Performance Computing Platform

Abhishek K, S. Balaji

Abstract: Docking has become the most important in-silico technique in the process of in-silico drug discovery. The complexes produced because of protein-ligand and the protein-protein interaction are predicted using docking techniques. Hence, it is very important but at the same time it is quite challenging owing to the huge computational costs and the complexity of the computational techniques. In our previous work [1], we had studied the role of the FFTs in increasing the efficiency of Scoring Functions (SF) in heterogeneous parallel processing based virtual screening pipeline for effective rescoring in protein-ligand docking. In this work, we propose multi-layered polar transformation functions to search a multi-dimensional space of a rigid-body model. These functions enhance the efficient use of the spherical co-ordinates to improve the scoring function, thereby improving the overall efficiency of the docking process.

Keywords: spherical transforms, protein-protein interaction, protein-ligand interaction, multi-dimensional space.

I. INTRODUCTION

Protein-protein interactions are very central to any biological functions, so much so that it has become imperative to gain the knowledge of the structure of the target complex in such studies. Although there exist several techniques like NMR, X-ray Crystallography which help to fetch the structure of the target complex, actuating all the structures of interest remains a challenge due to various factors which involves the tradeoff between cost and efficiency. In this respect, it can be said that the in-silico techniques employed to predict the target complex (known as protein-protein docking) has gathered a lot of traction from the scientific community with the hope that it will provide the required structural information that the in-vivo / in-vitro methods fail to provide.

In general, protein-protein docking is characterized by the 3-D (three dimensional) structure of the target complex by leveraging the information from the unbound monomers. In most of the literature, it has become the de-facto standard to assume that all the information that can be leveraged in a docking process is nothing but the co-ordinates of the monomers sans any data which refers to the binding sites of any protein.

The works in [2], [3], [4] give a very comprehensive summary of several docking methods that have been proposed over time. One of the daunting challenges that the scientific community is facing is the computational complexity of the problem and also the high degree of the freedom the target complex system comes with. Hence, it becomes imperative to be more agile and adapt techniques that can scale up so much so that we can reach a solution in real time.

The general approach towards this problem is two-fold in the sense that:

1. We must perform a holistic search and then predict the probable candidates. This usually relies on techniques like scoring function and assumption of a rigid body model to reduce the search space.
2. This stage is where we refine the results or outcome of the preceding stage and refine the probable candidates that have been predicted in phase 1. This phase includes techniques that are more computationally intense since they deal with pose ranking and the structural parameters. The initial phase of the search and predicting the probable candidates is very crucial for the overall success of the docking approach.

Surface Feature point matching techniques as explained in [5], [6], [7], [8], [9] and the techniques based on the energy minimization as discussed in [10], [11], [12] and also the algorithms which performs a global search that leverages Fast Fourier Transforms (FFT) as discussed in [13], [14], [15] have been leveraged and been experimented with for performing a holistic search and for predicting the probable candidates which is the first phase as mentioned above.

Though there have been lot of studies conducted on the aforementioned techniques and algorithms, these algorithms suffer with the traditional tradeoff between the computational complexity (time) and accuracy of the predictions. The works described in [13],[14],[15] leverage the concept of FFTs where the authors claim that FFTs help them achieve an equilibrium between the time complexity and the accuracy which means that the scoring algorithm can be designed with agility and also we can achieve a fair accuracy.

One of the other approaches to overcome the trade-off as suggested in the works [16], [17], [18] leverage the spherical aspects of the protein molecules unlike the other works which leverage the FFTs. The FFTs being very efficient though, cannot tap the multi-dimensional aspect of a molecule and hence we base our current work on the spherical aspects.
Ritchie’s work [17], [18] uses a radial basis function which is reported to be successful in optimizing the time complexity of the docking algorithm and that makes it a very promising method. This work further states that, as the distance from origin \( r \) increases owing to its radial basis function, the accuracy of field expression drastically reduces. This intuitively means that it becomes increasingly difficult to apply it on larger protein molecules. To overcome these shortcomings the authors have proposed to leverage the spherical harmonics and modified Legendre polynomials in combination which forms the radial basis function. This means that there is no decay [19] for \( r \) which is the distance from origin.

In our present work, we have extended this method by dividing \( r \) in multiple regions and ported the entire multi-dimensional pipeline onto GPU. It is to be noted that all these regions have a custom radial basis function. To put this into perspective, imagine \( \mathbb{R}^3 \) space. Now this space will be divided into multiple layers. Each layer has a scalar field which is characterized by the basis functions which in turn is composed of a combination of spherical harmonics and the radial basis function for that particular layer.

The computational pipeline for all these layers will be executed in parallel on an HPC (High Performance Computing) platform. Given that each layer will have its own coefficients which are less in number compared to the entire structure as a whole, the computation will be much efficient and faster.

II. METHODS

A. Scoring Functions

Scoring Function is characterized by the interaction energy between two molecules. In the present work we determine the scoring function based in terms of dot product of the two scalar fields that are associated with the molecules.

Let \( f_1(x), f_2(x), f_3(x), \ldots, f_d(x) \) be the scalar field functions for molecule A and \( g_1(x), g_2(x), g_3(x), \ldots, g_d(x) \) be the scalar field functions for molecule B. The scoring function would be formulated as follows:

\[
E(T, T') = \sum_{i=1}^{d} w_i f_i^T(x) g_i^{T'}(x) dx 
\]

where \( w_i \) represents the weight of the \( i \)th term,

\( T \) = rotational or translational operation on a field for molecule x and

\( f_i(x) \) is the field generated by applying the operation \( T \) to x.

B. Basis Functions

By representing the scalar fields in terms of orthogonal basis functions, the score computation becomes much faster. The basis function for \( \mathbb{R}^3 \) can be expressed

\[
B_{k,n,l}(x) = B_{k,n,l}(r, \theta, \phi) \equiv S_{k,n}(r) Y_{l,m}(\theta, \phi) \quad \text{(eq 2)}
\]

where,

\[
Y_{l,m}(\theta, \phi) \quad \text{— normalized spherical harmonics which is the angular part of basis function.}
\]

As discussed earlier the radial part \( r \) of the basis function is split into multiple intervals \( I_i \) of widths ‘\( a \)’ i.e.

\[
I_k = [ka, (k+1)a) \quad \text{for} \quad k=0,1, \ldots, \text{then}
\]

\[
S_{k,n}(r) \text{ for each region can be defined as}
\]

\[
S_{k,n}(r) = 0 \quad \text{if} \quad r \notin [ka, (k+1)a)
\]

\[
\int_0^\infty S_{k,n}(r)S_{k',n'}(r) r^2 dr = \delta_{nn'} \quad \text{—— (eq 3)}
\]

By leveraging the Gram-Schmidt process we can satisfy the aforementioned conditions:

\[
S_{k,n}(r) = \sum_{k,n} \frac{1}{N_{k,n}^2 \alpha^3} h_{k,n} \left( \frac{2}{(a)} r - 2k - 1 \right), r \in [ka, (k+1)a)
\]

\[
h_{k,n}(x) = \text{orthogonal polynomials characterized using Gram-Schmidt Process}
\]

The weight function and the intervals used for the Gram-Schmidt process are \((x+2k+1)^2\) and \([-1,1]\), respectively.

C. Fast Rotational and Translational Operators.

In order to achieve better performance, we need to obtain the following transformed coefficients \(a^R_{k,n,l,m} \) and \(b^R_{k,n,l,m} \) which facilitate configuration space search. The original coefficients characterize the new transformed coefficients and hence computing the re-expansion of the fields become much faster.

1. Rotational Operation on coefficients

Let \( a_{k,n,l,m} \) be the original coefficients and let \( R \) be the rotational operator on the field. Let \( a^R_{k,n,l,m} \) be the new rotational coefficients. As previously mentioned, we can derive the rotational coefficients using the original coefficients as follows:

\[
a^R_{k,n,l,m} = \sum_{l'=0}^{l} (-1)^{l-l'} a_{k,n,l,m} R_{l'm'}(R^{-1})
\]

\( R_{l'm'}(R) \) represents the rotational matrices for real spherical harmonics.

2. Translational Operation on coefficients

Let \( a^T_{k,n,l,m} \) represent the coefficients of a translated field.

Note that \( S_{a} \) intuitively indicates that the translation operation has been applied along the Z-axis.

The new translated coefficients \( a^T_{k,n,l,m} \) with an offset of \((0,0,\Delta z)\) can be determined as follows:

\[
\int_0^\infty S_{k,n}(r)S_{k',n'}(r) r^2 dr = \delta_{nn'}
\]

In the present work we have explored the option of porting the spherical transforms on the GPU and in the future computational references. This step significantly reduces the computational complexity and hence the computational time has significantly improved.

### III. RESULTS

We have used NVIDIA GeForce GT 710 and measured the computation time required on the GPU. The computation times are as shown in Table I.

**Table I. Computation Time on GeForce GT 710**

<table>
<thead>
<tr>
<th>Mol. A</th>
<th>Mol. B</th>
<th>Computation Time (in sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1AKZ</td>
<td>1UG(A)</td>
<td>42.4</td>
</tr>
<tr>
<td>1BRA</td>
<td>6PTI</td>
<td>51.3</td>
</tr>
<tr>
<td>1SUP</td>
<td>3SSI</td>
<td>61.1</td>
</tr>
<tr>
<td>3PTN</td>
<td>6PTI</td>
<td>67.3</td>
</tr>
</tbody>
</table>

We further measured the computation on Intel Core i7 2.3GHz and the computation times are as shown in Table II.

**Table II. Computation Time on Intel Core i7 2.3 GHz**

<table>
<thead>
<tr>
<th>Mol. A</th>
<th>Mol. B</th>
<th>Computation Time (in sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1AKZ</td>
<td>1UG(A)</td>
<td>89.2</td>
</tr>
<tr>
<td>1BRA</td>
<td>6PTI</td>
<td>103.8</td>
</tr>
<tr>
<td>1SUP</td>
<td>3SSI</td>
<td>138.3</td>
</tr>
<tr>
<td>3PTN</td>
<td>6PTI</td>
<td>115.7</td>
</tr>
</tbody>
</table>

Figure 2 below indicates the comparison of the computational times.

![Figure 2 Computational Times on GPU and CPU](image)

**IV. CONCLUSION**

We have extended out previous work [1] where we did an empirical study on porting the entire FFT pipeline for docking onto the GPU and in the present work we have explored the option of porting the spherical transforms on the HPC platform. It can be noted from (eq. 8) that the computation of the overlap is independent of the scalar fields and can be done using numerical methods and hence it is pre-computed at each step and stored in a look up table for future computational references. This step significantly reduces the computational complexity and hence the computational time has significantly improved.

### REFERENCES

Protein-Protein Docking Using Multi-Dimensional Spherical Basis Functions on High Performance Computing Platform


