F^2Dock: A Fast and Fourier Based Error-Bounded Approach to Protein-Protein Docking

Chandrajit Bajaj * Vinay Siddavanahalli †

November 3, 2006

Abstract

The functions of proteins is often realized through their mutual interactions. Determining a relative transformation for a pair of proteins and their conformations which form a stable complex, reproducible in nature, is known as docking. It is an important step in drug design, structure determination and understanding function and structure relationships. We provide a model for rigid docking and error-bounded approximation algorithms to solve the model and predict docking sites. Translational search is sped up using the Fourier domain. Shape based interactions is shown to give good results for a large range of pairs of proteins.

1 Introduction

Proteins, together with sugars, fats, oils, RNA and DNA are molecules which form the structural and functional building blocks in a cell. Through X-ray diffraction we are able to obtain near atomic resolution information of individual proteins and sometimes complexes of proteins. The RCSB Protein Data Bank [8] is a database describing proteins and RNA. It usually contains at least a list of atom coordinates and atom types. Structural interactions between these building blocks and especially between proteins is known to be responsible for their functions. For example, the actions of inhibitors and enzymes can be described through structural interactions between proteins. Hence, interactions between potential drugs and proteins or viruses form an important step in structure-based drug design. Complexes of any required pair of proteins is hard to create, crystallize and image. Also, proteins are known to be flexible and have been found in various shapes, usually known as conformations. Hence, we are interested in computationally modeling the interactions between pairs of proteins. Specifically, we are interested in obtaining the relative transformation and conformation where two proteins *fit* best, i.e., have a high affinity to each other. This is known as the *protein-protein docking* problem. If we consider large, fairly inflexible proteins, we can perform *rigid protein-protein docking* as an initial step. Rigid protein-protein docking based on structure alone has shown to be adequate for a range of proteins. But there are many other factors which contribute to the formation of a complex: electrostatics, hydrophobicity, hydrogen bonds, solvation energy etc. These, together with shape complementarity are known as affinity functions. The docking problem can be considered as a search for minimum energy complex conformations. The different terms in the energy are the Lennard Jones potential: $p(\mathbf{x}) = c_1\mathbf{x}^{-12} - c_2\mathbf{x}^{-6}$ ($\mathbf{x}, c_1, c_2$ are the distance between atoms and constants depending on the atom

*bajaj@cs.utexas.edu
†skvinay@cs.utexas.edu

type). The electrostatic interaction is modeled through the poisson equation: $-\nabla.(\epsilon\nabla\phi = \rho)$, where ϕ, ρ are the potential and charge density at a point. Electrostatics plays a role in long range interaction due to partially charged protein and solvent atoms. The change in energy due to displacing water molecules from the interface is known as desolvation energy. It is modeled as the sum of desolvation energies of individual atoms involved. For moving an atom of charge q, radius r from a region of dielectric ϵ_1 to a region of dielectric ϵ_2, the desolvation free energy is given as $\frac{q^2}{r}(\frac{1}{\epsilon_1} - \frac{1}{\epsilon_2})$. Docking energy computations also involve hydrophobicity computations, hydrogen bond formation and energies involved in conformational changes. Shape based complementarity, coupled with electrostatic compatibility is used as an initial step to obtain possible docking sites. These sites are further ranked using other energy terms. The few remaining potential docking sites are tested using energy minimization routines. We present a non-equispaced fast Fourier based algorithm for efficiently computing the initial docking search (based on shape and electrostatics complementarity). We show that it is able to accurately predict docking sites for proteins extracted from docked complexes accurately. Specifically, in this paper, we present a sum of Gaussians based model for proteins, and describe a new specification of the rigid protein-protein docking problem. An error-bounded approximation algorithm is presented and evaluated over a variety of complexes. We call our software F^2Dock, for Fast and Fourier based docking. Given 2 proteins with M_1, M_2 atoms respectively, we present an $O(max(M_1, M_2) + n^3 \log n + \rho n^3)$ algorithm to find the top ρ peaks in the docking profile. For rigid docking, both, redocking proteins obtained from a complex, and docking two proteins have the same computational complexity. We also show that for a summation of Gaussians model for the molecule where atoms are represented as Gaussian kernels, n^3 varies as $O(max(M_1, M_2))$. Our new algorithm is presented in section §3. Compared to traditional grid based algorithms, we see that our algorithm has lower computational complexity and lower memory requirements [60]. We have compared our algorithm with a traditional FFT based method on a large list of complexes, and show that our algorithm works well in practise, and accurately captures docking sites (see section §4). Our conclusions and future work in section §6.

2 Related work

There have been a wide range of both flexible and rigid docking algorithms, based on geometry or sequence or both. Some assume that the active sites (regions where other proteins interact) are already known.

Rigid docking approaches Contact surfaces between domains of methemoglobin was studied in [30] and proposed as a affinity function to use in docking. A rigid docking search carried out in [39], resulting in the program DOCK, which has evolved over the years [59] to include flexibility docking and other functions. They used spheres to represent grooves in one protein and the density of the other. It was later used in a geometric hashing scheme [23, 50, 51, 22, 18] extended to a parallel version [43]. A search strategy based on matching pairs of consistent spheres, one from each protein was used, instead of a full combinatorial search. The combinatorial search is reduced to a clique finding problem by constructing a graph of distances between atom pairs in [38]. Another graph based approach was studied in DOCK 4.0 [19]. A knob and hole detection and matching algorithm [16] was used to successfully redock the α, β subunits of hemoglobin. This was extended in [62] to allow for a further sampling along the axis containing the matched knob and hole. He performs an optimization using a *grid-based double skin layer approach* in 2D. We will further discuss this double skin layer approach later as we use a variation of it in our algorithm. A full 6D grid based search was used in [32]. They also provide a method to *uniformly* sample 3D rotational space. The grid based double skin layer approach was sped up using the Fast Fourier Transform in [35], and became the base of many variations and softwares [26], DOT [45], ZDOCK [12, 11, 13] RDOCK [44]. Hydrogen

bonds were used [46] to reduce the rotational sampling space and improve the scoring function. Spherical harmonics based approached were studied in [55, 56, 57] and [37, 36]. We have compared our algorithm to previous grid based Fourier transform and Spherical harmonics approaches in [60]. There have also been other approaches: including building webs over the surfaces and matching them using least squares fit [3], a slice based matching scheme [61], mapping surfaces to 2D matrices and detection of matching sub matrices [31] and fixing anchors and searching over other degrees of freedom: TreeDock [20]. A simulated annealing method, by choosing angles in discrete 45 degree steps and translations of 2A is used in [65] to perform a random walk and dock proteins. In [14], residues are approximated as spheres and the docking broken down as finding 5 rotations and a translation. The rotational space is sampled using simulated annealing.

Flexible docking approaches Many proteins are known to be flexible (see [15] for an example of the HIV-1 protease flexibility simulation). *Global search methods*: Global search strategies have been based on energy minimizations, heuristics based search methods, and geometric identifications of cavities indicating possible active sites. In DOCK [39] and later [17], receptor binding sites were identified as cavities and the complementary space represented as spheres. Fragments of the ligand[1] were separately bound to the active site using various distance heuristics between atoms and spheres. Fragments were then incrementally selected to form the entire ligand. An incremental approach based on shape [41], [64](HammerHead) and properties of the molecules FLEXX [54] is used to dock fragments, pruning the exponential search by retaining only a fixed set of possible conformers at each step. Other global search techniques include hydrogen bond pattern based search [47], genetic algorithms [33](GOLD), [25, 52, 34], monte-carlo/simulated annealing [29](AUTODOCK),[9],[2], molecular dynamics [49] and evolutionary programming [28]. See [48] for the performance of different algorithms in AUTODOCK. Steered molecular dynamics, using a visualization and feedback toolkit has also been studied in SMD [42]. *Backbone and domain movements*: Hinge bending in either protein or ligand is also used in docking in [58], accounting for domain movements. Conformations are sampled using a coarse set of values for torsion angles of rotating bonds in [63]. Those conformations which are sterically correct are used in a rigid body docking. Torsion and bond angles are sampled and matched using the α-shapes of the molecules [4]. *Side chain*: Using rotamer libraries, and a greedy heuristic or branch and cut algorithm, [1] performs docking of proteins with flexible side chains as a second step to rigid docking. Similar discrete side chain conformations were searched in [40]. A combination of the pseudo-brownian Monte Carlo minimization followed by flexible side chain docking using ICM was tested on a variety of bound and unbound complexes in [21]. Apart from backbone and side chain movements, loop flexibility at known active sites is handled using a Monte carlo, simulated annealing based docking approach in [7]. See the *connexions project* at http://cnx.rice.edu/content/m11464/latest/ for a summary of flexible docking.

3 Algorithm details

Consider two proteins A and B, with M_1, M_2 atoms respectively. We represent the molecules using Gaussian kernels, construct the double skins used for complementary space docking and derive a new model for docking.

3.1 Affinity functions

The affinity functions are modeled as Radial Basis Functions (RBFs) to facilitate using Fourier transforms to efficiently solve the docking problem.

[1] Often, one of the proteins is smaller and termed as a ligand, and generally taken to be more flexible.

Molecule representation We use the sum of Gaussian's representation to model our proteins. An atom centered at $\mathbf{x}_c$, with a van der Waal's radius of r, is modeled as an isotropic Gaussian kernel: $g(\mathbf{x}) = e^{-\beta(\frac{(\mathbf{x}-\mathbf{x}_c)^2}{r^2}-1)}$. The decay rate of the kernel is controlled by β. A value of 2.3 is used in the literature [27] to approximate the solvent excluded surface at an isovalue of 1. By lowering this parameter, we can model molecules at lower resolutions [5].

3.1.1 Shape complementarity

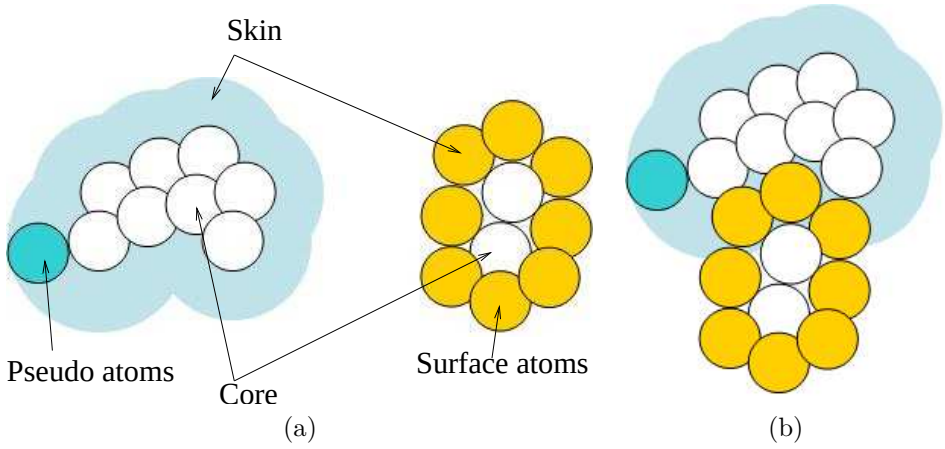


Figure 1: (a) Skin and Core regions for complementary space docking. Atoms are drawn as solid circles. The skins regions are colored while the core regions are white. (b) A possible docking of the molecules show a large overlap between the grown layer of the first and the surface atoms of the second.

For shape based docking we will try to maximize the overlap of the surface of protein B with the complementary space of A. The *double skin layer* approach is used here. It was introduced in [62] for 2D, [32] for 3D, sped up using Fast Fourier Transforms in [35], and extended to complex space in [11]. We define two *skin regions*: 1). The *surface skin* of B, which is the density function of the set of surface atoms of B, and 2). The complementary region of A, defined by a *grown skin region*, by introducing a 1-layer of pseudo-atoms on the surface of A. The atoms of A and the inner atoms of B form *core regions*. These regions are shown in figure 1. We used an adaptive grid based algorithm to construct these regions [60]. To maximize skin overlaps and to minimize overlaps of the cores, we assign positive imaginary weights to the core atoms and positive real weights to the skin atoms/pseudo-atoms. An integral of the superposition of the molecules has two real contributions: the core overlaps contribute negatively and the skin overlaps contribute positively. Since the symmetry is broken in the imaginary part of the integral (one contribution is due to atom - atom overlap and another from pseudo-atom - atom overlap), we currently do not use this value, although others in the literature assign this a 'smaller' negative potential. The weighted sum of Gaussians function definition of a molecule (of M atoms), with its associated *skin region* can be expressed as a sum of two functions: $f_{SC}(\mathbf{x}) = f^{Re}(\mathbf{x}) + f^{Im}(\mathbf{x}) = \sum_{k=1}^{M_{Re}} c^{Re} g(\mathbf{x} - \mathbf{x}_{Re,k}) + \sum_{k=1}^{M_{Im}} c^{Im} g(\mathbf{x} - \mathbf{x}_{Im,k}) = \sum_{k=1}^{M} c_k g(\mathbf{x} - \mathbf{x}_k)$. Here, g is the Gaussian function located at each atom (or pseudo atom) and (SC) stands for shape complementarity. The

weights $\{c_k \in \{c^{Im}, c^{Re}\}, k = 1..M\}$ are either positive imaginary or positive real. See [13] for an extension to shape complementarity to *pairwise shape complementarity*.

3.1.2 Electrostatics interactions

Similar to the procedure used for shape complementarity, Gabb et. al. [26] have shown how to introduce the electrostatics term. The first protein's electric potential is computed and matched against the charges in the other. This can also be sped up using a Fourier based algorithm. Charge assignments are made using APBS [6]. A new affinity function f_E^1 is defined as $\sum_k q_k \frac{1}{E(\mathbf{x}-\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k)}$, where $E(\mathbf{x})$ is the distance dependent dielectric constant [26]. The corresponding function for the second protein is $f_E^2 = \sum_{k=1}^{M_2} q_k \delta(\mathbf{x} - \mathbf{x}_k)$. In [11], they use these functions multiplied with a imaginary and a negative imaginary weight respectively.

3.2 Rigid docking model specification

Let A, B's affinity functions sum be denoted as f_1, f_2, ($f = f_{SC} + f_E$, or $f = f_{SC}$ depending on whether we use electrostatic interactions or not.) and T, Δ be translational and rotational operators. If the user considers a potential docking site as one where the overlap potential is over a threshold τ, then the rigid protein-protein docking solution, using our affinity functions definition, is expressed as the set of triplets:

$$\{(\mathbf{t}, \mathbf{r}, s) : (s = Re(\int_{\mathbf{x}} f_1(\mathbf{x}) T_{\mathbf{t}}(\Delta_{\mathbf{r}}(f_2(\mathbf{x})))d\mathbf{x})) \geq \tau\} \tag{3.1}$$

3.3 Search

We solve equation (3.1) using Fourier series expansions. First, we express the integral as a uniform sum of compactly supported functions and provide an adaptive algorithm to search for regions where the scoring function exceeds the threshold provided by the user.

3.3.1 Fourier series expansions

Any periodic bounded function can be expanded as a Fourier series. For example, a periodic function in $[-1/2, 1/2]$ can be expressed as: $q(x) = \sum_{j=-\infty}^{\infty} \omega_j e^{2\pi i j x}$, where the coefficients $\omega_j = \int_{-1/2}^{1/2} q(x) e^{-2\pi i j x} dx$. Let I_n denote a 3D grid of indices: $\{k : [-n/2..n/2)^3, k \in I\}$. Let us expand the kernel function in its fourier series form: $g(\mathbf{x} - \mathbf{x}_k) = \sum_{\boldsymbol{\omega} \in I_\infty} G_{\boldsymbol{\omega}} e^{2\pi i (\mathbf{x}-\mathbf{x}_k).\boldsymbol{\omega}}$. Hence, the affinity function $f(\mathbf{x}) = \sum_{k=1}^{M} c_k g(\mathbf{x} - \mathbf{x}_k)$ can be expressed as $f(\mathbf{x}) = \sum_{k=1}^{M} c_k (\sum_{\boldsymbol{\omega} \in I_\infty} G_{\boldsymbol{\omega}} e^{2\pi i (\mathbf{x}-\mathbf{x}_k).\boldsymbol{\omega}})$. Rearranging terms, we obtain: $f(\mathbf{x}) = \sum_{\boldsymbol{\omega} \in I_\infty} G_{\boldsymbol{\omega}} e^{2\pi i \mathbf{x}.\boldsymbol{\omega}} \sum_{k=1}^{M} c_k e^{-2\pi i \mathbf{x}_k.\boldsymbol{\omega}}$. Let us denote the second terms by $C_{\boldsymbol{\omega}}$. Hence, $f(\mathbf{x}) = \sum_{\boldsymbol{\omega} \in \mathbf{I}_\infty} \mathbf{G}_{\boldsymbol{\omega}} \mathbf{C}_{\boldsymbol{\omega}} \mathbf{e}^{\mathbf{2\pi i x}.\boldsymbol{\omega}}$. Similarly: $f(\mathbf{x} - \mathbf{y}) = \sum_{\boldsymbol{\omega} \in \mathbf{I}_\infty} \mathbf{G}_{\boldsymbol{\omega}} \mathbf{C}_{\boldsymbol{\omega}} \mathbf{e}^{\mathbf{2\pi i (x-y)}.\boldsymbol{\omega}}$.

Expanding f_1, f_2 using the above series, for a given rotation, with the molecules scaled to lie in $\pi^3 = (-0.5..0.5]^3$ for simpler mathematical notation, the scoring integral in equation (3.1) reduces to

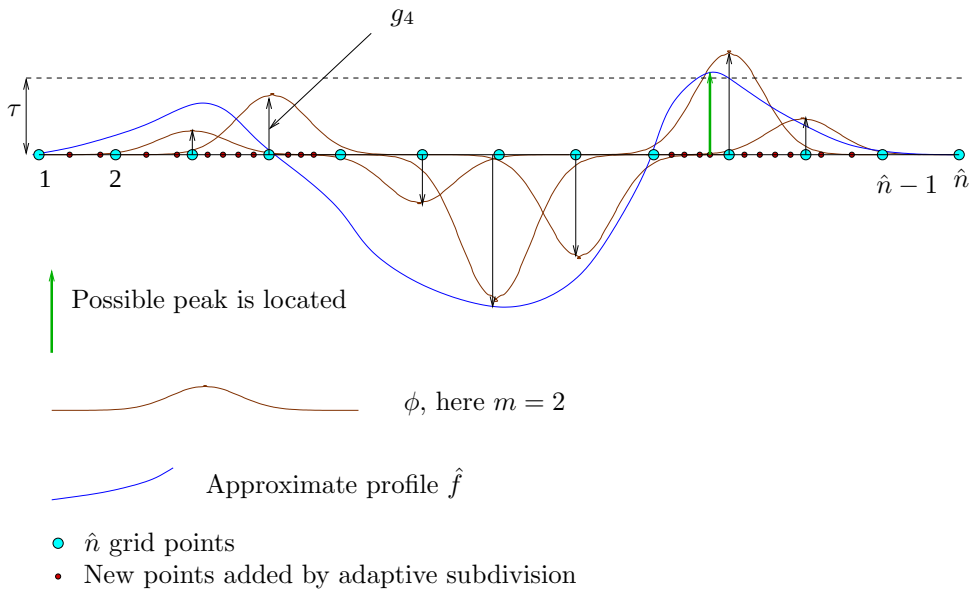


Figure 2: The docking peak search can be represented as finding the peak positions and values in a grid of overlapping splines.

$\int_{\mathbf{y}\in\pi^3} f_1(\mathbf{y})(\Delta_{R_f}(f_2))(\mathbf{x}-\mathbf{y})d\mathbf{y},\ \forall\ \mathbf{x} = \int_{\mathbf{y}\in\pi^3} \sum_{\boldsymbol{\omega}_1\in I_\infty} G_{\boldsymbol{\omega}_1} C_{\boldsymbol{\omega}_1} e^{2\pi i \mathbf{y}.\boldsymbol{\omega}_1} \sum_{\boldsymbol{\omega}_2\in I_\infty} G_{\boldsymbol{\omega}_2} C'_{\boldsymbol{\omega}_2} e^{2\pi i(\mathbf{x}-\mathbf{y}).\boldsymbol{\omega}_2} d\mathbf{y}$. Since $\int_{-1/2}^{1/2} e^{2\pi i y(a-b)} = 1$ if $a=b$ and 0 otherwise, the integral reduces to $\sum_{\boldsymbol{\omega}\in I_\infty} G^2_{\boldsymbol{\omega}} C_{\boldsymbol{\omega}} C'_{\boldsymbol{\omega}} e^{2\pi i \mathbf{x}.\boldsymbol{\omega}}$.

3.3.2 Approximations

We make three approximations in computing the above coefficients. Since the truncated Gaussian is a decaying kernel, we choose to compute only the first $(-n/2..n/2]^3$ Fourier coefficients. The parameter n is chosen to satisfy a user required accuracy in the docking profile. If we include electrostatics, the decay should be even slower, and hence, the same bounds derived for shape complementarity should be sufficient. The current analysis, though, is based on shape complementarity. The Fourier coefficients of the atoms centers, $C_{\boldsymbol{\omega}}, C'_{\boldsymbol{\omega}}$ are approximated as $\hat{C}_{\boldsymbol{\omega}}, \hat{C}'_{\boldsymbol{\omega}}$, computed using a Non-equispaced Fourier Transform (NFFT) algorithm given in [53] (Very briefly, the NFFT algorithm computes an approximation to Fourier coefficients when input data is not uniformly sampled). The truncated Gaussian is a tensor product kernel. We compute the Fourier coefficients of a 1D Gaussian kernel of size n using MAPLE numerically. The Fourier coefficients of the truncated Gaussians are now approximated as the tensor product $\hat{G}_{\boldsymbol{\omega}}$. Hence, we approximate the scoring integral as $\sum_{\boldsymbol{\omega}\in I_n} \hat{G}^2_{\boldsymbol{\omega}} \hat{C}_{\boldsymbol{\omega}} \hat{C}'_{\boldsymbol{\omega}} e^{2\pi i \mathbf{x}.\boldsymbol{\omega}} = \sum_{\boldsymbol{\omega}\in I_n} \hat{F}_{\boldsymbol{\omega}} e^{2\pi i \mathbf{x}.\boldsymbol{\omega}}$.

3.3.3 Inverse peak search

Given the function $\hat{f}(\mathbf{x}) = \sum_{\boldsymbol{\omega} \in I_n} \hat{F}_{\boldsymbol{\omega}} e^{2\pi i \mathbf{x}.\boldsymbol{\omega}}$, we are required to compute $\{(\mathbf{x}, s) : s = Re(\hat{f}(\mathbf{x})) \geq \tau\}$. A 3D IFFT of $\hat{F}_{\boldsymbol{\omega}}$ yields the docking profile $\hat{f}(\mathbf{x})$ at a uniform sampling. If we have prior knowledge on the smoothness of the profile, we can zero pad $\hat{F}_{\boldsymbol{\omega}}$ (if necessary) and obtain the profile at a sufficient sampling. This would generally lead to high computational and memory requirements. Instead, we perform an adaptive computation of $\hat{F}_{\boldsymbol{\omega}}$, progressively zooming in on regions where the threshold τ is satisfied. Using the NFFT algorithm in [53], we make the following approximation: $\hat{f}(\mathbf{x}) \approx \hat{g}(\mathbf{x}) = \sum_{\mathbf{k} \in I_{\hat{n},m}(\boldsymbol{\omega}_\mathbf{j})} g_k \phi(\boldsymbol{\omega}_\mathbf{j} - \mathbf{k}/\hat{n}),\ (\mathbf{j} \in I_n,\ \hat{n} = \alpha n,\ \alpha \approx 2, I_{\hat{n},m}(\boldsymbol{\omega}_\mathbf{j}) = \{\mathbf{j} \in I_{\hat{n}} : \hat{n}\boldsymbol{\omega}_\mathbf{j} - m \leq \mathbf{j} \leq \hat{n}\boldsymbol{\omega}_\mathbf{j} + m\})$. This is schematically represented in 1D in figure 2. Obtaining regions which are above a certain threshold is now reduced to finding roots of the previous polynomial. If we use a cubic bspline function for ϕ with a support width of 5, it requires the root of a 7x7x7 system of degree 5 equations. We instead adaptively compute regions which satisfy our docking threshold using an adaptive search algorithm. We initially start with the $\hat{n}^3$ grid of ϕ as a set of intervals. We determine using a simple procedure if any interval can potentially contain a value greater than the docking threshold and, if so, subdivide and recursively search the sub intervals. Consider any interval I. There are multiple ϕ functions whose summation determine the function in I. If we change these ϕ, such that positive ones centered outside I come closer by one interval width, negative ones shift away from I by one interval width and positive ones centered inside I are given its maximum value, the sum of the new function at the interval endpoints defines an upper bound for the true function inside I. This gives us a criterion as to whether we need to further subdivide and check an interval or not.

Using a FFT for the 1st step: The docking profile is usually a thin closed surface with zeros on the outside and large negatives on the inside. Hence, in the very first step of the algorithm, a large number of regions are removed from further consideration. We are able to convert the algorithm in the first level into an FFT of size n^3. This is an efficient way of speeding up algorithm 1. We provide the analysis in 1D, which can be easily extended to 3D. Consider an interval $[i, i+1]$, with gaussian functions ϕ_k, where $i - m \leq k \leq i + 1 + m$, both positive and negative. Let the extent of the ϕ_k be m on each side of k. Let us construct a new function ψ_k by raising the value of ϕ_k to $max(\phi_k, \phi_{k+1}, \phi_{k-1})$ on the $\hat{n}^3$ grid. This gives us the following simple observation:

Lemma The summation of the ψ at a point k in the low resolution grid of the gaussian centers is always greater than the summation of ϕ at any point in any interval which includes k.

The summation of functions ψ does not include any shifts. Hence, we can consider this as a convolution of ψ with g, the input to algorithm 1. Convolutions can be quickly computed in $O(n^3 \log n)$ using the FFT in a single step. This step eliminates most regions outside the overlap of molecules and core clashes from the docking profile. Hence, the adaptive search is limited to a narrow region where the surface contacts occur.

3.4 Error and complexity analysis

The exact docking profile is given as $\sum_{\boldsymbol{\omega} \in I_\infty} G^2_{\boldsymbol{\omega}} C_{\boldsymbol{\omega}} C'_{\boldsymbol{\omega}} e^{2\pi i \mathbf{x}.\boldsymbol{\omega}}$ while we compute $\sum_{\boldsymbol{\omega} \in I_n} \hat{G}^2_{\boldsymbol{\omega}} \hat{C}_{\boldsymbol{\omega}} \hat{C}'_{\boldsymbol{\omega}} e^{2\pi i \mathbf{x}.\boldsymbol{\omega}}$ approximately using the NFFT. Given a user defined error ϵ, we derive the value of n such that $|| \sum_{\boldsymbol{\omega} \in I_\infty} G^2_{\boldsymbol{\omega}} C_{\boldsymbol{\omega}} C'_{\boldsymbol{\omega}} e^{2\pi i \boldsymbol{\omega}.\mathbf{x}} - \sum_{\boldsymbol{\omega} \in I_n} \hat{G}^2_{\boldsymbol{\omega}} \hat{C}_{\boldsymbol{\omega}} \hat{C}'_{\boldsymbol{\omega}} e^{2\pi i \boldsymbol{\omega}.\mathbf{x}} + \epsilon_1 ||_2 \leq \epsilon$. The error ϵ_1 is due to the NFFT algorithm, which converges exponentially

Algorithm 1 Inverse adaptive peak search

Inputs are:
    -$\hat{n}^3$: number of frequencies
    -h: accuracy of peak position
    -ϕ: Compactly supported smooth decaying function at each $k \in I_{\hat{n}}$
    -g_k: coefficients of ϕ
    -τ: threshold for docking score
    -$\{(val, pos)\}$: Current output peak regions and scores.
Preprocessing: [Interval set: $I = intervals(k)$]

while $I \neq \emptyset$ **do**
  $interval \leftarrow I.next()$

  if interval.isLowRes() **then**
    $t \leftarrow 0$
    $\{\phi\} \leftarrow interval.overlapping\phi()$

    for $\phi \in \{\phi\}$ **do**
      if $interval.isOutside(\phi)$ **then**
        if $\phi > 0$ **then**
          $t \leftarrow t + \phi(interval.cIdx(\phi.center))$
        else
          $t \leftarrow t - \phi(interval.fIdx(\phi.center))$
        end if
      else
        if $\phi > 0$ **then**
          $t \leftarrow t + \phi_{max}$
        else
          $t \leftarrow t - \phi(interval.fIdx(\phi.center))$
        end if
      end if
    end for

    if (t $> \tau$) **then**
      $I \leftarrow I \cup interval.subIntervals()$
    end if
  else
    $update(\{(val, pos)\}, interval)$
  end if

end while

Output: $[\{(val, pos)\}]$

in their sampling parameters. Since for docking, we are satisfied with errors around 1% to 0.1%, we do not consider the cost due to sampling. From [5], lemma 1, the number of Fourier coefficients n required for a relative accuracy ϵ is:

$$n = min(\hat{n}) : \sum_{\boldsymbol{\omega} \in I_{\hat{n}}} G_{\boldsymbol{\omega}}^4 \geq \frac{V}{2\pi} - \frac{Mmin_j(|(c^A c^B)_j|^2)V}{||c^A c^B||_1^2}(\frac{\epsilon}{3})^2,\ V = \int g^2\ in\ (-0.5..0.5]^3,\ M = max(M_1, M_2)$$

Size and resolution vs. cost: The size of the molecule and the resolution we are interested affects the cost of the algorithm. We propose to do a hierarchical search using multiple resolutions. The resolution parameter can affect the rate of decay of the Gaussian (b) and the number of centers being considered. Using [5], lemma 3, n varies as $M^{1/3}\sqrt{b/2}\epsilon^{3/2}$.

Cost of the algorithm: The Fourier coefficients of the truncated Gaussian $G_{\boldsymbol{\omega}}$ are precomputed to full precision using MAPLE. The NFFT algorithm has been implemented to compute $C_{\boldsymbol{\omega}}, C'_{\boldsymbol{\omega}}$, the Fourier coefficients of the sum of centers in $O(M_1 + n\log n), O(M_2 + n\log n)$, respectively, where n, computed using our error analysis, is shown to be $O(max(M_1^{1/3}, M_2^{1/3}))$. Computation of the product of the coefficients is costs $O(n^3)$. An IFFT of the product yields the docking profile on a n^3 resolution grid and costs $n^3 \log n$. To obtain the peaks in a higher resolution, without using a larger grid, we can perform the inverse peak search algorithm described in §3.3.3. If there are η regions which satisfy the threshold τ in the docking profile, they can be located and computed in a grid of size 2^h in $O(\eta h n)$. Hence, the computational cost of our docking algorithm grows linearly with the number of atoms in the molecules. Since each rotation is performed independently, the total cost is $O(max(M_1, M_2)N_R^3)$, where N_R^3 is the number of sampled rotations. The memory cost is $O(max(M_1, M_2))$. Compared to traditional grid based algorithms, we see that our algorithm has lower computational costs and lower memory requirements [60].

4 Results

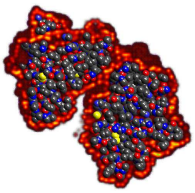
We have computed the docking predictions for a set of 71 complexes using different affinity functions and flexible models. We have also taken a set of three test cases to compare, at each step, with a traditional grid based approach. The three complexes we use to compare are: Hyhel-5 fab complexed with bobwhite quail lysozyme (PDB:1BQL.PDB), Idiotype-anti-idiotype fab complex (PDB: 1IAI. PDB) and an influenza virus hemagglutinin complexed with a neutralizing antibody (PDB: 2VIR .PDB). We will simply refer to these complexes as **complex 1, complex 2, complex 3** respectively (see figure 3).

5 Soft Docking

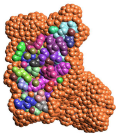
For soft docking, we first use shape complementary as the only affinity function in scoring. Then we investigate the effects of introducing electrostatics interactions.

5.1 Comparison with Grid-based FFT Algorithm

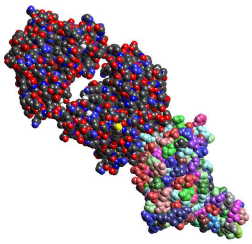
Difference in docking profiles: We compared the difference in the energy of the docking profile we obtain to that obtained from a 256^3 grid. From tables 5.1.1,5.1.2 and 5.1.3, we see that the differences are very small for relatively fewer Fourier coefficients. A slice from the docking profiles the two methods are shown in figure 4. From the figure, we can see that the shape of the profile and the location of the peaks are well conserved in our algorithm.



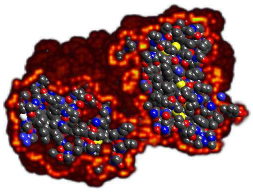
(a) An immunoglobulin from the influenza virus (IGG1, LAMBDA) (PDB:2VIR.PDB)



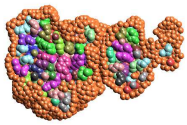
(b) Hemagglutinin (PDB:2VIR.PDB)



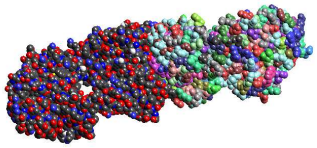
(c) **Complex 3:**influenza virus hemagglutinin complexed with a neutralizing antibody (PDB:2VIR.PDB)



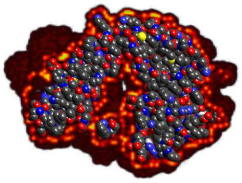
(d) Idiotypic FAB 730.1.4 (IGG1) of virus neutralizing antibody(PDB:1IAI.PDB)



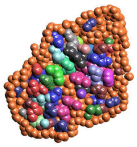
(e) Anti-Idiotypic Fab 409.5.3 (Igg2A)(PDB:1IAI.PDB)



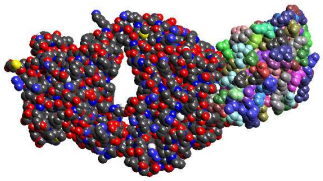
(f) **Complex 2:**Idiotype-anti-idiotype fab complex(PDB:1IAI.PDB)



(g) Hyhel-5 FAB (PDB:1BQL.PDB)



(h) Bobwhite Quail Lysozyme (PDB:1BQL.PDB)



(i) **Complex 1:**Hyhel-5 fab complexed with bobwhite quail lysozyme (PDB:1BQL.PDB)

Figure 3: The three complexes we have used as test cases. In the first column, we show one protein of the complex with the grown surface in red. The second column shows the surface atoms in light brown. We show a cut away to reveal the two skins. In the last column, the complexed structures are shown. The first molecule is colored using standard atom colors while the atoms in the second molecule are colored by their residue type to differentiate the two molecules in the complex. The three molecules/skins in the first column had 3263/4519, 3342/4555, 3243/4308 atoms/kernel centers respectively. In the second column, there were 988/1087, 4956/1719 and 5293/469 surface and interior atoms respectively.

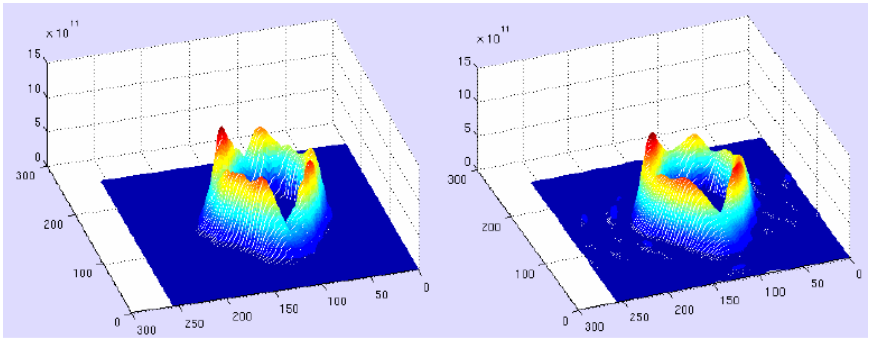


Figure 4: Comparison of a slice from our docking profile compared with that of a FFT based algorithm on a 256^3 grid. The shape and location of peaks is shown to be well conserved.

Number of freq.	$\beta = -0.5$		$\beta = -1$	
	l^2	l^∞	l^2	l^∞
16^3	6.3364	3.0409	9.9454	3.5909
20^3	3.9761	1.2994	7.9016	1.7434
32^3	1.1991	0.2889	5.3285	0.5909

Table 5.1.1: Difference in energy, in %, for **complex 1**, with $\alpha = m = 2$ as the NFFT parameters

Number of freq.	$\beta = -0.5$		$\beta = -1$	
	l^2	l^∞	l^2	l^∞
16^3	4.5203	3.5743	6.8897	4.2208
20^3	2.5131	1.4592	5.1096	1.8793
32^3	0.8462	0.2480	3.6941	0.5297

Table 5.1.2: Difference in energy, in %, for **complex 2**, with $\alpha = m = 2$ as the NFFT parameters

Number of freq.	$\beta = -0.5$		$\beta = -1$	
	l^2	l^∞	l^2	l^∞
16^3	4.8228	2.0457	7.7806	2.3983
20^3	2.7570	0.8029	6.0601	1.0721
32^3	0.9504	0.2017	4.6343	0.4111

Table 5.1.3: Difference in energy, in %, for **complex 3**, with $\alpha = m = 2$ as the NFFT parameters

Comparison with FFT grid-based algorithm using larger set of complexes: We compare results for redocking 71 complexes using shape complementarity to a traditional, expensive 128^3 grid FFT based docking. We find where the true position lies in our ranking of peaks. We use an accuracy of 2 Åbetween what we have and the true docked complex position while searching for the peaks. We see that the best results are obtained with a rate of decay around the recommended value of -2.3. These results have been plotted for comparison in figures 5, 6 and 7. Our new algorithm uses far lesser time and memory than the FFT grid-based method. In this experiment, we used Euler angles as they are used by many groups who perform the FFT grid-based docking search.

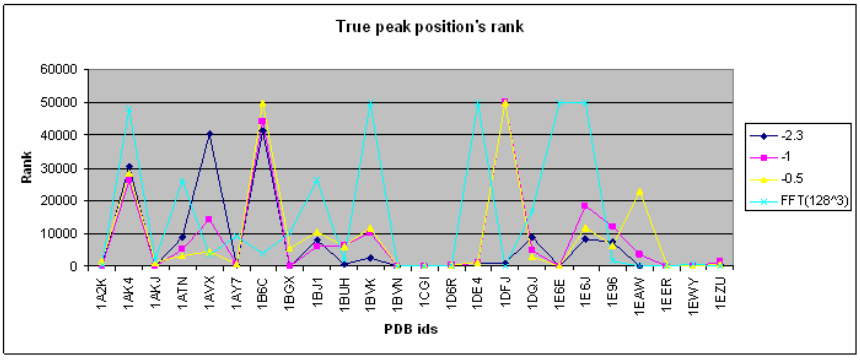


Figure 5: Comparison with docking with various rates of decay using 12 degrees rotational sampling, 32 fourier coefficients and a 128^3 FFT

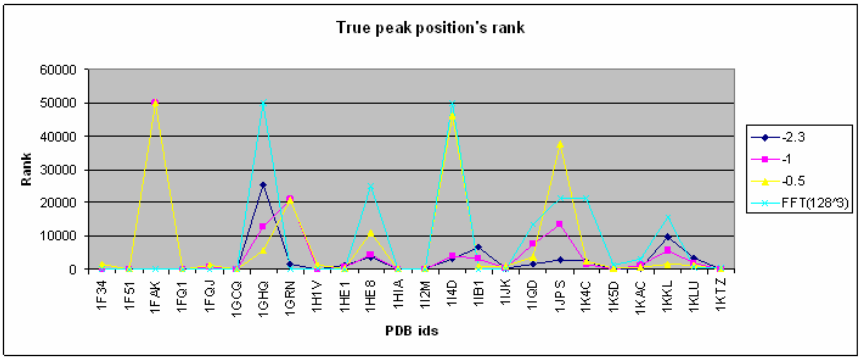


Figure 6: Comparison with docking with various rates of decay using 12 degrees rotational sampling, 32 fourier coefficients and a 128^3 FFT

For comparisons to other grid based methods, please see [60].

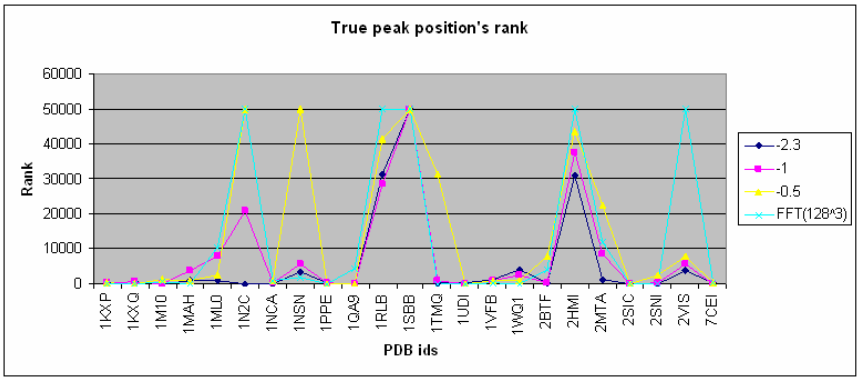


Figure 7: Comparison with docking with various rates of decay using 12 degrees rotational sampling, 32 fourier coefficients and a 128^3 FFT

5.2 Redocking

In redocking, the two proteins taken from the bound complex are computationally docked. From the list of 71 complexes, 1E96.PDB and 1F51.PDB could not be assigned charges using APBS [6] and were not considered any further. Our results for redocking, using shape complementarity is shown in tables 5.2.1, 5.2.2. We used a rotational sampling of 20°. The number of Fourier coefficients were around 32^3, which is seen to retain around 95% of the energy in the docking profile. From the table, we see that we were able to predict good peaks in the top 2000 for 38 complexes, and could not predict any good positions for 1VFB.PDB, 1EER.PDB, 1E6J.PDB and 1HE8.PDB. We present the results sorted according to the surface area of the complex. It is interesting to note that smaller surface areas resulted in more number of good predictions. To compute the RMSD, we used all atoms of the ligand in the interface. The cutoff RMSD used was 5Å. But 22 complexes were found within 2ÅRMSD, 47 within 3Åand 58 within 4Å. In this thesis, we do not further refine the search from these set of peaks to predict an actual complex using energetics.

5.3 Bound-unbound Docking

Since we are interested in flexible protein-protein docking, we first consider the effectiveness of soft docking on bound-unbound docking. For this set of experiments, we take one protein from the docked complex and a known independent structure of the other protein. On doing an analysis of the interface of the files obtained from APBS, we found that 1EAW, 1BVK, 1ATN, 1EZU, 2BTF, 2VIS, 1KXP and 1BGX had too many missing atoms in the region. Hence, we removed them from the list of 69 complexes and present results for the remaining in tables 5.3.1 and 5.3.2. We used the same parameters as the redocking case. Out of the 60 complexes tested, soft docking found peaks (within 5ÅRMSD) for 23 within the top 2000 predictions, 39 within the top 10000, 53 within the top 50000 and failed for 8 cases (1GCQ, 1I2M, 1DQJ, 1FAK, 1EER, 1ML0, 2HMI and 1N2C). Unlike redocking, we get only 2 complexes with peaks within 5ÅRMSD of the actual, 23 within 3Åand 42 within 4Å.

PDB ID	Rank	N Peaks	Best RMSD	PDB ID	Rank	N Peaks	Best RMSD
1GCQ	12	22	2.159996	1AVX	6931	5	2.442265
1AY7	484	20	1.664378	1BUH	282	7	2.709604
1PPE	15	12	1.703695	1GRN	167	6	2.641065
1KTZ	539	22	2.008881	1EWY	793	11	2.065424
1QA9	1877	7	2.355009	1F34	1	3	3.562068
7CEI	2419	12	1.297948	1B6C	154	9	2.197851
1D6R	114	9	2.007833	1IJK	4423	4	1.838018
1HIA	249	12	1.545003	1BVN	827	8	2.572958
1CGI	53	9	1.96838	1A2K	2165	6	2.092011
1EAW	73	4	2.380758	1TMQ	1624	7	1.992915
2SNI	0	12	1.556862	1GHQ	1611	2	3.700142
1UDI	262	10	2.188662	1M10	1029	1	4.026118
1KAC	5967	8	1.685013	1FQJ	395	6	2.61232
2SIC	22	16	1.81577	1I2M	67	15	1.903976
1HE1	2	12	1.023882	1WQ1	22	3	3.372812
1VFB	-	0	5.527322	1KXQ	105	2	2.524465
1AK4	1146	11	1.148174	2BTF	21725	1	3.261152
1BVK	6957	1	4.543537	1MAH	7466	1	4.040359

Table 5.2.1: Protein-protein redocking results using shape complementarity ..1. 'Rank' is the best rank among all predicted positions whose RMSD was less than 5Å. 'N Peaks' is the number of peaks in the predicted set which were less than 5ÅRMSD from the known position. 'Best RMSD' is the lowest RMSD among all the peaks that were shortlisted. If there were no good predictions in the top 50,000 that we choose to keep, we enter a '-' in the column for 'Rank'. The proteins are ranked by their surface areas.

PDB ID	Rank	N Peaks	Best RMSD	PDB ID	Rank	N Peaks	Best RMSD
1FQ1	933	7	2.005644	1EER	-	0	8.973606
1DFJ	0	11	1.80656	1RLB	13822	2	1.174494
1SBB	3139	10	1.528597	1IB1	3910	2	3.451666
1DQJ	4432	2	1.907411	1AKJ	886	7	1.110888
2MTA	2266	5	2.753511	2VIS	16781	4	2.465282
1EZU	39397	1	2.363247	1K5D	120	4	2.453245
1IQD	10752	5	3.519911	1H1V	1962	4	1.288332
1K4C	11343	4	2.095662	1E6J	-	0	5.554778
1FAK	17	6	1.588706	1NCA	5692	5	2.245905
1E6E	18	5	2.605772	1ML0	32582	1	4.319139
1ATN	2815	5	1.767566	1KXP	236	4	3.785924
1NSN	31399	1	4.938059	1HE8	-	0	6.315094
1KKL	11	2	1.919542	1BGX	43	1	3.330236
1I4D	17649	2	4.391554	1DE4	28305	2	3.448057
1JPS	1327	3	2.838963	2HMI	47242	1	3.468522
1KLU	11294	7	2.061819	1N2C	4929	2	4.784001
1BJ1	17946	1	3.221278				

Table 5.2.2: Protein-protein redocking results using shape complementarity..2. 'Rank' is the best rank among all predicted positions whose RMSD was less than 5Å. 'N Peaks' is the number of peaks in the predicted set which were less than 5ÅRMSD from the known position. 'Best RMSD' is the lowest RMSD among all the peaks that were shortlisted. If there were no good predictions in the top 50,000 that we choose to keep, we enter a '-' in the column for 'Rank'. The proteins are ranked by their surface areas.

PDB ID	Rank	N Peaks	Best RMSD	PDB ID	Rank	N Peaks	Best RMSD
1GCQ	-	0	-	1AK4	354	8	2.257936
1AY7	2758	14	2.579163	1AVX	42215	1	4.285829
1PPE	61	22	2.823658	1BUH	773	5	2.471546
1KTZ	12874	10	3.418981	1GRN	3218	1	4.973951
1QA9	3192	3	4.286313	1EWY	446	8	3.288623
7CEI	99	15	2.663766	1F34	4	2	3.923744
1D6R	4383	9	2.399734	1B6C	571	5	2.589875
1HIA	2902	28	2.943805	1IJK	9540	3	3.40416
1CGI	9488	1	4.497934	1BVN	1382	3	4.057453
2SNI	2	8	1.637314	1A2K	1137	3	3.538182
1UDI	15306	5	3.524703	1TMQ	499	2	2.616295
1KAC	6659	4	3.198683	1GHQ	16520	4	4.210506
2SIC	9	10	2.057806	1M10	11367	2	3.410315
1HE1	1	9	2.770525	1FQJ	5862	6	3.556944
1VFB	24581	2	3.448647	1I2M	-	0	-

Table 5.3.1: Bound-unbound docking results using shape complementarity..1. 'Rank' is the best rank among all predicted positions whose RMSD was less than 5Å. 'N Peaks' is the number of peaks in the predicted set which were less than 5ÅRMSD from the known position. 'Best RMSD' is the lowest RMSD among all the peaks that were shortlisted. If there were no good predictions in the top 50,000 that we choose to keep, we enter a '-' in the column for 'Rank'. The proteins are ranked by their surface areas.

PDB ID	Rank	N Peaks	Best RMSD	PDB ID	Rank	N Peaks	Best RMSD
1WQ1	34	5	2.422902	1JPS	9655	5	2.807827
1KXQ	86	2	2.694885	1KLU	16336	10	2.598834
1MAH	26	5	2.486886	1BJ1	17610	1	3.300814
1FQ1	345	3	3.59622	1EER	-	0	-
1DFJ	38	6	3.405731	1RLB	16708	3	3.499713
1SBB	491	7	1.878689	1IB1	1047	1	4.426045
1DQJ	-	0	-	1AKJ	9009	4	3.14449
2MTA	2489	4	2.255889	1K5D	4714	2	3.937634
1IQD	40224	1	3.324618	1E6J	14233	3	2.838202
1K4C	18016	6	2.286136	1NCA	24641	1	4.647226
1FAK	-	0	-	1ML0	-	0	-
1E6E	26	3	3.238776	1HE8	29673	1	3.445586
1NSN	214	45	2.157744	1DE4	30009	1	4.606551
1KKL	1722	3	4.509084	2HMI	-	0	-
1I4D	3036	1	4.605807	1N2C	-	0	-

Table 5.3.2: Bound-unbound docking results using shape complementarity..2. 'Rank' is the best rank among all predicted positions whose RMSD was less than 5Å. 'N Peaks' is the number of peaks in the predicted set which were less than 5ÅRMSD from the known position. 'Best RMSD' is the lowest RMSD among all the peaks that were shortlisted. If there were no good predictions in the top 50,000 that we choose to keep, we enter a '-' in the column for 'Rank'. The proteins are ranked by their surface areas.

5.4 Electrostatics Interactions

Electrostatics based affinity function is defined using a model by Gabb [26]. The dielectric value is set as 4 for distances less than 6 Åfrom the center of atoms, 80 for ¿ 8 and a linear interpolation in between. We add this term to our docking score and tabulate the new results in tables 5.4.1 and 5.4.1. For each complex, we used a cutoff of 5Åas the RMSD required between the locations of ligand interface atoms in the predicted position vs the known crystal structure. This time, we use a more reasonable cutoff of 4000 positions only. Similar values of 4000, 4000-7000 have been cited in [21, 10]. We see that adding electrostatics enables us to get hits in the top 4000 positions, reducing the computations required in finer docking stages.

PDB ID	Rank	N Peaks	Best RMSD	PDB ID	Rank	N Peaks	Best RMSD
1GCQ	788	5	4.410048	1AK4	2641	9	2.105113
1AY7	473	16	2.660349	1AVX	588	2	3.300274
1PPE	1677	12	2.858134	1BUH	299	6	2.857323
1KTZ	822	11	3.367324	1GRN	-	0	-
1QA9	41	7	3.695077	1EWY	463	9	3.258474
7CEI	1532	11	2.118427	1F34	643	3	3.879021
1D6R	1413	6	2.522827	1B6C	804	7	2.342338
1HIA	73	11	2.120566	1IJK	419	6	2.530123
1CGI	2974	1	4.757587	1BVN	1894	3	3.729099
2SNI	392	11	1.892177	1A2K	330	5	3.103228
1UDI	3603	8	3.123136	1TMQ	59	7	2.068443
1KAC	2615	7	2.097499	1GHQ	1636	2	3.876798
2SIC	58	15	2.332064	1M10	399	1	4.582629
1HE1	4	11	2.314517	1FQJ	577	5	3.346168
1VFB	-	0	-	1I2M	-	0	-

Table 5.4.1: Bound-unbound docking results using electrostatics and shape complementarity..1. 'Rank' is the best rank among all predicted positions whose RMSD was less than 5Å. 'N Peaks' is the number of peaks in the predicted set which were less than 5ÅRMSD from the known position. 'Best RMSD' is the lowest RMSD among all the peaks that were shortlisted. If there were no good predictions in the top 4,000 that we choose to keep, we enter a '-' in the column for 'Rank'. The proteins are ranked by their surface areas.

5.5 Timing

We compare the time taken to perform the convolutions using our Fourier method vs a FFT method using 128^3 and 256^3 grids in table 5.5.1. We see that computing approximate Fourier coefficients outperforms the traditional FFT algorithm. The number of coefficients tabulated correspond to those used in showing the error in table 5.1.1

PDB ID	Rank	N Peaks	Best RMSD	PDB ID	Rank	N Peaks	Best RMSD
1WQ1	97	1	3.897003	1JPS	2538	3	3.095251
1KXQ	1492	2	3.168111	1KLU	124	6	2.75031
1MAH	1094	1	4.654828	1BJ1	1121	1	3.071451
1FQ1	52	3	3.168911	1EER	-	0	-
1DFJ	288	7	2.555477	1RLB	2182	2	2.293018
1SBB	1011	7	2.157762	1IB1	1	2	4.703256
1DQJ	1923	1	3.19577	1AKJ	2159	1	2.951469
2MTA	627	5	2.723326	1K5D	266	2	4.417297
1IQD	120	4	3.334374	1E6J	-	0	-
1K4C	783	4	2.105856	1NCA	139	10	2.801303
1FAK	-	0	-	1ML0	-	0	-
1E6E	135	4	3.228511	1HE8	-	0	-
1NSN	1016	1	4.203168	1DE4	-	0	-
1KKL	2421	1	3.711123	2HMI	-	0	-
1I4D	1010	2	4.441025	1N2C	-	0	-

Table 5.4.2: Bound-unbound docking results using electrostatics and shape complementarity..2. 'Rank' is the best rank among all predicted positions whose RMSD was less than 5Å. 'N Peaks' is the number of peaks in the predicted set which were less than 5ÅRMSD from the known position. 'Best RMSD' is the lowest RMSD among all the peaks that were shortlisted. If there were no good predictions in the top 4,000 that we choose to keep, we enter a '-' in the column for 'Rank'. The proteins are ranked by their surface areas.

Frequencies, (α, β)	$\alpha = 2, \beta = 2$	FFT(256^3)
4096	0.114369	16.798823
8000	0.170260	16.798823

Table 5.5.1: Time in seconds taken to estimate Fourier coefficients with the NFFT′ for different over-sampling factors α and β for a molecule with 1100 atoms. The time to perform the FFT for a 256^3 grid is also given.

5.6 Memory cost

The experimental results closely followed the theoretical memory requirement (linear in the number of centers). We used a memory over-sampling factor of 2 in the NFFT steps. Hence for our three test cases which had 10000 to 15000 atoms, we needed approximately 5MB of space. This is in contrast to 268 MB for a 256^3 grid for the FFT Grid Based approaches. For comparisons to other grid based methods, please see [60].

6 Conclusion

Our main contribution lies in expressing the docking of proteins as a convolution of functions, and providing approximation algorithms to find peaks in the docking score. Both shape complementarity and electrostatics were used to obtain the docking positions. Our experiments show that the model accurately predicts docking sites for a large number of protein pairs. We used the FFTW package [24] for computing FFT and the inverse FFT.

References

[1] Ernst Althaus, Oliver Kohlbacher, Hans-Peter Lenhof, and Peter Mller. A combinatorial approach to protein docking with flexible side chains. *Journal of Computational Biology*, 9(4):597–612, August 2002.

[2] Joannis Apostolakis, Andreas Plückthun, and Amedeo Caflisch. Docking small ligands in flexible binding sites. *Journal of Computational Chemistry*, 19(1):21–37, 1998.

[3] David J. Bacon and John Moult. Docking by least-squares fitting of molecular surface patterns. *Journal of Molecular Biology*, 225(3):849–858, June 1992.

[4] Chandrajit Bajaj, Fausto Bernardini, and Kokichi Sugihara. A geometric approach to molecular docking and similarity. Technical Report CSD-TR-94-017, Computer Sciences, Purdue University, March 1994.

[5] Chandrajit Bajaj and Vinay Siddavanahalli. Fast error-bounded surfaces and derivatives computation for volumetric particle data. ICES Report 06-03, The University of Texas at Austin, January 2006.

[6] Nathan A. Baker, David Sept, Simpson Joseph, Michael J. Holst, and J. Andrew McCammon. Electrostatics of nanosystems: application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences of the United States of America*, 98(18):10037–10041, August 2001.

[7] Karine Bastard, Aurelien Thureau, Richard Lavery, and Chantal Prevost. Docking macromolecules with flexible segments. *Journal of Computational Chemistry*, 24(15):1910–1920, November 2003.

[8] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland1, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[9] Amedeo Caflisch, Stephan Fischer, and Martin Karplus. Docking by monte carlo minimization with a solvation correction: Application to an fkbp-substrate complex. *Journal of Computational Chemistry*, 18(6):723–743, 1997.

[10] Carlos J. Camacho, David W. Gatchell, S. Roy Kimura, and Sandor Vajda. Scoring docked conformations generated by rigid-body protein-protein docking. *Proteins: Structure, Function, and Genetics*, 40(3):525–537, July 2000.

[11] Rong Chen, Li Li, and Zhiping Weng. Zdock: An initial-stage protein-docking algorithm. *Proteins: Structure, Function, and Genetics, Special Issue: CAPRI - Critical Assessment of PRedicted Interactions . Issue Edited by Jol Janin*, 52(1):80–87, May 2003.

[12] Rong Chen and Zhiping Weng. Docking unbound proteins using shape complementarity, desolvation, and electrostatics. *Proteins: Structure, Function, and Genetics*, 47(3):281–294, March 2002.

[13] Rong Chen and Zhiping Weng. A novel shape complementarity scoring function for protein-protein docking. *Proteins: Structure, Function, and Genetics*, 51(3):397–408, March 2003.

[14] Jacqueline Cherfils, Stephane Duquerroy, and Joel Janin. Protein-protein recognition analyzed by docking simulation. *Proteins: Structure, Function, and Genetics*, 11(4):271–280, 1991.

[15] Jack R. Collins, Stanley K. Burt, and John W. Erickson. Flap opening in hiv-1 protease simulated by 'activated' molecular dynamics. *Nature structural biology.*, 2(4):334–338, April 1995.

[16] Michael L. Connolly. Shape complementarity at the hemoglobin $\alpha_1\beta_1$ subunit interface. *Biopolymers*, 25(7):1229–1247, February 1986.

[17] Renee L. DesJarlais, Robert P. Sheridan, J. Scott Dixon, Irwin D. Kuntz, and R. Venkataraghavan. Docking flexible ligands to macromolecular receptors by molecular shape. *Journal of medicinal chemistry*, 29(11):2149–2153, November 1986.

[18] Dina Duhovny, Ruth Nussinov, and Haim J. Wolfson. Efficient unbound docking of rigid molecules. In R. Guigo and D. Gusfield, editors, *Proceedings of the Fourth International Workshop on Algorithms in Bioinformatics*, pages 185–200, Springer-Verlag GmbH Rome, Italy, September 2002.

[19] Todd J. A. Ewing and Irwin D. Kuntz. Critical evaluation of search algorithms for automated molecular docking and database screening. *Journal of Computational Chemistry*, 18(9):1175–1189, December 1998.

[20] Amr Fahmy and Gerhard Wagner. Treedock: A tool for protein docking based on minimizing van der waals energies. *Journal of the American Chemical Society*, 124(7):1241–1250, February 2002.

[21] Juan Fernndez-Recio, Maxim Totrov, and Ruben Abagyan. Soft proteinprotein docking in internal coordinates. *Protein Science*, 11:280–291, 2002.

[22] Daniel Fischer, Shuo Liang Lin, Haim L. Wolfson, and Ruth Nussinov. A geometry-based suite of molecular-docking processes. *Journal of Molecular Biology*, 248(2):459–477, 1995.

[23] Daniel Fischer, Raquel Norel, Ruth Nussinov, and Haim J. Wolfson. 3-d docking of protein molecules. In *CPM '93: Proceedings of the 4th Annual Symposium on Combinatorial Pattern Matching*, pages 20–34, London, UK, 1993. Springer-Verlag.

[24] Matteo Frigo and Steven G. Johnson. The design and implementation of fftw3. *Proceedings of the IEEE. Invited paper, Special Issue on Program Generation, Optimization, and Platform Adaptation*, 93(2):216–231, february 2005.

[25] Jones G, Willett P, and Glen RC. Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation. *Journal of molecular biology*, 245(1):43–53, January 1995.

[26] Henry A. Gabb, Richard M. Jackson, and Michael J. E. Sternberg. Modelling protein docking using shape complementarity, electrostatics and biochemical information. *Journal of Molecular Biology*, 272(1):106–120, September 1997.

[27] R. Gabdoulline and R. Wade. Analytically defined surfaces to analyze molecular interaction properties. *J. of Molecular Graphics*, 14(6):341–353, December 1996.

[28] Daniel K. Gehlhaar, Gennady Verkhivker, Paul A. Rejtoand David B. Fogel, Lawrence J. Fogel, and Stephan T. Freer. Docking conformationally flexible small molecules into a protein binding site through simulated evolution. In J.R. McDonnell, R.G. Reynolds, and D.B. Fogel, editors, *Evolutionary Programming IV: The Proc. of Fourth Annual Conference on Evolutionary Programming*, pages 615–627, MIT Press, Cambridge, MA, 1995.

[29] D. S. Goodsell and Arthur J. Olson. Automated docking of substrates to proteins by simulated annealing. *Proteins:Structure, Function and Genetics*, 8(3):195–202, 1990.

[30] Jonathan Greer and Bruce L. Bush. Macromolecular shape and surface maps by solvent exclusion. *Proceedings of the National Academy of Sciences of the United States of America*, 75(1):303–307, January 1978.

[31] Manuela Helmer-Citterich and Anna Tramontano. Puzzle: A new method for automated protein docking based on surface shape complementarity. *Journal of Molecular Biology*, 235(3):1021–1031, January 1994.

[32] Fan Jianga and Sung-Hou Kim. "soft docking": Matching of molecular surface cubes. *Journal of Molecular Biology*, 219(1):79–102, May 1991.

[33] Gareth Jones, Peter Willett, Robert C. Glen, Andrew R. Leach, and Robin Taylor. Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology*, 267(3):727–748, April 1997.

[34] Richard S. Judson, E. P. Jaeger, and Adi M. Treasurywala. A genetic algorithm based method for docking flexible molecules. *Journal of Molecular Structure: THEOCHEM*, 308:191–206, May 1994.

[35] Ephraim Katchalski-Katzir, Isaac Shariv, Miriam Eisenstein, Asher A. Friesem, Claude Aflalo, and Ilya A. Vakser. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences of the United States of America*, 89(6):2195–2199, March 1992.

[36] Julio A. Kovacs, Pablo Chacn, Yao Cong, Essam Metwally, and Willy Wriggers. Fast rotational matching of rigid bodies by fast fourier transform acceleration of five degrees of freedom. *Acta Crystallographica, Biological Crystallography*, D59(8):1371–1376, August 2003.

[37] Julio A. Kovacs and Willy Wriggers. Fast rotational matching. *Acta Crystallographica, Biological Crystallography*, D58(8):1282–1286, August 2002.

[38] F. S. Kuhl, G. M. Crippen, and D. K. Friesen. A combinatorial algorithm for calculating ligand binding. *Journal of Computational Chemistry*, 5(1):24–34, 1984.

[39] Irwin D. Kuntz, Jeffrey M. Blaney, Stuart J. Oatley, Robert Langridge, and Thomas E. Ferrin. A geometric approach to macromolecule-ligand interactions. *Journal of Molecular Biology*, 161(2):269–288, October 1982.

[40] Andrew R. Leach. Ligand docking to proteins with discrete side-chain flexibility. *Journal of Molecular Biology*, 235(1):345–356, January 1994.

[41] Andrew R. Leach and Irwin D. Kuntz. Conformational analysis of flexible ligands in macromolecular receptor sites. *Journal of Computational Chemistry*, 13(6):730–748, 1992.

[42] Jonathan Leech, Jan F. Prins, and Jan Hermans. Smd: Visual steering of molecular dynamics for protein design. *IEEE Computational Science and Engineering*, 3(4):38–45, Winter 1996.

[43] Hans-Peter Lenhof. New contact measures for the protein docking problem. In *RECOMB '97: Proceedings of the first annual international conference on Computational molecular biology*, pages 182–191, New York, NY, USA, 1997. ACM Press.

[44] Li Li, Rong Chen, and Zhiping Weng. Rdock: Refinement of rigid-body protein docking predictions. *Proteins: Structure, Function, and Genetics*, 53(3):693–707, September 2003.

[45] Jeffrey G. Mandell1, Victoria A. Roberts, Michael E. Pique, Vladimir Kotlovyi, Julie C. Mitchell, Erik Nelson, Igor Tsigelny, and Lynn F. Ten Eyck. Protein docking using continuum electrostatics and geometric fit. *Protein Engineering*, 14(2):105–113, February 2001.

[46] Michael Meyer, Peter Wilson, and Dietmar Schomburg. Hydrogen bonding and molecular surface shape complementarity as a basis for protein docking. *Journal of Molecular Biology*, 264(1):199–210, November 1996.

[47] Miho Yamada Mizutani, Nobuo Tomioka, and Akiko Itai. Rational automatic search method for stable docking models of protein and ligand. *Journal of Molecular Biology*, 243(2):310–326, October 1994.

[48] Garrett M. Morris, David S. Goodsell, Robert S. Halliday, Ruth Huey, William E. Hart, Richard K. Belew, and Arthur J. Olson. Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry*, 19(14):1639–1662, 1998.

[49] Alfredo Di Nola, Danilo Roccatano, and Herman J. C. Berendsen. Molecular dynamics simulation of the docking of substrates to proteins. *Proteins*, 19(3):174–182, July 1994.

[50] Raquel Norel, Daniel Fischer, Haim J. Wolfson, and Ruth Nussinov. Molecular surface recognition by a computer vision-based technique. *Protein engineering*, 7(1):39–46, January 1994.

[51] Raquel Norel, Shuo L. Lin, Haim J. Wolfson, and Ruth Nussinov. Molecular surface complementarity at protein-protein interfaces: The critical role played by surface normals at well placed, sparse, points in docking. *Journal of Molecular Biology*, 252(2):263–273, September 1995.

[52] Dixon JS Oshiro CM, Kuntz ID. Flexible ligand docking using a genetic algorithm. *Journal of computer-aided molecular design*, 9(2):113–130, April 1995.

[53] Daniel Potts, Gabriele Steidl, and Manfred Tasche. *Fast fourier transform for nonequispaced data: A tutorial, in Modern Sampling Theory: Mathematics and Applications*, chapter 12, pages 249–274. 1998.

[54] Matthias Rarey, Bernd Kramer, Thomas Lengauer, and Gerhard Klebe. A fast flexible docking method using an incremental construction algorithm. *Journal of Molecular Biology*, 261(3):470–489, August 1996.

[55] David W. Ritchie. *Parametric Protein Shape Recognition*. Phd thesis, Departments of Computer Science & Molecular and Cell Biology, University of Aberdeen, King's College, Aberdeen, UK, September 1998.

[56] David W. Ritchie and Graham J. L. Kemp. Fast computation, rotation, and comparison of low resolution spherical harmonic molecular surfaces. *Journal of Computational Chemistry*, 20(4):383–395, February 1999.

[57] David W. Ritchie and Graham J.L. Kemp. Protein docking using spherical polar fourier correlations. *Proteins: Structure, Function, and Genetics*, 39(2):178–194, March 2000.

[58] Bilha Sandak, Ruth Nussinov, and Haim J. Wolfson. An automated computer vision and robotics-based technique for 3-d flexible biomolecular docking and matching. *Computer applications in the biosciences : CABIOS*, 11(1):87–99, February 1995.

[59] Brian K. Shoichet and Irwin D. Kuntz. Protein docking and complementarity. *Journal of Molecular Biology*, 221(1):327–346, September 1991.

[60] Julio Castrillon-Candas Vinay Siddavanahalli and Chandrajit Bajaj. Nonequispaced fourier transforms for protein-protein docking. ICES Report 05-44, The University of Texas at Austin, Austin TX USA, October 2005.

[61] Peter H. Walls and Michael J. E. Sternberg. New algorithm to model protein-protein recognition based on surface complementarity. applications to antibody-antigen docking. *Journal of Molecular Biology*, 228(1):277–297, November 1992.

[62] Huajun Wang. Grid-search molecular accessible surface algorithm for solving the protein docking problem. *Journal of Computational Chemistry*, 12(6):746–750, 1991.

[63] Jian Wang, Peter A. Kollman, and Irwin D. Kuntz. Flexible ligand docking: A multistep strategy approach. *Proteins: Structure, Function, and Genetics*, 36(1):1–19, October 1999.

[64] William Welch, Jim Ruppert, and Ajay N. Jain. Hammerhead: fast, fully automated docking of flexible ligands to protein binding sites. *Chemistry & biology*, 3(6):449–462, June 1996.

[65] Shi-Yi Yue. Distance-constrained molecular docking by simulated annealing. *Protein engineering.*, 4(2):177–184, December 1990.