

GPU.proton.DOCK: Genuine Protein Ultrafast proton equilibria consistent DOCKing

Alexander A. Kantardjiev*

Biophysical Chemistry Group, Institute of Organic Chemistry, Bulgarian Academy of Sciences, Sofia-1113 and Department of Physics, Sofia University, Sofia, Bulgaria

Received February 28, 2011; Revised April 21, 2011; Accepted May 7, 2011

ABSTRACT

GPU.proton.DOCK (Genuine Protein Ultrafast proton equilibria consistent DOCKing) is a state of the art service for *in silico* prediction of protein–protein interactions via rigorous and ultrafast docking code. It is unique in providing stringent account of electrostatic interactions self-consistency and proton equilibria mutual effects of docking partners. GPU.proton.DOCK is the first server offering such a crucial supplement to protein docking algorithms—a step toward more reliable and high accuracy docking results. The code (especially the Fast Fourier Transform bottleneck and electrostatic fields computation) is parallelized to run on a GPU supercomputer. The high performance will be of use for large-scale structural bioinformatics and systems biology projects, thus bridging physics of the interactions with analysis of molecular networks. We propose workflows for exploring *in silico* charge mutagenesis effects. Special emphasis is given to the interface-intuitive and user-friendly. The input is comprised of the atomic coordinate files in PDB format. The advanced user is provided with a special input section for addition of non-polypeptide charges, extra ionogenic groups with intrinsic pK_a values or fixed ions. The output is comprised of docked complexes in PDB format as well as interactive visualization in a molecular viewer. GPU.proton.DOCK server can be accessed at <http://gpudock.orgchm.bas.bg/>.

INTRODUCTION

Evaluation of putative protein–protein interactions is a crucial requirement for modern structural bioinformatics and systems biology research. Understanding protein networks and the application of drug design techniques

is intimately related to the availability of fast and reliable docking methods that should account for all major aspects of molecular interaction physics. Enormous effort is underway to decipher and predict *in silico* protein–protein interactions (1–3). The introduction of Fourier correlation method (4) paved the way to development of reasonably fast algorithms for rigid body docking (the alternative being geometric hashing). Still the speed is a limiting factor for modern day large-scale effort and the availability of massively parallel GPU (Graphic Processing Unit) supercomputer systems might be the breakthrough in this class of molecular modeling techniques (5). But more important is to delve deeper in the physics of protein–protein interactions. Long-range electrostatic interactions are domineering in protein molecules—a determinant of the structure–function relationship (6–10). Therefore, docking algorithms should be able to account for self-consistency of the long-range electrostatic interactions and mutual effects of the partners on each other protonation states—till now ignored, but essential step toward accurate and reliable docking predictions.

The field of protein–protein docking prediction algorithms is very active. An essential step of any docking workflow is to find a list of ranked mutual orientations based on a scoring measure for shape complementarity and long-range interactions. Such approaches are the popular ZDOCK (11), Hex (12), PIPER (13), GRAMM-X (14) and Symm-Dock (15). A subsequent step is aimed at refinement of rigid docking results by taking into account short-range interactions. A precise treatment requires accounting for flexibility (16), e.g. RosettaDock (17) and HadDock (18).

However, a basic point is missed by all modern methods—the self-consistency of electrostatic interactions and the mutual influence of docking partners on their protonation states. It is widely recognized that electrostatics is a crucial determinant for protein interactions, but no modern docking algorithm goes beyond simplistic formal Coulomb treatment.

*To whom correspondence should be addressed. Tel: +359 2 9606 123; Fax: +359 2 986 27 95; Email: alexkant@orgchm.bas.bg, alexander.kantardjiev@gmail.com

Our contribution is the implementation of this essential missing link and its realization on a massively parallel GPU supercomputer via Compute Unified Device Architecture (CUDA)/C/C++ programming environment. Thus, we have developed ultrafast docking code with a strong potential for large-scale systems biology projects. Concurrently, we have put on a rigorous basis the concerted action of protein electric fields and the ionization states influence upon molecules encounter.

On the docking algorithmic side, we make use of the significant speed-up of the Fast Fourier Transform (FFT) parallelized effectively under CUDA environment. However, the Fourier transform is not used in the spirit of the traditional grid based Katchalski-Katzir algorithm (4). We implement a version of 6D correlation search, which makes use of spherical polar functions (19). It is a grid-less method implemented via spherical polar fourier representation of docking partners and several 1D Fourier Transforms (1D-FFTs).

On the electrostatics side, we apply a proven and reliable self-consistent and rigorous method PHEPS/PHEMTO (20,21). PHEPS electrostatics algorithms proper are fast, with reasonable, sound physics background and reliability proven by numerous benchmarks—unequivocal validation by comparison with experimental studies as shown in a number of peer-reviewed publications over the years (20–26). The estimation of protein electrostatic potential distribution is based on GPU parallelization via CUDA kernels. Thus our intrinsic fast electrostatics becomes ultrafast—an essential breakthrough since each sampling step of the 6D translation–rotation space (5 rotational and 1 translational degree of freedom) requires estimation of electrostatic energies, update of pKa values and reassignment of protonation charges.

METHODS

Docking and Fourier Transform correlations—rigor of tradition and modern flavor

Tradition in the field revolves around 3D Cartesian grid representations of the proteins and eventual properties encoded on a grid. A subsequent estimation of the resulting grid correlation functions is estimated by Fourier Transform representation (4). Evaluation of the correlation using the Convolution Theorem and Fourier Transforms reduces algorithmic complexity to $N \log N$. Still the high computational complexity of this approach is an obstacle for real-time applications in structural bioinformatics and systems biology projects requiring all-to-all comparisons. Even FFT algorithms on a modern platform do not bring the required speed-up. A breakthrough is the combination of a Fourier Transform-based approach, Grid representation substitution by a set of orthogonal polar basis functions and the emerging GPU supercomputer technology. Instead of a 3D-FFT, which makes use of translational correlations only, our docking correlations are based on gridless (grid free) representation—a 3D-polynomial

expansion of spherical polar basis functions (spherical harmonic functions) (14). Then sampling docking correlations is reduced to estimation of coefficient vectors of the docking partners. The correlation *Corr* is a scalar product of the vectors of expansion coefficients for ‘receptor’ $R_{xyz}^*(Tr, \alpha_r, \beta_r)$ and ‘ligand’ molecule $L_{xyz}(\alpha_l, \beta_l)$:

$$Corr = \text{Re} \left\langle \sum_{xyz} R_{xyz}^*(TR, \alpha_r, \beta_r) L_{xyz}(\alpha_l, \beta_l) \right\rangle \quad (1)$$

Rotations and translations are reduced to transforming expansion coefficients for rotation $r_{xyz}(\lambda, \alpha, \beta)$ and expansion coefficients for translation $r_{xyz}(TR)$. Rotation is via matrix elements of the real Wigner rotation matrices $Rot_{zz'}^{(y)}$ (22):

$$r_{xyz}(\lambda, \alpha, \beta) = \sum_{z'} Rot_{zz'}^{(y)}(\lambda, \alpha, \beta) r_{xyz'} \quad (2)$$

Translation is performed via the following matrix elements $Tr a_{xy,pq}^{(z)}$ in Gauss–Laguerre basis functions (more details in Supplementary Data S2):

$$r_{xyz}(TR) = \sum_{pq} Tr a_{xy,pq}^{(z)}(TR) r_{xyz} \quad (3)$$

The complementarity is calculated conveniently via a series of 1D Fast Fourier Transforms:

$$FT_{RL}(\gamma_L) = \sum_z e^{-iz\gamma_L} \sum_{xy} R_{xyz}^*(TR, \alpha_r, \beta_r) \times L_{xyz}(\alpha_l, \beta_l) \quad (4)$$

The corresponding docked complexes are ranked according obtained scores. Though a rigid docking algorithm, GPU.proton.DOCK gives some flexibility by inclusion of a softer scoring function. That is why some structures seem to penetrate each other in visualization mode.

Electrostatic fields and Fourier Transform correlations

In order to account for the long-range electrostatic interactions an additional correlation function is required. Our implementation uses polynomial expansion to encode both protein surface and electrostatic potential field of the protein molecules. The pH-dependent electrostatic energy of a protein complex can be expressed as the following multiple integral of converged electrostatic potential distribution $\Phi^R(x, y, z, pH)$ of ‘receptor’ molecule and reassigned protonation charge distribution $Q^L(x, y, z, pH)$ of the ‘ligand’:

$$E(pH) = \iiint_{xyz} \Phi^R(x, y, z, pH) Q^L(x, y, z, pH) dx dy dz \quad (5)$$

Our task is to correlate protein electrostatic fields after a self-consistent iterative procedure, which can be applied either at every sampling step or pre-computed, which is still a good approximation relevant to standard formal treatment of electrostatics. In order to apply grid free correlation, the electrostatic potential is represented as

an expansion of spherical polar function basis functions. Again the orthogonality property gives the overlap of spherical polar functions as a scalar product of the expansion coefficients. This convenient formalism gives us the tool to express electrostatic energy as a scalar product of transformed expansion coefficients for converged electrostatic potential distribution R_{xyz}^ϕ of receptor and reassigned protonation charge distribution of the 'ligand' L_{xyz}^q :

$$E(pH) = \text{Re} \left\langle \sum_{xyz} R_{xyz}^\phi(pH) L_{xyz}^q(pH) \right\rangle \quad (6)$$

A step further is to account for the mutual influence of the docking partners. Such a calculation requires a separate self-consistent electrostatics run, which includes mutual effect of docking partners on each other ionization sites and hence proton equilibria. In this case, we implement an additional GPU kernel to make our traditionally fast electrostatics ultrafast. The details of the electrostatic algorithms application to this problem domain (docking) and its parallelization is given in Supplementary Data S2 and at our GPU.proton.DOCK server site.

Proton equilibria algorithms—a self-consistent treatment

The model accepts experimentally measured pK_a of model compounds (e.g. *N*-acetyl amides of each *i*-th ionogenic amino acids) ($pK_{\text{mod},i}$) and evaluates Born term—a linear response approximation. Partial charges assume values from molecular mechanics parameterization sets—AMBER and PARSE. Hydrogen atom charges have been accounted for in the framework of all atom force field models.

The pair-wise interaction between any *i*-th and *j*-th ionic groups can be simulated by an empirical three term curve: $W_{ij}(r, a_k) = \sum_k (a_k / r_{ij}^k)$, $k = 3$. The a_k were estimated by a nonlinear procedure by minimizing the functional $F(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$ (Supplementary Data S1).

At a stage before accounting for ionization, the procedure calculates intrinsic constants: $pK_{\text{int},i} = pK_{\text{mod},i} + \Delta pK_{\text{Born},i} + \Delta pK_{\text{par},i}$, where $pK_{\text{mod},i}$ is the pK_a of the *i*-th site according to model compounds; $\Delta pK_{\text{Born},i}$ is the Born self-energy of the *i*-th and $\Delta pK_{\text{par},i}$ is the contribution of the *i*-th site interacting with the set of partial (permanent, fixed) atomic charges. For each protonation group and at each step of the iterative self-consistent method, we estimate:

$$pK_{a,i}(pH) = pK_{\text{int},i} + pK_{\text{tit},i} = pK_{\text{int},i} + \frac{1}{RT \ln 10} \sum_i \sum_{j < i} [Q_j(pH)(W_{ij} - C)(1 - SA_{ij})], \quad (7)$$

where C is the Debye–Hückel term for ionic strength (Is). The term $pK_{\text{tit},i}$ is the pK_a shift of the *i*-th site caused by interactions with all other proton binding groups. This Tanford-Roxby style procedure is a well-controlled approximation of the strict statistical-mechanics

treatment (See Supplementary Data S1). In resume the key relation is:

$$\langle pK_j \rangle = \frac{\sum_{\mu=1}^{2^M} pK_{\mu,j} e^{-\beta G_\mu}}{\sum_{\mu} e^{-\beta G_\mu}} = \frac{\sum_{\mu=1}^{2^M} pK_{\text{intr},j} e^{-\beta G_\mu} - \sum_{\mu=1}^{2^M} e^{-\beta G_\mu} \left[\sum_{j=1}^M (p_{\mu,j} - p_j^0) E \right]}{\sum_{\mu} e^{-\beta G_\mu}}, \quad (8)$$

Here, \mathbf{p} is the protonation vector, \mathbf{G} is free energy of the corresponding ionization state, \mathbf{M} is number of proton binding groups and E is site–site electrostatic interaction energy. This relation can be derived in reverse order starting from the canonical Tanford-Roxby equation by trivial substitutions.

When the self-consistent iterative procedure meets convergence criteria, the new charge distribution is applied for calculation of the electrostatic potential grid. At this point, we have accelerated the code by applying direct summation GPU kernel (Supplementary Data S2). We are testing a fast multipole GPU kernel and multilevel summation techniques for achieving higher performance. See implementation and Supplementary Data sections for more details.

Staircase of sophistication—GPU.proton.Dock modes

GPU.proton.DOCK server attempts to empower the user to compare and interpret complementarily several approaches of increasing detail and sophistication in exploring protein–protein docking mechanism. All of them take into account the subtle issues in accounting for ionization states—appropriate treatment of pH dependence and self-consistence. Dissection of individual residues contributions to docking results (through *in silico* mutagenesis—see below) is also among the features worth consideration.

Upon coming at a stage to evaluate electrostatic interactions of the charge system and face the contribution of protonation-dependent electrostatics to correlation functions, GPU.proton.DOCK server provides three alternatives to cope with the diverse needs and specific requirements for electrostatic docking calculation by the protein scientist:

- (i) A standard, straightforward method, which relies on simple Coulomb electrostatics and immutable fields. This is the fastest approach. Each sampling step uses a pre-computed electrostatic field.
- (ii) A step toward improvement—still immutable field at each step but a preliminary computation is performed via self-consistent iterative electrostatics. Thus, we have a converged protonation charge distribution after the iterative procedure for a

given pH value, but no update at each sampling step.

- (iii) Mutual electrostatic influence of the docking partners. We consider this step an essential and crucial contribution to the docking algorithms field. Each sampling step in the 6D docking space requires re-evaluation of electrostatic potential and reassignment of protonation charges.

Whatever mode for calculation is chosen, the user can define a range of pH values to 'titrate' docking results. The user is provided with interactive Jmol Java applet to view docked structures. The results are also available as PDB formatted complexes enlisted according to the docking score. The user can download all predictions in NMR/MODEL PDB format as well as archives of differently numbered sets of single PDB files. Such type of output can be readily used for visualization using convenient molecular modelling software for rendering protein 3D structure—VMD (<http://www.ks.uiuc.edu/Research/vmd/>), Chimera (28) etc. The final pages of the GPU.proton.DOCK workflows provide interactive visualization for each of the predicted complexes.

***In silico* electrostatics mutagenesis—yet another GPU.proton.DOCK bonus**

Another GPU.proton.DOCK option that is of use for the structural bioinformatician is *in silico* mutagenesis at the level of single amino acids residues. We have already applied this methodology for the evaluation of charge mutants effects on fundamental molecular electrostatics and the way specific charge sites exert effect on electric/dipole moments (both scalar and vector values). Now our docking service offers such a tool to find effects of charge mutants on protein–protein interactions. It is useful to make explicit the meaning of a charge mutant—elimination (ignoring) of a titratable site in the self-consistent iterative procedure. What follows is a brief description of this new functionality—valuable information that our service is going to provide with ease.

The mutagenesis mode branch of GPU.proton.DOCK workflow requires electrostatic self-consistent computations with 'charge' mutated structures—one for the each docking protein multipole. The user is given the possibility to mutate a single ionizable residue for each of the docking partners. The interface is friendly and intuitive. The next stage of the *in silico* mutagenesis workflow bears resemblance to a normal run—a choice of pH value and launching of the Fourier Transform docking algorithm. Finally, the user is provided with visualization tools via molecular viewers and one is given access to docked structures in standard PDB format.

IMPLEMENTATION

The algorithms implementing docking algorithms, electrostatics modeling and protein structure handling are written in C/C⁺⁺/CUDA, Perl and Haskell by the author (Alexander Kantardjiev). C/CUDA codes computationally demanding algorithms, which are the bottleneck

in computing time. The heart of the acceleration is comprised of GPU kernels. GPU supercomputers are based on massively parallel and multithreaded hardware architecture and thus achieve their limit with fine-grained parallel decompositions. Our application of GPU parallelization is both at the level of electrostatic potential grid calculation and the evaluation of the correlations by Fourier Transforms—FFT algorithm. The direct approach is of quadratic time complexity $O(mn)$ for n charge sites and m grid points. Our GPU kernel gave 60-fold speedup over a single core CPU. Kernel development for electrostatic potential distribution via direct summation is straightforwardly parallelized (actually the outer loop of the serial implementation) (Supplementary Data S2). We have developed GPU versions of the linear computational complexity algorithms for the electrostatic potential computation (multilevel summation and fast multipole method) but they are not included in the current Server implementation. More details about our GPU electrostatics effort are available at our GPU.proton.DOCK site and Supplementary Data sections.

The bottleneck of the docking run is the Fourier transform. We make use of the FFT algorithm provided by CUFFT library. Our method relies on multiple 1D-FFT instead of 3D-FFT.

Perl excels at efficient and elegant protein structure parsing and convenient data structure manipulation. The web implementation itself is driven by CGI/PERL routines with Java employed to run molecular viewer for interactive visualization of dipole/electric moments relative to 3D protein structure. This Java applet is part of Jmol applet molecular viewer distribution (<http://jmol.sourceforge.net>). GPU.proton.DOCK server expects as an input two coordinate files in PDB format. Protein structure files, containing HETATM records, are given special attention—an option is present to account for ligand/cofactors/ions charge properties explicitly in the electrostatic interaction calculation. As an additional asset, the user is given relevant information about the protein molecule and warned about certain inconsistencies in protein structure, that might impact adversely ensuing calculation, e.g. interruption in residue numbering, which influences electrostatics through the appearance of terminal amino positive and carboxy negative charge sites with intrinsic pKs. The user is given the possibility to edit initial setup of ionogenic groups (attention to cystein residues in disulfide bonds and excluding covalently modified groups). This is accomplished by user-friendly panel selection of ionizable groups that are going to be accounted for in the consequent self-consistent electrostatic calculation, alleviating the efforts of the user to customize input protein structure. Direct edit of PDB file allows for a range of options aimed at the advanced user: adding missing terminal charges, fixed (non-titratable) integer or partial charges and titratable groups with user defined pKa intrinsic. We consider such rich electrostatic setup a significant practical boost for our GPU.proton.DOCK server. Reasonably acquainted users could address a number of subtle issues, e.g. effects of ligands, cofactors, inhibitors and ions. All other

parameters used as input are predefined or automatically calculated. These steps complete the initial setup. Calculation proceeds through aforementioned stages—evaluation of solvent accessibilities and the linear response Born term $\Delta pK_{\text{Born},i}$, perturbation of pK_a by partial charges $\Delta pK_{\text{par},i}$ and finally the iterative procedure for self-consistent evaluation of titratable $\Delta pK_{\text{tit},i}$.

This is the place to mention default values for sampling rotation–translation space. We sample a 6D space—1 translational and 5 rotational degrees of freedom. The traditional sampling is also 6D, but consists of 3 rotational and 3 translational degrees of freedom. The sampling step for translation is 0.9 Å. Rotational steps—6 angular degrees. Default polynomial expansion order is 20. The overall mutual orientations of docking partners in sampling is on the order of billions— 10^9 .

Just for reminder—to estimate and compare electrostatic energies and potentials, the following energy conversion units were used: 1 kcal = 4.186 kJ = 1.68 RT units (at 298 K) = 0.735 pKa units. The units of $\phi_i(\text{pH})$ in kcal/mol·e = 43.176 mV or 30.24 mC/m².

BENCHMARKS AND EXTENSIVE TESTS

It is a coalescence of a FFT-based rigid docking, long term of experience with protein electrostatics and proton equilibria as well as the emergence of extremely powerful GPU parallel architectures that gives us the confidence to present the service to the wide protein community—from the accomplished protein docking experts and adept structural bioinformaticians to the novice systems biology practitioners. Approaches outlined above were applied to diverse cases of protein–protein interactions (see corresponding table, which is uploaded at GPU.proton.DOCK server site Supplementary Data page). Extensive tests for reliability and accuracy on standard benchmarks were performed as well as comparative analysis in relation to other docking algorithms (especially Hex method). However, direct comparison with other docking algorithms should be careful. One should take into account the difference in scoring functions, the strategy for sampling search space, the step parameter for the search etc. Our approach is comparable with Hex at the level of representation and sampling the search space (spherical polar fourier). The core of the acceleration is the sampling of the mutual orientation space and Supplementary Data S2 contains a table with GPU.proton.DOCK speeds of sampling (millions of orientations per second) against Hex performance for different polynomial expansion order. However, inclusion of sophisticated treatment of electrostatics and protonation equilibria makes direct comparisons in speed inconsistent: though our method is slower than Hex [but on the same order—several tens of seconds per run, e.g. the 15s Hex run (19), against the 24s GPU.proton.DOCK], our development effort raises the level of realism in protein–protein interactions treatment and contributes to the science of docking. Still the method falls in the category ‘ultrafast’ and our intentions are to apply it in large-scale systems biology/structural bioinformatics

projects. For the contemporary status of docking accuracy GPU.proton.DOCK is adequate and consistent. Identification of near-native docking results proves reliability of the method.

CONCLUSION AND FUTURE DEVELOPMENT

We are convinced that GPU.proton.DOCK server will be of favor to anyone who needs fast and comprehensive analysis of protonation-dependent docking results as well as *in silico* charge mutagenesis effects on the interaction mechanisms. At the same time, we work toward improvements, extensions and new functionality. A compendious agenda follows:

- Explicit backbone flexibility algorithm and side chain optimization algorithm;
- Eliciting interplay of dipole/electric moments in protein recognition and complex formation;
- Potential of mean force models—toward knowledge-based potentials in docking algorithms;
- Many-body docking—assembling multimeric proteins;
- Treatment of desolvation and explicit modeling of water molecules effect on docking;
- Development of ultrafast ligand–protein docking—toward virtual screening;
- Development of docked structures databases—application of docking on a large scale;
- Quantum effects in protein–protein/ligand recognition—quantum entanglement contribution: toward quantum non-locality concepts in explaining structure–function relation in the context of structural bioinformatics and for the improvement of docking algorithms.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGMENTS

Alexander Kantardjiev dedicates this work to the memory of his aristocratic grandfather Zdenko Kantardjiev and to the memory of his fine and sophisticated grandgrandmother Dr. Ivanka Bardarova-Kantardjieva, among the first Bulgarians with Doctoral degree in Transcendental Philosophy (Kant), Zurich University and Music studies, Prague Conservatory. We are thankful to Columbia University Professor Barry Honig and Professor Emil Alexov for kind donation of a computer cluster, which boosted our development effort in high performance parallel programming.

FUNDING

The work is partially supported by (grant D-002-126) of National Fund ‘Scientific Research’, Sofia, Bulgaria, which allowed assembling a GPU-based supercomputer system. Funding for open access charge: Waived by Oxford University Press.

Conflict of interest statement. None declared.

REFERENCES

- Halperin, I., Ma, B., Wolfson, H. and Nussinov, R. (2002) Principles of docking: an overview of search algorithms and a guide to scoring functions. *Proteins: Struct. Func. Bioinf.*, **47**, 409–443.
- Smith, G.R. and Sternberg, M.J.E. (2002) Prediction of protein-protein interactions by docking methods. *Curr. Opin. Struct. Biol.*, **12**, 28–35.
- Ritchie, D. (2008) Recent progress and future directions in protein-protein docking. *Curr. Prot. Prot. Sci.*, **9**, 1–15.
- Katchalski-Katzir, E., Shriv, I., Eisenstein, M., Friesem, A.A., Aflalo, C. and Vakser, I.A. (1992) Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proc. Natl Acad. Sci. USA*, **89**, 2195–2199.
- Stone, J., Hardy, D., Ufimtsev, I. and Schulten, K. (2010) GPU-accelerated molecular modeling coming of age. *J. Mol. Graphics and Mod.*, **29**, 116–125.
- Warshel, A. (1981) Electrostatic basis of structure-function correlation in proteins. *Acc. Chem. Res.*, **14**, 284–290.
- Honig, B. and Nicholls, A. (1995) Classical electrostatics in biology and chemistry. *Science*, **268**, 1144–1149.
- Antosiewicz, J., McCammon, J. and Gilson, M. (1994) Prediction of pH-dependent properties of proteins. *J. Mol. Biol.*, **238**, 415–436.
- Bashford, D. and Karplus, M. (1990) pK_{as} of ionization groups in proteins: atomic detail from a continuum electrostatic model. *Biochemistry*, **29**, 10219–10225.
- Warshel, A. and Papazyan, A. (1998) Electrostatic effects in macromolecules: fundamental concepts and practical modeling. *Curr. Opin. Struct. Biol.*, **8**, 211–217.
- Chen, R., Li, L. and Weng, Z. (2003) ZDOCK: an initial-stage protein-docking algorithm. *Proteins: Struct. Func. Bioinf.*, **52**, 80–87.
- Macindoe, G., Mavridis, L., Venkatraman, V., Devignes, M. and Ritchie, D. (2010) HexServer: an FFT-based protein docking server powered by graphics processors. *Nucleic Acids Res.*, **38**, W445–W449.
- Kozakov, D., Brenke, R., Comeau, S. and Vajda, S. (2006) PIPER: An FFT-Based Protein Docking Program with Pairwise Potentials. *Proteins: Struct. Func. Bioinf.*, **65**, 392–406.
- Tovchigrechko, A. and Vakser, I.A. (2006) GRAMM-X public web server for protein-protein docking. *Nucleic Acids Res.*, **34**, W310–W314.
- Schneidman-Duhovny, D., Inbar, Y., Nussinov, R. and Wolfson, H.J. (2005) PatchDock and SymmDock: servers for rigid and symmetric docking. *Nucleic Acids Res.*, **33**, W363–W367.
- Andrusier, N., Mashlach, E., Nussinov, R. and Wolfson, H.J. (2008) Principles of flexible protein-protein docking. *Proteins: Struct. Func. Bioinf.*, **73**, 271–289.
- Lyskov, S. and Gray, J.J. (2008) The RosettaDock server for local protein-protein docking. *Nucleic Acids Res.*, **36**, W233–W238.
- Dominguez, C., Boelens, R. and Bonvin, A.M.J.J. (2003) HADDOCK: a protein-protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.*, **125**, 1731–1737.
- Ritchie, D., Kozakov, D. and Vajda, S. (2008) Accelerating and focusing protein-protein docking correlations using multi-dimensional rotational FFT generating functions. *Bioinformatics*, **24**, 1865–1873.
- Kantardjiev, A.A. and Atanasov, B.P. (2006) PHEPS: web-based pH-dependent Protein electrostatics Server. *Nucl. Acid Res.*, **134**, W43–W47.
- Kantardjiev, A.A. and Atanasov, B.P. (2009) PHEMTO: protein pH-dependent electric moment tools. *Nucl. Acid Res.*, **134**, W422–W427.
- Atanasov, B., Mustafi, D. and Makinen, M.W. (2000) Protonation of the β-lactam nitrogen is the trigger event in the catalytic action of class A β-lactamases. *Proc. Natl Acad. Sci. USA*, **97**, 3160–3165.
- Karshikov, A.D., Engh, R., Bode, W. and Atanasov, B.P. (1989) Electrostatic interactions in proteins: calculations of the electrostatic term of free energy and the electrostatic potential field. *Eur. Biophys. J.*, **17**, 287–297.
- Spasov, V.Z., Karshikov, A.D. and Atanasov, B.P. (1989) Electrostatic interactions in proteins: a theoretical analysis of lysozyme ionization. *Biochim. Biophys. Acta*, **999**, 1–6.
- Roumenina, L.T., Kantardjiev, A.A., Atanasov, B.P., Waters, P., Gadjeva, M., Reid, K.B.M., Mantovani, A., Kishore, U. and Kojouharova, M.S. (2005) Role of Ca²⁺ in the Electrostatic Stability and the Functional Activity of the Globular Domain of the Human C1q. *Biochemistry*, **44**, 14097–14109.
- Roumenina, L.T., Bureeva, S., Kantardjiev, A.A., Karlinsky, D., Andia-Pravdivy, J.E., Sim, R., Kaplun, A., Popov, M., Kishore, U. and Atanasov, B.P. (2007) Complement C1q-target proteins recognition is inhibited by electric moment effectors. *J. Mol. Recognit.*, **20**, 405–415.
- Biedenharn, L.C. and Louck, J.C. (1981) *Angular Momentum in Quantum Physics*. Addison-Wesley, Reading, MA.
- Pettersen, E.F., Goddard, T.D., Huang, C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C. and Ferrin, T.E. (2004) UCSF Chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem.*, **25**, 1605–1612.