

Computing electrostatic binding energy with the TABI Poisson-Boltzmann solver*

LEIGHTON WILSON, JINGZHEN HU, JIAHUI CHEN, ROBERT KRASNY, AND WEIHUA GENG

Computations of the electrostatic binding energy $\Delta\Delta G_{\text{elec}}$ are presented for 51 solvated biomolecular complexes using the treecode-accelerated boundary integral (TABI) Poisson-Boltzmann solver. TABI computes the electric potential on the triangulated molecular surface of a complex and its monomers, and further processing yields the solvation free energy ΔG_{solv} needed to compute $\Delta\Delta G_{\text{elec}}$. The accuracy of the TABI results was verified using the high-order finite-difference Matched Interface and Boundary (MIB) method as the reference. Among two codes used here for surface triangulation, MSMS and NanoShaper, the latter is found to be more accurate, efficient, and robust. It is shown that the accuracy of the computed $\Delta\Delta G_{\text{elec}}$ using TABI can be efficiently improved by extrapolating low triangulation density results to the high density limit. The calculations needed to compute $\Delta\Delta G_{\text{elec}}$ are susceptible to loss of precision due to cancellation of digits and this emphasizes the need for relatively higher accuracy in computing ΔG_{solv} .

1. Introduction

The binding of two monomers into a biomolecular complex is a fundamental process in biochemistry and it takes place in various different forms including enzyme-substrate [1–3], DNA-drug [4], and RNA-protein [5] interactions. Figure 1 depicts two monomers A, B and the bound complex AB they form. A key property of the complex is its binding energy,

*This work was supported in part by National Science Foundation grants DMS-1418957, DMS-1418966, DMS-1819094, DMS-1819193. Computing services were provided by the Center for Scientific Computation at Southern Methodist University and Advanced Research Computing at the University of Michigan. Jingzhen Hu was partially supported by the SMU Hamilton Scholarship. Leighton Wilson was supported by a National Defense Science and Engineering Graduate Fellowship.

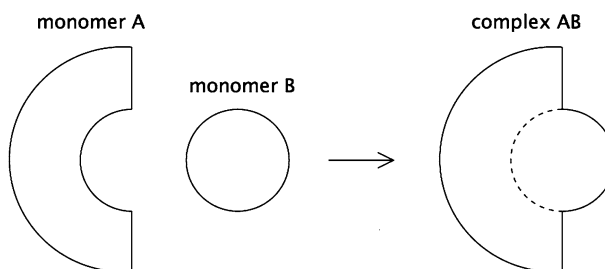


Figure 1: Binding of monomers A, B to form complex AB.

$$(1) \quad \Delta\Delta G = \Delta G^{\text{AB}} - (\Delta G^{\text{A}} + \Delta G^{\text{B}}),$$

where ΔG^{AB} is the free energy of the complex and $\Delta G^{\text{A}}, \Delta G^{\text{B}}$ are the free energies of the unbound monomers. The binding energy is the work done in assembling the complex; it is generally positive which means that the two monomers do not bind spontaneously, but instead require external energy to form the complex; the higher the value of the binding energy, the more stable is the complex. Methods of computing the binding energy are widely used to screen compounds for synthetic drug design.

Among several factors that determine the binding energy of a complex, electrostatic and solvation effects have particular importance. With this in mind, the present work focuses on computing the electrostatic binding energy of a solvated biomolecular complex,

$$(2) \quad \Delta\Delta G_{\text{elec}} = \Delta\Delta G_{\text{coul}} + \Delta\Delta G_{\text{solv}},$$

where the first term on the right arises from Coulomb attraction/repulsion of the solute atoms and the second term arises from the reaction field due to solvent polarization. The strength of these two terms determines when binding occurs.

Various models with different levels of physical accuracy and computational efficiency are used for electrostatic binding energy calculations. A common choice combines an atomistic representation of the solute with an explicit or implicit description of the solvent. A study of protein-ligand binding compared three explicit water models with generalized Born (GB) and Poisson-Boltzmann (PB) implicit solvent models, finding that the explicit and implicit computations of $\Delta\Delta G_{\text{elec}}$ have similar accuracy, yet the latter are significantly faster [6]. Another study showed that the PB model captures much of the nonspecific salt dependence of protein-protein complexation [7].

In finite-difference PB simulations, studies have examined the dependence of the computed $\Delta\Delta G_{\text{elec}}$ on the grid spacing [8–10], while other work showed that accuracy can be improved using a high-order discretization with rigorous treatment of the dielectric interface conditions [11]. In another study utilizing the PB model, the computed $\Delta\Delta G_{\text{elec}}$ was sensitive to the choice of force field used to assign atomic partial charges, but consistent ranking of complexes was obtained within each force field [12].

The present work uses the PB model to compute $\Delta\Delta G_{\text{elec}}$ for 51 solvated biomolecular complexes [8]. While several previous studies employed finite-difference methods for this purpose [6–12], here we use the treecode-accelerated boundary integral (TABI) solver [13]. TABI computes the electric potential and its normal derivative on the triangulated molecular surface separating the solute and solvent, and further processing yields the solvation free energy ΔG_{solv} needed to compute the binding energy. In this work the molecular surface is triangulated using MSMS [14] and NanoShaper [15], and we find that the latter is more accurate, efficient, and robust in TABI calculations. The calculations needed to compute $\Delta\Delta G_{\text{elec}}$ are susceptible to loss of precision due to cancellation of digits, and as a result, the computed $\Delta\Delta G_{\text{elec}}$ is sensitive to the accuracy of the PB solver, as known in the case of finite-difference calculations [6, 8–10]. In TABI calculations, a given level of accuracy in $\Delta\Delta G_{\text{elec}}$ can be achieved using sufficiently high triangulation density, and we show how accuracy can be improved more efficiently by extrapolating low density results to the high density limit.

The work is organized as follows. Section 2 introduces the PB implicit solvent model of a solvated biomolecule. Section 3 discusses numerical PB solvers. Section 4 describes the TABI solver. Section 5 explains the procedure for computing $\Delta\Delta G_{\text{elec}}$. Section 6 presents implementation details. Section 7 reports numerical results for the electrostatic binding energy of 51 biomolecular complexes [8]. The article ends with a section summarizing the main points.

2. Poisson-Boltzmann model

Figure 2 shows the PB model of a solvated biomolecule with the solute domain Ω_1 , solvent domain Ω_2 , and dielectric interface Γ which we take to be the molecular surface (or solvent-excluded surface). The solute is represented by partial charges q_k at atomic centers \mathbf{y}_k , $k = 1 : N_a$ and the solvent has a distribution of dissolved ions.

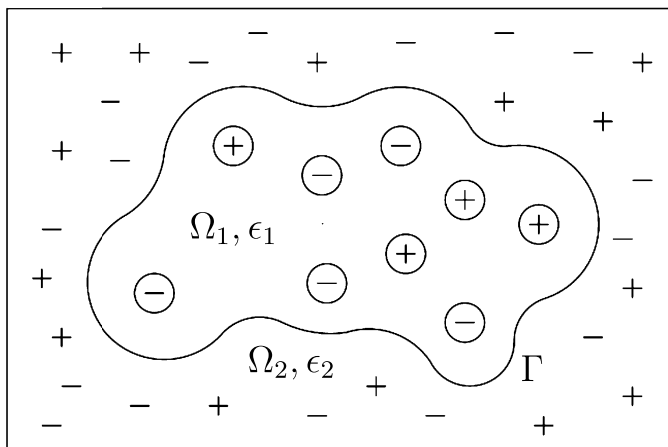


Figure 2: Poisson-Boltzmann model, solute domain Ω_1 , solvent domain Ω_2 , dielectric constants $\epsilon_{1,2}$, molecular surface Γ , solute is represented by atomic partial charges, solvent is continuum dielectric with dissolved ions.

In this work the solute domain has dielectric constant $\epsilon_1 = 1$ corresponding to a vacuum and the solvent domain has a high dielectric constant $\epsilon_2 = 80$ corresponding to water. The electric potential $\phi(\mathbf{x})$ satisfies the linear PB equation,

$$(3) \quad -\nabla \cdot \epsilon(\mathbf{x})\nabla\phi(\mathbf{x}) + \bar{\kappa}^2(\mathbf{x})\phi(\mathbf{x}) = \sum_{k=1}^{N_a} \delta(\mathbf{x} - \mathbf{y}_k)q_k, \quad \mathbf{x} \in \Omega_1 \cup \Omega_2,$$

subject to interface conditions for the potential and its normal derivative,

$$(4) \quad \phi_1(\mathbf{x}) = \phi_2(\mathbf{x}), \quad \epsilon_1 \frac{\partial \phi_1}{\partial \mathbf{n}}(\mathbf{x}) = \epsilon_2 \frac{\partial \phi_2}{\partial \mathbf{n}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma,$$

where subscripts 1,2 indicate the limiting value from either side of the interface. The screening parameter is $\bar{\kappa}^2 = \epsilon_2 \kappa^2$, where κ is the inverse Debye length; $\bar{\kappa} = 0$ in Ω_1 and has a nonzero constant value in Ω_2 . References give more detail about the definition and units of these coefficients [16, 17]. The potential also satisfies the far-field boundary condition,

$$(5) \quad \lim_{|\mathbf{x}| \rightarrow \infty} \phi(\mathbf{x}) = 0.$$

Equations (3)-(5) define a boundary value problem for the potential $\phi(\mathbf{x})$ which in general must be solved numerically.

3. Numerical PB solvers

Numerical PB solvers face several challenges; the solute biomolecule is represented by singular point charges, the molecular surface typically has complex geometry, the dielectric constant is discontinuous across the surface, and the domain is unbounded. As described in comprehensive reviews [18, 19], several types of numerical PB solvers have been developed including grid-based finite-difference and finite-element methods [20–23, 25–30], and boundary element methods [13, 31–38].

Grid-based PB solvers discretize the truncated volumetric domain and solve a sparse linear system for the potential at the grid points. These solvers are available in popular software packages such as APBS [39], AMBER [26, 40], CHARMM [23, 24], and DelPhi [21, 41]. In these methods, the singular charges are interpolated to the grid, the interface conditions are enforced implicitly, and the far-field boundary condition is implemented by a focusing technique applied on a sequence of truncated domains. The computational error can be reduced by refining the grid, and schemes with higher order accuracy have been developed that enforce the interface conditions more rigorously and account for the singular charges using a Green’s function based decomposition [42–44].

Boundary element methods (BEMs) convert the PB differential equation into an equivalent set of integral equations for the surface potential and its normal derivative which are solved on the triangulated molecular surface. BEMs account for the singular charges, interface conditions, and far-field boundary condition analytically. These advantages are offset by the cost of solving a dense linear system, but the solution can be accelerated using GMRES iteration [45] with matrix-vector products computed by the Fast Multipole Method [46–48] or a treecode [49–51].

4. TABI PB solver

This section describes the treecode-accelerated boundary integral (TABI) PB solver, starting with the boundary integral form of the PB implicit solvent model, then the discretization of the integral equations, and finally the treecode algorithm used to accelerate the matrix-vector product.

4.1. Boundary integral form of PB model

The TABI solver utilizes a well-conditioned boundary integral form of the PB implicit solvent model [32]. Applying Green's second identity to Eq. (3), the electric potential is expressed in terms of single and double layer potentials. In the solute domain, $\mathbf{x} \in \Omega_1$, this takes the form

$$(6) \quad \phi(\mathbf{x}) = \int_{\Gamma} \left[G_0(\mathbf{x}, \mathbf{y}) \frac{\partial \phi(\mathbf{y})}{\partial \mathbf{n}} - \frac{\partial G_0(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}_{\mathbf{y}}} \phi(\mathbf{y}) \right] dS_{\mathbf{y}} + \sum_{k=1}^{N_a} G_0(\mathbf{x}, \mathbf{y}_k) q_k,$$

while in the solvent domain, $\mathbf{x} \in \Omega_2$, it is

$$(7) \quad \phi(\mathbf{x}) = \int_{\Gamma} \left[-G_{\kappa}(\mathbf{x}, \mathbf{y}) \frac{\partial \phi(\mathbf{y})}{\partial \mathbf{n}} + \frac{\partial G_{\kappa}(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}_{\mathbf{y}}} \phi(\mathbf{y}) \right] dS_{\mathbf{y}},$$

where the Coulomb and screened Coulomb potentials are

$$(8) \quad G_0(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi|\mathbf{x} - \mathbf{y}|}, \quad G_{\kappa}(\mathbf{x}, \mathbf{y}) = \frac{e^{-\kappa|\mathbf{x} - \mathbf{y}|}}{4\pi|\mathbf{x} - \mathbf{y}|}.$$

Enforcing the interface conditions in Eq. (4) yields a set of coupled integral equations relating the surface potential $\phi_1(\mathbf{x})$ and its normal derivative $\partial\phi_1/\partial\mathbf{n}(\mathbf{x})$ on the molecular surface, $\mathbf{x} \in \Gamma$,

$$(9a) \quad \frac{\bar{\epsilon}}{\epsilon_1} \phi_1(\mathbf{x}) = \int_{\Gamma} \left[K_1(\mathbf{x}, \mathbf{y}) \frac{\partial \phi_1(\mathbf{y})}{\partial \mathbf{n}} + K_2(\mathbf{x}, \mathbf{y}) \phi_1(\mathbf{y}) \right] dS_{\mathbf{y}} + S_1(\mathbf{x}),$$

$$(9b) \quad \frac{\bar{\epsilon}}{\epsilon_2} \frac{\partial \phi_1(\mathbf{x})}{\partial \mathbf{n}} = \int_{\Gamma} \left[K_3(\mathbf{x}, \mathbf{y}) \frac{\partial \phi_1(\mathbf{y})}{\partial \mathbf{n}} + K_4(\mathbf{x}, \mathbf{y}) \phi_1(\mathbf{y}) \right] dS_{\mathbf{y}} + S_2(\mathbf{x}),$$

where $\bar{\epsilon} = (\epsilon_1 + \epsilon_2)/2$, and the kernels $K_{1,2,3,4}$ depend on G_0 , G_k and their first and second order normal derivatives [32], while the source terms are

$$(10) \quad S_1(\mathbf{x}) = \frac{1}{\epsilon_1} \sum_{k=1}^{N_a} G_0(\mathbf{x}, \mathbf{y}_k) q_k, \quad S_2(\mathbf{x}) = \frac{1}{\epsilon_1} \sum_{k=1}^{N_a} \frac{\partial G_0(\mathbf{x}, \mathbf{y}_k)}{\partial \mathbf{n}_{\mathbf{x}}} q_k.$$

4.2. Discretization of boundary integral equations

In this work the molecular surface Γ is triangulated using either MSMS [14] or NanoShaper [15]. As an example, Fig. 3 shows the triangulated surfaces for a barnase-barstar complex (PDB ID 1b2s) with MSMS density $d = 5$. Letting $\mathbf{x}_i, i = 1 : N$ denote the triangle centroids, the integrals in Eqs. (9a)-(9b) are discretized by centroid collocation, yielding a linear system of equations, $A\mathbf{x} = \mathbf{b}$, where the elements of matrix A involve kernel evaluations of type $K(\mathbf{x}_i, \mathbf{x}_j)$, vector \mathbf{x} contains the surface potentials $\phi_1(\mathbf{x}_i)$ and normal derivatives $\partial\phi_1/\partial\mathbf{n}(\mathbf{x}_i)$, and vector \mathbf{b} contains the source terms $S_1(\mathbf{x}_i), S_2(\mathbf{x}_i)$. The system is solved for vector \mathbf{x} by GMRES iteration which requires a matrix-vector product in each step [45], but the matrix A is dense and computing the product by direct summation requires $O(N^2)$ operations, which is prohibitively expensive when N is large. Next we describe the treecode used to accelerate the matrix-vector product.

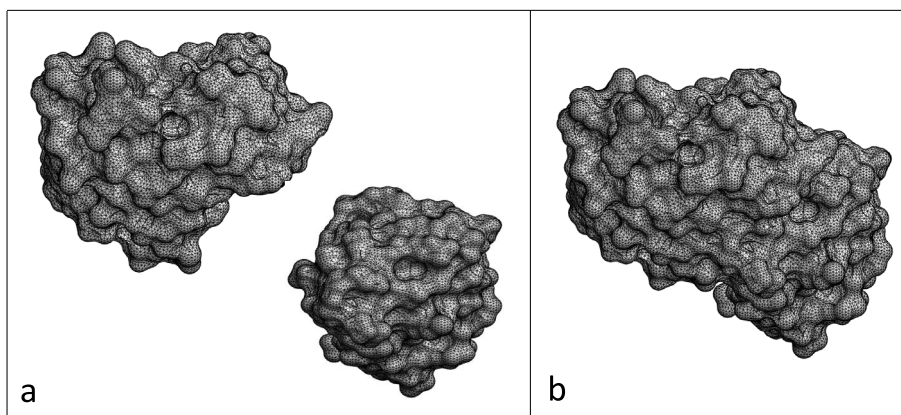


Figure 3: Triangulated molecular surfaces of barnase-barstar (PDB ID 1b2s) using MSMS density $d = 5$, (a) monomers, barnase (upper left), barstar (lower right), (b) complex.

4.3. Treecode

We summarize the treecode algorithm and refer to previous work for more detail [49–51]. The matrix-vector product for the discretization of Eqs. (9a)-(9b) requires computing N -body potentials of the form

$$(11) \quad V_i = \sum_{j=1, j \neq i}^N K(\mathbf{x}_i, \mathbf{x}_j) q_j, \quad i = 1 : N,$$

where $K(\mathbf{x}, \mathbf{y})$ is a kernel, $\mathbf{x}_i, \mathbf{x}_j$ are triangle centroids (also called particles in this context), and q_j is a charge associated with \mathbf{x}_j . The treecode starts by dividing the particles $\{\mathbf{x}_i\}$ into a hierarchy of clusters $\{C\}$ having a tree structure as shown schematically in Fig. 4a. Next the treecode loops through the particles and computes each potential as a sum of particle-cluster interactions,

$$(12) \quad V_i \approx \sum_{C \in N_i} \sum_{\mathbf{x}_j \in C} K(\mathbf{x}_i, \mathbf{x}_j) q_j + \sum_{C \in F_i} \sum_{\|\mathbf{k}\|=0}^p a^{\mathbf{k}}(\mathbf{x}_i, \mathbf{x}_c) m_C^{\mathbf{k}}, \quad i = 1 : N,$$

where N_i, F_i denote the near-field and far-field clusters of particle \mathbf{x}_i . Figure 4b shows a particle-cluster interaction.

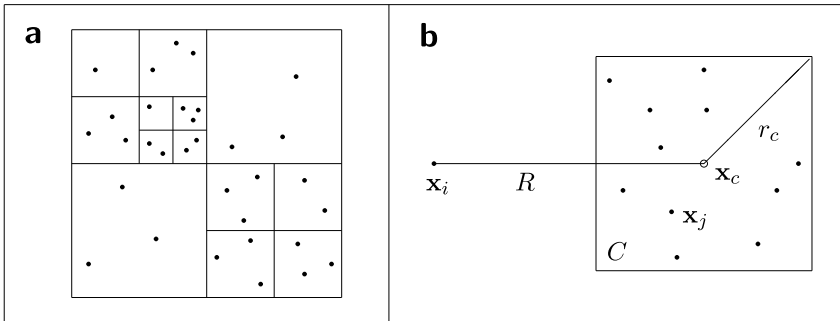


Figure 4: Schematic of treecode, (a) tree structure of particle clusters, (b) particle-cluster interaction between particle \mathbf{x}_i and cluster $C = \{\mathbf{x}_j\}$, \mathbf{x}_c : cluster center, r_c : cluster radius, R : particle-cluster distance.

The first term on the right of Eq. (12) is a direct sum for particles \mathbf{x}_j near \mathbf{x}_i , and the second term is a p th order Cartesian Taylor approximation for clusters that are well-separated from \mathbf{x}_i , where $a^{\mathbf{k}}(\mathbf{x}_i, \mathbf{x}_c)$ are the Taylor coefficients of kernel K about the cluster center \mathbf{x}_c and $m_C^{\mathbf{k}}$ are the cluster moments. Multi-index notation is used with $\mathbf{k} = (k_1, k_2, k_3)$, $\|\mathbf{k}\| = k_1 + k_2 + k_3$. A particle \mathbf{x}_i and a cluster C are defined to be well-separated if the multipole acceptance criterion (MAC) is satisfied, $r_c/R \leq \theta$, where r_c is the cluster radius, $R = |\mathbf{x}_i - \mathbf{x}_c|$ is the particle-cluster distance, and θ is a user-specified parameter. The accuracy of the treecode is controlled

by the order p and MAC parameter θ . Using the treecode, the operation count for the matrix-vector product is $O(N \log N)$, where the factor N is the number of particles \mathbf{x}_i , and the factor $\log N$ is the number of levels in the tree.

5. Energy computations

This section explains how the electrostatic free energy ΔG_{elec} and electrostatic binding energy $\Delta \Delta G_{\text{elec}}$ are computed in the framework of the TABI solver for the boundary integral form of the PB implicit solvent model.

5.1. Electrostatic free energy

The electrostatic free energy of a solvated biomolecule is given by

$$(13) \quad \Delta G_{\text{elec}} = \int_{\mathbb{R}^3} \left(\phi(\mathbf{x})\rho(\mathbf{x}) + \Delta\Pi(\mathbf{x}) - \frac{1}{2}\epsilon(\mathbf{x})|\mathbf{E}(\mathbf{x})|^2 \right) d\mathbf{x},$$

where ϕ is the electric potential, ρ is the fixed charge density of the solute, $\Delta\Pi$ is the excess osmotic pressure of the ions in the solvent, and $\frac{1}{2}\epsilon|\mathbf{E}|^2$ is the electrostatic stress [52]. The pressure and stress terms are typically small in magnitude, and their evaluation involves computationally challenging integrals, so in practice they are commonly omitted [52–54]. With this assumption the electrostatic free energy is

$$(14) \quad \Delta G_{\text{elec}} = \Delta G_{\text{coul}} + \Delta G_{\text{solv}},$$

where the first term is the Coulomb energy of the solute atoms in vacuum,

$$(15) \quad \Delta G_{\text{coul}} = \frac{1}{8\pi\epsilon_1} \sum_{\substack{j,k=1 \\ j \neq k}}^{N_a} \frac{q_j q_k}{|\mathbf{y}_j - \mathbf{y}_k|},$$

and the second term is the solvation energy due to solvent polarization, which in the boundary integral PB framework [32] is

$$(16) \quad \Delta G_{\text{solv}} = \frac{1}{2} \sum_{k=1}^{N_a} q_k \int_{\Gamma} \left(K_1(\mathbf{y}_k, \mathbf{y}) \frac{\partial \phi_1(\mathbf{y})}{\partial \mathbf{n}} + K_2(\mathbf{y}_k, \mathbf{y}) \phi_1(\mathbf{y}) \right) dS_{\mathbf{y}}.$$

Numerical solution of the PB boundary integral form in Eqs. (9a)-(9b) yields the electric potential and its normal derivative at the triangle centroids from which the solvation energy in Eq. (16) is computed.

5.2. Electrostatic binding energy

Figure 5 shows the thermodynamic loop which yields the electrostatic binding energy of a solvated complex,

$$(17) \quad \Delta\Delta G_{\text{elec}} = \Delta\Delta G_{\text{coul}} + \Delta\Delta G_{\text{solv}},$$

where the Coulomb binding energy and solvation binding energy are given in terms of the electrostatic free energy of the complex and its monomers,

$$(18a) \quad \Delta\Delta G_{\text{coul}} = \Delta G_{\text{coul}}^{\text{AB}} - (\Delta G_{\text{coul}}^{\text{A}} + \Delta G_{\text{coul}}^{\text{B}}),$$

$$(18b) \quad \Delta\Delta G_{\text{solv}} = \Delta G_{\text{solv}}^{\text{AB}} - (\Delta G_{\text{solv}}^{\text{A}} + \Delta G_{\text{solv}}^{\text{B}}).$$

In these calculations, the Coulomb energy ΔG_{coul} is computed directly using Eq. (15) and is free of numerical error, while the solvation energy ΔG_{solv} is computed by the TABI solver using Eq. (16), so the latter determines the numerical accuracy of the computed electrostatic binding energy $\Delta\Delta G_{\text{elec}}$. Note however that in addition to the discretization error arising from the TABI solver, the calculations needed to compute $\Delta\Delta G_{\text{elec}}$ are susceptible to loss of precision due to cancellation of digits; this is commented on further below.

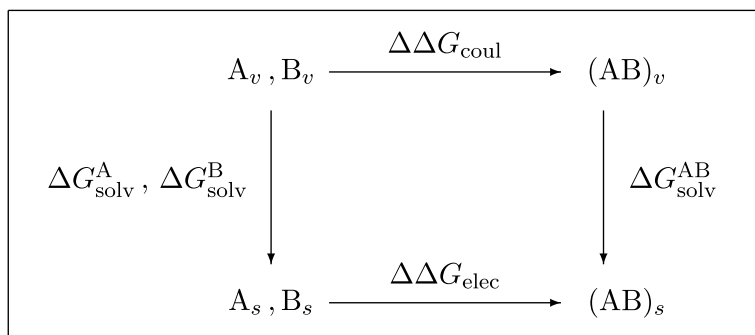


Figure 5: Thermodynamic loop for binding of monomers A, B to form complex AB, subscripts v, s denote vacuum and solvent, electrostatic binding energy $\Delta\Delta G_{\text{elec}}$ has contributions from Coulomb binding energy in vacuum and solvation free energy of monomers and complex.

6. Implementation details

This section describes the data sets of biomolecular complexes that were studied, the molecular surface triangulation codes used in the calculations, and an extrapolation scheme to efficiently improve the accuracy of the computed $\Delta\Delta G_{\text{elec}}$.

6.1. Data sets of biomolecular complexes

We consider three data sets comprising 51 biomolecular complexes. The PQR files containing the atomic coordinates and radii were originally obtained from Dr. Marcia Fenley at Florida State University [8, 9] and are also available from Dr. Guowei Wei at Michigan State University [55]. Table 1 gives the Protein Data Bank ID of each complex, and Fig. 6 displays the number of atoms and total charge of each monomer and complex.

Data set 1 includes 28 DNA-minor groove drug complexes comprising a large DNA segment (A) and a small drug molecule (B), where the DNA segment is negatively charged ($\approx -20e_c$), while the drug molecule is slightly positively charged. Data set 2 includes 9 wild-type and mutant barnase-barstar complexes in which the barnase (A) and barstar (B) are about the same size, while the barnase is slightly positively charged and the barstar is negatively charged. Data set 3 includes 14 RNA-peptide complexes, where in most cases the RNA (A) is larger than the peptide (B) and is highly negatively charged, while the peptide (B) is positively charged with relatively smaller magnitude.

Table 1: Protein Data Bank ID of biomolecular complexes [8, 9, 55]

data set 1	102d	109d	121d	127d	129d	166d	195d	1d30
	1d63	1d64	1d86	1dne	1eel	1fmq	1fms	1jtl
	1lex	1prp	227d	261d	264d	289d	298d	2dbe
	302d	311d	328d	360d				
data set 2	1b27	1b2s	1b2u	1b3s	1x1u	1x1w	1x1x	1x1y
	2za4							
data set 3	1a1t	1a4t	1biv	1exy	1g70	1hji	1i9f	1mnb
	1nyb	1qfq	1ull	1zbn	2a9x	484d		

6.2. Molecular surface triangulation codes

Two molecular surface triangulation codes were employed, MSMS [14] and NanoShaper [15]. In MSMS the user specifies the density d which sets the

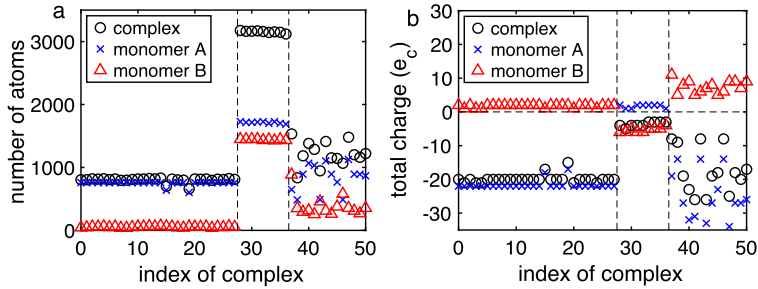


Figure 6: Biomolecular complexes, (a) number of atoms, (b) total charge (e_c), complex (\circ), monomer A (\times), monomer B (\triangle), data set 1 (DNA-drug, index 0-27), data set 2 (barnase-barstar, index 28-36), data set 3 (RNA-peptide, index 37-50).

number of triangle vertices per \AA^2 of surface area, while in NanoShaper the user specifies the scale s which sets the number of grid points per \AA in the marching cubes algorithm used for surface generation. Table 2 lists five values of the MSMS density d_i and NanoShaper scale s_i yielding triangulations of comparable size, where the relation $d_i \approx 1.6s_i^2$ is found empirically. For each density/scale value, Table 2 also gives the corresponding number of triangles N for representative complexes (102d, 1b27, 1a1t) from each data set. For some complexes, MSMS did not produce a valid triangulation for high density $d_{4,5} = 40, 80$, while NanoShaper had no such limitation, so we mainly consider $d_{1,2,3} = 5, 10, 20$, with partial results for $d_{4,5} = 40, 80$ given in the supplementary material. For simplicity in some instances below, the term “density” and notation “ d_i ” refers to both the MSMS density and the corresponding NanoShaper scale in Table 2.

Table 2: Molecular surface triangulation, values of MSMS density d_i and NanoShaper (NS) scale s_i for $i = 1 : 5$ yielding triangulations of comparable size N for representative complexes from each data set

i	d_i	s_i	102d, N		1b27, N		1a1t, N	
			MSMS	NS	MSMS	NS	MSMS	NS
1	5	1.76	33859	33248	79846	78188	55800	55592
2	10	2.47	65945	65580	152996	154400	106729	109728
3	20	3.49	137079	131552	318333	309168	223542	219748
4	40	4.94	276275	264004	639682	621140	449744	440900
5	80	7.13	566091	550432	1305099	1295040	921589	919476

6.3. Extrapolation for improved accuracy

The accuracy of the computed electrostatic binding energy $\Delta\Delta G_{\text{elec}}$ depends on the triangulation density used in solving the PB equation, where higher density yields better accuracy. However, higher density is also more computationally expensive, so here we consider an alternative approach in which the binding energy is computed for two low density values and these are extrapolated to the high density limit. In particular, we assume the computed binding energy depends smoothly on the triangulation density,

$$(19a) \quad \Delta\Delta G_{\text{elec}}(d) \sim c_0 + c_1 d^{-1}, \quad d \rightarrow \infty,$$

$$(19b) \quad \Delta\Delta G_{\text{elec}}(s) \sim c_0 + c_1 s^{-2}, \quad s \rightarrow \infty,$$

where Eq. (19a) is used for MSMS calculations and Eq. (19b) is used for NanoShaper calculations. These asymptotic relations are justified empirically from the numerical results, and a similar approach was previously used in boundary element PB calculations of electrostatic solvation energy [13, 38]. In either case, the constants c_0, c_1 are obtained by interpolating the two $\Delta\Delta G_{\text{elec}}(d)$ values computed using TABI with density d_1, d_2 or d_2, d_3 , and then c_0 is our estimate of the exact binding energy.

7. Numerical results

The results presented below were obtained on a single core of the University of Michigan FLUX cluster with Intel Xeon CPUs running at either 2.5 or 2.8GHz. In this system the type of processor could not be specified, so the timing results were averaged over multiple runs. The code was compiled with the GCC Fortran compiler using the `-O2` optimization flag.

The original TABI code was utilized [56] with atomic partial charges set by the CHARMM force field [24], solvent temperature $T = 298\text{K}$, dielectric constant $\epsilon_1 = 1$ in the solute and $\epsilon_2 = 80$ in the solvent, and ionic strength $\kappa = 0.1\text{M NaCl}$. The treecode used Taylor approximation order $p = 3$, MAC parameter $\theta = 0.5$, and maximum number of particles in a leaf $N_0 = 500$; these values ensure that the treecode approximation error is smaller than the boundary element discretization error.

Results for the three data sets are given in the supplementary material. Tables S1-S3 give the computed $\Delta\Delta G_{\text{elec}}$ using MSMS and NanoShaper with density $d_i/\text{scale } s_i$, and Tables S4-S6 give the extrapolated values using $d_{1,2}, d_{2,3}/s_{1,2}, s_{2,3}$, where for example the notation $d_{1,2}$ means that extrapolation with density d_1 and d_2 was used. Tables S4-S6 also give $\Delta\Delta G_{\text{elec}}$ values

computed using the Matched Interface and Boundary (MIB) PB solver [11], a finite-difference scheme with rigorous treatment of the interface jump conditions [42], geometric singularities [43], and charge singularities [44]. The MIB calculations used fine mesh spacing $h = 0.2\text{\AA}$ and they serve as the reference for the accuracy of the TABI results.

7.1. Accuracy and efficiency of TABI

Table 3 presents the relative deviation (%) in computed electrostatic binding energy $\Delta\Delta G_{\text{elec}}$ between TABI and reference MIB results, where the deviation is averaged over the complexes in each data set. For low triangulation density/scale, TABI+MSMS is more accurate than TABI+NanoShaper for data set 1 and data set 3, but TABI+NanoShaper is more accurate for data set 2. As previously mentioned, for some complexes MSMS did not produce a valid triangulation for high density d_4, d_5 , but NanoShaper was able to provide better accuracy for high scale s_4, s_5 . Extrapolation improves the accuracy for both TABI+MSMS and TABI+NanoShaper, but the latter is more accurate for each data set and for the combined set of all 51 complexes.

Table 3: Relative deviation (%) in electrostatic binding energy $\Delta\Delta G_{\text{elec}}$ between TABI and reference MIB results, MSMS/NanoShaper triangulation density d_i /scale s_i from Table 2, extrapolation with $d_{1,2}, d_{2,3}/s_{1,2}, s_{2,3}$

TABI+MSMS							
	d_1	d_2	d_3	d_4	d_5	$d_{1,2}$	$d_{2,3}$
data set 1	36.8	19.6	14.6	na	na	12.8	11.1
data set 2	52.5	44.0	40.9	na	na	34.6	38.0
data set 3	56.6	34.8	25.4	na	na	13.5	17.0
all	45.0	28.1	22.2	na	na	16.9	17.4
TABI+NanoShaper							
	s_1	s_2	s_3	s_4	s_5	$s_{1,2}$	$s_{2,3}$
data set 1	84.1	44.7	25.0	15.0	10.2	8.1	6.5
data set 2	34.6	18.6	12.3	8.4	7.9	5.6	6.6
data set 3	71.1	39.1	22.3	13.4	10.0	8.7	6.7
all	71.8	38.6	22.0	13.4	9.8	7.8	6.6

Table 4 presents the TABI computational run time (hr) which is the sum of the run times for each complex and its two monomers in a given data set. For extrapolated results, the run time is the sum of the run times for the two density/scale values used in the extrapolation. The results show that for a given density/scale, TABI+NanoShaper is faster than TABI+MSMS,

in agreement with previous solvation energy calculations [57]. In addition, TABI+NanoShaper is faster and more accurate using extrapolation of two low scale values than one high scale calculation; for example considering the combined set of all 51 complexes, scale s_5 yields 9.8% deviation and 100.1 hr run time, while extrapolation $s_{1,2}$ yields 7.8% deviation and 10.7 hr run time.

Table 4: Run time (hr) for electrostatic binding energy calculations using TABI+MSMS/NanoShaper, triangulation density d_i /scale s_i from Table 2, extrapolation with $d_{1,2}, d_{2,3}/s_{1,2}, s_{2,3}$

TABI+MSMS							
	d_1	d_2	d_3	d_4	d_5	$d_{1,2}$	$d_{2,3}$
data set 1	1.2	2.5	6.4	na	na	3.7	8.9
data set 2	2.2	6.5	14.4	na	na	8.6	20.9
data set 3	1.4	4.0	8.8	na	na	5.4	12.8
all	4.7	12.9	29.7	na	na	17.7	42.6
TABI+NanoShaper							
	s_1	s_2	s_3	s_4	s_5	$s_{1,2}$	$s_{2,3}$
data set 1	0.9	2.1	4.8	10.4	24.2	3.0	6.9
data set 2	1.3	3.1	8.1	19.2	49.2	4.4	11.3
data set 3	1.0	2.3	5.2	11.3	26.8	3.2	7.5
all	3.2	7.4	18.2	41.0	100.1	10.7	25.7

Hence extrapolation improves the accuracy and efficiency of electrostatic binding energy calculations, and TABI performs better with NanoShaper than with MSMS. Figure 7 shows this graphically, where the TABI run time (hr) is plotted versus the relative deviation (%) from the reference MIB results for each data set (1, 2, 3, all), and the results for MSMS density $d_{1,2}, d_{2,3}$ (\circ) and NanoShaper scale $s_{1,2}, s_{2,3}$ (\times) are connected by a line.

7.2. Sensitivity of electrostatic binding energy

Several previous studies have noted the sensitivity of electrostatic binding energy calculations to numerical errors [6, 8–10, 12]. Among various terms needed in these calculations, the Coulomb free energy ΔG_{coul} in Eq. (15) is computed directly and has no numerical error, while the solvation free energy ΔG_{solv} in Eq. (16) is subject to the discretization error in the TABI solver. However another issue is that the calculations needed to compute $\Delta\Delta G_{\text{elec}}$ are susceptible to loss of precision due to cancellation of digits.

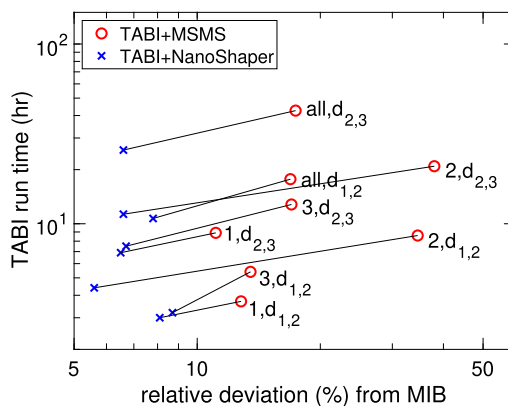


Figure 7: Electrostatic binding energy calculations, TABI run time (hr) versus relative deviation (%) from MIB for each data set (1, 2, 3, all), extrapolation with MSMS density $d_{1,2}$, $d_{2,3}$ (\circ), NanoShaper scale $s_{1,2}$, $s_{2,3}$ (\times).

This can be seen in Table 5 which gives the terms used in calculating $\Delta\Delta G_{\text{elec}}$ for three representative complexes, one from each data set. For example consider complex 102d, where according to Eq. (17), $\Delta\Delta G_{\text{elec}} = 10.52$ is the result of adding $\Delta\Delta G_{\text{coul}} = -1146.38$ and $\Delta\Delta G_{\text{solv}} = 1156.90$; in this case, the two leading digits of the components have been cancelled and the 1st digit of $\Delta\Delta G_{\text{elec}}$ is determined by the 3rd digit of the components. Another instance occurs for complex 1b27, where according to Eq. (18b), $\Delta\Delta G_{\text{solv}} = 589.43$ is the result of subtracting $\Delta G_{\text{solv}}^A + \Delta G_{\text{solv}}^B = -2858.91$ from $\Delta G_{\text{solv}}^{\text{AB}} = -2269.48$; in this case the leading digit of the components has been cancelled and the 1st digit of $\Delta\Delta G_{\text{solv}}$ is determined by the 2nd digit of the components. Cancellation of digits can also occur for the Coulomb energy calculation in Eq. (18a), which even though it is done directly can still be sensitive to the choice of force field used to assign the partial atomic charges [12].

The issue arises when leading digits are cancelled in the calculations needed to compute a $\Delta\Delta G$ value, and it is difficult to know in advance when the problem will occur. Hence to ensure that the electrostatic binding energy $\Delta\Delta G_{\text{elec}}$ is accurate to a certain number of digits, the solvation free energy ΔG_{solv} should be accurate to several more digits. This is consistent with the results of finite-difference calculations of binding energy [6, 8–10]. In the case of TABI calculations, it emphasizes the utility of extrapolating low triangulation density results to the high density limit to efficiently improve accuracy.

Table 5: Terms used in calculating $\Delta\Delta G_{\text{elec}}$ for representative complexes from each data set, energy in kcal/mol, solvation energy and binding energy are extrapolated using NanoShaper scale $s_{1,2}$

	102d	1b27	1alt
$\Delta G_{\text{coul}}^{\text{A}}$	-18772.59	-34506.17	-16493.30
$\Delta G_{\text{coul}}^{\text{B}}$	-439.32	-27780.19	-16729.86
$\Delta G_{\text{coul}}^{\text{A}} + \Delta G_{\text{coul}}^{\text{B}}$	-19211.91	-62286.36	-33223.16
$\Delta G_{\text{coul}}^{\text{AB}}$	-20358.29	-62786.71	-37697.99
$\Delta\Delta G_{\text{coul}}$	-1146.38	-500.35	-4474.83
$\Delta G_{\text{solv}}^{\text{A}}$	-6235.19	-1273.09	-5402.25
$\Delta G_{\text{solv}}^{\text{B}}$	-149.91	-1585.82	-2499.25
$\Delta G_{\text{solv}}^{\text{A}} + \Delta G_{\text{solv}}^{\text{B}}$	-6385.10	-2858.91	-7901.50
$\Delta G_{\text{solv}}^{\text{AB}}$	-5228.20	-2269.48	-3357.10
$\Delta\Delta G_{\text{solv}}$	1156.90	589.43	4544.40
$\Delta\Delta G_{\text{elec}}$	10.52	89.08	69.57

8. Conclusion

This work presented computations of electrostatic binding energy $\Delta\Delta G_{\text{elec}}$ for 51 solvated biomolecular complexes using the treecode-accelerated boundary integral (TABI) Poisson-Boltzmann solver. TABI computes the electric potential on the triangulated molecular surface of a complex and its monomers, and further processing yields the solvation free energy ΔG_{solv} needed to compute the binding energy. The accuracy of the TABI results was verified using the high-order finite-difference Matched Interface and Boundary (MIB) method as the reference. Among two codes used here for surface triangulation, MSMS and NanoShaper, the latter is found to be more accurate, efficient, and robust. It was shown that the accuracy of the computed $\Delta\Delta G_{\text{elec}}$ using TABI can be efficiently improved by extrapolating low triangulation density results to the high density limit. The calculations needed to compute $\Delta\Delta G_{\text{elec}}$ are susceptible to loss of precision due to cancellation of digits and this emphasizes the need for relatively higher accuracy in computing ΔG_{solv} . An updated version of the TABI-PB solver is available in Github [58] or as a contributed module in APBS [59, 60]. It is hoped that the capability demonstrated here in computing electrostatic binding energy of solvated biomolecular complexes may facilitate biophysical modeling and screening studies for synthetic drug design.

Appendix: Supplementary material

Table S1: Data set 1, electrostatic binding energy $\Delta\Delta G_{\text{elec}}$ (kcal/mol), MSMS density d_i , NanoShaper scale s_i from Table 2

PDB ID	NanoShaper					MSMS				
	s_1	s_2	s_3	s_4	s_5	d_1	d_2	d_3	d_4	d_5
102d	16.1	13.3	11.6	10.7	10.3	14.5	11.5	10.6	10.4	10.1
109d	7.6	5.0	4.2	3.5	3.2	5.2	3.7	3.6	3.3	na
121d	35.4	29.8	27.2	26.1	25.5	29.6	26.9	25.7	25.2	na
127d	37.0	33.8	32.0	30.9	30.6	34.2	32.2	31.3	na	na
129d	48.3	44.6	42.9	41.8	41.6	45.8	43.1	42.5	41.7	na
166d	21.6	18.5	17.0	16.4	15.9	17.9	16.7	16.2	15.9	15.7
195d	11.3	7.9	5.8	5.1	4.4	7.0	5.5	4.7	na	na
1d30	19.3	15.3	13.0	11.7	11.5	15.5	12.7	11.8	11.3	na
1d63	24.2	18.5	15.9	14.2	13.8	18.0	15.4	14.1	11.8	12.0
1d64	20.9	17.6	16.2	15.6	15.3	16.7	16.1	15.4	15.2	15.1
1d86	37.7	32.2	28.9	27.8	26.9	31.4	28.9	27.2	27.0	26.5
1dne	34.1	28.5	25.9	24.9	24.3	28.4	25.8	24.5	24.3	24.0
1eel	21.6	18.3	16.7	15.8	15.2	18.5	16.3	15.5	na	na
1fmq	21.9	18.1	16.4	15.9	15.5	18.5	16.5	15.7	na	na
1fms	34.0	29.7	28.0	27.2	26.2	30.2	28.0	27.2	26.3	na
1jtl	18.8	14.6	12.8	11.7	11.4	13.6	11.6	11.5	11.1	na
1lex	20.9	15.1	12.5	11.3	10.7	14.5	11.8	11.0	12.2	na
1prp	18.6	15.2	13.3	12.7	12.3	14.6	13.2	12.4	12.2	12.0
227d	15.7	11.7	9.1	7.8	7.4	11.2	9.0	7.9	7.1	na
261d	11.8	7.4	5.1	3.6	3.0	4.2	1.4	0.8	na	na
264d	37.9	35.3	33.9	33.5	33.1	35.8	33.5	32.7	31.9	na
289d	23.6	20.0	18.8	18.0	17.8	19.6	18.3	17.7	17.5	na
298d	21.8	18.5	16.9	16.5	16.0	18.4	16.6	16.0	15.8	na
2dbe	15.5	11.0	8.5	7.4	6.9	10.4	8.4	9.1	na	na
302d	32.1	28.6	27.0	26.1	25.7	28.7	26.6	25.9	na	na
311d	20.4	14.7	12.2	11.1	10.4	14.3	11.6	10.4	10.1	na
328d	23.0	20.4	19.1	18.6	18.2	20.5	18.6	17.9	na	na
360d	88.5	82.9	80.4	79.3	79.1	58.5	57.2	56.1	55.8	na

Table S2: Data set 2, electrostatic binding energy $\Delta\Delta G_{\text{elec}}$ (kcal/mol), MSMS density d_i , NanoShaper scale s_i from Table 2

PDB ID	NanoShaper					MSMS				
	s_1	s_2	s_3	s_4	s_5	d_1	d_2	d_3	d_4	d_5
1b27	118.3	103.9	97.9	93.6	92.6	125.9	118.6	116.1	113.7	na
1b2s	102.7	89.4	82.5	78.9	78.3	108.4	102.4	99.8	na	na
1b2u	102.6	91.7	88.0	84.6	84.8	117.6	112.8	111.5	110.0	na
1b3s	78.6	63.3	59.0	55.2	54.6	76.1	69.5	67.0	67.2	na
1x1u	105.0	94.5	87.0	83.1	82.6	107.4	98.2	94.9	93.3	na
1x1w	123.9	113.2	105.9	102.2	101.5	135.7	128.5	126.4	na	na
1x1x	147.1	132.5	125.8	121.6	121.0	162.8	156.0	153.0	148.0	na
1x1y	116.2	105.3	99.6	96.4	95.4	140.7	134.1	132.3	na	na
2za4	86.4	76.4	70.5	68.2	67.4	141.8	135.9	133.2	na	na

Table S3: Data set 3, electrostatic binding energy $\Delta\Delta G_{\text{elec}}$ (kcal/mol), MSMS density d_i , NanoShaper scale s_i from Table 2

PDB ID	NanoShaper					MSMS				
	s_1	s_2	s_3	s_4	s_5	d_1	d_2	d_3	d_4	d_5
1a1t	103.8	86.9	76.6	72.0	69.0	108.9	97.0	91.1	87.5	na
1a4t	99.9	87.4	80.4	77.5	75.7	89.5	80.4	77.0	75.4	na
1biv	75.6	59.4	53.4	48.7	46.8	60.6	51.8	48.3	46.7	45.9
1exy	219.3	200.9	191.5	185.7	184.3	206.7	193.4	185.9	na	na
1g70	169.3	153.1	144.7	140.2	139.0	158.4	148.4	145.9	160.3	na
1hji	84.9	69.7	63.1	59.1	57.0	81.4	71.1	66.8	64.2	68.0
1i9f	13.6	-0.6	-9.8	-13.5	-16.4	2.1	-7.2	-12.9	-14.7	na
1mnb	165.4	148.4	140.1	136.4	133.4	153.2	141.2	135.7	na	na
1nyb	17.7	4.7	-2.4	-5.8	-7.2	9.6	1.3	-2.2	-4.7	na
1qfq	45.8	35.0	29.8	25.8	25.3	48.9	40.3	38.0	35.4	36.0
1ull	-12.9	-35.8	-46.5	-53.5	-56.3	-21.2	-38.8	-46.8	na	na
1zbn	250.3	234.6	225.3	221.3	219.2	239.6	228.7	224.5	221.7	na
2a9x	413.4	399.8	395.3	389.2	387.8	402.3	395.3	390.4	387.1	na
484d	189.8	166.5	153.9	147.2	143.6	191.6	174.9	168.8	na	na

Table S4: Data set 1, extrapolated binding energy $\Delta\Delta G_{\text{elec}}$ (kcal/mol), MSMS density d_i , NanoShaper scale s_i from Table 2, MIB grid spacing $h = 0.2\text{\AA}$

PDB ID	NanoShaper		MSMS		MIB
	$s_{1,2}$	$s_{2,3}$	$d_{1,2}$	$d_{2,3}$	h
102d	10.5	9.8	8.3	9.8	10.3
109d	2.4	3.4	2.1	3.5	2.7
121d	24.0	24.6	24.0	24.6	23.9
127d	30.5	30.2	30.1	30.5	29.1
129d	40.8	41.3	40.3	41.9	40.2
166d	15.4	15.5	15.4	15.7	15.7
195d	4.4	3.8	3.9	4.0	2.7
1d30	11.3	10.6	9.8	11.0	10.6
1d63	12.8	13.2	12.6	12.9	12.4
1d64	14.3	14.7	15.5	14.7	14.6
1d86	26.5	25.7	26.3	25.6	25.5
1dne	22.7	23.4	23.0	23.3	22.8
1eel	14.9	15.1	14.0	14.7	15.1
1fmq	14.2	14.7	14.4	15.0	15.4
1fms	25.5	26.3	25.7	26.5	25.7
1jtl	10.3	11.0	9.5	11.4	11.5
1lex	9.1	9.9	9.0	10.3	9.7
1prp	11.9	11.4	11.7	11.7	11.6
227d	7.5	6.6	6.7	6.9	5.6
261d	3.0	2.9	-1.6	0.2	2.9
264d	32.7	32.4	31.1	32.0	32.3
289d	16.4	17.5	16.9	17.1	16.6
298d	15.1	15.4	14.7	15.4	15.4
2dbe	6.4	6.1	6.3	9.8	5.8
302d	24.9	25.5	24.4	25.2	25.2
311d	8.8	9.7	8.8	9.3	9.3
328d	17.7	17.9	16.6	17.2	17.5
360d	77.0	77.9	55.8	55.1	55.6

Table S5: Data set 2, extrapolated binding energy $\Delta\Delta G_{\text{elec}}$ (kcal/mol), MSMS density d_i , NanoShaper scale s_i from Table 2, MIB grid spacing $h = 0.2\text{\AA}$

PDB ID	NanoShaper		MSMS		MIB
	$s_{1,2}$	$s_{2,3}$	$d_{1,2}$	$d_{2,3}$	h
1b27	89.1	91.9	110.6	113.8	87.1
1b2s	76.0	75.5	95.8	97.4	72.1
1b2u	80.5	84.4	107.5	110.3	78.6
1b3s	47.6	54.8	62.2	64.7	49.3
1x1u	83.8	79.5	88.2	91.8	76.0
1x1w	102.2	98.6	120.6	124.5	95.3
1x1x	117.6	119.2	148.6	150.2	114.7
1x1y	94.2	94.0	126.9	130.6	89.2
2za4	66.2	64.7	129.4	130.7	74.4

Table S6: Data set 3, extrapolated binding energy $\Delta\Delta G_{\text{elec}}$ (kcal/mol), MSMS density d_i , NanoShaper scale s_i from Table 2, MIB grid spacing $h = 0.2\text{\AA}$

PDB ID	NanoShaper		MSMS		MIB
	$s_{1,2}$	$s_{2,3}$	$d_{1,2}$	$d_{2,3}$	h
1a1t	69.6	66.3	84.0	85.7	63.0
1a4t	74.6	73.4	70.5	73.8	72.3
1biv	43.0	47.3	42.2	45.0	44.8
1exy	182.4	182.1	178.9	178.9	177.4
1g70	136.5	136.4	137.4	143.5	133.5
1hji	54.0	56.5	59.8	62.9	51.2
1i9f	-15.2	-19.0	-17.5	-18.2	-19.2
1mnb	130.9	131.8	128.1	130.6	128.2
1nyb	-8.7	-9.5	-8.0	-5.4	-12.6
1qfq	24.0	24.7	30.8	35.9	20.3
1ull	-59.3	-57.2	-58.2	-54.2	-52.8
1zbn	218.5	215.9	216.9	220.6	215.7
2a9x	385.6	390.9	387.7	385.8	385.4
484d	142.5	141.3	156.8	163.1	133.4

References

- [1] F. Dong, M. Vijaykumar, H.-X. Zhou, *Comparison of calculation and experiment implicates significant electrostatic contributions to the binding stability of barnase and barstar*. Biophys. J., **85**:49–60, 2003.
- [2] L. P. Lee, B. Tidor, *Barstar is electrostatically optimized for tight binding to barnase*. Nat. Struct. Mol. Biol., **8**:73–76, 2001.
- [3] L. P. Lee, B. Tidor, *Optimization of binding electrostatics: Charge complementarity in the barnase-barstar protein complex*. Protein Science, **10**:362–377, 2001.
- [4] D. G. Brown, M. R. Sanderson, E. Garman, S. Neidle, *Crystal structure of a berenil-d(CGCAAATTTGCG) complex: An example of drug-DNA recognition based on sequence-dependent structural features*. J. Mol. Biol., **226**:481–490, 1992.
- [5] T. C. Leeper, Z. Athanassiou, R. L. A. Dias, J. A. Robinson, G. Varani, *TAR RNA recognition by a cyclic peptidomimetic of Tat protein*. Biochemistry, **44**:1236–12372, 2005.
- [6] S. Izadi, B. Aguilar, A. V. Onufriev, *Protein–ligand electrostatic binding free energies from explicit and implicit solvation*. J. Chem. Theory Comput., **11**:4450–4459, 2015.
- [7] C. Bertoni, B. Honig, E. G. Alexov, *Poisson–Boltzmann calculations of nonspecific salt effects on protein–protein binding free energies*. Biophys. J., **92**:1891–1899, 2007.
- [8] R. C. Harris, A. H. Boschitsch, M. O. Fenley, *Influence of grid spacing in Poisson–Boltzmann equation binding energy estimation*. J. Chem. Theory Comput., **9**, 3677–3685, 2013.
- [9] R. C. Harris, T. Mackoy, M. O. Fenley, *Problems of robustness in Poisson–Boltzmann binding free energies*. J. Chem. Theory Comput., **11**:705–712, 2015.
- [10] A. Li, K. Gao, *Accurate estimation of electrostatic binding energy with Poisson–Boltzmann equation solver DelPhi program*. J. Theor. Comput. Chem., **15**:1650071, 2016.
- [11] D. D. Nguyen, B. Wang, G. W. Wei, *Accurate, robust, and reliable calculations of Poisson–Boltzmann binding energies*. J. Comput. Chem., **38**:941–948, 2017.

- [12] A. Chakavorty, L. Li, E. G. Alexov, *Electrostatic component of binding energy: Interpreting predictions from Poisson–Boltzmann equation and modeling protocols*. J. Comput. Chem., **37**:2495–2507, 2016.
- [13] W. Geng, R. Krasny, *A treecode-accelerated boundary integral Poisson–Boltzmann solver for electrostatics of solvated biomolecules*. J. Comput. Phys., **247**:62–78, 2013. [MR3066162](#)
- [14] M. F. Sanner, A. J. Olson, J.-C. Spohner, *Reduced surface: An efficient way to compute molecular surfaces*. Biopolymers, **38**:305–320, 1996.
- [15] S. Decherchi, W. Rocchia, *A general and robust ray-casting-based algorithm for triangulating surfaces at the nanoscale*. PLoS ONE, **8**:e59744, 2013.
- [16] M. J. Holst, *The Poisson-Boltzmann Equation: Analysis and Multi-level Numerical Solution*. PhD thesis, University of Illinois Urbana-Champaign, 1994. [MR2690378](#)
- [17] W. Geng, *A boundary integral Poisson-Boltzmann solver package for solvated bimolecular simulations*. Mol. Based Math. Biol., **3**:43–58, 2015. [MR3372488](#)
- [18] N. A. Baker, *Improving implicit solvent simulations: a Poisson-centric view*. Curr. Opin. Struct. Biol., **15**:137–143, 2005.
- [19] B. Lu, Y. C. Zhou, M. J. Holst, J. A. McCammon, *Recent progress in numerical methods for the Poisson–Boltzmann equation in biophysical applications*. Commun. Comput. Phys., **3**:973–1009, 2008.
- [20] J. Warwicker, H. C. Watson, *Calculation of the electric potential in the active site cleft due to α -helix dipoles*. J. Mol. Biol., **157**:671–679, 1982.
- [21] B. Honig, A. Nicholls, *Classical electrostatics in biology and chemistry*. Science, **268**:1144–1149, 1995.
- [22] M. E. Davis, J. D. Madura, J. Sines, B. A. Luty, S. A. Allison, J. A. McCammon, *Diffusion-controlled enzymatic reactions*. Methods Enzymol., **202**:473–497, 1991.
- [23] W. Im, D. Beglov, B. Roux, *Continuum solvation model: electrostatic forces from numerical solutions to the Poisson–Boltzmann equation*. Comput. Phys. Commun., **111**:59–75, 1998.
- [24] A. D. MacKerell Jr., D. Bashford, M. Bellott, J. D. Dunbrack, M. J. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Mich-

- nick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, M. Karplus, *All-atom empirical potential for molecular modeling and dynamics studies of proteins*. J. Phys. Chem. B, **102**:3586–3616, 1998.
- [25] N. A. Baker, D. Sept, S. Joseph, M. J. Holst, J. A. McCammon, *Electrostatics of nanosystems: application to microtubules and the ribosome*. Proc. Natl. Acad. Sci. U.S.A., **98**:10037–10041, 2001.
- [26] R. Luo, L. David, M. K. Gilson, *Accelerated Poisson–Boltzmann calculations for static and dynamic systems*. J. Comput. Chem., **23**:1244–1253, 2002.
- [27] Q. Cai, J. Wang, H.-K. Zhao, R. Luo, *On removal of charge singularity in Poisson–Boltzmann equation*. J. Chem. Phys., **130**:145101, 2009.
- [28] D. Chen, Z. Chen, C. Chen, W. Geng, G. W. Wei, *MIBPB: A software package for electrostatic analysis*. J. Comput. Chem., **32**:756–770, 2011.
- [29] D. Xie, *New solution decomposition and minimization schemes for Poisson–Boltzmann equation in calculation of biomolecular electrostatics*. J. Comput. Phys., **275**:294–309, 2014. [MR3250333](#)
- [30] W. Deng, X. Zhufu, J. Xu, S. Zhao, *A new discontinuous Galerkin method for the nonlinear Poisson–Boltzmann equation*. Appl. Math. Lett., **257**:1000–1021, 2015. [MR3361706](#)
- [31] R. J. Zauhar, R. S. Morgan, *A new method for computing the macromolecular electric potential*. J. Mol. Biol., **186**:815–820, 1985.
- [32] A. H. Juffer, E. F. F. Botta, B. A. M. van Keulen, A. van der Ploeg, H. J. C. Berendsen, *The electric potential of a macromolecule in a solvent: a fundamental approach*. J. Comput. Phys., **97**:144–171, 1991.
- [33] A. H. Boschitsch, M. O. Fenley, H.-X. Zhou, *Fast boundary element method for the linear Poisson–Boltzmann equation*. J. Phys. Chem. B, **106**:2741–2754, 2002.
- [34] B. Lu, X. Cheng, J. Huang, J. A. McCammon, *Order N algorithm for computation of electrostatic interactions in biomolecular systems*. Proc. Natl. Acad. Sci. U.S.A., **103**:19314–19319, 2006.
- [35] B. Lu, X. Cheng, J. A. McCammon, *“New-version-fast-multipole-method” accelerated electrostatic calculations in biomolecular systems*. J. Comput. Phys., **226**:1348–1366, 2007. [MR2356376](#)

- [36] C. Bajaj, S.-C. Chen, A. Rand, *An efficient higher-order fast multipole boundary element solution for Poisson–Boltzmann-based molecular electrostatics*. SIAM J. Sci. Comput., **33**:826–848, 2011. [MR2801191](#)
- [37] B. Zhang, B. Lu, X. Cheng, J. Huang, N. P. Pitsianis, X. Sun, J. A. McCammon, *Mathematical and numerical aspects of the Adaptive Fast Multipole Poisson–Boltzmann solver*. Commun. Comput. Phys., **13**:107–128, 2013.
- [38] C. D. Cooper, J. P. Bardhan, L. A. Barba, *A biomolecular electrostatics solver using Python, GPUs and boundary elements that can handle solvent-filled cavities and Stern layers*. Comput. Phys. Commun., **185**:720–729, 2014. [MR3161134](#)
- [39] N. A. Baker, *Poisson–Boltzmann methods for biomolecular electrostatics*. In “Numerical Computer Methods”, Chapter 5, Methods Enzymol., **383**:94–118, 2004.
- [40] Q. Lu, R. Luo, *A Poisson–Boltzmann dynamics method with nonperiodic boundary condition*. J. Chem. Phys., **119**:11035–11047, 2003.
- [41] W. Rocchia, S. Sridharan, A. Nicholls, E. G. Alexov, A. Chiabrera, B. Honig, *Rapid grid-based construction of the molecular surface and the use of induced surface charge to calculate reaction field energies: Applications to the molecular systems and geometric objects*. J. Comput. Chem., **23**:128–137, 2002.
- [42] Y. C. Zhou, S. Zhao, M. Feig, G. W. Wei, *High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources*. J. Comput. Phys., **213**:1–30, 2006. [MR2186592](#)
- [43] S. Yu, W. Geng, G. W. Wei, *Treatment of geometric singularities in implicit solvent models*. J. Chem. Phys., **126**:244108, 2007.
- [44] W. Geng, S. Yu, G. W. Wei, *Treatment of charge singularities in implicit solvent models*. J. Chem. Phys., **127**:114106, 2007.
- [45] Y. Saad, M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*. SIAM J. Sci. Stat. Comput., **7**:856–869, 1986. [MR0848568](#)
- [46] L. Greengard, V. Rokhlin, *A fast algorithm for particle simulations*. J. Comput. Phys., **73**:325–348, 1987. [MR0918448](#)

- [47] H. Cheng, L. Greengard, V. Rokhlin, *A fast adaptive multipole algorithm in three dimensions*. J. Comput. Phys., **155**:468–498, 1999. [MR1723309](#)
- [48] L. Greengard, J. Huang, *A new version of the Fast Multipole Method for screened Coulomb interactions in three dimensions*. J. Comput. Phys., **180**:642–658, 2002. [MR1917861](#)
- [49] J. E. Barnes, P. Hut, *A hierarchical $O(N\log N)$ force-calculation algorithm*. Nature, **324**:446–449, 1986.
- [50] Z.-H. Duan, R. Krasny, *An adaptive treecode for computing nonbonded potential energy in classical molecular systems*. J. Comput. Chem., **22**:184–195, 2001.
- [51] P. Li, H. Johnston, R. Krasny, *A Cartesian treecode for screened Coulomb interactions*. J. Comput. Phys., **228**:3858–3868, 2009. [MR2511077](#)
- [52] K. A. Sharp, B. Honig, *Calculating total electrostatic energies with the nonlinear Poisson–Boltzmann equation*. J. Phys. Chem., **94**:7684–7692, 1990.
- [53] M. K. Gilson, M. E. Davis, B. A. Luty, J. A. McCammon, *Computation of electrostatic forces on solvated molecules using the Poisson–Boltzmann equation*. J. Phys. Chem., **97**:3591–3600, 1993.
- [54] W. Geng, G. W. Wei, *Multiscale molecular dynamics using the matched interface and boundary method*. J. Comput. Phys., **230**:435–457, 2011. [MR2737637](#)
- [55] Biomolecular complex PQR files (from M. O. Fenley, available at website of G. W. Wei, Michigan State University) <http://users.math.msu.edu/users/wei/Data/bindingdata.tar.gz>
- [56] Code used in this work, www.sourceforge.net/projects/tabipb
- [57] L. Wilson, R. Krasny, *Comparison of the MSMS and NanoShaper molecular surface triangulation codes in the TABI Poisson–Boltzmann solver*. J. Comput. Chem., 2021. DOI: [10.1002/jcc.26692](https://doi.org/10.1002/jcc.26692)
- [58] www.github.com/Treecodes/TABI-PB
- [59] E. Jurrus, D. Engel, K. Star, K. Monson, J. Brandi, L. E. Felberg, D. H. Brookes, L. Wilson, J. Chen, K. Liles, M. Chun, P. Li, D. W. Gohara, T. Dolinsky, R. Konecny, D. R. Koes, J. E. Nielsen, T. Head-Gordon, W. H. Geng, R. Krasny, G.-W. Wei, M. J. Holst, J. A. McCammon,

N. A. Baker, *Improvements to the APBS biomolecular solvation software suite*, Protein Science, **27**:112–128, 2018.

[60] www.github.com/Electrostatics/apbs

LEIGHTON WILSON
DEPARTMENT OF MATHEMATICS
UNIVERSITY OF MICHIGAN
ANN ARBOR, MI 48109, USA
E-mail address: lwilson@umich.edu

JINGZHEN HU
DEPARTMENT OF MATHEMATICS
DUKE UNIVERSITY
DURHAM, NC 27710, USA
E-mail address: jingzhen.hu@duke.edu

JIAHUI CHEN
DEPARTMENT OF MATHEMATICS
MICHIGAN STATE UNIVERSITY
EAST LANSING, MI 48824, USA
E-mail address: chenj159@msu.edu

ROBERT KRASNY
DEPARTMENT OF MATHEMATICS
UNIVERSITY OF MICHIGAN
ANN ARBOR, MI 48109, USA
E-mail address: krasny@umich.edu

WEIHUA GENG
DEPARTMENT OF MATHEMATICS
SOUTHERN METHODIST UNIVERSITY
DALLAS, TX 75275, USA
E-mail address: wgeng@smu.edu

RECEIVED SEPTEMBER 17, 2021