

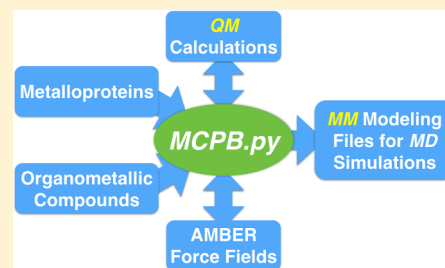
MCPB.py: A Python Based Metal Center Parameter Builder

Pengfei Li and Kenneth M. Merz, Jr.*

Department of Chemistry, Department of Biochemistry and Molecular Biology, Michigan State University, East Lansing, Michigan 48824-1322, United States

Supporting Information

ABSTRACT: MCPB.py, a python based metal center parameter builder, has been developed to build force fields for the simulation of metal complexes employing the bonded model approach. It has an optimized code structure, with far fewer required steps than the previous developed MCPB program. It supports various AMBER force fields and more than 80 metal ions. A series of parametrization schemes to derive force constants and charge parameters are available within the program. We give two examples (one metalloprotein example and one organometallic compound example), indicating the program's ability to build reliable force fields for different metal ion containing complexes. The original version was released with AmberTools15. It is provided via the GNU General Public License v3.0 (GNU_GPL_v3) agreement and is free to download and distribute. MCPB.py provides a bridge between quantum mechanical calculations and molecular dynamics simulation software packages thereby enabling the modeling of metal ion centers. It offers an entry into simulating metal ions in a number of situations by providing an efficient way for researchers to handle the vagaries and difficulties associated with metal ion modeling.



INTRODUCTION

Metal ions are ubiquitous in biological systems with estimates that over 50% of proteins containing metal ions.¹ Force fields (FFs) are used to represent the relationship between the energy and configuration of a molecule.² FFs have a significant speed advantage over quantum mechanical (QM) methods and are powerful tools to study macromolecular systems at the atomic level. FFs are generally coupled with Monte Carlo (MC)³ or molecular dynamics (MD)⁴ sampling methods to study the statistical properties of a system. There are several different modeling strategies for metal ions in FFs: the bonded model approach,⁵ nonbonded model approach,^{6,7} cationic dummy atom approach,⁸ polarizable model approach,⁹ etc. The bonded model is widely used in contemporary force fields like AMBER,¹⁰ CHARMM,¹¹ GROMOS¹² and OPLS-AA,¹³ and the generic functional form is shown in eqs 1–3 (CHARMM has an additional 1-3 Urey–Bradley nonbonded term, which is not shown in these eqs). The total energy of a system is represented by its bonded part and nonbonded part (see eq 1). The bonded energy consists of the bond, angle and torsion terms (see eq 2). The bond and angle terms are described using harmonic equations while the torsion potentials is represented by a Fourier expansion. In eq 2, k_r , r_{ij} and $r_{ij,eq}$ are the bond force constant, bond length and equilibrium bond length; k_θ , θ_{ij} and $\theta_{ij,eq}$ are the angle force constant, angle amplitude and equilibrium angle values; V_n , n , ω and γ are the torsion barrier, periodicity, torsion angle and phase, respectively. The nonbonded energy is made up of electrostatic and van der Waals (VDW) interactions (see eq 3). A Coulombic equation is usually utilized to describe the electrostatic interactions and a 12-6 Lennard-Jones (LJ) potential is usually used for the VDW interactions. In eq 3 q_i and q_j are the partial charges of the two

particles, $R_{min,ij}$ and ϵ_{ij} are the distance of minimum energy and well depth of the 12-6 LJ potential.

$$U(r_{ij}) = U_{\text{bonded}} + U_{\text{nonbonded}} \quad (1)$$

$$U_{\text{bonded}}(r_{ij}) = \sum_{\text{all bonds}} k_r (r_{ij} - r_{ij,eq})^2 + \sum_{\text{all angles}} k_\theta (\theta_{ij} - \theta_{ij,eq})^2 + \sum_{\text{all torsions}} \sum_n \frac{1}{2} V_n [1 + \cos(n\omega - \gamma)] \quad (2)$$

$$U_{\text{nonbonded}}(r_{ij}) = \sum_{i,j \neq i} \left\{ \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} + \epsilon_{ij} \left[\left(\frac{r_{ij}}{R_{min,ij}} \right)^{12} - 2 \left(\frac{r_{ij}}{R_{min,ij}} \right)^6 \right] \right\} \quad (3)$$

Modeling transition metal ion containing systems with FFs is challenging due to the intrinsic chemistries of these species. They can have several coordination modes¹⁴ and the possible binding modes of metal complexes are more numerous than what is seen in typical organic molecules. For example, zinc can have four-, five- and six- coordination binding modes.¹⁵ If we consider each kind of ligand that could potentially bind to zinc, the number of possible complexes would be quite large, making

Received: November 6, 2015

Published: February 25, 2016

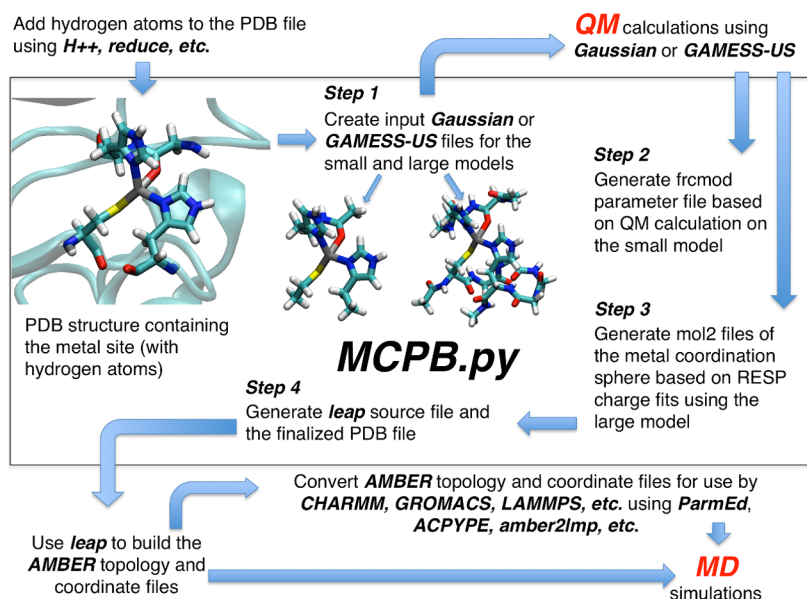


Figure 1. Workflow of the MCPB.py program.

their modeling even more challenging if there is a need to develop force fields for different metal complexes individually.

There are several ways to obtain force constants parameters: through empirical methods,¹⁶ through experiments (e.g., X-ray, NMR, normal-mode analysis of spectra, etc.),^{17,18} or based on theoretically calculated Hessian matrices.^{19,20} In general, the empirical method could be applied more broadly but usually offers limited accuracy. Deriving parameters from experimental information is very challenging and time-consuming, thereby restricting it to specific systems. Parameter determination based on quantum calculations offers considerable accuracy and is broadly applicable to a wide range of molecular configurations.

Seminario proposed a method that uses the Cartesian Hessian matrix to calculate the force constant (referred as the Seminario method) and validated it through a series of small organic molecules.²⁰ Nilsson et al. developed the Hess2FF software to calculate the force constants through the Seminario method. They applied it to 5 different systems with some containing either a Fe or Zn ion.²¹ Lin and Wang applied the Seminario method and the restrained electrostatic potential (RESP) fitting scheme to zinc complexes with the general AMBER force field (GAFF).²² They showed that the bonded model with RESP fitted charges showed the best performance among the models investigated.²² Peters et al. developed the metal center parameter builder (MCPB) software based on the MTK++ software package using the C++ language.⁵ It is a semiautomatic tool for the parametrization of metal ion containing molecules. The zinc AMBER force field (ZAFF) has been developed for four-coordinated zinc complexes in protein system using the Seminario/ChgModB combination (shown as the best combination) using MCPB.⁵ After the availability of MCPB, a broad range of metal ion containing systems have been parametrized using this package.^{23–26} However, even with the considerable time saving afforded by MCPB, the process is still overly complicated for the nonexpert.

Herein we introduce the MCPB.py software, a python based metal center parameter builder, which streamlines much of the functionality of MCPB into a much easier to use package. It was built on the python metal site modeling toolbox (pyMSMT) in AmberTools15.²⁷ It uses a much more optimized workflow,

offering a more user-friendly experience with far fewer steps requiring user input. It supports Z-matrix, Seminario and empirical methods for the parametrization. In the current version, it supports a variety of AMBER force fields, more than 80 ions and two widely used QM software packages (Gaussian²⁸ and GAMESS-US²⁹). Together with ParmEd²⁷ or ACPYPE,³⁰ the topology and coordinate files of AMBER can be converted to the formats used by CHARMM³¹ or GROMACS.³² We expect the MCPB.py application to expand further and expedite the modeling of ions in metalloproteins and organometallic compounds using a range of packages and FFs.

CODE STRUCTURE

MCPB.py is built using the molecule and atom classes in the pyMSMT package. There are different parsers (e.g., PDB, mol2) available that read and write different format files. Users can build their own program based on these data structures as well. The installed main program of MCPB.py is found in the \$AMBERHOME/bin directory. The source code for the MCPB.py program is freely available in AmberTools for those interested in delving into the details. The original version of the code is available as part of the AmberTools15 package and updated versions can be obtained from GitHub (<https://github.com/Amber-MD/pysmt>). We suggest users that begin with the two examples in the SI to familiarize themselves with the program and then check the source code for particulars.

WORKFLOW

Figure 1 illustrates the workflow of the MCPB.py program. It facilitates metal site modeling in classical FFs with parameters derived from QM calculations. It is designed as a bridge between several QM software packages (including Gaussian²⁸ and GAMESS-US²⁹) and MD software employing FFs (including AMBER,³³ CHARMM,³¹ GROMACS,³² NAMD,³⁴ OpenMM,³⁵ LAMMPS,³⁶ etc.) for the modeling of metal-containing systems. The current version of MCPB.py supports the following AMBER force fields: ff94, ff99, ff99SB, ff03, ff03.r1, ff10, ff12SB, ff14SB and GAFF. The default is ff14SB for the protein whereas GAFF is used for small organic ligands.

Table 1. Small Model Capping Schemes for Coordinated Amino Acid Residues

atom bound to central metal ion	original residue	small model
side chain atom in an amino acid	NH-CHR-CO	CH ₃ R
backbone O atom in a nonterminal residue	NH-CHR-CO	CH ₃ -CO or CH ₂ R-CO (if this residue also has a side chain atom bound to the metal ion) *C-terminal converted to NH-CH ₃ (if it does not bond to the metal ion), or to NH-CH ₂ R (if it has a side chain atom bound to the metal ion)
backbone N atom in a nonterminal residue	NH-CHR-CO	NH-CH ₃ or NH-CH ₂ R (if this residue also has a side chain atom bound to the metal ion) *N-terminal converted to CH ₃ -CO (if it does not bond to the metal ion), or to CH ₂ R-CO (if it has a side chain atom bound to the metal ion)
backbone CO ₂ ⁻ in the C-terminal residue	NH-CHR-CO ₂ ⁻	CH ₃ -CO ₂ ⁻
backbone NH ₂ in the N-terminal residue	NH ₂ -CHR-CO	NH ₂ -CH ₃
backbone N and O in one residue	NH-CHR-CO	NH-CH ₂ -CO *N-terminal modeled as CH ₃ -CO (or to CH ₂ R-CO if it has a side chain atom bound to the metal ion) and the C-terminal to NH-CH ₃ (or to NH-CH ₂ R if it has a side chain atom bound to the metal ion)

The program should be used with AmberTools15 or higher versions because it uses code, data files and the resp program from the AmberTools15 software package.²⁷

Similar to MCPB,⁵ MCPB.py uses two models to do the parametrization in order to strike a balance between accuracy and speed (as shown in Figure 1): a smaller one to obtain the metal associated bond and angle parameters and a larger one to parametrize the partial charges. Besides supporting the parametrization of metal centers that only have amino acid side chains and ligands coordinating to the metal ion, MCPB.py further supports modeling for complexes, which have backbone, terminal oxygen or nitrogen atoms coordinated to the metal ions, or a system with a mixed side chain/backbone/terminal/ligand binding mode. To differentiate it from our earlier MCPB program, we use “small model” instead of “side chain model” because MCPB.py is more versatile in dealing with ligands coordinating a metal center.

MCPB.py builds the small model based on the schemes shown in Table 1. The non amino acid residues are kept during the modeling process. Generally, we use three approaches to build the small model: (1) MCPB.py uses the CH₃R (where R represents a side chain group) group to mimic residues that have side chain atoms bound to the metal ion; (2) a ACE (CO-CH₃) or NME (NH-CH₃) residue (together with a capped ACE or NME group, if necessary) to mimic the backbone when there is a backbone atom bound to the metal ion; or (3) the CH₂R-CO or NH-CH₂R group (with a capped ACE or NME group, if necessary) if both backbone and side chain atoms of one residue are bound to the central metal ion. These approaches will mimic the chemical environment at an affordable computational cost. Several examples are shown in Figure 2. Similar to MCPB, MCPB.py supports modeling for those metal sites with multiple metal centers as well.

For the large model, MCPB.py uses an approach adapted from MCPB, with the metal site amino acid residues capped by ACE and NME groups. If there are two residues that are both bound to the metal ion with less than 5 residues between them, the intermediate residues will be retained and simplified to GLY residues. There is a variable (*large_opt*) in MCPB.py to control whether a full geometry optimization, hydrogen-only optimization or no optimization is specified in the Gaussian input file of the large model. It is not suggested to carry out full geometry optimization for the large model due to the computational expense and the chance that the structure might undergo large structural changes in the absence of the protein environment constraints. In general, we recommend a hydrogen-only geometry optimization if some structural relaxation is necessary, but a full optimization can be done if

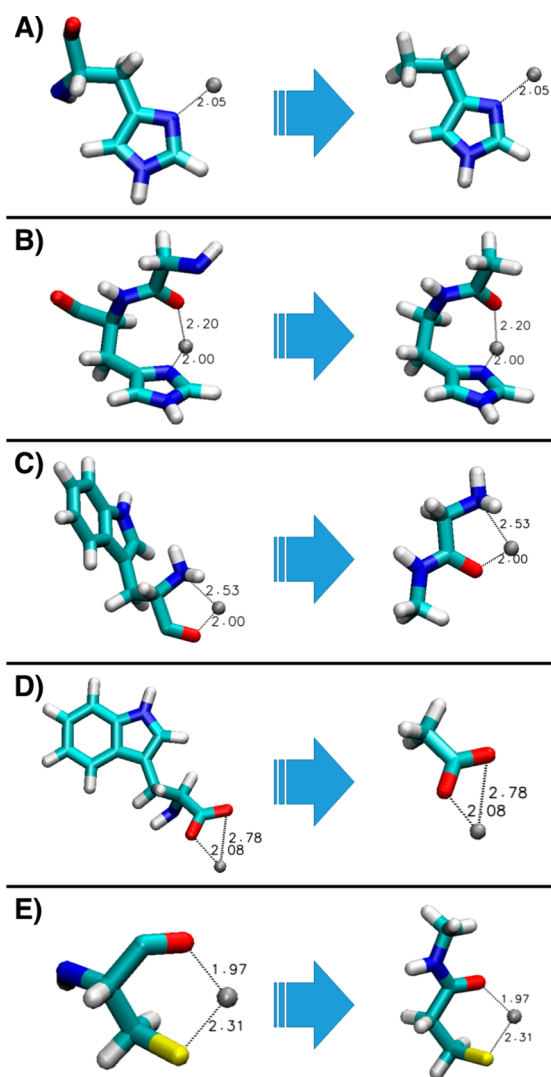


Figure 2. Examples of capping different metal bound amino acid residues in the small models. Structures on the left are the coordinated amino acids in the protein, and the structures on the right are capped residues in the small models built by MCPB.py. Panels A and B are for PDB entry 1E67, panels C, D, and E are for PDB entries 1AK0, 1Y79 and 2BZS, respectively. Zinc coordination is shown as a dotted line with the bond lengths given in Ångstroms.

the computational resources allow and if it is confirmed that the structure does not dramatically change upon optimization.

To conserve the total charge of the metal site, RESP charge fits³⁷ for the large model are performed with the capped ACE,

H																	He
Li	Be											B	C	N	O	F	Ne
Na	Mg											Al	Si	P	S	Cl	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba	La	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra	Ac															
		Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu		
		Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr		

Figure 3. Metal ions currently supported by the MCPB.py program. The metals with a blue background use the VDW parameters from Li et al.,^{6,39,40} whereas the metals with a green background use the VDW parameters adapted from UFF.⁴¹

NME and intermediate GLY residues in their neutral forms. Similar to antechamber,³⁸ there are two stages of charge fitting performed in MCPB.py: the first stage is carried out with a small restraint on the heavy atoms (0.0005) and no restraints on the hydrogen atoms. The second stage refits the charges on the CH₂ and CH₃ group (while the charges on the other groups are kept at the values from the first stage), including symmetrizing the charges on the hydrogen atoms bound to the same carbon atom, with a 0.001 restraint on the carbon atoms and no restraints on the hydrogen atoms.

Figure 3 shows the 81 metal ions presently supported by the MCPB.py program. Most of the VDW parameters (VDW radii for the RESP fits and Lennard-Jones parameters) of the related metal ions are from the recent work of Li et al.,^{6,39,40} whereas the remaining are adapted from UFF.⁴¹

■ USAGE

The MCPB.py code was designed with an optimized structure and offers a more user-friendly experience than the MCPB program. The command line input for MCPB.py is

```
MCPB.py -i input file -s/--step number
[--fchk Gaussian fchk file][--logf Gaussian/GAMESS-US log file]
```

(4)

Note: The --fchk and --logf options are not necessary, in the case where the fchk file or log file do not use the default name (users can consult the manual for further details).

There are a series of input variables available to meet different users' demands. In the examples (see SI), we kept the input files simple. For more advanced cases, users can consult the software manual for full details. The modeling process of MCPB.py involves 4 steps (as shown in the workflow of Figure 1), which are indicated as steps 1, 2, 3 and 4, respectively. The first step is preparing the input files for the small and large models (input files for the standard model will also be generated for the subsequent steps). The second step is generating the frcmod file for the system, which contains the bond, angle, torsion and VDW force field parameters. The bond and angle parameters can be generated based on the Seminario method (default, step 2s), Z-matrix method (using step 2z) and an empirical method (using step 2e, only supports zinc currently). As in MCPB,⁵ MCPB.py assigns zero torsion barriers for dihedral angles.

The third step involves the RESP charge fitting steps and building the mol2 files for each metal site residue based on the

fitted RESP charges. The program will rename these residues automatically to differentiate them from the amino acid residues in the standard AMBER force field library. There are several options (with steps labeled as 3a, 3b, 3c (default) and 3d) available for performing the RESP charge fits. They correspond to charge fitting approaches involving (1) all charges treated flexibly (ChgModA); (2) the charges of the backbone heavy atoms (CA, N, C and O) are fixed (ChgModB); (3) charges of all backbone atoms (N, H, C, O, CA, HA) are fixed (ChgModC); and (4) charges of all the backbone atoms and the CB atom are fixed (ChgModD). These fixed charges are assigned values from the AMBER force field employed. If a residue has a backbone atom bound to the metal ion, all atomic charges in it will be flexible no matter which algorithm is chosen. This is because metal binding may have a strong influence on the charge distribution of the atoms in the backbone.

The fourth step is generating the new PDB file with the renamed metal site residues and creating the leap input file to build the AMBER topology and coordinate files. If there are any missing parameters needed by leap, users can use the Carthess2FC.py program (available at <https://github.com/Amber-MD/pymomt>) to calculate all the bond and angle parameters of the small model and manually add the related parameters into the frcmod file. If there are any metal related dihedral parameters missing, users can give them a zero barrier in the frcmod file.

After completion of these steps, users can transfer the AMBER topology and coordinate files to other formats if they want to use alternate software to run the minimization and MD simulations. There are a number of packages available to do this conversion: for example, ParmEd²⁷ converts the AMBER format into CHARMM's format; ACPYPE³⁰ converts the AMBER format into that of GROMACS; finally, amber2lmp converts the AMBER format into LAMMPS's.

■ CONCLUSION

Metal ion modeling in FFs remains a challenging issue due to the range of coordination modes available to transition metal ions. MCPB.py is an efficient tool that facilitates the construction of reliable force field parameters for metal ions utilizing the bonded model. It supports two widely used QM software packages in Gaussian and GAMESS-US for the parametrization process. It has an optimized code structure and has more options than the original C++ based MCPB program.

Far fewer steps overall and fewer steps requiring user intervention are capable of affording a reliable metal ion force field. The ability of MCPB.py to handle FF design for different metal centers has been shown by two examples outlined in the SI. It has a GNU_GPL_v3 license and is free to download and distribute. It lowers the barrier for the molecular modeling of metal ion containing systems and offers a clean interface for nonexpert users interested in modeling metal ion containing systems.

■ ASSOCIATED CONTENT

■ Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.5b00674.

Full references for AMBER 2015 and Gaussian 03, details for Examples 1 and 2, Table S1, Figures S1–10, and associated references (PDF).

The MCPB.py input file, original PDB file, Gaussian com, log and fchk files, frmod files, mol2 files, and tleap input file for each of the two examples (ZIP).

■ AUTHOR INFORMATION

Corresponding Author

*K. M. Merz, Jr. E-mail: kmerz1@gmail.com.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank Dr. David Case (Rutgers), Dr. Jason Swails (Rutgers) and Dr. Hai Nguyen (Rutgers) for help with the coding. We thank Dr. Junmei Wang (UTexas) for providing the normal-mode analysis code. We thank users who have reported bugs to us. We acknowledge computational support from the high-performance computing centers at the University of Florida and Michigan State University. We also thank suggestions from the reviewers.

■ ABBREVIATIONS

MM, molecular mechanics; QM, quantum mechanics; FF, force field; MC, Monte Carlo; MD, molecular dynamics; RESP, restrained electrostatic potential; MCPB, metal center parameter builder; pyMSMT, python metal site modeling toolbox; GAFF, general AMBER force field; ZAFF, zinc AMBER force field

■ REFERENCES

- (1) Thomson, A. J.; Gray, H. B. Bio-inorganic chemistry. *Curr. Opin. Chem. Biol.* **1998**, *2*, 155–158.
- (2) Lifson, S.; Warshel, A. Consistent Force Field for Calculations of Conformations, Vibrational Spectra, and Enthalpies of Cycloalkane and n-Alkane Molecules. *J. Chem. Phys.* **1968**, *49*, 5116–5129.
- (3) Metropolis, N.; Ulam, S. The monte carlo method. *J. Am. Stat. Assoc.* **1949**, *44*, 335–341.
- (4) Alder, B. J.; Wainwright, T. Studies in molecular dynamics. I. General method. *J. Chem. Phys.* **1959**, *31*, 459–466.
- (5) Peters, M. B.; Yang, Y.; Wang, B.; Füsti-Molnár, L.; Weaver, M. N.; Merz, K. M., Jr Structural Survey of Zinc-Containing Proteins and Development of the Zinc AMBER Force Field (ZAFF). *J. Chem. Theory Comput.* **2010**, *6*, 2935–2947.
- (6) Li, P.; Roberts, B. P.; Chakravorty, D. K.; Merz, K. M., Jr Rational Design of Particle Mesh Ewald Compatible Lennard-Jones Parameters for + 2 Metal Cations in Explicit Solvent. *J. Chem. Theory Comput.* **2013**, *9*, 2733–2748.

(7) Li, P.; Merz, K. M., Jr Taking into Account the Ion-induced Dipole Interaction in the Nonbonded Model of Ions. *J. Chem. Theory Comput.* **2014**, *10*, 289–297.

(8) Aqvist, J.; Warshel, A. Free energy relationships in metal-loenzyme-catalyzed reactions. Calculations of the effects of metal ion substitutions in staphylococcal nuclease. *J. Am. Chem. Soc.* **1990**, *112*, 2860–2868.

(9) Grossfield, A.; Ren, P.; Ponder, J. W. Ion Solvation Thermodynamics from Simulation with a Polarizable Force Field. *J. Am. Chem. Soc.* **2003**, *125*, 15671–15682.

(10) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M., Jr; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc.* **1995**, *117*, 5179–5197.

(11) MacKerell, A. D.; Bashford, D.; Bellott, M.; Dunbrack, R.; Evanseck, J.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S. a. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J. Phys. Chem. B* **1998**, *102*, 3586–3616.

(12) van Gunsteren, W. F.; Daura, X.; Mark, A. E. GROMOS Force Field. In *Encyclopedia of Computational Chemistry*; John Wiley & Sons, Ltd: Hoboken, NJ, 2002.

(13) Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *J. Am. Chem. Soc.* **1996**, *118*, 11225–11236.

(14) Frenking, G.; Fröhlich, N. The Nature of the Bonding in Transition-Metal Compounds. *Chem. Rev.* **2000**, *100*, 717–774.

(15) Bock, C. W.; Katz, A. K.; Glusker, J. P. Hydration of zinc ions: A comparison with magnesium and beryllium ions. *J. Am. Chem. Soc.* **1995**, *117*, 3754–3765.

(16) Vedani, A.; Dobler, M.; Dunitz, J. D. An empirical potential function for metal centers: application to molecular mechanics calculations on metalloproteins. *J. Comput. Chem.* **1986**, *7*, 701–710.

(17) Nakagawa, I.; Shimanouchi, T. Infrared absorption spectra of aquo complexes and the nature of co-ordination bonds. *Spectrochim. Acta* **1964**, *20*, 429–439.

(18) Sacconi, L.; Sabatini, A.; Gans, P. Infrared Spectra from 80 to 2000 Cm⁻¹ of Some Metal-Ammine Complexes. *Inorg. Chem.* **1964**, *3*, 1772–1774.

(19) Pulay, P.; Fogarasi, G.; Pang, F.; Boggs, J. E. Systematic ab initio gradient calculation of molecular geometries, force constants, and dipole moment derivatives. *J. Am. Chem. Soc.* **1979**, *101*, 2550–2560.

(20) Seminario, J. M. Calculation of intramolecular force fields from second-derivative tensors. *Int. J. Quantum Chem.* **1996**, *60*, 1271–1277.

(21) Nilsson, K.; Lecerof, D.; Sigfridsson, E.; Ryde, U. An automatic method to generate force-field parameters for hetero-compounds. *Acta Crystallogr., Sect. D: Biol. Crystallogr.* **2003**, *59*, 274–289.

(22) Lin, F.; Wang, R. Systematic derivation of AMBER force field parameters applicable to zinc-containing systems. *J. Chem. Theory Comput.* **2010**, *6*, 1852–1870.

(23) Chakravorty, D. K.; Wang, B.; Lee, C. W.; Giedroc, D. P.; Merz, K. M., Jr Simulations of Allosteric Motions in the Zinc Sensor CzrA. *J. Am. Chem. Soc.* **2012**, *134*, 3367–3376.

(24) Chakravorty, D. K.; Wang, B.; Ucisik, M. N.; Merz, K. M., Jr Insight into the Cation- π Interaction at the Metal Binding Site of the Copper Metallochaperone CusF. *J. Am. Chem. Soc.* **2011**, *133*, 19330–19333.

(25) Roberts, B. P.; Miller, B. R., III; Roitberg, A. E.; Merz, K. M., Jr Wide-open flaps are key to urease activity. *J. Am. Chem. Soc.* **2012**, *134*, 9934–9937.

(26) Ucisik, M. N.; Chakravorty, D. K.; Merz, K. M., Jr Structure and Dynamics of the N-Terminal Domain of the Cu (I) Binding Protein CusB. *Biochemistry* **2013**, *52*, 6911–6923.

(27) Case, D. A. et al. AMBER 2015; University of California: San Francisco, 2015.

(28) Frisch, M. J. et al. Gaussian 03, Revision E.01; Gaussian Inc.: Wallingford, CT, 2004.

- (29) Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S. General atomic and molecular electronic structure system. *J. Comput. Chem.* **1993**, *14*, 1347–1363.
- (30) Sousa da Silva, A. W.; Vranken, W. F. ACPYPE - AnteChamber PYthon Parser interface. *BMC Res. Notes* **2012**, *5*, 1–8.
- (31) Brooks, B. R.; Brooks, C. L.; MacKerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S. CHARMM: the biomolecular simulation program. *J. Comput. Chem.* **2009**, *30*, 1545–1614.
- (32) van der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. GROMACS: fast, flexible, and free. *J. Comput. Chem.* **2005**, *26*, 1701–1718.
- (33) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M., Jr; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber biomolecular simulation programs. *J. Comput. Chem.* **2005**, *26*, 1668–1688.
- (34) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kale, L.; Schulten, K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781–1802.
- (35) Eastman, P.; Pande, V. OpenMM: a hardware-independent framework for molecular simulations. *Comput. Sci. Eng.* **2010**, *12*, 34–39.
- (36) Plimpton, S. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.* **1995**, *117*, 1–19.
- (37) Bayly, C. I.; Cieplak, P.; Cornell, W.; Kollman, P. A. A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges: the RESP model. *J. Phys. Chem.* **1993**, *97*, 10269–10280.
- (38) Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A. Automatic atom type and bond type perception in molecular mechanical calculations. *J. Mol. Graphics Modell.* **2006**, *25*, 247–260.
- (39) Li, P.; Song, L. F.; Merz, K. M., Jr Parameterization of Highly Charged Metal Ions Using the 12–6-4 LJ-Type Nonbonded Model in Explicit Water. *J. Phys. Chem. B* **2015**, *119*, 883–895.
- (40) Li, P.; Song, L. F.; Merz, K. M., Jr Systematic Parameterization of Monovalent Ions Employing the Nonbonded Model. *J. Chem. Theory Comput.* **2015**, *11*, 1645–1657.
- (41) Rappé, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A., III; Skiff, W. M. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *J. Am. Chem. Soc.* **1992**, *114*, 10024–10035.