Electronic Supplementary Information

A parallelized molecular collision cross section package with optimized accuracy and efficiency for trajectory method calculations

Christian Ieritano,[a] Jeff Crouse,[a] J. Larry Campbell,[a,b] W. Scott Hopkins*[a]

[a] Department of Chemistry, University of Waterloo, Waterloo, ON, N2L 3G1, Canada. E-mail: shopkins@uwaterloo.ca

[b] SCIEX, 71 Four Valley Drive, Concord, Ontario, L4K 4V8, Canada

Table of contents

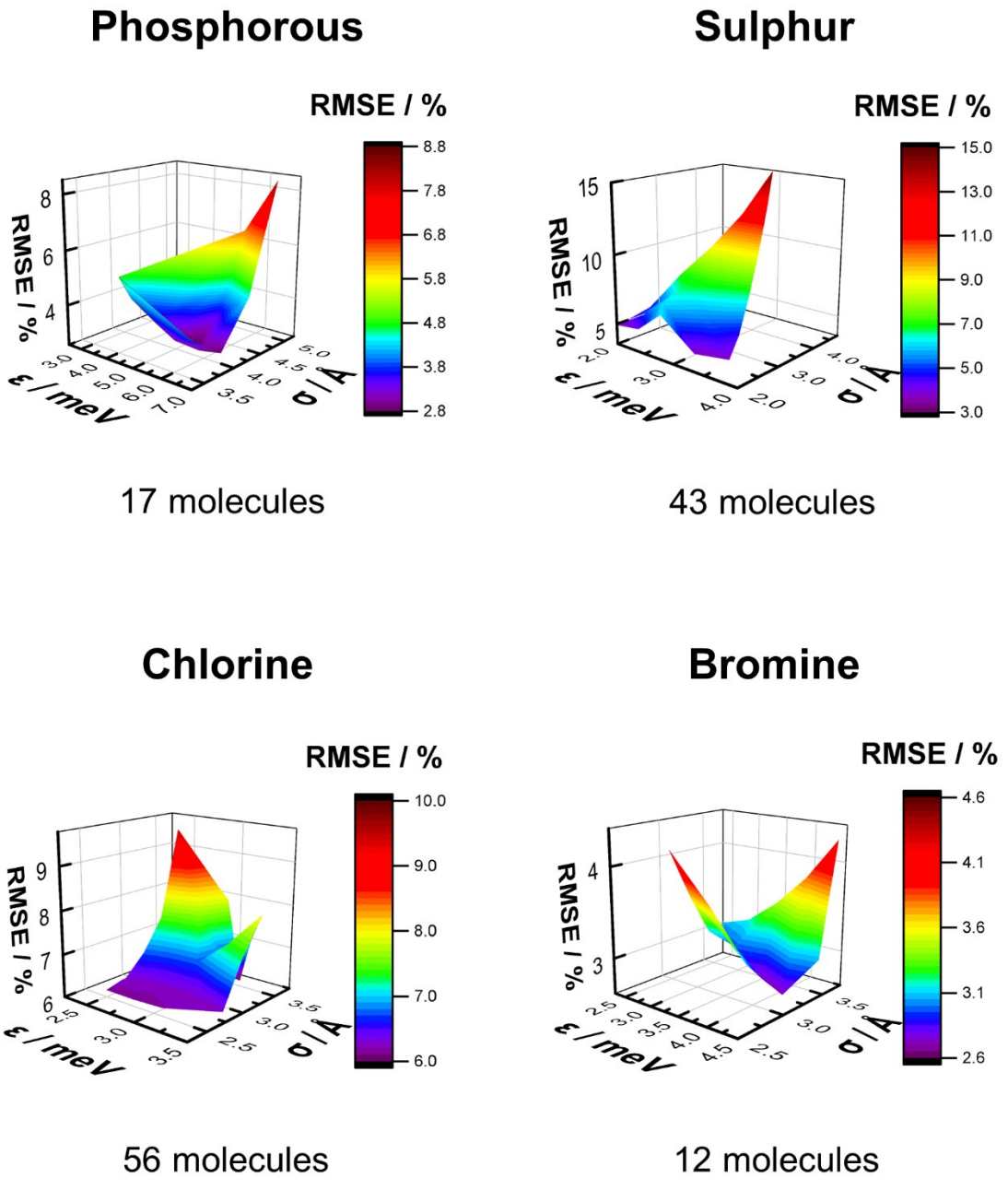
Generation of macromolecule .mfj input files

Conducting ion mobility measurements of macromolecular ions is an important application with relevance to metabolomics and proteomics. As a result, it would be pertinent to explore the applicability of MobCal-MPI towards CCS calculations of macromolecules with the now parallelized program. In this regard, we have calculated the CCS of several macromolecular ions. The structures of various angiotensins (2JP8, 1N9V, and 1N9U), bradykinin (6F3W), enolase-T35 (I-tasser),[1–3] bovine ubiquitin (2ZCC), equine cytochrome C (1HRC), and β-lactoglobulin (3NPO, 2Q2M) were obtained from the PDB, unless otherwise stated. Hydrogens were added explicitly to structures derived from X-ray crystallography using Chimera, and relevant acidic/basic residues to selectively protonated to match the charge state of the experimentally detected ion. As the MK charge scheme cannot be applied for such large molecules without significant computational cost, we instead turned to H++,[4-6] an automated tool catered towards proteins that computes the pKa values of ionizable groups and assigns both partial charges and missing hydrogens to the complex. Charges were manually inserted into a .key file, and converted to a .mfj input using the Python package accompanying this manuscript.

It should be noted that without a methodology to obtain gas-phase structural data correct to the atomic level for macromolecular ions, we are unable to conduct a comparison between theory and experiment. Instead, our aim is to demonstrate the feasibility of the now parallelized code and identify a methodology to be able to calculate collision cross sections once an appropriate model can be generated. Calculation times for macromolecules are reflected in Figure 2.

Optimization of LJ parameters for use in HPCCS

Using the molecules employed in this study, chemical species were selected and binned such that they only contain a single additional heteroatom (*i.e.* one of P, S, Cl or Br) in addition to any combination of C, H, N, O, and F. Using the LJ parameters previously derived by Wu and coworkers for C, H, N, O, and F,[7] ε and σ were empirically optimized to minimize the RMSE between calculated and experimental Ω_{N2} for the additional heteroatom. Optimizations surfaces for P, S, Cl, and Br are shown below. Derived LJ parameters are summarized in Table S5 and were used in the benchmark of HPCCS. Note that this methodology is independent from the empirical optimization used for the calibration of MobCal-MPI. Parameters in Table S5 are only intended for use in CCS packages evaluating the ion-N_2 interaction potential with contributions from V_{vdW} (12-6 LJ potential), V_{IID}, and V_{IQ} using non-specific atom types.



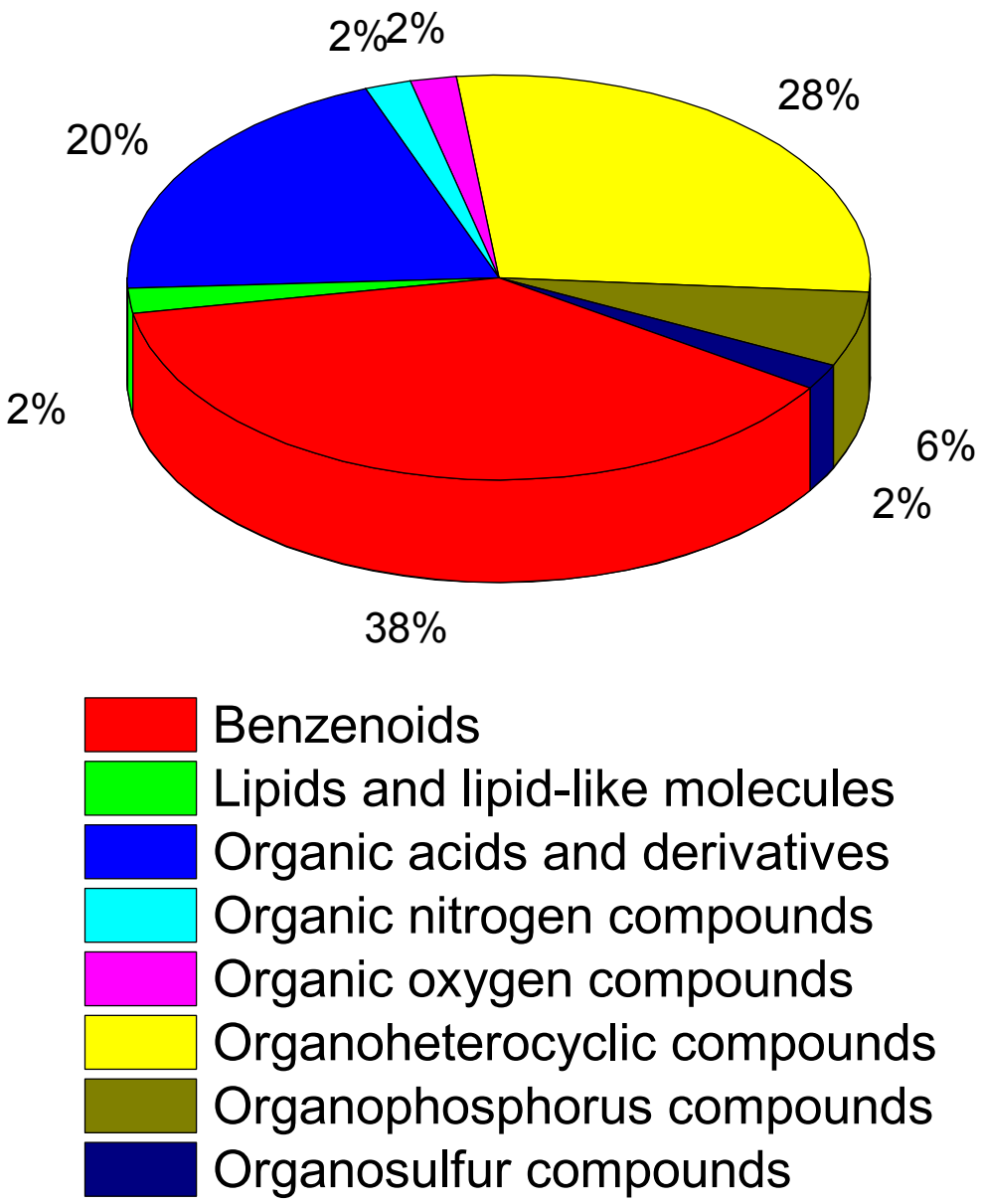


Figure S1. Superclasses of validation set as determined by ClassyFire.

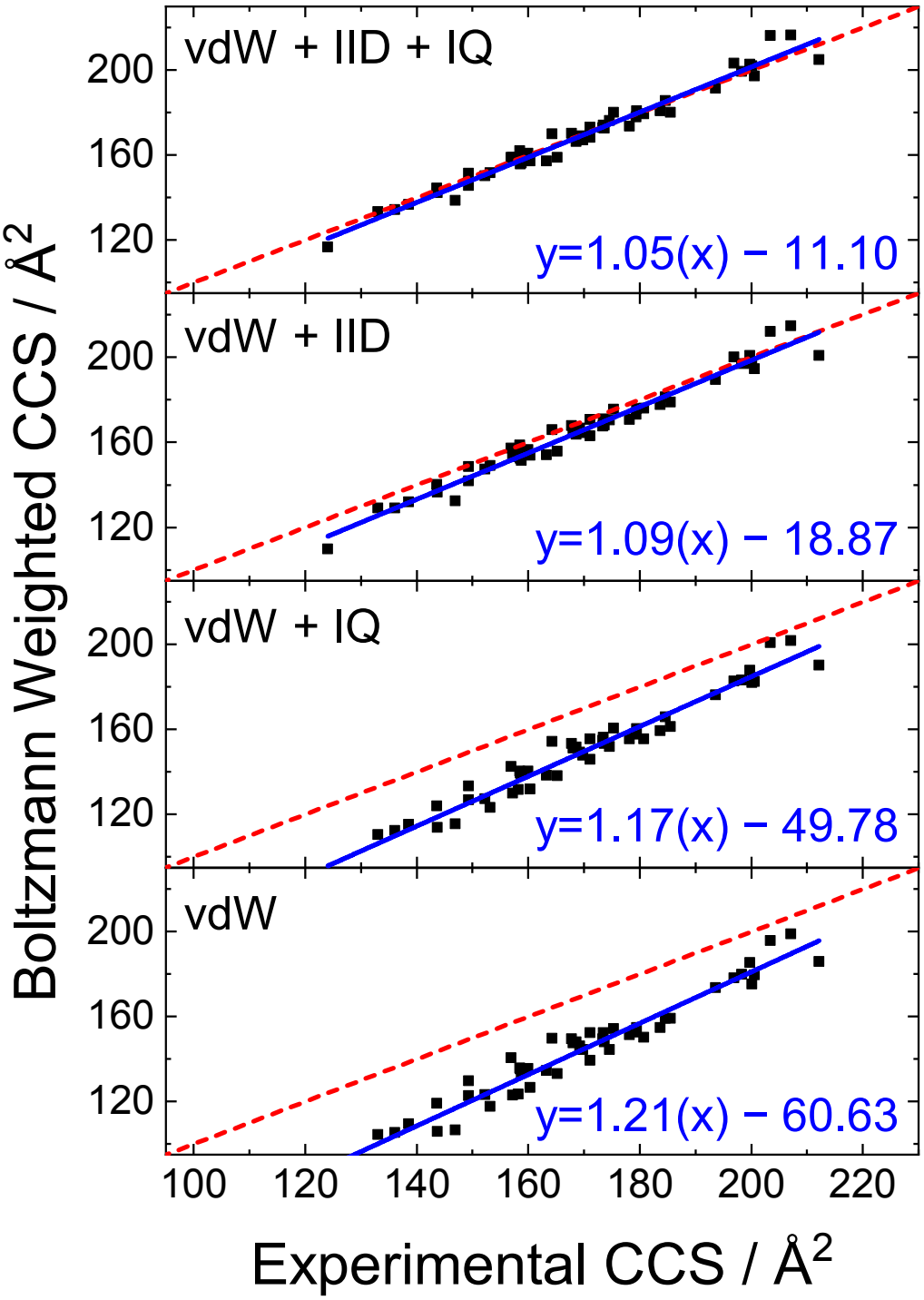


Figure S2. Linear regression and relative error analysis of validation set molecule CCSs using different terms in the ion-gas potential. All calculations were performed using the trajectory method for 10 cycles, taking 48 points of velocity integrations and 512 points of impact parameter integrations.

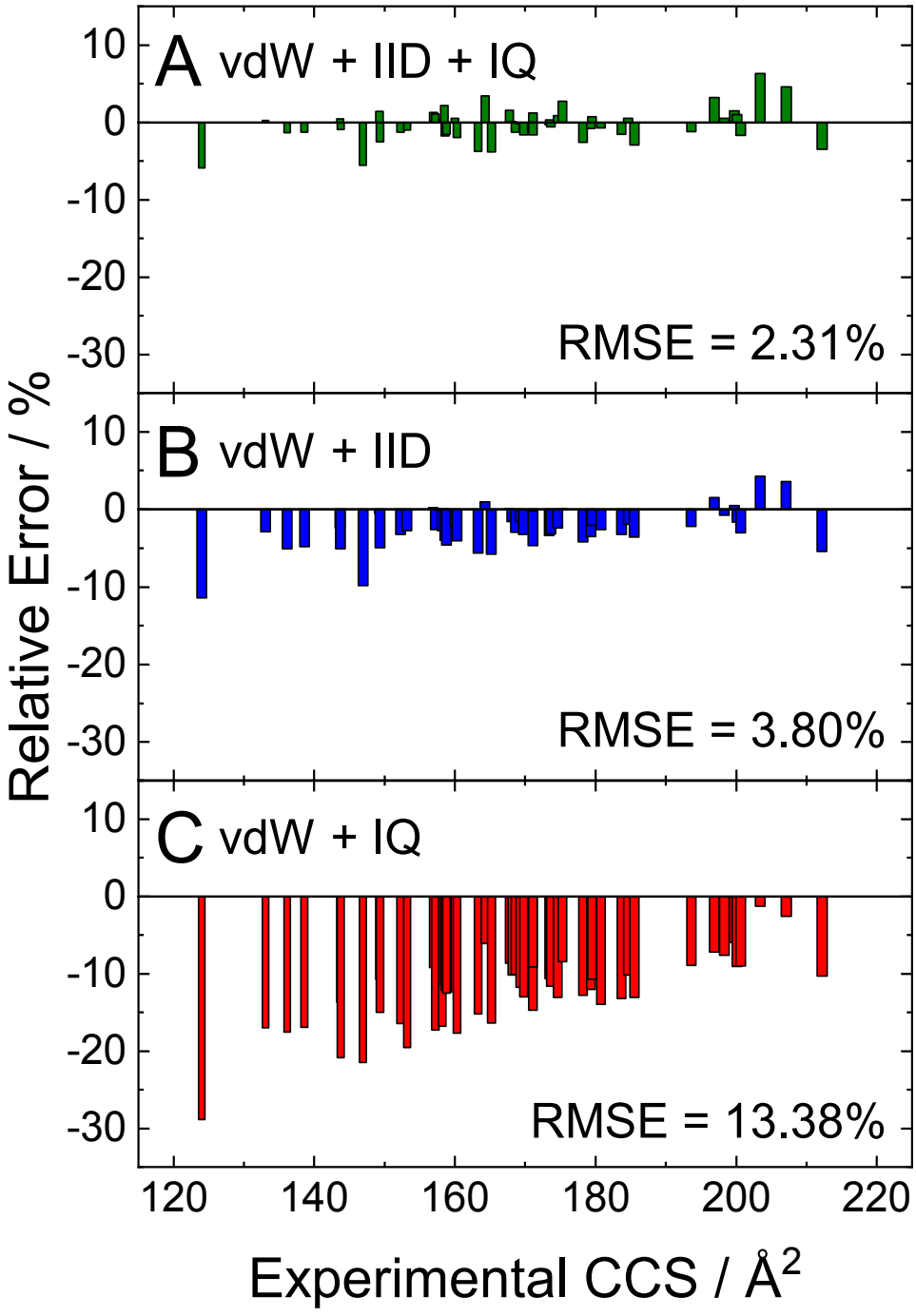


Figure S3. Relative error for CCS calculations with (**A**) V_{vdW}, V_{IID}, and V_{IQ} terms, (**B**) V_{vdW} and V_{IID}, and (**C**) V_{vdW} and V_{IQ} with optimized scaling parameters. All calculations were performed using the trajectory method for 10 cycles, taking 48 points of velocity integrations and 512 points of impact parameter integrations.

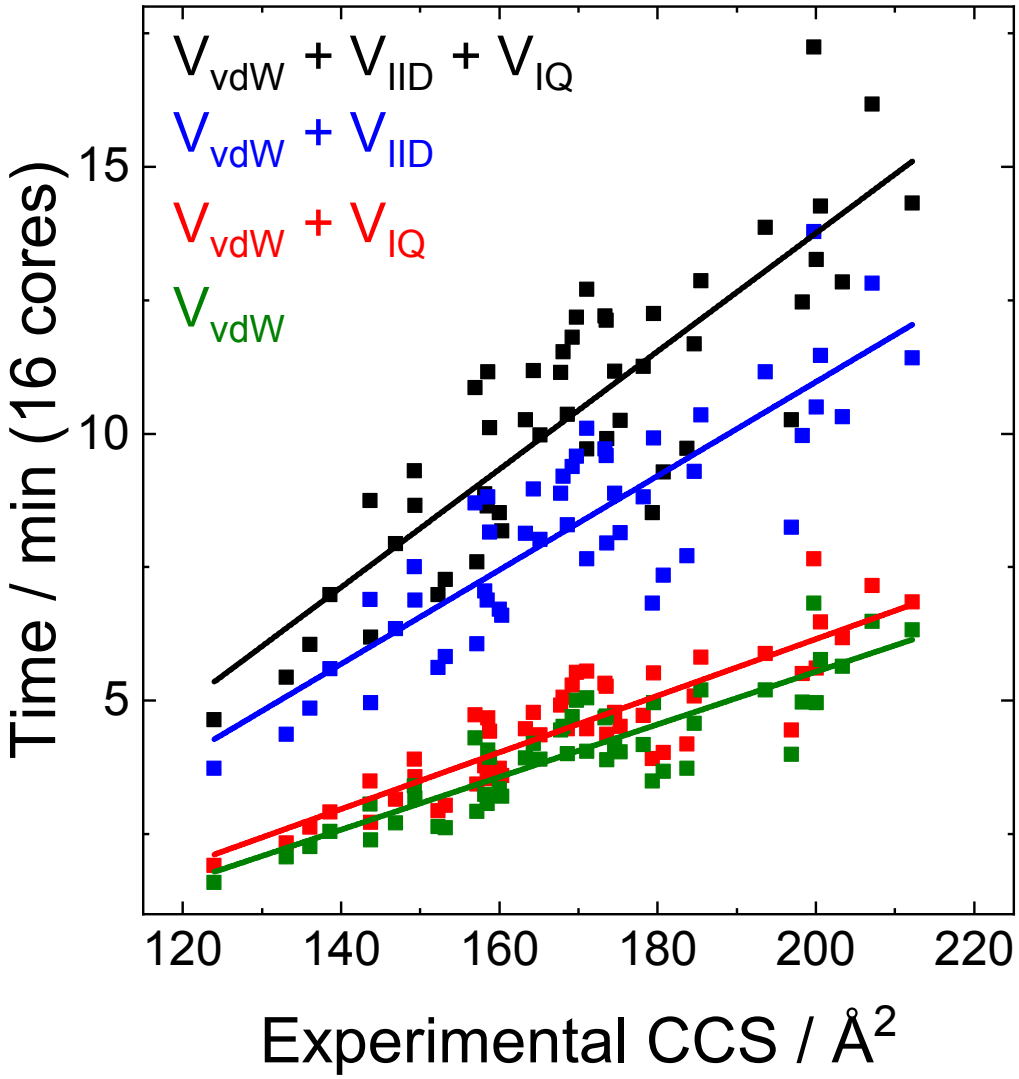


Figure S4. Times for trajectory method calculations using different ion-neutral interactions on 16 cores. All calculations were performed using the trajectory method for 10 cycles, taking 48 points of velocity integrations and 512 points of impact parameter integrations.

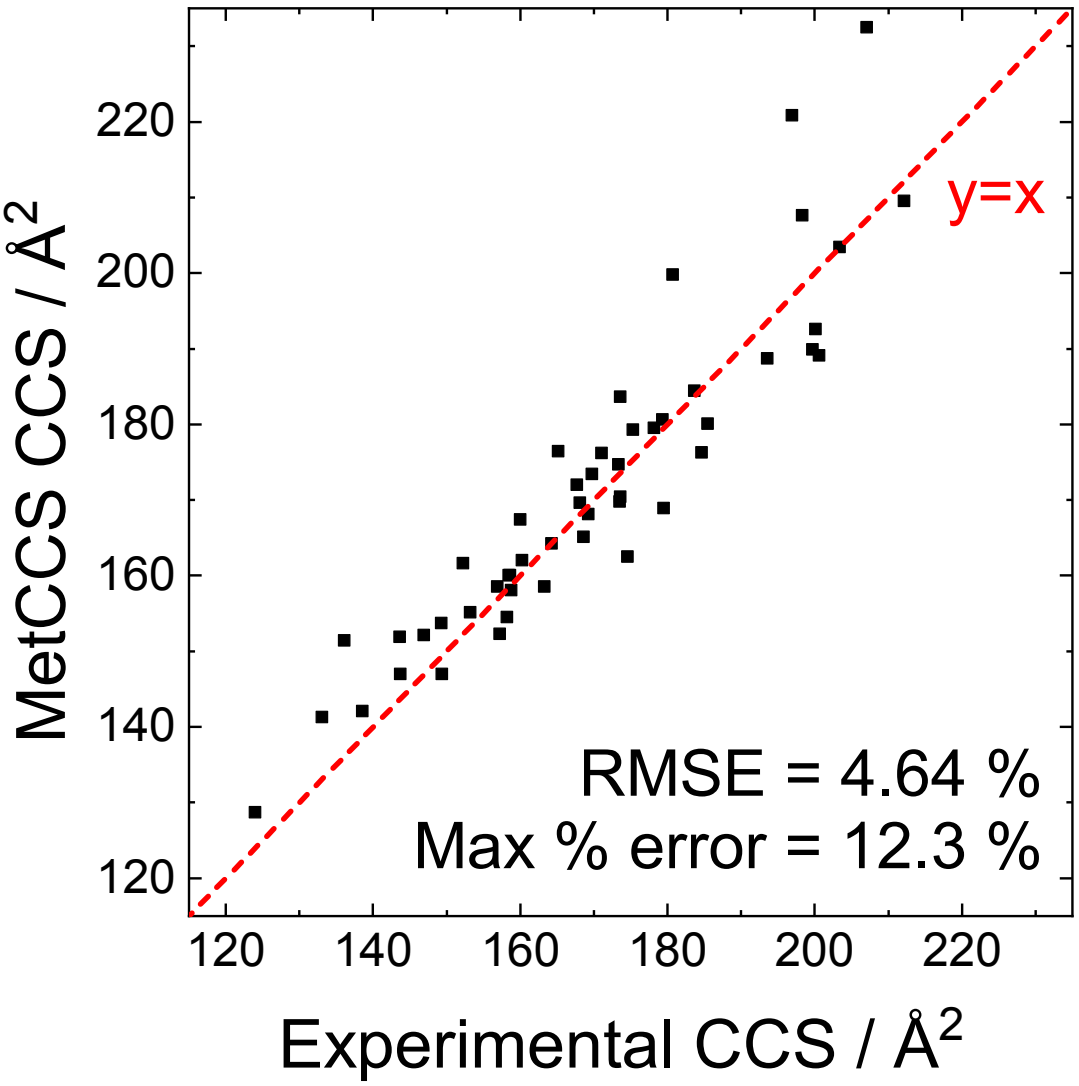


Figure S5. Comparison of MetCCS predicted CCS to experiment for validation set compounds.

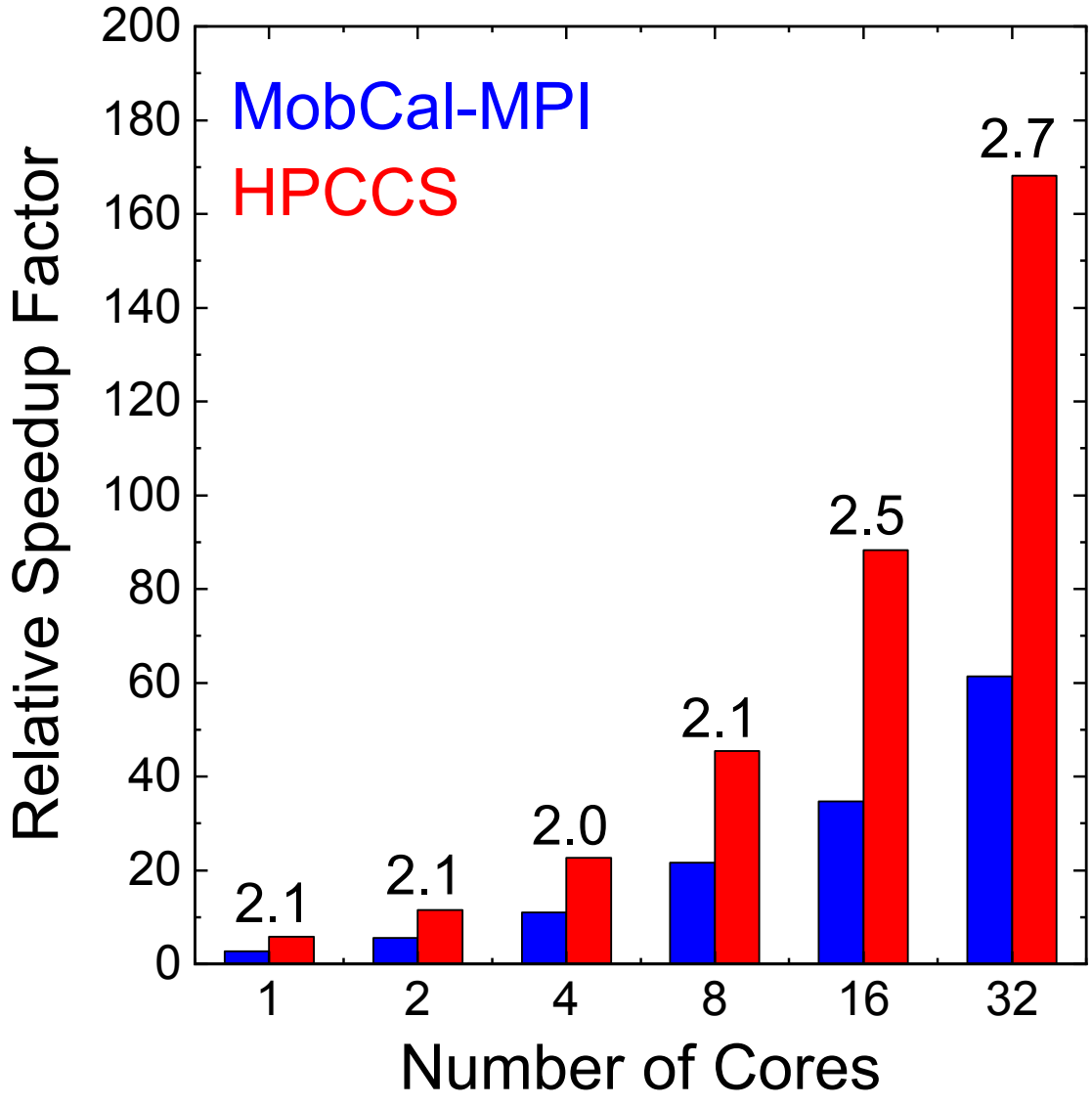


Figure S6. Comparison of the speedups relative to serial, unoptimized implement of MobCal using the Exp-6 potential outlined in Ref. 8 between HPCCS and MobCal-MPI using 1, 2, 4, 8, 16, and 32 cores. Mobcal-MPI is using the Exp-6 vdW potential, and HPCCS is using the 12-6 LJ potential. On average, HPCCS performs 2.3-fold more efficiently relative to MobCal-MPI. Individual speedup factors of HPCCS over MobCal-MPI are shown above the bars in black. All calculations were performed using the trajectory method for 10 cycles, taking 48 points of velocity integrations and 512 points of impact parameter integrations.

Table S1. Performance and relative speedups for MobCal-MPI. All calculations were performed using the trajectory method for 10 cycles, taking 48 points of velocity integrations and 512 points of impact parameter integrations.

Cores	Average Speedup factor	Efficiency
1	1.00	1.00
2	2.00	1.00
4	3.98	0.99
8	7.82	0.98
16	12.58	0.79
32	22.57	0.71

Table S2. Performance and relative speedups for MobCal-MPI relative to modified MobCal as outlined in Ref. 7. All calculations were performed using the trajectory method for 10 cycles, taking 48 points of velocity integrations and 512 points of impact parameter integrations.

Cores	Average Speedup factor
1 (MobCal implement from Ref. 8)	1.00
1	2.86
2	5.71
4	11.39
8	22.38
16	36.03
32	64.61

Table S3. Average speedups when V_{IID} or V_{IQ} terms are removed relative to inclusion of all potentials (ie. $V_{vdw} + V_{IID} + V_{IQ}$). All calculations were performed using the trajectory method for 10 cycles, taking 48 points of velocity integrations and 512 points of impact parameter integrations.

Potential	Average speedup factor
$V_{vdw} + V_{IID} + V_{IQ}$	1.00
$V_{vdW} + V_{IID}$	1.25
$V_{vdw} + V_{IQ}$	2.30
V_{vdw}	2.60

Table S4. Boltzmann-weighted CCSs calculated using different basis sets and charge schemes at the B3LYP level of theory.

Filename	ChelpG	ChelpG (Dipole)	MK	MK (Dipole)
6-31++G(d,p)				
Amifostine	138.79	138.79	138.87	138.75
Carboxin	145.02	145.08	145.58	145.58
Diazinon	173.28	173.24	173.11	174.03
6-311++G(d,p)				
Amifostine	138.33	138.33	138.36	138.36
Carboxin	144.83	144.83	145.46	145.46
Diazinon	173.54	173.54	173.26	173.26
6-311G(d,p)				
Amifostine	138.28	138.26	138.36	138.36
Carboxin	144.88	144.82	145.53	145.53
Diazinon	173.85	173.49	173.73	173.73
aug-cc-pVTZ[a]				
Amifostine	137.88	137.94	137.99	137.93
Carboxin	144.79	144.83	145.28	145.29
Diazinon	172.92	172.91	172.72	172.74

[a] Geometries, energies, and thermochemistry used from B3LYP/6-31++G(d,p)

Table S5. 12-6 LJ parameters for use in trajectory method CCS calculations

Atom	ε / meV	σ / Å
C[a]	3.574044	3.225487
H[a]	1.571967	1.898617
O[a]	2.701032	3.074995
N[a]	3.290241	3.571906
F[a]	2.465693	3.014650
P[b]	5.04013	4.08297
S[b]	2.25343	2.61403
Cl[b]	3.09101	3.24624
Br[b]	3.58203	3.33650

[a] Taken from ref. 9. Also utilized in original HPCCS implement.

[b] Derived by manually fitting LJ parameters to reproduce experimental CCS containing only C, H, O, N, F, and the atom to be parameterized using a similar methodology outlined in ref. 7.

Table S6. Performance and speedups of HPCCS relative to serial execution of itself. All calculations were performed using the trajectory method for 10 cycles, taking 48 points of velocity integrations and 512 points of impact parameter integrations.

Cores	Average Speedup factor	Efficiency
1	1.00	1.00
2	2.00	1.00
4	3.93	0.98
8	7.88	0.98
16	15.32	0.96
32	29.16	0.91

Table S7. Performance and speedups of HPCCS relative to modified MobCal as outlined in Ref. 8. All calculations were performed using the trajectory method for 10 cycles, taking 48 points of velocity integrations and 512 points of impact parameter integrations.

Cores	Average Speedup factor
1	5.99
2	12.01
4	23.53
8	47.18
16	91.73
32	174.58

References

1. Roy, A.; Kucukural, A.; Zhang, Y. I-TASSER: A Unified Platform for Automated Protein Structure and Function Prediction. *Nat. Protoc.* **2010**, *5* (4), 725–738.
2. Zhang, Y. I-TASSER Server for Protein 3D Structure Prediction. *BMC Bioinformatics* **2008**, *9*, 1–8.
3. Yang, J.; Yan, R.; Roy, A.; Xu, D.; J, P.; Zhang, Y. The I-TASSER Suite: Protein Structure and Function Prediction. *Nat Methods* **2015**, *12* (1), 7–8.
4. Anandakrishnan, R., Aguilar, B. & Onufriev, A. V. *Nucleic Acids Res.*, 2012, **40**, 537-541.
5. Myers, J., Grothaus, G., Narayanan, S. & Onufriev, A. *Proteins*, 2006, **63**, 928-938.
6. Gordon, J. C., Myers, J. B., Folta, T., Shoja, V., Heath, L. S. & Onufriev, A. *Nucleic Acids Res.*, 2005, **33**, 368-371.
7. T. Wu, J. Derrick, M. Nahin, X. Chen and C. Larriba-Andaluz, *J. Chem. Phys.*, 2018, **148**, 074102.
8. J. W. Lee, H. H. L. Lee, K. L. Davidson, M. F. Bush and H. I. Kim, *Analyst*, 2018, **143**, 1786–1796.
9. V. Shrivastav, M. Nahin, C. J. Hogan and C. Larriba-Andaluz, *J. Am. Soc. Mass Spectrom.*, 2017, **28**, 1540–1551.