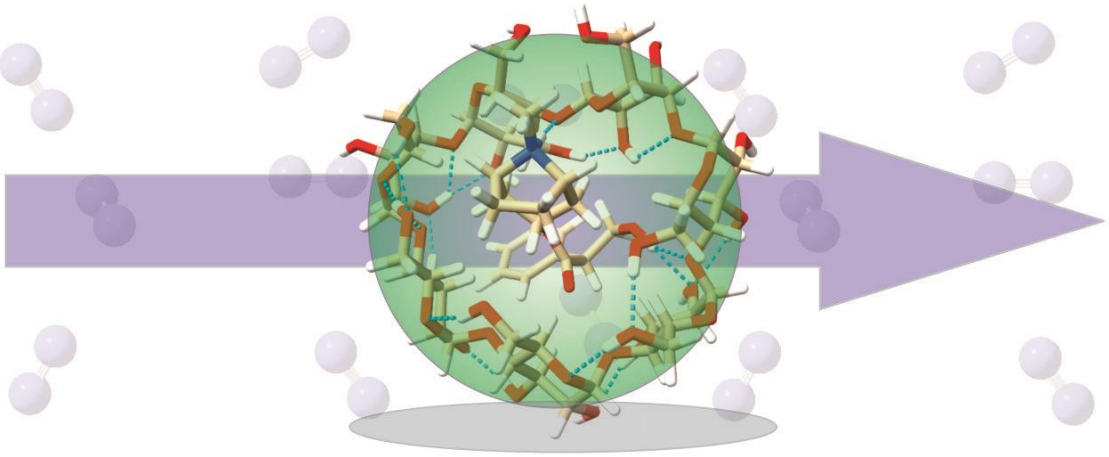
HOPKINS GROUP – UNIVERSITY OF WATERLOO

MobCal-MPI User Guide



Hopkins Lab

Revised March 10, 2021

Table of Contents

1. Introduction

Welcome and thank you for choosing MobCal-MPI to compliment your ion-mobility experiments.

This guide is meant to provide a brief overview of the MobCal-MPI methodology (trajectory method) used to calculate ion-neutral collision cross sections (CCSs) in addition to the data-processing scripts. The heart of the code was first reported by Shvartsburg, Jarrold, *et al.* in the following publications:

M. F. Mesleh, J. M. Hunter, A. A. Shvartsburg, G. C. Schatz, and M. F. Jarrold, Structural Information from Ion Mobility Measurements: Effects of the Long Range Potential, *J. Phys. Chem.* **1996**, 100, 16082-16086; Erratum, *J. Phys. Chem. A* **1997**, 101, 968.
A. A. Shvartsburg and M. F. Jarrold, An Exact Hard Spheres Scattering Model for the Mobilities of Polyatomic Ions, *Chem. Phys. Lett.* **1996**, 261, 86-91.

MobCal-MPI refines the work above to enable calculation of gas trajectories on a parallelized framework that makes use of atom-specific vdW parameters, as summarized in Section 2. It should be noted that Section 2 is not a literature review of CCS calculations, nor is it intended to be. We provide the reader with a refresher on the details of the trajectory method pertinent to MobCal-MPI. We recommend several recent (2019-2020) reviews to complement this outline for the interested reader, as well as to the papers cited therein.

Section 3 outlines the installation procedure for the MobCal-MPI Suite GUI, which is used for generation of MobCal-MPI input files (.mfj) and processing data from completed MobCal-MPI output files (.mout). Section 4 outlines the compilation procedure for the MobCal-MPI Fortran code and how to run CCS calculations using MobCal-MPI. Appendix A discusses the use of Gaussian, a commercial package for conducting quantum-chemical calculations, to generate input files for MobCal-MPI. Information for citation of the MobCal-MPI suite can be found in Appendix B. References cited throughout the guide can be found in Appendix C.

1.1 Terms of use

MobCal-MPI is provided free of use to academic users to validate ion-mobility experiments with computational methods. MobCal-MPI is freely available to the public for use in academic research. Any attempt to use MobCal-MPI for profit without the written consent of the authors is considered a breach of the usage terms and may be subject to litigation.

1.2 Change log

This release of MobCal-MPI (v 1.2) is equipped with a Python-based Graphical User Interface (GUI) for:

- *Conversion of Gaussian output files (.log) into MobCal-MPI input files (.mfj)*
- *Calculation of Boltzmann weights of isomer populations for the generation of Boltzmann-weighted CCSs*
- *Extraction of CCSs from MobCal-MPI output files (.mout)*
- Multiple temperatures can now be specified in a single .mfj input file
- The reporting of CCS now occurs at the end of each .mout file in a formatted table

2. Methods within MobCal-MPI

Details of the trajectory method, as implemented in MobCal-MPI, are described in this section. Each subsection is derived from excerpts from the original MobCal-MPI publication in Analyst.

Ieritano, C.; Crouse, J.; Campbell, J. L.; Hopkins, W. S. A Parallelized Molecular Collision Cross Section Package with Optimized Accuracy and Efficiency. *Analyst* **2019**, *144* (5), 1660–1670. https://doi.org/10.1039/c8an02150c.

Further reading recommendations for ion mobility calculations are provided in Section 2.5 for the interested reader.

2.1 Trajectory method

Ion mobility spectrometry (IMS) coupled to mass spectrometry (MS) is a powerful tool with applications in numerous fields of research. The success of the IMS approach relies on both rigorous experimental calibration and accurate theoretical modelling of ion collision cross sections (CCSs). With regard to modelling ion mobility, one is generally concerned with the rate of collision between an ion and a neutral buffer gas under specific electric field, pressure, and temperature conditions. Under low field conditions, ion mobility (K) can be described by the Mason-Schamp relation in the free molecular regime as shown in equation 1.[1]

$$K = \frac{\sqrt{18\pi}}{16}\sqrt{\frac{1}{m_{ion}}+\frac{1}{m_{gas}}}\frac{ze}{\sqrt{k_bT}}\frac{1}{\Omega_{avg}}\frac{1}{N} \qquad \text{Eq 1}$$

Where m_{ion} is the ion molecular mass, m_{gas} is the molecular mass of the buffer gas, z is the charge, e is the elementary charge, k_b is the Boltzmann constant, T is the temperature, and N is the number density of the gas. The orientationally averaged CCS of an ion is approximated by modelling collisions between the ion and buffer gas. This can be accomplished using one of several existing methods, which include the projection approximation,[2] elastic hard sphere scattering,[3] and the trajectory method.[3] The MobCal code,[2,3] originally produced by Shvartsburg and Jarrold, has been developed and refined over the past twenty years to conduct calculations using all three methods, although the trajectory method is generally accepted to be the most accurate.

Within the trajectory method, ion CCSs are evaluated through momentum transfer integrals, which are averaged over all possible velocities and geometries of the ion and buffer gas as per equation 2.[2,3]

$$\Omega_{avg} = \frac{1}{8\pi^2}\int_0^{2\pi} d\theta \int_0^{\pi} \sin\varphi \, d\varphi \int_0^{2\pi} \frac{\pi}{8}\left(\frac{\mu}{k_bT}\right)^3 d\gamma \int_0^{\infty} \mathrm{g}^5 \cdot \exp\left(-\frac{\mu g^2}{2k_bT}\right) dg \int_0^{\infty} 2b(1-\cos\chi(\theta,\varphi,\gamma,g,b))\, db \qquad \text{Eq 2}$$

Where θ, φ, and γ define the orientation of the ion with respect to the ion-collision gas centre of mass axis, g is the relative velocity, b is the impact parameter, μ is the reduced mass, and χ is the angle at which buffer gas is scattered upon interaction with the ion. Owing to the dependence of χ on molecular orientation and relative velocity, it can only be evaluated numerically as outlined in equation 3.[4]

$$\chi(\theta,\varphi,\gamma,g,b) = (\pi - 2b)\int_{r_{min}}^{\infty}\left[r^2\sqrt{1-\frac{b^2}{r^2}-\frac{\Phi(r)}{\frac{1}{2}m_{red}g^2}}\right]^{-1} dr \qquad \textit{Eq 3}$$

Where r_{min} is the distance of closest approach between the ion and buffer gas, which has position r. Gas trajectories are ultimately determined by the intermolecular potential $\Phi(r)$, which is composed of three contributions: van der Waals (V_{vdW}), ion-induced dipole (V_{IID}), and ion-quadrupole (V_{IQ}) interactions (see equation 4).

$$\Phi(r) = V_{vdW} + V_{IID} + V_{IQ} \qquad \textit{Eq 4}$$

The dominant term in modelling trajectories arises from the vdW interaction. While traditionally this takes the form of a 12-6 Lennard-Jones (LJ) potential, it was shown that the Exp-6 potential (see Equation 5) employed in the MM3 forcefield yields more accurate CCS predictions.[5,6]

$$V_{vdW}(r_i) = \sum_{i=1}^{n}\varepsilon_i\left[1.84\times10^5\exp\left(\frac{12r_i}{r_i^*}\right) - 2.25\left(\frac{r_i^*}{r_i}\right)^6\right] \qquad \textit{Eq 5}$$

Here, r_i^* is the equilibrium distance between the buffer gas and interaction partner and ε_i is the depth of the potential well. Typically, vdW parameters used in trajectory method calculations are empirically optimized to best reproduce experimentally obtained CCSs in a specific buffer gas; the optimized parameters are available only for the common elements (H, C, N, O, and F) from a handful of sources.[7–9] Moreover, atom types for a particular element are treated equivalently (*e.g.,* sp^3 *versus* sp^2 hybridized carbons) in traditional implementations. Further still, these parameter sets have limited support for heteroatoms (*i.e.*, those other than H, C, N, O, F); this is often circumvented by utilizing the same parameters for multiple atom types (*e.g.*, assignment of sulphur and phosphorous vdW parameters to those of parameterized atoms such as silicon), or by using vdW parameters from the UFF forcefield.[10–13] This lack of a generalized set of vdW parameters was addressed in 2017 by Lee and coworkers, who incorporated vdW parameters from molecular mechanics forcefields into CCS calculations.[5,14] Errors between experimental and calculated CCSs were ultimately minimized with the use of the Merck Molecular Force Field (MMFF94)[15] vdW parameters in the Exp-6 potential.

Inclusion of additional potentials beyond the vdW potential enables a more accurate description of molecular interactions. These additional terms become increasingly prominent components of the ion-neutral potential as the polarizability of the buffer gas increases. The ion-induced dipole interaction is described by Equation 6.[16,17]

$$V_{IID}(r_i) = -\frac{\alpha}{2}\left(\frac{ze}{n}\right)^2\left[\left(\sum_{i=1}^{n}\frac{x_i}{r_i^3}\right)^2 + \left(\sum_{i=1}^{n}\frac{y_i}{r_i^3}\right)^2 + \left(\sum_{i=1}^{n}\frac{z_i}{r_i^3}\right)^2\right] \quad \text{Eq 6}$$

The terms x_i, y_i, z_i, and r_i define the distance between each of the n atoms and centre of mass of the buffer gas atom/molecule, which has a polarizability α. When the buffer gas possesses a quadrupole moment (e.g., N_2), inclusion of an ion-quadrupole potential is also necessary for accurate calculation of gas trajectories. The quadrupole moment of molecular nitrogen ($4.65 \pm 0.08 \times 10^{-40}$ C·cm^2)[18] is effectively reproduced through separation of charges by negative z on each nitrogen ($-0.4825e$), which is balanced by a point charge of $2z$ ($+0.965e$) at the centre of mass of N_2.[8,16,17] This facilitates a relatively simple calculation of the ion-quadrupole potential by Equation 7, where the j index denotes each partial charge of molecular nitrogen, and i indicates atomic partial charges of the ion.

$$V_{IQ}(r_{ij}) = \sum_{j=1}^{3}\sum_{i=1}^{n}\frac{z_i z_j e^2}{r_{ij}} \quad \text{Eq 7}$$

2.2 Optimization of atom-specific vdW parameters in MobCal-MPI

MMFF94 vdW parameters were obtained from the original publication.[15] In this forcefield, atom types are defined by four unique parameters (α_i, A_i, N_i, and G_i) related to physicochemical properties of the atom. Combination of these parameters with those of molecular nitrogen yields the values of r_{ij}^* and ε_{ij} used in this study. Since the interaction partner in this work is always N_2, we simply refer to r_i^* and ε_i. Although it is possible to fit individual parameters, the most straight-forward procedure (owing to the rather complex combination rules associated with the MMFF94 forcefield) was to fit final r_i^* and ε_i values. Since the atom types outlined in the forcefield are inherently parameterized, optimization of vdW parameters can be easily accomplished through the application of a uniform scaling factor to the distance (r_i^*) and energy (ε_i) related terms as per equations 8 and 9.

$$r_{i\,scaled}^* = r_i^* \times \rho_{dist} \quad \text{Eq 8}$$

$$\varepsilon_{i\,scaled} = \varepsilon_i \times \rho_{ener} \quad \text{Eq 9}$$

For i = C, H, O, N, F, P, S, Cl, Br, I, …

Optimization of scaling factors was completed iteratively, screening combinations of scaling factors between 0.70 and 1.00. For the final iteration, scaling factors were stepped in increments of 0.01 to ensure optimal parameterization of vdW interaction potentials. It should be noted that this procedure returned numerous scaling factors that produced similar root mean square errors (RMSEs), indicating that a comprehensive search of the scaling factor surface had to be completed.

To assess the CCS calculation accuracy, accurate structural models must be accompanied by reproducible CCS measurements. To accomplish this, calibrant ions with highly reproducible experimental CCS measurements were selected. The conformational space of these species was then mapped using the BH algorithm at the molecular mechanics level and refined using DFT. Note that all CCS values were obtained using N_2 as a collision gas. With refined structural models, multiple low-energy isomers for a given calibrant ion were submitted for CCS calculation with the new parallelized code. Figure 1 shows a contour plot of the RMSEs obtained through a comparison of experimental CCS with the Boltzmann-weighted CCS of each calibrant ion at specific values of the linear scaling parameters ρ_{dist} and ρ_{ener}. While numerous combinations of scaling factors return RMSEs of <3 %, the combination of ρ_{dist} = 0.78 and ρ_{ener} = 0.80 returns a minimum RMSE of 2.60 %. Note that the parallelization, benchmarking, and optimization of the vdW parameters that we report here have been conducted for CCS calculations employing the trajectory method, which is generally accepted as the 'gold standard' for computed CCS values.[8,19–23] Further to this, the option of utilizing helium as the collision partner is available in MobCal-MPI, but we have not completed an extensive benchmarking or refitting of scaling parameters beyond the previous work of Lee et al.[5]

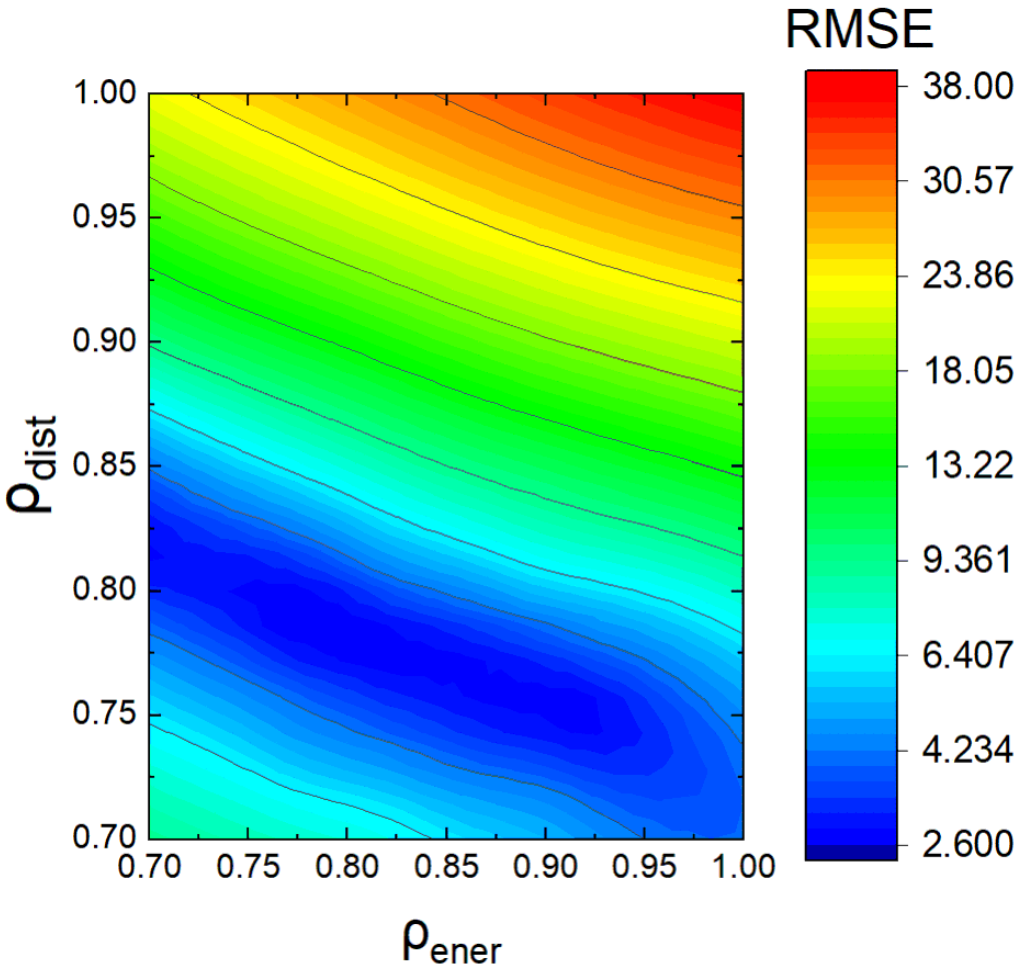


Figure 1. Contour plot indicating RMSE of the calibration set at specific combinations of the scaling parameters ρ_{dist} and ρ_{ener}.

2.3 Assessing the accuracy of MobCal-MPI

2.3.1 – Accuracy of the trajectory method

Empirical optimization methods are often accompanied by systematic bias towards the calibration set, even if the calibration set encompasses a structurally diverse set of compounds. Thus, it is necessary to test our scaled parameters on an equally diverse chemical set that does not contain compounds found in the calibration set. To this end, we constructed a validation set consisting of 50 compounds from the same compendiums, which fall within 8 molecular superclasses and 18 molecular classes based on ClassyFire assignment (Figure S1). Figures 2A and 2B show the correlations between calculated and measured CCS for both the calibration and validation sets; CCS calculations for both sets return excellent accuracy (2.60 % and 2.31 % RMSE, respectively) and no outliers can be readily identified.

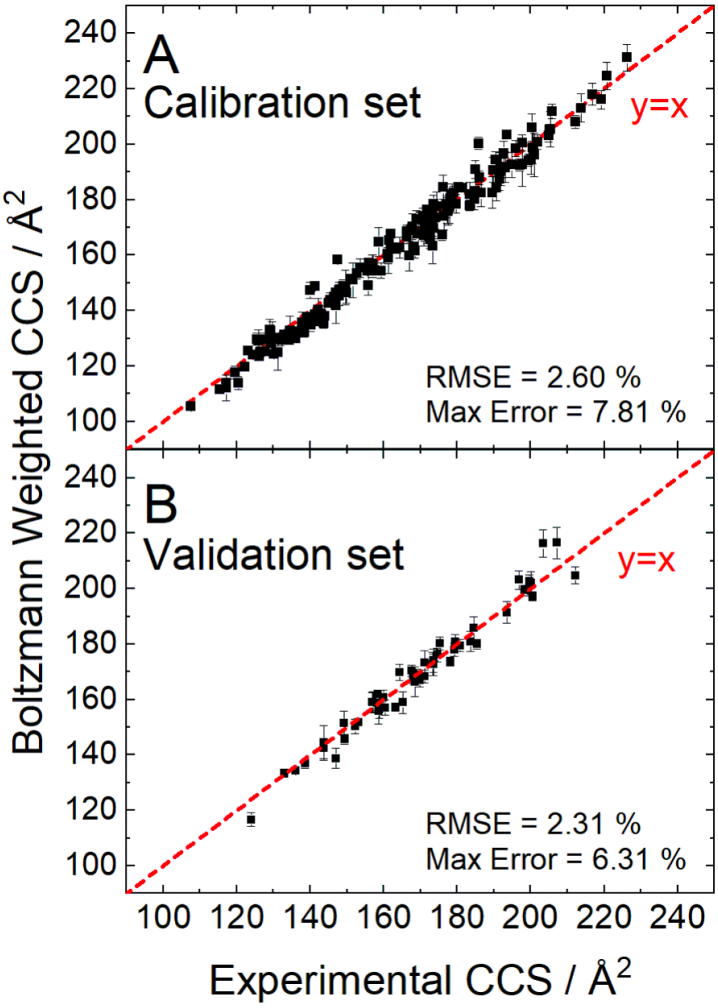


Figure 2. Comparison of the calculated Boltzmann-weighted CCSs using MobCal-MPI from the calibration set (**A**) and validation set (**B**) to experiment.

An additional factor regarding the accuracy of the CCS calculations stems from the trajectory method itself. To reiterate, the CCS depends on the orientation of the molecule relative to the buffer gas, the relative impact velocity of the buffer gas, and the scattering angle. Each of these are evaluated through repeated simulations of the collision trajectories between the ion and gas molecules, where the user can specify a definitive number of velocity and impact parameter (scattering angle) integrations to be performed. The number of velocity (*inp*) and impact parameter (*imp*) integrations could potentially be optimized, but it is useful to constrain the number of integration points such that they can be efficiently distributed across parallel architecture. For example, the recommended number of points for velocity integration is 40, but this does not scale well with traditional number of cores commonly employed (*i.e.*, 8 or 16 cores). For that reason, we recommend that 48 points of velocity integration are used for even distribution across cores.

2.3.2 – Influence of trajectory method parameters on calculated CCS accuracy

To investigate the effect of changing the frequency of impact parameter integration, we calculated the CCS of the validation set using 112, 256, 512, 752, and 1008 *imp* integration points. The results of this investigation are plotted in Figure 3, which shows that CCS values are relatively unaffected by the choice of *imp* integration frequency (RMSE varies between 2.20 % and 2.36 %). However, significant variance in a single CCS calculation is observed when *imp* is set below 500 points. This is somewhat concerning for prior parameterization work, which employed 25 – 100 *imp* integration points.[5,14] When an *imp* of 112 is specified, a mean standard deviation of 3.87 $Å^2$ and a maximum standard deviation of 8.22 $Å^2$ was found for our chosen set of compounds. In contrast, when 1008 *imp* integration points were used, both the mean and maximum standard deviations decreased to 1.20 $Å^2$ and 2.36 $Å^2$, respectively. There is, of course, a trade-off in terms of computational time when increasing the size of the integration grid; using 1008 *imp* points takes 10 times longer than using 112 points. Our testing suggests that 512 *imp* integration points offers a significantly reduced standard deviation in the determination of an ion's CCS with only a moderate increase in computational cost compared to smaller grid sizes. Of course, this number must increase as ion size increases beyond those in the calibration/validation set to fully explore the surface of the ion's collision landscape.

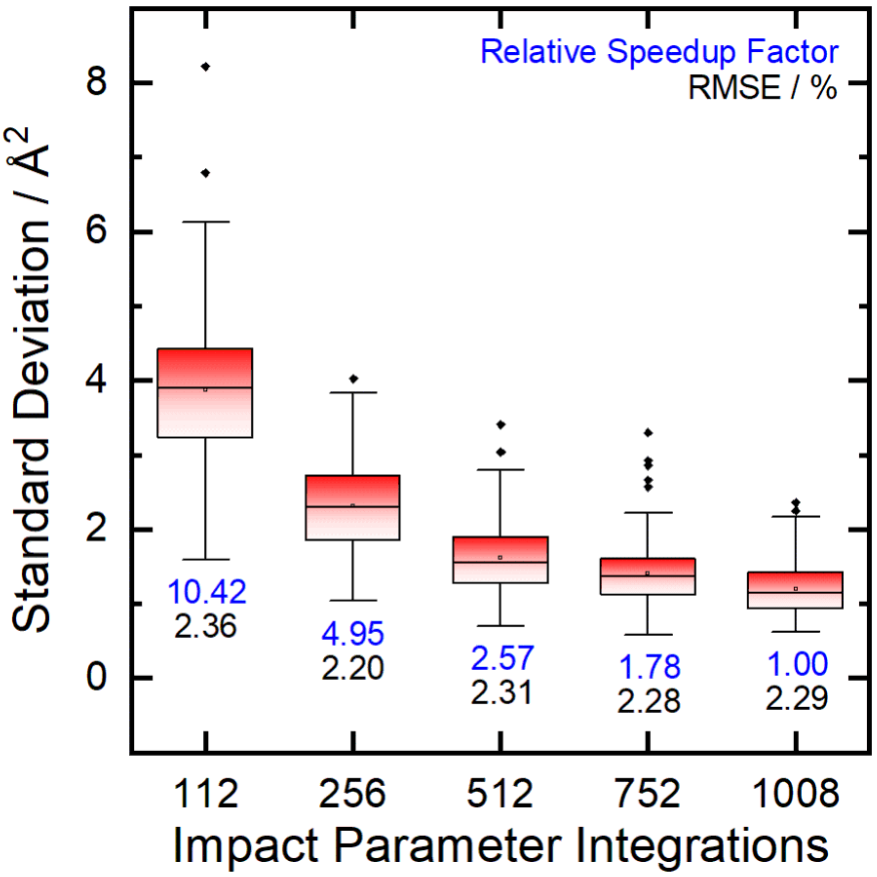


Figure 3. Box plot indicating the effect of the number of impact parameter integration points on the relative standard deviation of an individual CCS calculation for all isomers within the validation set. Speedups in calculation time are reported relative to the times for completion using 1008 *imp* points in blue, and RMSE (accuracy) for each *imp* routine is shown in black.

2.3.3 – Influence of charge-neutral interactions in CCS calculations

Since ion-induced dipole (V_{IID}) and ion-quadrupole (V_{IQ}) interactions are significant contributors to the total ion-N_2 interaction potential, a rigorous partial charge assignment scheme must be employed. Numerous methodologies for the assignment of atomic partial charges exist; however those that determine partial charges by fitting point charges to reproduce the electrostatic potential of the molecule are generally accepted to be the most accurate (e.g., ChelpG, MK).[24–26] Calculated CCSs can vary dramatically when alternative atomic charge assignment schemes are used (e.g., those based on MM forcefield assignments or Mulliken charges). In principle, one could avoid this issue and decrease computational time by ignoring V_{IID} and V_{IQ} terms. However, caution must be exercised in this regard because functional groups centered around the same atom type can have similar vdW interaction descriptions but may possess dissimilar V_{IID} and V_{IQ} terms. For instance, the nitrogen atom in an amide exhibits very different electronic character (*viz.* partial charge) than does a sulfonamide nitrogen atom due to donation/resonance effects with carbonyl *versus* sulfonyl groups. However, the nitrogen

atoms in the amide and sulfonamide groups have nearly identical vdW parameters and means that differentiation can only occur through atomic charge. Owing to the dependence of atomic charge on V_{IID} and V_{IQ} used to evaluate Ω_{N2}, TM calculations should employ partial charge assignments based on electrostatic potential mapping as opposed to the less reliable Mulliken population analysis or assignment of equivalent partial charges to all atoms in a particular ion (equal). Figure 4 shows impact of partial charge assignment on the calculated CCSs of the validation set with unique partial charges.

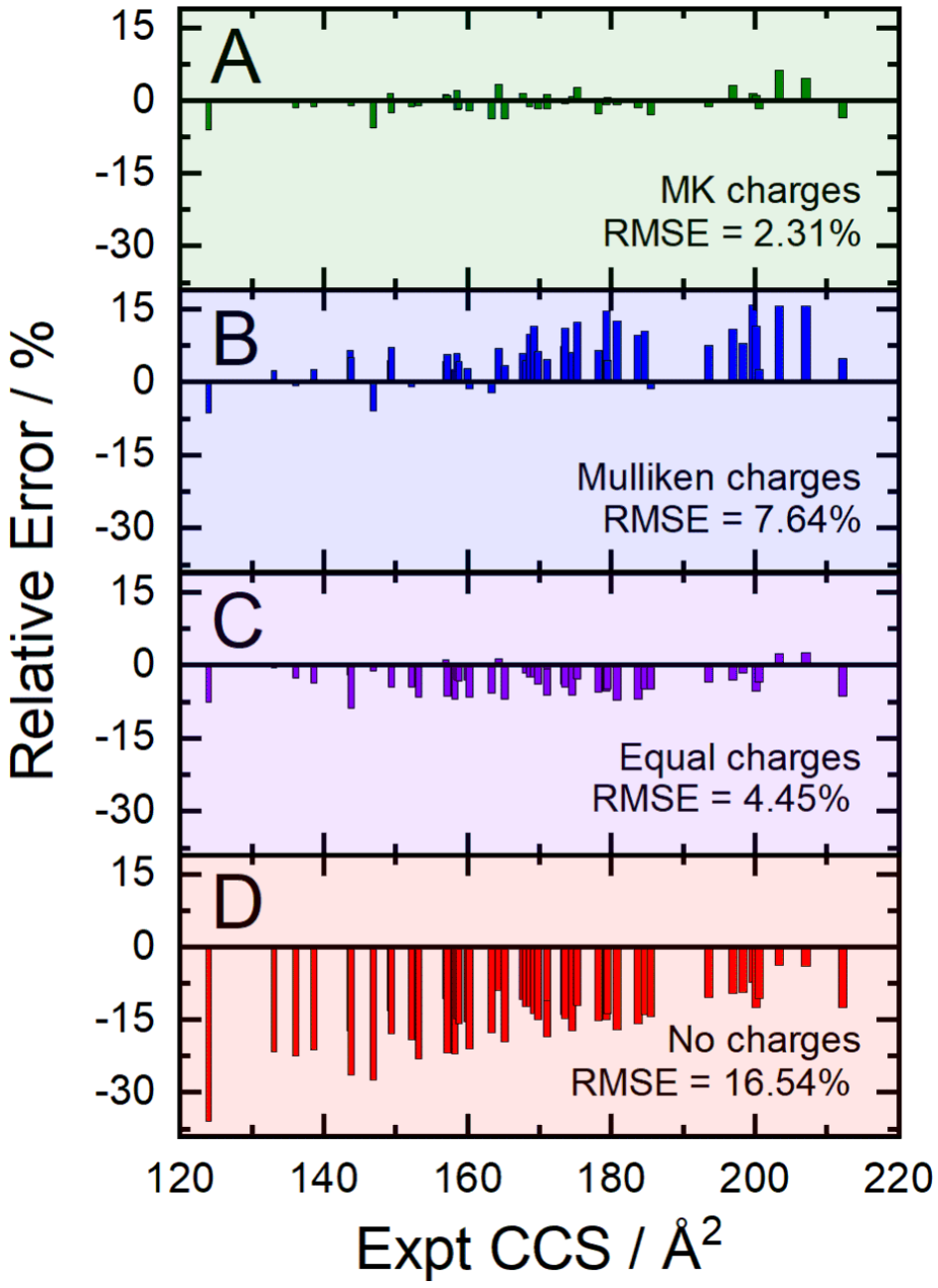


Figure 4. Relative error (%) in CCS predictions associated with assignment of partial charges by various methodologies: (**A**) MK charges constrained to reproduce dipole moment (green), (**B**) Mulliken charges (blue), (**C**) equal partial charges (purple), and (**D**) no partial charges (red).

Analysis of CCSs computed with the newly optimized scaling factors for the vdW potentials indicates that optimal accuracy is achieved when partial charges are assigned based on electrostatic potential mapping (i.e., CHelpG, MK), with the stipulation that partial charges be constrained to reproduce the molecular dipole moment. For example, the RMSE of CCSs determined using Mulliken and equal charge assignment schemes are 7.63 % and 4.45 %, respectively. Note that this effect is likely to be more important for calculations of Ω_{N2} than Ω_{He} owing to the relatively large polarizability of molecular nitrogen. We recommend that partial charge calculations should be computed at the B3LYP/6-31++G(d,p) level of theory to maintain consistency with the calibration set. However, a change in basis set should not have a major effect unless the relative energies of local minima are significantly changed, since this would influence Boltzmann-weighted CCSs.

It has been previously suggested that inclusion of the V_{IQ} term in the ion-N_2 potential could be circumvented by fitting appropriate scaling factors to the V_{vdW} potential.[9] While exclusion of V_{IQ} would be beneficial in terms of reducing computational cost, its exclusion will also reduce the accuracy of the calculated CCS values. To quantify the importance of the various terms in the interaction potential, we explored the effect of removal of the V_{IID} and V_{IQ} terms on computational time, calculation accuracy, and trends with experimental values. Removal of both V_{IID} and V_{IQ} returned an average speedup of 2.6-fold, while removal of V_{IQ} returned a speedup of 1.2-fold (Table S3). In terms of accuracy, the red bars in Figure 6D show that removal of both V_{IID} and V_{IQ} terms returns a RMSE of 16.54 %. Figures S2 and S3 shows that individual removal of V_{IID} and V_{IQ} terms yield RMSEs of 13.38 % and 3.80 %, respectively. Based on the data, it is unlikely that removal of either potential can be compensated by modifications to the vdW potential for small ions. Moreover, the slight gain in computational efficiency results in significantly larger variance of CCS measurements, in addition to increased deviation from experiment. While there is some merit to replacement of the V_{IQ} potential with appropriate vdW parameters, linear regression indicates a significant non-unity relationship between CCSs predicted by MobCal-MPI and experiment (Figure S2). This correlates with the observation that the measurable difference in experimental Ω_{N2} of diastereomeric species,[8] tertiary/quaternary ammonium species,[16,17] and proteins[27] depends on accurate modelling of short- (V_{vdW}) and long-range (V_{IID} and V_{IQ}) interactions using appropriate potentials. If such a method were desired, individual empirical optimization of each atom type in the vdW potential would be required, which is not feasible owing to the 100 unique vdW parameters defined in the MMFF94 forcefield.

The exclusion of the V_{IID} and V_{IQ} terms creates a dependence of error on ion size; deviations from experimental CCSs diminish as ion size increases. In other words, V_{IID} and V_{IQ} are particularly important for accurate calculation of small molecule CCSs. This result can be rationalized in terms of surface exposure and has implications in the development of a method for macromolecular CCS calculations. In the case of small species, which effectively expose all atoms at the outer molecular surface, any changes to parameters that affect ion radii (*i.e.*, V_{vdW}, V_{IID}, and V_{IQ} terms) will have a substantial influence on buffer gas trajectories (and CCS by association). As the size of the molecular entity increases, atoms can become buried within the effective ion core, and thus will exhibit little influence on buffer gas trajectories. Continual ion growth will eventually reach a size regime where only a small fraction of the total number of atoms is exposed at the molecular surface. When this size regime is reached, changes in effective ion radii will minutely increase the size of the exposed surface and will essentially leave the volume of the ion core unchanged. Bearing this in mind, we expect the effect of corrections from V_{IID} and V_{IQ} terms to diminish to the point where macromolecular ions (e.g., proteins, DNA/RNA, etc.) can be treated relatively accurately using only a vdW interaction potential.

2.4 Benchmarking CCS calculation speed

Prior to this work, the updated MobCal code was only available for serial execution.[14] This severely limited its performance on High Performance Computing (HPC) architectures capable of parallelization. To remedy this, we have modified the original code to improve its computational efficiency. Briefly, modifications to the updated MobCal code now include: (i) message passing interface (MPI) to support parallel computing architectures, (ii) replacement of *goto* looping with *do* and *endo* functionality, enabling more efficient optimization, (iii) significant reworking of the *dljpot* and *mobil2* subroutines, including loop reorganization, removal of repetitive calculations of identical arrays and unused variables, (iv) consistent double precision used throughout, and (v) comments and patch notes for future use and debugging purposes. When operating on a single core (i.e., in serial), our re-worked MPI code was found to perform 2.9-fold more efficiently than the serial implementation developed by Lee and coworkers.[14] It should be noted that there are miniscule differences in CCSs calculated with the two codes owing to differences created by the pseudo-random number generator RANLUX.

Extensive benchmarking of the new parallelized code has been conducted on species ranging in size from 9 to 7,029 atoms (135 Da to 50 kDa). Benchmarking was performed on the Graham HPC cluster, which is part of the Sharcnet consortium of Compute Canada. Each CPU node is dual socket, equipped with 2 x 16 core Intel E5-2683 v4 chipsets, operating at 2.1 GHz for a total of 32 cores per node. Benchmarking was performed against serial execution of MobCal-MPI and the non-parallelized implementation developed by Lee and coworkers.[14] Speedups and overall efficiency are shown in Figure 5. Our updated MPI code substantially reduces average CCS computation time, providing 36- and 64-fold increases in efficiency when deployed on 16 and 32 cores, respectively, relative to the most recent serial version of MobCal.

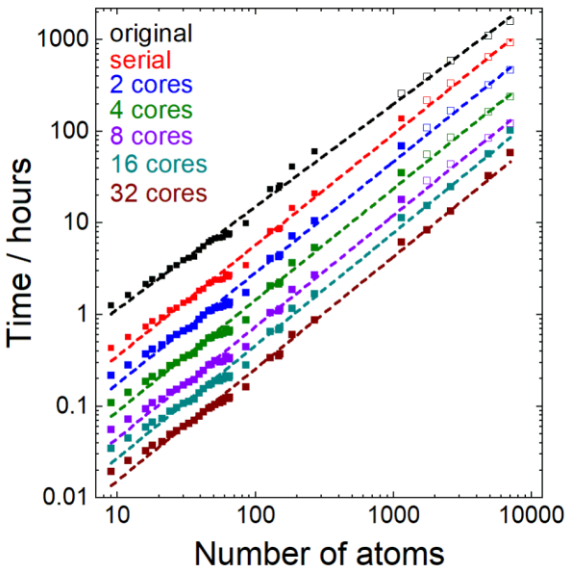


Figure 5. Total execution time in hours for numerous species ranging in size from 9 to 7,029 atoms using the most recent serial MobCal code (black),[14] and MobCal-MPI using 1 (red, serial), 2 (blue), 4 (green), 8 (purple), 16 (teal), and 32 (maroon) CPU cores for calculation. Open squares indicate extrapolated points.

2.5 Additional Reading

1. An excellent review on prior methods and future directions for ion mobility calculations by Prof. Carlos Larriba-Andaluz and Prof. James S. Prell

   Larriba-Andaluz, Carlos., Prell, James S. Fundamentals of ion mobility in the free molecular regime. Interlacing the past, present and future of ion mobility calculations. *Int. Rev. Phys. Chem.* **2020**, 39 (4), 569-623. https://doi.org/10.1080/0144235X.2020.1826708

2. Another excellent review by Prof. James S. Prell on the use of molecular dynamics simulations in tandem with ion mobility calculations to compliment native IMS experiments.

   Rolland, A. D.; Prell, J. S. Computational Insights into Compaction of Gas-Phase Protein and Protein Complex Ions in Native Ion Mobility-Mass Spectrometry. *TrAC - Trends Anal. Chem.* **2019**, *116*, 282–291. https://doi.org/10.1016/j.trac.2019.04.023.

3. Installing the MobCal-MPI Suite Graphical User Interface (GUI)

MobCal-MPI is equipped with a Graphical User Interface (GUI) to enhance functionality without use of the command prompt or Python environments (IDLE, etc.). The GUI is equipped with subroutines that enable:

1. Generation of MobCal-MPI input files (.mfj) from Gaussian output files (.log)
2. Automated Boltzmann-weighting for calculation of Boltzmann-weighted CCSs
3. Extraction of CCSs from MobCal-MPI output files (.mout)

3.1 Installing prerequisite packages

The MobCal-MPI GUI requires the OpenBabel 2.4.1 and SDF2XYZ2SDF programs in order to function. Installation of these packages is detailed below.

Babel 2.4.1

The GUI is compatible has been tested with OpenBabel 2.4.1

N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison. Open Babel: An open chemical toolbox. *J. Cheminf.* (2011), 3, 33.

OpenBabel 2.4.1 can be downloaded from:

https://sourceforge.net/projects/openbabel/files/openbabel/2.4.1/

Follow the on-screen instructions and install in the default directory.

SDF2XYZ2SDF

The GUI requires the SDF2XYZ2SDF program. For details on the functionality of this program, please see P. Tosco and T. Balle, *Journal of Molecular Modeling*, 2011, 17, 3021-3023.

SDF2XYZ2SDF can be downloaded from:

https://sourceforge.net/projects/sdf2xyz2sdf/files/binaries/windows/

Follow the on-screen instructions and install in the default directory.

Adding Babel and SDF2XYZ2SDF directories to PATH

If you specified the default install directories for both packages, these need to be added to PATH.

To add the directories to the PATH:

1. Enter 'Environment Variables' in the Windows Search bar
2. Select 'Edit the system environment variables'
3. Select the 'Environment Variables…' radio button at the bottom of the screen.
4. Highlight the Path variable in the top box and select the 'Edit…' radio button.
5. Select the 'New' radio button and add the following two entries:
    a. C:\Program Files\OpenBabel-2.4.1
    b. C:\open3dtools\bin

3.2 Installing Python and Python libraries

Python 3.7 or 3.8

The GUI is compatible with all Python 3.7 and 3.8 releases. Download the version to suite your OS from https://www.python.org/downloads/. When installing, ensure that the radio button to 'Add Python 3.X to PATH' is selected; it should be located at the bottom of the installer window.

It is also beneficial to install do a custom installation of Python such that you can modify the install directory. A convenient location is C:/Python37 (for Python 3.7) or C:/Python38 (for Python 3.8)

Numpy and PyQt5

The MobCal-MPI Suite GUI requires the numpy and PyQt5 libraries. To install these:

1. Open up the command prompt (or equivalent for Mac/Linux). For Windows users, type *cmd* into the search bar.
2. Type the following commands:

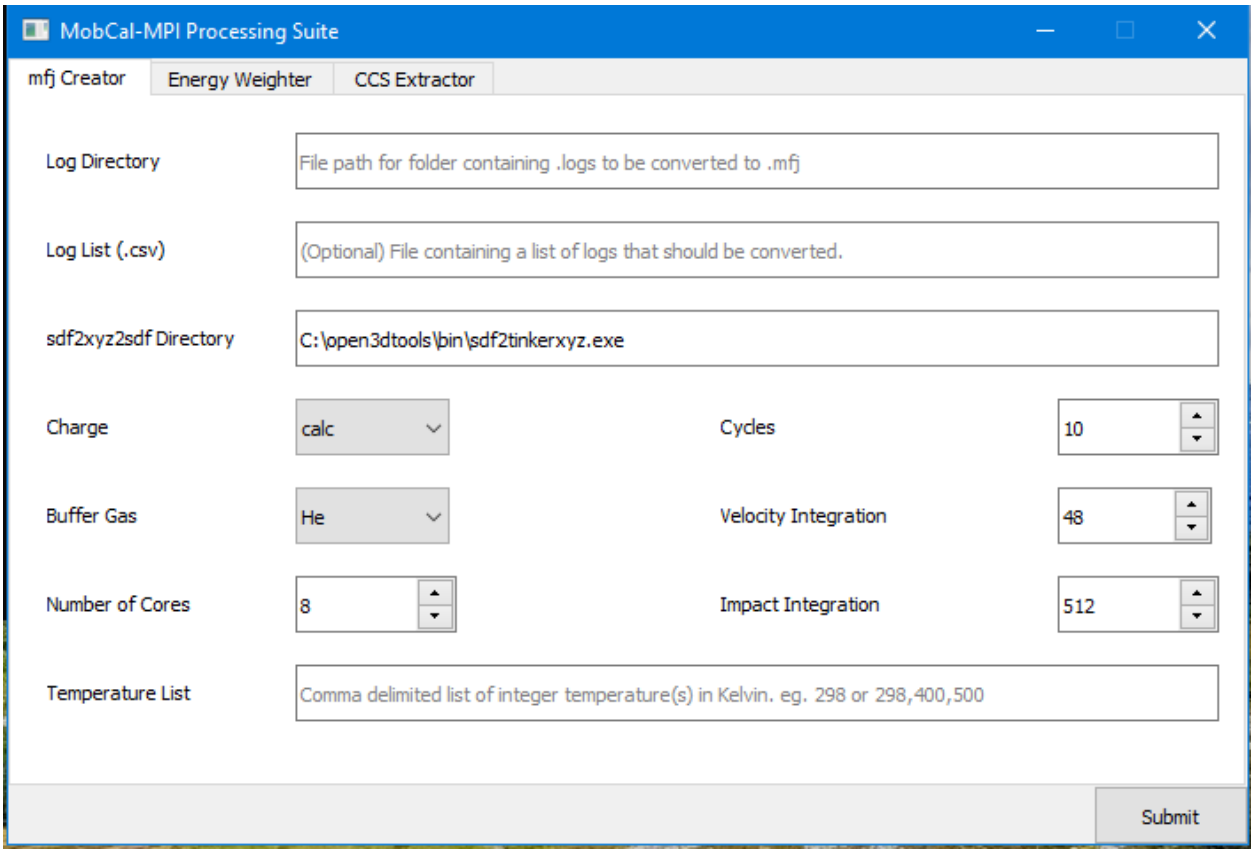
    pip install numpy
    pip install pyqt5

    Note that by running either of these commands, you may be prompted to update pip. This is normal behaviour.

3.3 Verifying installation

From the GUI folder, launch the Launcher.py module. If any libraries are missing upon execution, the user will be prompted to install these. Any bugs or suggestions to the GUI should be reported to Issues via GitHub or to cieritano@uwaterloo.ca

Successful installation will be met with the following interface:



3.4 Using the MobCal-MPI Suite GUI

3.4.1 mfj Creator

The mfj Creator automates the generation of MobCal-MPI input files (.mfj) from Gaussian output files. For specifics on how to set up Gaussian calculations to produce output files compatible with MobCal-MPI, please see Appendix A.

Within the mfj creator, all fields are editable. The functionality of each field is described below:

Log Directory: The directory that contains the .log files for conversion

Log List (.csv): An optional entry specifying the file directory and filename of a .csv containing a list of .log files for conversion to .mfj. This is often useful for excluding CCS calculations of high-energy isomers without having to move files around. A sample .csv is provided in the GUI_Samples_Files folder of the MobCal-MPI release (**Test_CSV_List.csv**).

sdf2xyz2sdf Directory: If sdf2xyz2sdf was not installed in the default directory, you can specify the location of it here. By default, the default install location is filled in. If the sdf2tinkerxyz.exe is not found in this location, specify it here.

If you installed in the default directory, nothing needs to be changed here.

Charge: This variable allows for specification of the partial charge assignment scheme. The options available are Calc, Equal, and None

charge = 'calc'
This specifies that a unique partial charge scheme is going to be read. Currently this MFJ creator can read charges predicted by electrostatic potential mapping methods (eg. ChelpG, MK) by Gaussian directly from the .log output file. For more details on this, see Appendix A.
Recommended.

charge = 'equal'
This specifies that equivalent partial charges are assigned to each atom by the relationship partial charge = (ion charge) / (number of atoms)
Not recommended.

charge = 'none'
Assign all atomic charges as zero, which effectively removes the ion-induced dipole and ion-quadrupole terms for the ion-gas potential.
Not recommended.

Note that regardless of charge schemes, this script was built in such a way such that Gaussian .log files must contain partial charges calculated by either the MK/ChelpG schemes. If a .log file is missing partial charges, the .mfj creator will not execute regardless of the charge scheme selected.

Buffer Gas: MobCal-MPI is parameterized for CCS calculations in N_2 and He. Here, you can specify the buffer gas you intend to perform CCS calculations in.

Temperature List: As of MobCal-MPI v1.2, you can now specify custom temperatures in the input line. You can also specify multiple temperatures to perform multiple temperature-dependent CCS calculations in one MobCal-MPI calculation. Input for this line is a comma separated list. Up to 20 temperatures can be specified, and they must be integers, e.g.,

298

298,400,500,800

Number of cores: The number of cores intended to use for MobCal-MPI calculations. This variable is not used to generate .mfj input files, but is used for error handing to ensure the number of calculations is equality partitioned over the number of cores. The number of cores requested must be divisible by the number of trajectories, as determined by the product of the number of cycles, velocity point integrations, and impact parameter integrations. The recommended value is **8** or **16**, both of which are compatible with the recommended values below.

Cycles: The number of cycles of CCS calculations MobCal-MPI will perform. The calculated CCS will be the average CCS calculated across each cycle. The recommended number of cycles is **10**.

Velocity Integration: The number of relative velocities (*inp*) of the buffer gas to sample during one cycle of CCS calculation. The recommended value is **48.**

Impact Integration: The number of collisions to sample for each relative velocity of the colliding neutral gas. For moderate sized molecules (< 400 Da), **512** points are recommended. For larger molecules, we recommend a greater number of collisions to sample to minimize the variance of the CCS (**752 or 1008** points).

Once all parameters are filled in, click the 'Submit' button to start the conversion process. The code will run sequentially through files in alphabetical order to convert Gaussian .log files to MobCal-MPI input files. Upon successful completion, contain a header that looks like:

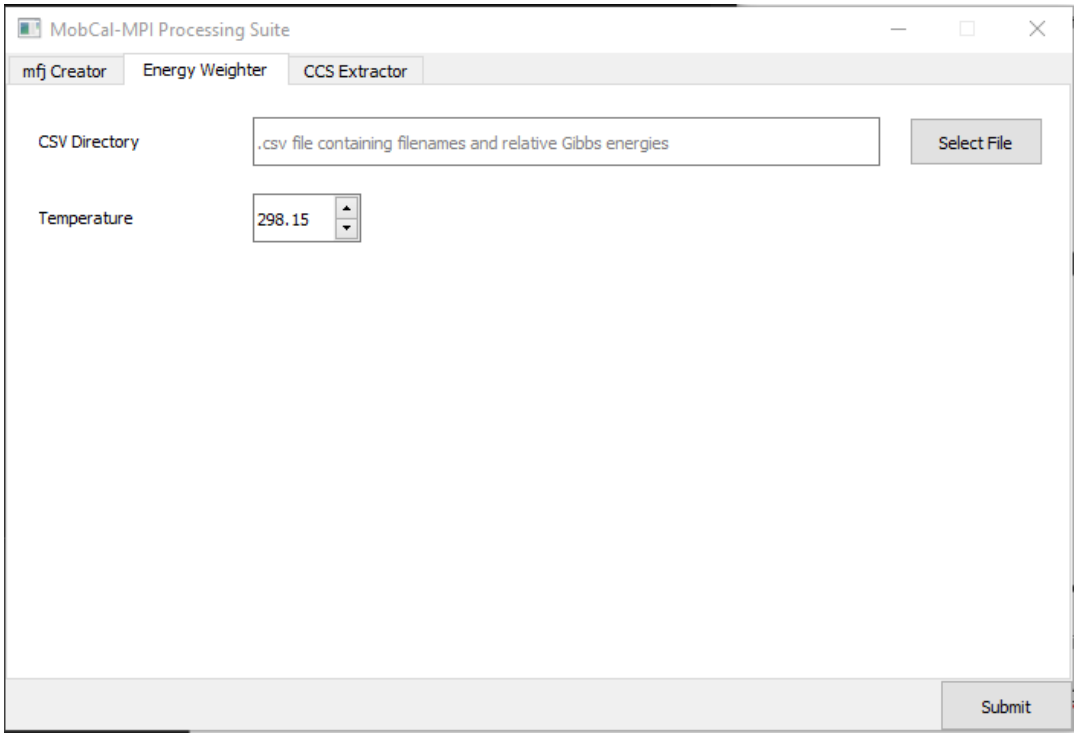
```
AMIFOSTINE_3          Title
1                     Number of conformers (always 1)
28                    Number of atoms
ang                   Specifies coordinates to be read in angstroms (ang)
calc                  Specifies if a calculated (calc), equal (equal), or no (none) partial charge scheme is used
1.0000                Scaling factor for xyz coordinates. Set to 1.0000
10 48 512 2 -529376712 298 400 500   (# cycles) (# of velocity int points) (# of impact parameter int points) (gas; 1=He, 2=N2)
                                     (random seed) (temp(s) list)
```

And a body that contains all the pertinent molecular information:

```
AMIFOSTINE_3
1
28                  xyz coordinates        Atomic mass  Partial charge   MMFF vdw parameters
ang
calc
1.0000
10 48 512 2 -529376712 298 400 500
  1.290700    0.833300   -1.273600    31.972   -0.193666    3.00  4.800  3.320  1.345
  1.835900   -0.567900    0.215300    30.974    0.928334    1.600 4.500  3.320  1.345
  1.601500    0.228400    1.605900    15.995   -0.608579    0.70  3.150  3.890  1.282
  3.441700   -0.700200    0.177400    15.995   -0.621605    0.70  3.150  3.890  1.282
  1.123800   -1.872800    0.078400    15.995   -0.623803    0.75  3.150  3.890  1.282
 -1.323000    0.767800    0.692000    14.003   -0.039906    1.15  2.820  3.890  1.282
 -1.458700   -1.483100   -0.811300    14.003   -0.302638    1.00  2.820  3.890  1.282
 -2.757700    0.544900    1.014800    12.000   -0.136973    1.050 2.490  3.890  1.282
 -3.480200   -0.274000   -0.071900    12.000   -0.019077    1.050 2.490  3.890  1.282
 -1.054200    1.971400   -0.111000    12.000   -0.040526    1.050 2.490  3.890  1.282
  0.439900    2.194000   -0.338400    12.000   -0.014244    1.050 2.490  3.890  1.282
 -2.860900   -1.659300   -0.288600    12.000    0.095374    1.050 2.490  3.890  1.282
 -2.799000   -0.005900    1.960400     1.008    0.091435    0.250 0.800  4.200  1.209
 -3.284100    1.496100    1.171900     1.008    0.080214    0.250 0.800  4.200  1.209
 -3.500200    0.275800   -1.022000     1.008    0.046225    0.250 0.800  4.200  1.209
 -4.525400   -0.406500    0.225100     1.008    0.065223    0.250 0.800  4.200  1.209
 -1.458100    2.875500    0.371900     1.008    0.042858    0.250 0.800  4.200  1.209
 -1.565300    1.866600   -1.074800     1.008    0.082288    0.250 0.800  4.200  1.209
  0.974400    2.358200    0.598800     1.008    0.076472    0.250 0.800  4.200  1.209
  0.589100    3.075000   -0.968200     1.008    0.131005    0.250 0.800  4.200  1.209
 -3.432300   -2.270100   -0.990200     1.008    0.053687    0.250 0.800  4.200  1.209
 -2.779200   -2.204400    0.655600     1.008    0.072372    0.250 0.800  4.200  1.209
 -1.429400   -1.306500   -1.816500     1.008    0.278131    0.150 0.800  4.200  1.209
 -0.795400   -2.237000   -0.596000     1.008    0.326918    0.150 0.800  4.200  1.209
  2.343200    0.155300    2.228600     1.008    0.491602    0.150 0.800  4.200  1.209
  3.745900   -1.567900   -0.135000     1.008    0.495074    0.150 0.800  4.200  1.209
 -0.776100    0.804000    1.549800     1.008    0.178081    0.150 0.800  4.200  1.209
 -1.083700   -0.620300   -0.299900     1.008    0.065725    0.150 0.800  4.200  1.209
```

3.4.2 Energy Weighter

The energy weighter calculates the relative population of an isomer present in an ensemble for the purpose of calculating Boltzmann-weighted CCSs through the following interface.



Relative populations (ρ) for the k^{th} isomer are defined by Equation 10

$$\rho_i = \frac{\exp(G_i \cdot (RT)^{-1})}{\sum_k \exp(G_i \cdot (RT)^{-1})} \qquad \textit{Eq 10}$$

where R is the gas constant, G^{ik} is the Gibbs energy of the i^{th} isomer of the analyte of interest, and T is the temperature in Kelvin.

To calculate relative populations using the Energy weighter, a .csv file containing a list of filenames and their Gibbs corrected energies, as calculated using computational chemistry packages (see Appendix A), must be specified in the **CSV directory** entry slot. Isomer populations can be calculated at any temperature (Kelvin) through editing the **Temperature** field.

An example input .csv file is provided with the GUI in GUI_Sample_Files (**Gibbs.csv**).

Basically, the .csv contains 2 columns with a title at the top. The .csv format is designed to work in tandem with the Featherstone Suite (https://github.com/jrjfeath/Featherstone_Labs_Suite), a Python package designed for post-processing of Gaussian calculations (Appendix A). Extracting thermochemical data using the Featherstone suite generates a .csv that contains the Gibbs corrected energies in addition to other thermochemical corrections. Only the Filename and Sum of electronic

and thermal Free Energies columns (including the titles) should be retained. Energy units are irrelevant since values are taken and recalculated relative to the global minimum energy (leaving this in units of Hartree is perfectly fine). To run the code, click **Submit**.

The script discovers identical molecules by their name prior to detecting an underscore in the file name. Using the energy weighter assumes your files are appropriately named; eg. Molecule1_1.log, Molecule1_2.log; Molecule2_1.log, Molecule2_2.log ...and so on. The code will recognize and split molecule 1 from 2, and calculate the Boltzmann weight for the 2 conformations of each independent of other molecules.

An example output file is provided with the GUI (**Rel_E.csv**). Boltzmann weighted CCSs are calculated as per equation 11:

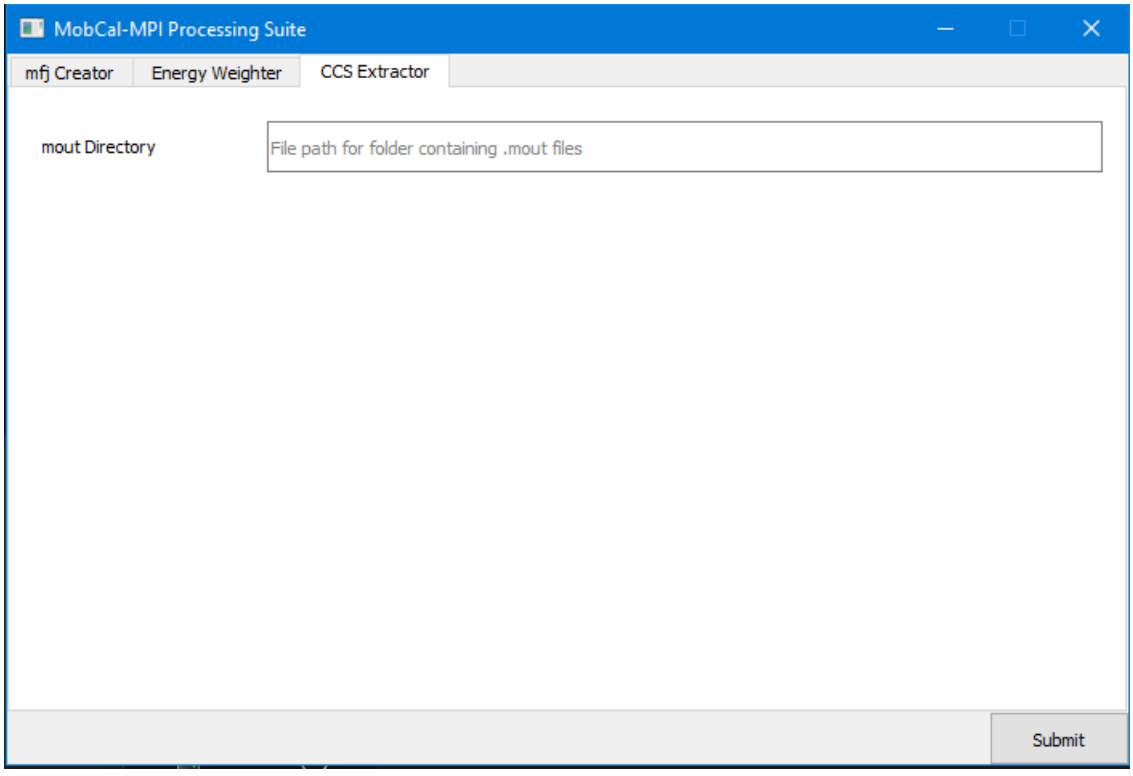
$$CCS_{BW} = \sum_i CCS_i \cdot \rho_i \qquad \textit{Eq 11}$$

Where CCS_i and ρ_i are the CCS and relative population of the i^{th} isomer of an analyte, respectively.

3.4.3 CCS Extractor

The CCS extractor returns the mass, CCS, CCS standard deviation, and the temperature that the CCS calculation was performed at. To run the CCS extractor, paste the directory containing the .mout files into the **mout Directory** field and click **Submit**. Example files and outputs are provided.

The CCS is output as a .csv titled **CCS_DATA.csv** to the same directory.



4. Running CCS Calculations in MobCal-MPI

4.1 Compiling MobCal-MPI

4.1.1 Compiler version

MobCal-MPI is written in Fortran, and as such, requires the use of legacy Fortran compilers. Most publicly-funded high-performance computing architectures (UNIX platforms) will be equipped with these and the associated documentation. We recommend using MobCal-MPI on such frameworks due to the general difficulty associated with installing Fortran compilers on local systems.

Note that we do not provide support related to installion of Fortran compilers. However, we recommend matching the compiler version to the version in which the benchmarks were conducted. Specifically, MobCal-MPI was tested on ComputeCanada clusters (https://docs.computecanada.ca/wiki/MPI) using the mpifort compiler (Intel 2016.4). The version info of mpifort is shown below:

[cieritan@gra-login2]$ mpifort --version
ifort (IFORT) 16.0.4 20160811
Copyright (C) 1985-2016 Intel Corporation. All rights reserved.

4.1.2 Compilation of MobCal-MPI

Once the compilers are implemented, compilation MobCal-MPI is done using the following command:

mpifort -o MobCal_MPI.exe MobCal_MPI.f -Ofast

Correct compilation of MobCal-MPI should produce a file named MobCal-MPI.exe without any error messages displayed in the terminal.

4.2 Running MobCal-MPI

4.2.1 Submitting a MobCal-MPI job

Without a job scheduler:
Beyond the .mfj input file, an additional file is required to submit a MobCal-MPI job. This file can be named anything and does not require a specific extension. The first line of the file should be the name of the .mfj input file found in the same directory. The second line should be the name of the output file you want the results to print to (.mout). For example, if I had an input file called AMIFOSTINE_3.mfj and I wanted to calculate its CCS, the job submission file should look like:

AMIFOSTINE_3.mfj
AMIFOSTINE_3.mout

Each CCS calculation requires its own job submission file. Scripts are useful in this regard.

To submit the file to MobCal-MPI **on a single core**, use the following command. Assuming the job submission file is called AMIFOSTINE_3.sub:

./MobCal_MPI.exe AMIFOSTINE_3.sub

To submit the file to MobCal-MPI **using multiple cores**, use the syntax native to your system to request multiple computing cores. For example, to run on 8 cores:

./MobCal_MPI.exe -n 8 AMIFOSTINE_3.sub

With a job scheduler:
Submission of MobCal-MPI jobs with a job scheduler is just as easy as without one. Using the same format for the job submission file, a sample bash script compatible with SLURM schedulers, as used on Compute Canada (https://docs.computecanada.ca/wiki/Running_jobs), is provided (**runmob_MPI.sh**).

Note that execution of bash scripts requires changing file permissions to enable execution. Using runmob_MPI.sh as the example submission script, changing permissions and execution is done by the following commands:

chmod a+x runmob_MPI.sh
./runmob_MPI.sh

***4.2.2 Verifying proper performance of MobCal-MPI

It has been reported that the use of optimization flags when compiling MobCal-MPI can cause erroneous behaviour. This bug has been fixed in **Version 1.2**. However, we still recommend you conduct CCS calculations on the test files provided and verify that the CCSs correspond with that of the reference file to within standard deviation.

Test files are provided with MobCal-MPI in both the **GUI_Samples_files** folder (under **mfj_creator**) and in the **Compiler_flags_test** folder using mpifort and mpif90 (an older compiler) with several optimization flags (O1, O2, O3, Ofast).

To verify that MobCal-MPI is performing correctly, please look at the tables of calculated values for:

1. pgst
2. wgst
3. v
4. ke/kt
5. gst^5 * exp(gst^2/tst)
6. frac of sum
7. gst
8. b2max/ro2
9. b/A

average TM cross section (*i.e.*, CCS)

If these entries and the CCSs to not match with these test files, please report this in the Issues section and/or contact us by email cieritan@uwaterloo.ca and shopkins@uwaterloo.ca. When contacting us, please ensure you attach the .mfj input, .mout output, and include the compiler type and version number used.

MobCal-MPI can still be used if this compilation error persists, but will require recompilation without the use of compilation flags. To do this, use the following command on **MobCal_MPI.f**:

mpifort -o MobCal_MPI.exe MobCal_MPI.f

Appendix A – Setting up Gaussian calculations for use with MobCal-MPI

Appendix A outlines the basic procedure for exploring the potential energy surface (PES) of a molecule to generate a series of local minima. The geometries of these local minima and their electrostatic potentials are refined at higher levels of computational theory, namely using Density Functional Theory (DFT), to generate candidate structures for the gas-phase ensemble. These DFT optimized structures serve as the basis for generation of MobCal-MPI input files.

This guide walks you through use of the Basin-Hopping (BH) algorithm, a custom code developed by the Hopkins group, for exploring the potential energy landscape of a molecule (A1). Local minima identified by this routine are ported to the Gaussian16 computational chemistry suite for refinement at higher level of theory (A2). A discussion of appropriate levels of computational theory that are compatible with MobCal-MPI is provided (A3), with specific recommendations for calculation methods provided in Appendix A2.2.

A1. Exploring the potential energy surface (PES)

A1.1 Basin-Hopping (BH) to generate local minima

Since the ion swarm probed during an IMS experiment consists of an ensemble of structures, a series of local minima should contribute to the calculated CCS. In the regard, weighting the contribution of an isomer's CCS by their relative population (*i.e.*, Boltzmann-weighted CCS) allows for trajectory-method CCS calculations using frozen-core inputs.

Candidate structures are generated by exploring the PES of the analyte using the basin-hopping (BH) algorithm. The BH method can be envisioned as a guided walk along the potential energy landscape, where molecules are distorted along their rotatable dihedral angles and subsequently optimized at a low-level of computational theory (molecular mechanics or semi-empirical).

Basin-hopping is used to locate minima on a PES through applying local minimization of a molecular system given the initial coordinates X (Equation 12).[28]

$$\tilde{V}(X) = \min[V(X)] \qquad \textit{Eq 12}$$

$\tilde{V}(X)$ is associated with the local minimum energy found by performing a geometry optimization at any point along the PES (*i.e.*, X). Sampling and optimizing at various points allows us to project the potential energy landscape into a series of catchment basins (Figure S1), where each plateau corresponds to a unique local minimum. Instead of a formal Monte Carlo[29] based sampling of the PES, random distortions along the nuclear coordinates are instead directed towards lower energy conformations. In other words, the sampling of the PES can be thought of a 'random, but targeted' walk along the potential energy landscape towards the global minimum (GM).

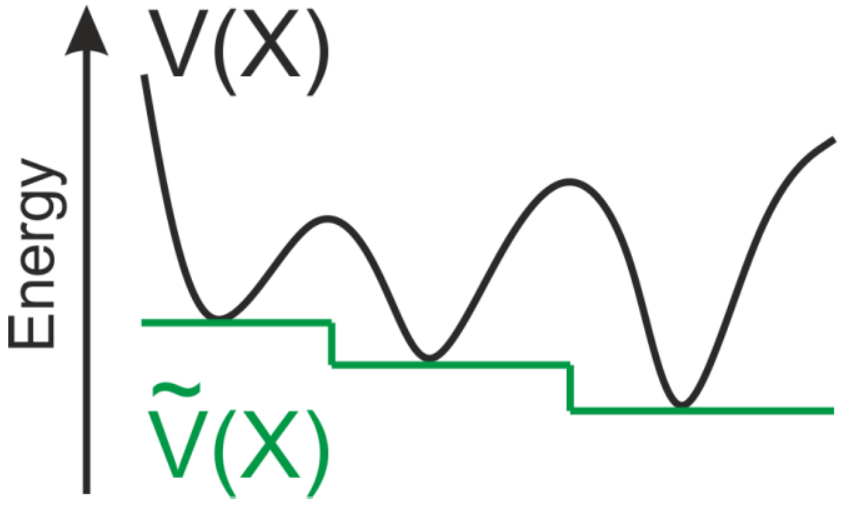


Figure S1. Projection of the potential energy landscape generates a series of catchment basins which correspond to energies of various local minima

To accomplish this targeted behaviour, each iteration of the BH routine perturbs the current cluster coordinates and performs a geometric optimization to obtain a new minimum with energy V_i (Figure S2). The optimized structure is accepted if it meets one of the following criteria:

1. The energy of the current step is lower than the energy of the current GM (*i.e.*, $V_i < V_{GM}$)
2. If $V_i > V_{GM}$, the step is accepted with a probability proportional to a Boltzmann distribution at a pre-specified temperature (*i.e.*, $e^{\frac{V_{GM}-V_i}{k_bT}}$)

If a step fails to meet either criteria, the iteration is discarded, and a new step is proposed by geometric distortion from the original structure. Owing to the second acceptance condition, the steps between adjacent minima are not completely biased to go in any particular direction along the PES. Thus, this approach allows for random sampling of the PES, but with a low-energy trajectory guided towards a global minimum candidate.

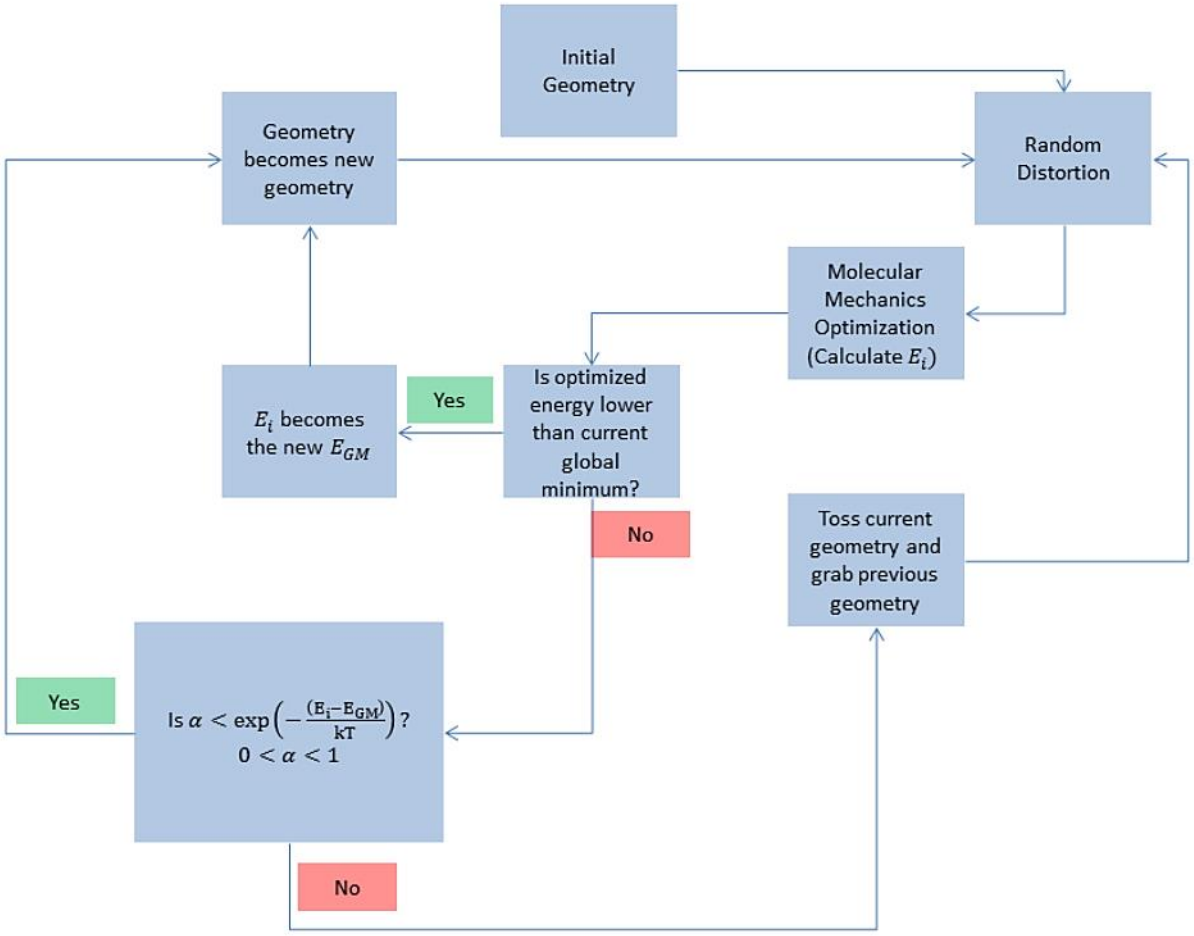


Figure S2. A flowchart of the BH algorithm. The routine loops through this cycle until the number of steps specified are completed.

To achieve an effective sampling of the PES, a relatively large number of structural perturbations must be performed. Depending on cluster size, the number of structural candidates required can range from 10 000 to 25 000, or even higher for larger systems. Due to this high volume of inputs, geometry optimizations are completed using molecular mechanics, as these methods are computationally inexpensive.

We are currently working on documentation for using the BH algorithm. Until then, the code can be requested from our group by emailing cieritano@uwaterloo.ca or shopkins@uwaterloo.ca.

A1.2 Post-processing of BH results and DFT optimization

Post processing of a BH routine, which includes identification of unique minima, is accomplished through sorting of isomers based on energetics and geometry. A typical routine will return (at most) 250 unique, low energy, and viable candidate geometries for further structural refinement at higher levels of computational theory (Figure S3).

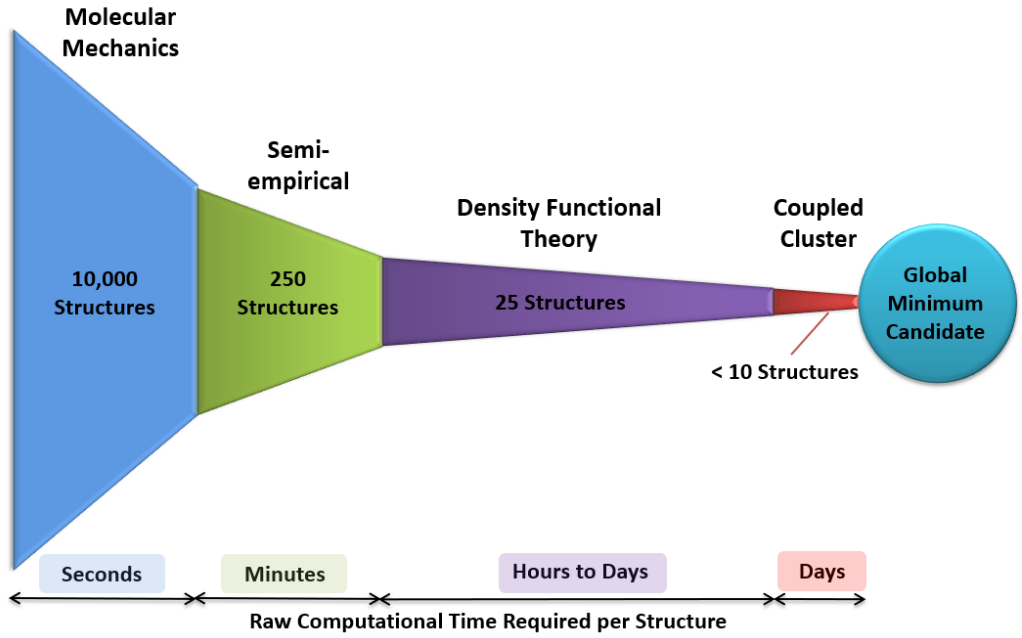


Figure S3. Calculations to identify a cluster global minimum are performed using a hierarchal method which refines isomer sets at progressively higher levels of theory.

Sorting isomers based on geometry and energetics is seamless when conducting using the Featherstone Suite (https://github.com/jrjfeath/Featherstone_Labs_Suite). It should be noted that the Featherstone suite is designed for use with the Gaussian computational chemistry package. Basically, several tools are implemented within that are extensively documented. These tools allow you to sort unique isomers based on energy and cosine similarities[30] using a GUI.

Semi-empirical methods, such as Density Functional based Tight Binding (DFTB)[31,32] or the recently developed parameterized model 7 (PM7)[33] often serve as a pre-filter prior to *ab initio* calculations to further refine isomer sets obtained from the BH routine. These methods employ approximations to the Hartree-Fock (HF) formalism,[34] where certain 2-electron Hamiltonians are replaced by a set of parameters identified for analogous systems, or are completely omitted. While the cluster geometries predicted by the efficient semi-empirical methods reminisce those of *ab initio* calculations, they produce unreliable electronic energies and thermochemistry. The ability to sort unique BH files and convert then to new Gaussian inputs is all possible within the Featherstone Suite.

Once the PES of the analyte has been thoroughly explored and refined with PM7, representative local minima are ready for refinement at higher levels of computational theory. For this, we recommend the use of Density Functional Theory (DFT). Selecting a DFT functional and basis can seem as daunting as reading Fortran code, but rest assured, there is a method to the madness. Martin Head-Gordon wrote an excellent review on the performance of 200 density functional and scored their performance based on their ability to accurately calculate non-covalent interactions, isomerization energies, thermochemistry, and barrier heights based on experimental values.

To spare you the details, the functionals that are 1) available in Gaussian16 and 2) demonstrate optimal performance for calculation of ion thermochemistry are B3LYP-D3, ωB97X-D, and M06-2X-D3

when paired with basis sets from Ahlrichs and Weigand (e.g., def2-TZVPP). Specifically, **we recommend ωB97X-D/def2-TZVPP** using partial charges computed using the Merz-Singh Kollman (MK) partition scheme and constrained to reproduce the molecular dipole moment (pop=mk,dipole). This option is requested in Gaussian16 using the following calculation line:

 # opt=maxcycles=100 freq wb97xd/def2tzvpp int=ultrafine pop=(mk,dipole)

The output files generated from DFT calculations performed in Gaussian are imported to the **mfj creator** of the MobCal-MPI GUI to create MobCal-MPI input files (Section 3.4.1).

Other density functional and basis set combinations perform optimally for CCS calculations in MobCal-MPI. This performance is outlined in our latest publication in *Materials Communications Today.*

Ieritano, Christian, Hopkins, W. Scott. Assessing collision cross section calculations using MobCal-MPI with a variety of commonly used computational methods. *Mat. Comm. Today*. **2021**, 102226.

We have nevertheless summarized the contents of this article in Section A2.

A2. Which calculation method should I use? Explained!

A2.1 – Benchmarking CCS calculations using MobCal-MPI with a variety of computational methods

The following section is derived from a recent MobCal-MPI publication in *Materials Today Communications*.

Ieritano, C.; Hopkins, W. S. Assessing Collision Cross Section Calculations Using MobCal-MPI with a Variety of Commonly Used Computational Methods. *Mater. Today Commun.* **2021**, *27*, 102226. https://doi.org/10.1016/j.mtcomm.2021.102226.

To assess which computational method is best suited for use with MobCal-MPI, a series of structurally diverse molecules with reliable experimental CCSs must be examined. The validation set used in this regard is the same as the original MobCal-MPI publication (50 molecules, 238 total isomers).[35] The CCS of each of the molecules was procured from reliable literature sources acquired on either drift-tube IMS or travelling-wave IMS systems.

Given the variety of computational methods and improved functionals available in commercially available computational chemistry packages, we performed an additional validation of the performance of MobCal-MPI using Hartree-Fock theory and three different density functionals, each tested with six different basis sets, as well as using the semi-empirical PM7 method. [33,36] DFT calculations on each of the 238 isomers were performed using Gaussian 16 C.01[37] using the B3LYP-D3,[38–40] ωB97X-D,[41] and M06-2X-D3 functionals.[40,42] Three Pople basis sets (6-31G, 6-31G(d,p), and 6-31++G(d,p)) and three sets from Ahlrichs and Weigand (def2-SVP, def2-TZVP, and def2-TZVPP)[43,44] were used with the HF and DFT methods, which comprises methods of low-medium

computational cost. Normal mode analyses were also conducted to ensure that structures correspond to true minima and to calculate thermochemical corrections. Each of the 238 isomers calculated was supplemented with partial charges determined using the MK partition scheme and constrained to reproduce the molecular dipole moment.[24,25] Mulliken charges were used with the PM7 method, as this is the only partial charge assignment scheme available.

Theoretical N_2 CCSs (Ω_{N2}) were calculated using the trajectory method implementation in MobCal-MPI.[10, 12] All CCS calculations employed 10 complete cycles of mobility calculations that used 48 points of velocity integration and either 512 points Calculated Ω_{N2} are reported as average values with statistical errors assessed from the ten cycles of calculation. For ions exhibiting multiple low-energy conformations or prototropic isomers, a Boltzmann-weighted CCS is reported based on the standard Gibbs corrected energies (T = 298.15 K). Total error for a corresponding Boltzmann-weighted CCS was calculated from the standard errors for the Boltzmann-weighted CCSs of the low-energy isomers.

The relative deviations of MobCal-MPI CCSs compared to those measured experimentally are shown for each computational method as a violin plot in Figure S1. In all cases, including the PM7 method employing Mulliken charges, the majority of calculated CCSs are within 2.5% of the reported experimental value. The largest deviations of 4.0 % and 3.0 % were observed for the HF/6-31G and PM7 methods, respectively.

The agreement of MobCal-MPI to experiment across the 25 model chemistries employed indicates that any one of the methods can be used to calculate CCSs. This is because changing the basis set or level of computational theory within the 25 model chemistries explored here is unlikely to have a major effect on the calculated geometry. Thus, changing the model chemistry should only change the calculated CCS if either: 1) the relative energies of local minima differ drastically, since this would influence Boltzmann-weighted CCSs, or 2) the partial charge assignments differ. The latter issue is relatively insensitive to partial charge assignment schemes that map partial charges based on the electrostatic potential (*i.e.*, the MK partition scheme). However, since Mulliken charges depend on the basis set, errors in excess of 10 % have been observed when they are used with B3LYP-D3.[35]

In order to examine to robustness of MobCal-MPI, a comparison of the CCS determined at each model chemistry must be compared to the level of theory used in the original parameterization. Figure S2 assesses the deviation of each CCS calculation for the 238 isomers between one model chemistry and B3LYP-D3/6-31++G(d,p). Most model chemistries produces CCSs within 1% RMSD of B3LYP-D3/6-31++G(d,p). Notable exceptions include HF and any method employing the 6-31G basis set, which produces slightly overestimated CCSs when compared to other model chemistries. Nevertheless, the majority of calculated CCSs lie within ± 2.5% of the value calculated at the B3LYP/6-31++G(d,p) level of theory, further supporting that any of the 25 model chemistries can be coupled with MobCal-MPI. However, caution should be exercised when PM7 and HF methods are used with MobCal-MPI, in addition to any basis set that lacks polarization functions.

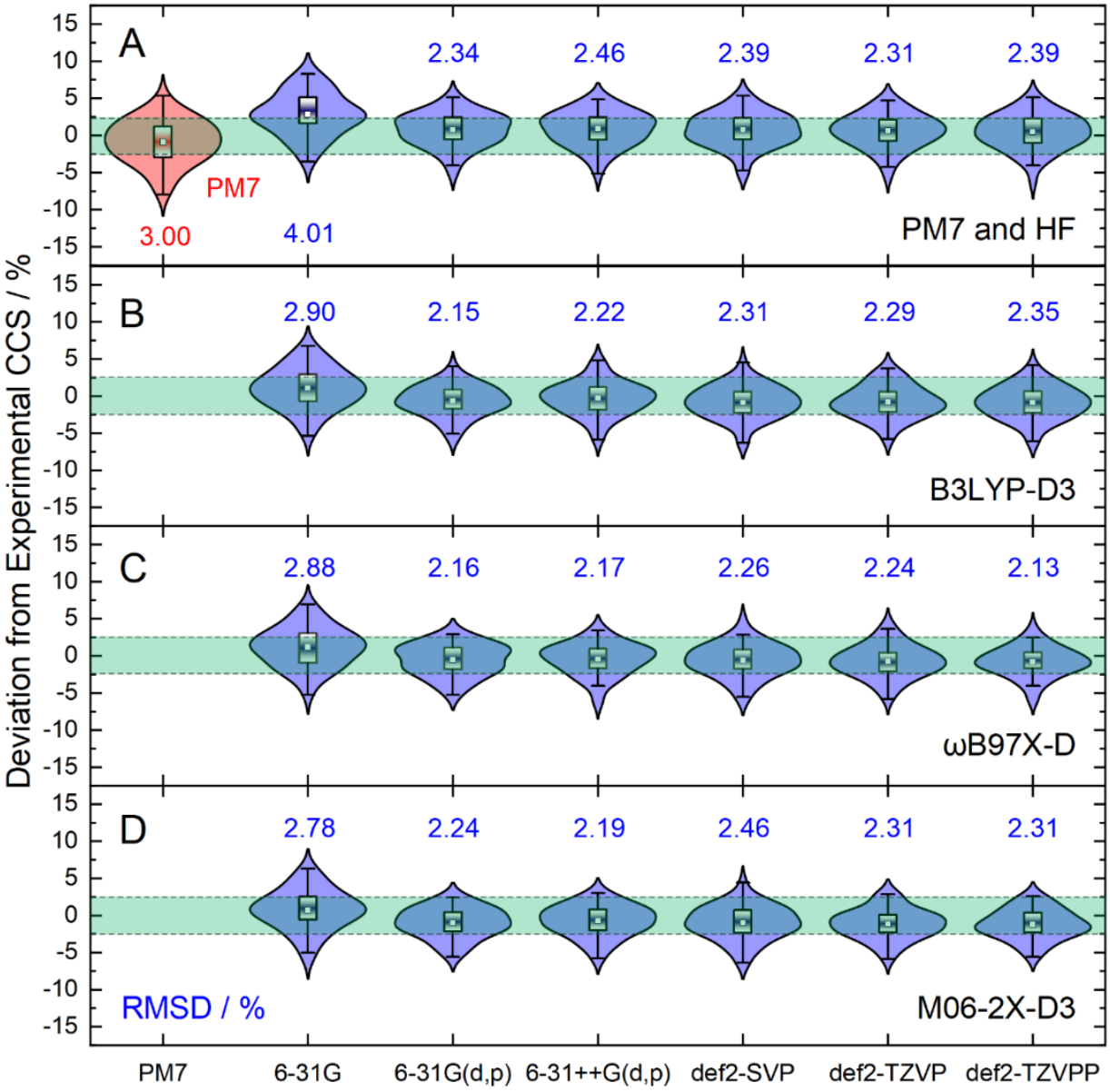


Figure S1. Violin plot showing the relative accuracy of MobCal-MPI when employing (A) Hartree-Fock, (B) B3LYP-D3, (C) wB97X-D, and (D) M06-2X-D3 computational methods with the basis sets provided across the bottom of the plot. Results for the PM7 method are shown in Panel A in red. The probability density of the dataset is overlayed with a box plot; the dot within the box represents the mean deviation for the set. The boundaries of the box indicate the 25th to 75th percentile and the whiskers indicate the highest and lowest observed relative standard deviations. The ± 2.5 % deviation region is highlighted in green. Root mean square deviations (RMSD) to experimental CCS for the various sets are provided in blue text on the plot.

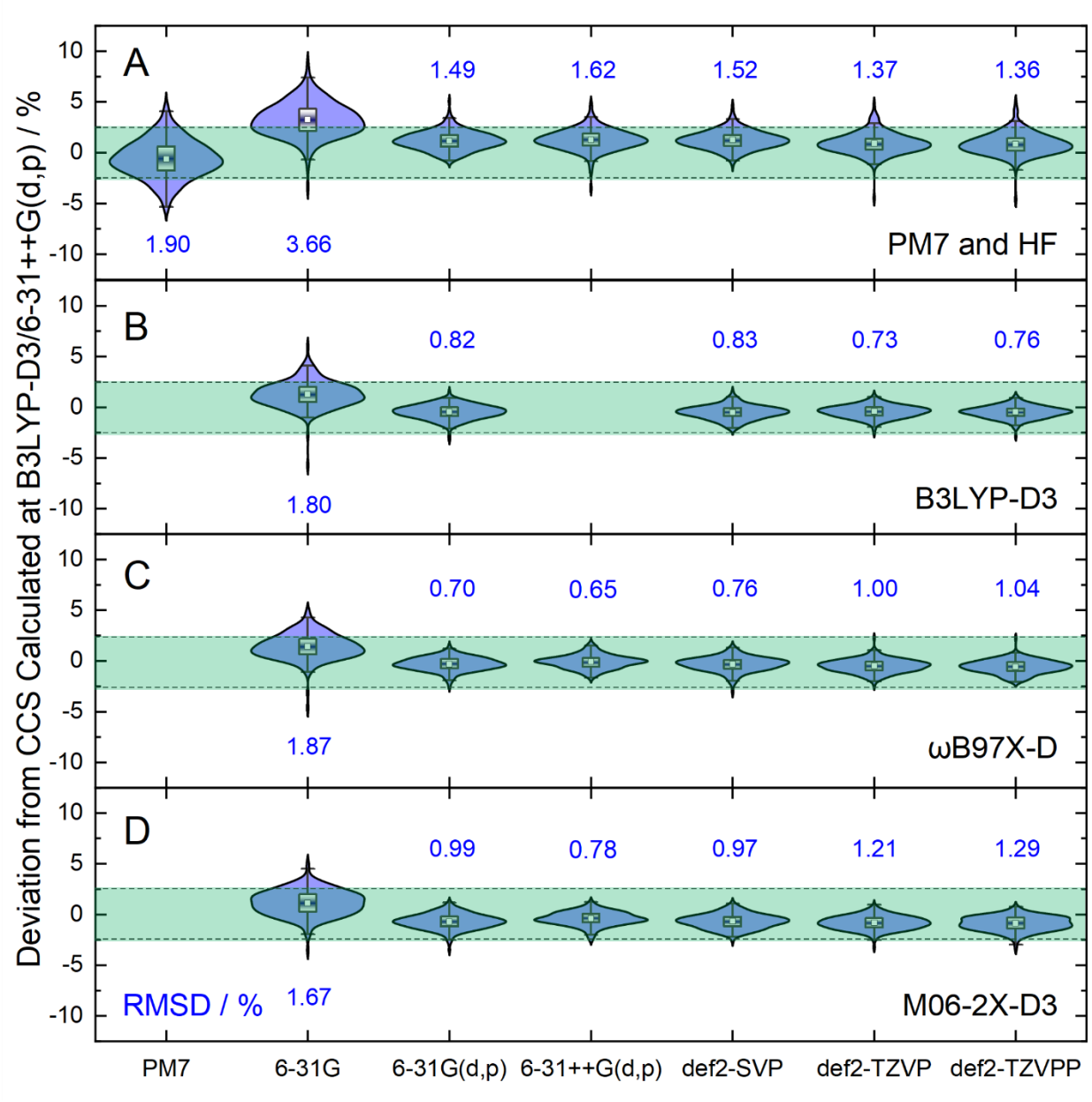


Figure S2. Violin plot showing the deviation of CCSs calculated by MobCal-MPI for each of the 238 isomers considered in this study when employing (A) Hartree-Fock, (B) B3LYP-D3, (C) wB97X-D, and (D) M06-2X-D3 computational methods with the basis sets provided across the bottom of the plot compared to the B3LYP-D3/6-31++G(d,p) level of theory. The probability density of the dataset is overlayed with a box plot; the dot within the box represents the mean deviation for the set. The boundaries of the box indicate the 25th to 75th percentile and the whiskers indicate the highest and lowest observed relative standard deviations. The ±2.5% deviation region is highlighted in green. Root mean square deviations (RMSD) of CCS calculated at the specified level of theory compared to B3LYP-D3/6-31++G(d,p) for the various sets are provided in blue text on the plot.

A2.2 – What DFT method should I use? Recommendations for MobCal-MPI compatibility

In summary, we have demonstrated that MobCal-MPI performs within the expected error of the method (ca. 2.5 %) for 25 different model chemistries. Even when employing the relatively low level PM7 semi-empirical method and Mulliken partial charges, MobCal-MPI calculated CCS values within 3% of those determined experimentally. Ultimately, users can employ MobCal-MPI with any one of the 25 model chemistries tested in this study and expect accurate CCSs, though caution should be exercised when using methods with systematic deviations from B3LYP/6-31++G(d,p). These include PM7, HF, and any basis set that does not include polarization functions.

Specifically, **we recommend the following options in order of confidence:**

(1) ωB97X-D/def2-TZVPP
(2) B3LYP-D3/6-31++G(d,p)
(3) M06-2X-D3/6-31++G(d,p)
(4) PM7 (Mulliken charges)

Appendix B – Citing MobCal-MPI

Any usage of MobCal-MPI should cite the following papers:

(1) Ieritano, C.; Crouse, J.; Campbell, J. L.; Hopkins, W. S. A Parallelized Molecular Collision Cross Section Package with Optimized Accuracy and Efficiency. *Analyst* **2019**, *144* (5), 1660–1670. https://doi.org/10.1039/c8an02150c.

(2) (1) Ieritano, C.; Hopkins, W. S. Assessing Collision Cross Section Calculations Using MobCal-MPI with a Variety of Commonly Used Computational Methods. *Mater. Today Commun.* **2021**, *27*, 102226. https://doi.org/10.1016/j.mtcomm.2021.102226.

Appendix C – References

(1) Mason, E. A.; McDaniel, E. W. *Transport Properties of Ions in Gases*; John Wiley and Sons: New York, 1988.

(2) Wyttenbach, T.; Von Helden, G.; Batka, J. J.; Carlat, D.; Bowers, M. T. Effect of the Long-Range Potential on Ion Mobility Measurements. *J. Am. Soc. Mass Spectrom.* **1997**, *8* (3), 275–282. https://doi.org/10.1016/S1044-0305(96)00236-X.

(3) Shvartsburg, A. A.; Jarrold, M. F. An Exact Hard-Spheres Scattering Model for the Mobilities of Polyatomic Ions. *Chem. Phys. Lett.* **1996**, *261* (1–2), 86–91. https://doi.org/10.1016/0009-2614(96)00941-4.

(4) Li, Z.; Wang, H. Drag Force, Diffusion Coefficient, and Electric Mobility of Small Particles. I. Theory Applicable to the Free-Molecule Regime. *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.* **2003**, *68* (6), 061206. https://doi.org/10.1103/PhysRevE.68.061206.

(5) Lee, J. W.; Davidson, K. L.; Bush, M. F.; Kim, H. I. Collision Cross Sections and Ion Structures: Development of a General Calculation Method via High-Quality Ion Mobility Measurements and Theoretical Modeling. *Analyst* **2017**, *142* (22), 4289–4298. https://doi.org/10.1039/c7an01276d.

(6) Lii, J. H.; Allinger, N. L. Molecular Mechanics. The MM3 Force Field for Hydrocarbons. 3. The van Der Waals' Potentials and Crystal Data for Aliphatic and Aromatic Hydrocarbons. *J. Am. Chem. Soc.* **1989**, *111* (23), 8576–8582. https://doi.org/10.1021/ja00205a003.

(7) Siu, C. K.; Guo, Y.; Saminathan, I. S.; Hopkinson, A. C.; Siu, K. W. M. Optimization of Parameters Used in Algorithms of Ion-Mobility Calculation for Conformational Analyses. *J. Phys. Chem. B* **2010**, *114* (2), 1204–1212. https://doi.org/10.1021/jp910858z.

(8) Campuzano, I.; Bush, M. F.; Robinson, C. V.; Beaumont, C.; Richardson, K.; Kim, H.; Kim, H. I. Structural Characterization of Drug-like Compounds by Ion Mobility Mass Spectrometry: Comparison of Theoretical and Experimentally Derived Nitrogen Collision Cross Sections. *Anal. Chem.* **2012**, *84* (2), 1026–1033. https://doi.org/10.1021/ac202625t.

(9) Wu, T.; Derrick, J.; Nahin, M.; Chen, X.; Larriba-Andaluz, C. Optimization of Long Range Potential Interaction Parameters in Ion Mobility Spectrometry. *J. Chem. Phys.* **2018**, *148* (7), 074102. https://doi.org/10.1063/1.5016170.

(10) Paglia, G.; Williams, J. P.; Menikarachchi, L.; Thompson, J. W.; Tyldesley-Worster, R.; Halldórsson, S.; Rolfsson, O.; Moseley, A.; Grant, D.; Langridge, J.; Palsson, B. O.; Astarita, G. Ion Mobility Derived Collision Cross Sections to Support Metabolomics Applications. *Anal. Chem.* **2014**, *86* (8), 3985–3993. https://doi.org/10.1021/ac500405x.

(11) Lavanant, H.; Tognetti, V.; Afonso, C. Traveling Wave Ion Mobility Mass Spectrometry and Ab Initio Calculations of Phosphoric Acid Clusters. *J. Am. Soc. Mass Spectrom.* **2014**, *25* (4), 572–580. https://doi.org/10.1007/s13361-013-0818-3.

(12) Poyer, S.; Lopin-Bon, C.; Jacquinet, J. C.; Salpin, J. Y.; Daniel, R. Isomer Separation and Effect of the Degree of Polymerization on the Gas-Phase Structure of Chondroitin Sulfate Oligosaccharides Analyzed by Ion Mobility and Tandem Mass Spectrometry. *Rapid Commun. Mass Spectrom.* **2017**, *31* (23), 2003–2010. https://doi.org/10.1002/rcm.7987.

(13) Zhang, X.; Krechmer, J.; Groessl, M.; Xu, W.; Graf, S.; Cubison, M.; Jayne, J. T.; Jimenez, J. L.; Worsnop, D. R.; Canagaratna, M. R. A Novel Framework for Molecular Characterization of Atmospherically Relevant Organic Compounds Based on Collision Cross Section and Mass-to-Charge Ratio. *Atmos. Chem. Phys.* **2016**, *16* (20), 12945–12959. https://doi.org/10.5194/acp-16-12945-2016.

(14) Lee, J. W.; Lee, H. H. L.; Davidson, K. L.; Bush, M. F.; Kim, H. I. Structural Characterization of Small Molecular Ions by Ion Mobility Mass Spectrometry in Nitrogen Drift Gas: Improving the Accuracy of Trajectory Method Calculations. *Analyst* **2018**, *143* (8), 1786–1796. https://doi.org/10.1039/c8an00270c.

(15) Halgren, T. A. Merck Molecular Force Field. II. MMFF94 van Der Waals and Electrostatic Parameters for Intermolecular Interactions. *J. Comput. Chem.* **1996**, *17* (5–6), 520–552. https://doi.org/10.1002/(SICI)1096-987X(199604)17:5/6<520::AID-JCC2>3.0.CO;2-W.

(16) Kim, H.; Kim, H. I.; Johnson, P. V.; Beegle, L. W.; Beauchamp, J. L.; Goddard, W. A.; Kanik, I. Experimental and Theoretical Investigation into the Correlation between Mass and Ion Mobility for Choline and Other Ammonium Cations in N2. *Anal. Chem.* **2008**, *80* (6), 1928–1936. https://doi.org/10.1021/ac701888e.

(17) Kim, H. I.; Kim, H.; Pang, E. S.; Ryu, E. K.; Beegle, L. W.; Loo, J. A.; Goddard, W. A.; Kanik, I. Structural Characterization of Unsaturated Phosphatidylcholines Using Traveling Wave Ion Mobility Spectrometry. *Anal. Chem.* **2009**, *81* (20), 8289–8297. https://doi.org/10.1021/ac900672a.

(18) Djoumbou Feunang, Y.; Eisner, R.; Knox, C.; Chepelev, L.; Hastings, J.; Owen, G.; Fahy, E.; Steinbeck, C.; Subramanian, S.; Bolton, E.; Greiner, R.; Wishart, D. S. ClassyFire: Automated Chemical Classification with a Comprehensive, Computable Taxonomy. *J. Cheminform.* **2016**, *8* (1), 1–20. https://doi.org/10.1186/s13321-016-0174-y.

(19) Akashi, S.; Downard, K. M. Effect of Charge on the Conformation of Highly Basic Peptides Including the Tail Regions of Histone Proteins by Ion Mobility Mass Spectrometry. *Anal. Bioanal. Chem.* **2016**, *408* (24), 6637–6648. https://doi.org/10.1007/s00216-016-9777-4.

(20) Boschmans, J.; Jacobs, S.; Williams, J. P.; Palmer, M.; Richardson, K.; Giles, K.; Lapthorn, C.; Herrebout, W. A.; Lemière, F.; Sobott, F. Combining Density Functional Theory (DFT) and Collision Cross-Section (CCS) Calculations to Analyze the Gas-Phase Behaviour of Small Molecules and Their Protonation Site Isomers. *Analyst* **2016**, *141* (13), 4044–4054. https://doi.org/10.1039/c5an02456k.

(21) Gonzales, G. B.; Smagghe, G.; Coelus, S.; Adriaenssens, D.; De Winter, K.; Desmet, T.; Raes, K.; Van Camp, J. Collision Cross Section Prediction of Deprotonated Phenolics in a Travelling-Wave Ion Mobility Spectrometer Using Molecular Descriptors and Chemometrics. *Anal. Chim. Acta* **2016**, *924*, 68–76. https://doi.org/10.1016/j.aca.2016.04.020.

(22) Lapthorn, C.; Pullen, F. S.; Chowdhry, B. Z.; Wright, P.; Perkins, G. L.; Heredia, Y. How Useful Is Molecular Modelling in Combination with Ion Mobility Mass Spectrometry for "small Molecule" Ion Mobility Collision Cross-Sections? *Analyst* **2015**, *140* (20), 6814–6823. https://doi.org/10.1039/c5an00411j.

(23) Weitzel, K.; Chemie, F.; Rev, M. S.; Introduction, I.; Reference, C. Bond-Dissociation Energies of Cations — Pushing The. *WHO Libr. Cat. Data* **2011**, *32* (i), 221–235. https://doi.org/10.1002/mas.

(24) Singh, U. C.; Kollman, P. A. An Approach to Computing Electrostatic Charges for Molecules. *J. Comput. Chem.* **1984**, *5* (2), 129–145. https://doi.org/10.1002/jcc.540050204.

(25) Besler, B. H.; Merz, K. M.; Kollman, P. A. Atomic Charges Derived from Semiempirical Methods. *J. Comput. Chem.* **1990**, *11* (4), 431–439. https://doi.org/10.1002/jcc.540110404.

(26) Breneman, C. M.; Wiberg, K. B. Determining Atom-centered Monopoles from Molecular Electrostatic Potentials. The Need for High Sampling Density in Formamide Conformational Analysis. *J. Comput. Chem.* **1990**, *11* (3), 361–373. https://doi.org/10.1002/jcc.540110311.
(27) Canzani, D.; Laszlo, K. J.; Bush, M. F. Ion Mobility of Proteins in Nitrogen Gas: Effects of Charge State, Charge Distribution, and Structure. *J. Phys. Chem. A* **2018**, *122* (25), 5625–5634. https://doi.org/10.1021/acs.jpca.8b04474.
(28) Wales, D. *Energy Landscapes*; Cambridge University Press, 2004. https://doi.org/10.1017/cbo9780511721724.
(29) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953**, *21* (6), 1087–1092. https://doi.org/10.1063/1.1699114.
(30) Zhou, C.; Ieritano, C.; Hopkins, W. S. Augmenting Basin-Hopping With Techniques From Unsupervised Machine Learning: Applications in Spectroscopy and Ion Mobility. *Front. Chem.* **2019**, *7*, 519. https://doi.org/10.3389/fchem.2019.00519.
(31) Porezag, D.; Frauenheim, T.; Köhler, T.; Seifert, G.; Kaschner, R. Construction of Tight-Binding-like Potentials on the Basis of Density-Functional Theory: Application to Carbon. *Phys. Rev. B* **1995**, *51* (19), 12947–12957. https://doi.org/10.1103/PhysRevB.51.12947.
(32) Elstner, M.; Porezag, D.; Jungnickel, G.; Elsner, J.; Haugk, M.; Frauenheim, T. Self-Consistent-Charge Density-Functional Tight-Binding Method for Simulations of Complex Materials Properties. *Phys. Rev. B - Condens. Matter Mater. Phys.* **1998**, *58* (11), 7260–7268. https://doi.org/10.1103/PhysRevB.58.7260.
(33) Throssel, K.; Frisch, M. J. Evaluation and Improvement of Semiempirical Methods I: PM7R8: A Variant of PM7 with Numerically Stable Hydrogen Bonding Corrections. *Manuscr. Prep.*
(34) Roothaan, C. C. J. New Developments in Molecular Orbital Theory. *Rev. Mod. Phys.* **1951**, *23* (2), 69–89. https://doi.org/10.1103/RevModPhys.23.69.
(35) Ieritano, C.; Crouse, J.; Campbell, J. L.; Hopkins, W. S. A Parallelized Molecular Collision Cross Section Package with Optimized Accuracy and Efficiency. *Analyst* **2019**, *144* (5), 1660–1670. https://doi.org/10.1039/c8an02150c.
(36) Stewart, J. J. P. Optimization of Parameters for Semiempirical Methods VI: More Modifications to the NDDO Approximations and Re-Optimization of Parameters. *J. Mol. Model.* **2013**, *19* (1), 1–32. https://doi.org/10.1007/s00894-012-1667-x.
(37) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; V. Barone, B.; Mennucci, G.; Petersson, A. Gaussian 09. Gaussian Incorporated: Wallingford, CT 2009.
(38) Becke, A. D. Density-Functional Thermochemistry. III. The Role of Exact Exchange. *J. Chem. Phys.* **1993**, *98* (7), 5648–5652. https://doi.org/10.1063/1.464913.
(39) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron Density. *Phys. Rev. B* **1988**, *37* (2), 785–789. https://doi.org/10.1103/PhysRevB.37.785.
(40) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. A Consistent and Accurate Ab Initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H-Pu. *J. Chem. Phys.* **2010**, *132* (15), 154104. https://doi.org/10.1063/1.3382344.
(41) Chai, J. Da; Head-Gordon, M. Long-Range Corrected Hybrid Density Functionals with Damped Atom-Atom Dispersion Corrections. *Phys. Chem. Chem. Phys.* **2008**, *10* (44), 6615–6620. https://doi.org/10.1039/b810189b.

(42) Zhao, Y.; Truhlar, D. G. The M06 Suite of Density Functionals for Main Group Thermochemistry, Thermochemical Kinetics, Noncovalent Interactions, Excited States, and Transition Elements: Two New Functionals and Systematic Testing of Four M06 Functionals and 12 Other Functionals (T. *Theor. Chem. Acc.* **2008**, *119* (5–6), 525. https://doi.org/10.1007/s00214-007-0401-8.
(43) Weigend, F.; Ahlrichs, R. Balanced Basis Sets of Split Valence, Triple Zeta Valence and Quadruple Zeta Valence Quality for H to Rn: Design and Assessment of Accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7* (18), 3297–3305. https://doi.org/10.1039/b508541a.
(44) Weigend, F. Accurate Coulomb-Fitting Basis Sets for H to Rn. *Phys. Chem. Chem. Phys.* **2006**, *8* (9), 1057–1065. https://doi.org/10.1039/b515623h.