University of Groningen

High performance collision cross section calculation-HPCCS

Zanotto, Leandro; Heerdt, Gabriel; Souza, Paulo C T; Araujo, Guido; Skaf, Munir S

Published in:
Journal of Computational Chemistry

DOI:
[10.1002/jcc.25199](https://doi.org/10.1002/jcc.25199)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2018

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Zanotto, L., Heerdt, G., Souza, P. C. T., Araujo, G., & Skaf, M. S. (2018). High performance collision cross section calculation-HPCCS. *Journal of Computational Chemistry*, 39(21), 1675-1681.
<https://doi.org/10.1002/jcc.25199>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

High Performance Collision Cross Section Calculation—HPCCS

Leandro Zanotto,^[a] Gabriel Heerdt ^[a] Paulo C. T. Souza,^[a,b] Guido Araujo,^[c] and Munir S. Skaf ^[a]

Since the commercial introduction of Ion Mobility coupled with Mass Spectrometry (IM-MS) devices in 2003, a large number of research laboratories have embraced the technique. IM-MS is a fairly rapid experiment used as a molecular separation tool and to obtain structural information. The interpretation of IM-MS data is still challenging and relies heavily on theoretical calculations of the molecule's collision cross section (CCS) against a buffer gas. Here, a new software (HPCCS) is presented, which performs CCS calculations using high performance computing techniques. Based on the trajectory

method, HPCCS can accurately calculate CCS for a great variety of molecules, ranging from small organic molecules to large protein complexes, using helium or nitrogen as buffer gas with considerable gains in computer time compared to publicly available codes under the same level of theory. HPCCS is available as free software under the Academic Use License at <https://github.com/cepid-cces/hpccs>. © 2018 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.25199

Introduction

Ion mobility coupled to mass spectrometry (IM-MS) is an analytical technique that separates gas-phase molecular ions at flight time based on their charge, size and shape. The technique has long been used for the detection of dangerous substances and to evidence the illegal use of chemical agents.^[1–6] The past 20 years have witnessed an exponential growth in the use of IM-MS experiments, especially since the advent of commercial ion-mobility devices in 2003.^[7–11]

Unlike many other biophysical techniques that provide structural information, such as X-ray crystallography and NRM spectroscopy, IM-MS is a fairly rapid experiment and requires no previous purification or crystallization of the target compound.^[7] However, the interpretation of IM-MS data is still a challenge and depends on accurate theoretical estimates of the molecular ion collision cross section (CCS) against a buffer gas in a low pressure drift chamber.^[7–12] For any given molecular geometry, the CCS of the target molecular ion can be accurately calculated, with an appropriate treatment of the collisions between the molecular ion and the buffer gas, for subsequent comparison with experimental values.

The most widely used program for theoretical CCS calculations is *Mobcal*,^[13,14] which was developed in 1996 and has been applied in a diversity of studies.^[15–20] This open source software computes the rotationally averaged cross sections from input atomic coordinate files of the target molecular ion - obtained from X-ray scattering, NMR data, quantum chemical calculations or molecular dynamics (MD) simulations - based on three different treatments of the ion-buffer gas collisions: the projection approximation (PA),^[21] the exact hard sphere scattering (EHSS),^[14,22] and the trajectory method (TM).^[14] EHSS and TM methods calculate CCS considering only helium as buffer gas. In 2011, Bush and collaborators

expanded the applicability of *Mobcal* TM method to nitrogen gas.^[23] TM is by far the most accurate method, being the best choice for CCS estimates for highly charged macromolecules, such as proteins and protein complexes.^[7] Currently, *Mobcal* is the only publicly available package capable of computing CCS for different types of molecular ions within the TM approximation.

Since *Mobcal*, efforts have been dedicated to design new algorithms capable of calculating CCS for biomacromolecular systems more rapidly and computationally less costly. In 2011, Bleiholder and co-workers presented the projected superposition approximation (PSA)^[24–27] algorithm to compute CCS for proteins, which is based on the projection approximation modified to account for collective size and shape effects. More recently, in 2015, Benesch and co-workers^[28] published the Ion Mobility Projection Approximation Calculation Tool (IMPACT), which provides CCS estimates within the PA framework at low computational costs. Both PSA and PA methods allow CCS calculations for systems containing a large number of atoms, but results can be highly inaccurate,^[7,24] by more than 30%, since, unlike TM, these methods neglect interactions between the

[a] L. Zanotto, G. Heerdt, P. C. T. Souza, M. S. Skaf
Institute of Chemistry and Center for Computational Engineering & Sciences, University of Campinas, Campinas, São Paulo 13083-852, Brazil
E-mail: skaf@iqm.unicamp.br

[b] P. C. T. Souza
Faculty of Mathematics and Natural Sciences, University of Groningen, Groningen 9747, AG, The Netherlands

[c] G. Araujo
Institute of Computing and Center for Computational Engineering & Sciences, University of Campinas, Campinas, São Paulo 13083-970, Brazil
Contract grant sponsor: Fundação de Amparo à Pesquisa do Estado de São Paulo; Contract grant numbers: 2012/24750-6, 2013/08293-7, 2016/04963-6

© 2018 Wiley Periodicals, Inc.

Figure 1. The accurate estimation of CCS requires the calculation of all possible collision angles between a buffer gas and a target molecule, as shown in A). Another approach to compute CCS of a molecule is the calculation of its projected average area taken over all possible orientations B). [Color figure can be viewed at wileyonlinelibrary.com]

gas and molecular ion, as well as the roughness of the molecular ion surface.

Despite its accuracy, theoretical computations of CCS for biomacromolecular systems under the TM approximation are inefficient with Mobcal because of its outdated program language and technology, thus limiting its usage to studies of small proteins and organic molecules. Here, we present a new software (HPCCS) capable of high performance CCS calculations for a large variety of molecular ions, ranging from small organic molecules to large protein complexes containing tens to hundreds of thousands of atoms. HPCCS is based on Mobcal,^[13,14] but focuses only on the Trajectory Method. Since the original Mobcal was written for sequential execution, it does not use the current parallel processing features we have in modern computers. To achieve that, the software was totally rewritten and improved with new features that enable application speedups using High Performance Computing (HPC). Large biomolecules can now be simulated with HPCCS within much shorter times and higher accuracy, which helps the interpretation of experimental IM-MS data.

HPCCS Code Overview

Trajectory method

To obtain structural information from ion-mobility measurements, cross sections can be calculated for conformers obtained from experiments or generated by molecular modeling methods and compared with experimental CCS results. The collision cross section gives an orientationally averaged result and the mobility, K , of a molecular ion in a low pressure buffer gas within the linear response regime can be calculated from:^[14,16]

$$K = \frac{\sqrt{18\pi}}{16} \left[\frac{1}{m} + \frac{1}{m_B} \right]^{1/2} \frac{ze}{(k_B T)^{1/2}} \frac{1}{\Omega_{\text{avg}}^{(1,1)}} \frac{1}{N} \quad (1)$$

where m is the molecular ion mass, m_B and N are the buffer gas mass and density, respectively, ze is the ionic charge and

$\Omega_{\text{avg}}^{(1,1)}$ is the orientationally averaged collision integral [eq. (2)]. Since the $\Omega_{\text{avg}}^{(1,1)}$ integral is related to the scattering angles χ , Figure 1A, the orientationally averaged collision cross section can be accurately determined by integrating over all possible collision geometries:^[14,16]

$$\Omega_{\text{avg}}^{(1,1)} = \frac{1}{8\pi^2} \int_0^{2\pi} d\theta \int_0^\pi d\phi \sin\phi \int_0^{2\pi} d\gamma \frac{\pi}{8} \left(\frac{\mu}{k_B T} \right)^3 \int_0^\infty dg e^{-\frac{\mu g^2}{2k_B T}} g^5 \times \int_0^\infty db 2b(1 - \cos\chi(\theta, \phi, \gamma, g, b)) \quad (2)$$

where μ is the reduced mass, g is the magnitude of the relative collision velocity, and b the impact parameter for all possible ion orientations defined by the three Euler angles θ , ϕ , and γ . Equation (2) can be calculated by numerical integration of the trajectories of buffer gas particles toward the molecular ion propagated in an intermolecular interaction potential V .^[23]

$$V(\theta, \phi, \gamma, b, r) = 4 \sum_i^n \epsilon_i \left[\left(\frac{\sigma_i}{r_i} \right)^{12} - \left(\frac{\sigma_i}{r_i} \right)^6 \right] - \frac{\alpha}{2} \left(\frac{ze}{n} \right)^2 \left[\left(\sum_i^n \frac{x_i}{r_i^3} \right)^2 + \left(\sum_i^n \frac{y_i}{r_i^3} \right)^2 + \left(\sum_i^n \frac{z_i}{r_i^3} \right)^2 \right] \quad (3)$$

The first term is a sum of Lennard-Jones (6–12) potential between the buffer gas and individual atom i of the molecular ion. The Lennard-Jones parameters, ϵ (depth) and σ (distance) are provided in the literature.^[13,14] The second term is the charge-induced dipole interaction with α being the gas polarizability and x_i , y_i , z_i and r_i the relative position of each individual atom i . For properly compute the interactions between ions and N_2 gas, these additional terms are required in eq. (3): an ion-quadrupole interaction and the orientation of the linear N_2 molecule during the collision process.^[23] This approach to the CCS calculation is called the Trajectory Method (TM) and provides an accurate prediction for the cross section of a

Table 1. Experimental and theoretical collision cross section given in Å² for a range of small molecules (group 1), considering He as the buffer gas.

Molecule	Exp. ^[a]	HPCCS (%error) ^[b]	Molecule	Exp. ^[a]	HPCCS (%error) ^[b]
N-ethylaniline	71.90	8.7	Ondansetron	105.73	0.3
Acetaminophen	74.04	4.7	Trichlormethiazide	110.06	14.2
Ephedrine	78.48	2.4	Clozapine N-oxide	114.31	3.6
Phenacetin	80.21	3.3	Sulpiride	116.28	3.4
Pregabalin	80.27	5.7	Cortisone	117.39	1.3
Naproxen	90.26	4.6	Amlodipine	121.91	2.9
Alprenolol	97.25	3.2	Doxazosin	134.85	14.0
MPPH ^[c]	100.54	2.1	Verapamil	140.95	8.0
Cinchonine	101.92	6.8	Sildenafil	141.60	3.7
Trimethoprim	105.63	3.8	Reserpine	179.95	9.6
			Average %error		5.3 ± 3

[a] Experimental results reported by Laphorn et al.^[39] [b] Percentage errors relative to the experimental CCS value. [c] 5-(p-Methylphenyl)-5-phenylhydantoin.

given candidate geometry, although it is computationally very expensive.^[7,14,24]

Development and code refactoring

Mobcal was developed in 1996 using the Fortran 77 language, a computer language available at that time, especially suited to numeric computation and extensively used on scientific community. The program was developed as a single file sequential code to be executed by a single processor core, the only desktop technology available at that time. Therefore, CCS calculations for proteins and protein complexes under the TM approximation become computationally expensive, given that at least 50,000 gas-target individual collision events are needed to accurately compute the CCS for such systems.

In the last decades, several features were added to the processors, like vector units and multi-cores, which enable to run tasks in parallel.^[29] The HPCCS software is an efficient parallel code, which takes advantage of the current computing technology to allow users calculate CCS for large molecules, thus providing an excellent tool to extract molecular structural information. HPCCS is a parallelized software that uses a refactored Mobcal as a basis. To have a more robust, cleaner and easier to support package, the original code was revisited and rewritten in C/C++. The main refactoring features are as follows:

- The original code, presented as a single unstructured file, was modularized, splitting the functions into files to facilitate support and debugging.
- Constant and global variables were declared on separate files for better support.
- All variables were declared as specific data types. In practice this allocates only the necessary amount of memory for each variable, thus reducing the memory footprint. In addition, unused variables were removed.
- The arrays were allocated dynamically to use only the necessary amount of memory.
- When possible, loops were fused for faster execution.
- The Mersenne Twister^[30] replaced the RANLUX^[31] random number generator.

- The input file was changed to use PQR, a protein structure file format with atomic charges and radii parameters frequently used in molecular dynamics simulations; this file format is more user-friendly compared to Mobcal's mff format.

Parallelization and optimization

Approximately 95% of the CPU time in CCS TM calculations is dedicated to computing the intermolecular interaction potential function. Motivated by this observation and considering that the trajectories are independent, we developed a parallel code for such computations. In Mobcal there is a triple loop that runs the simulations and calculates the scattering angles using the Monte Carlo method.^[32] In HPCCS, this triple loop was collapsed into a single loop using loop fusion.^[33] Variables and arrays were privatized, so trajectory variables have their own memory positions which are not shared with other trajectory variables, turning the parallelization possible and easy to apply.

The OpenMP^[34] library was used for parallelization: a powerful parallelization library that readily enables sequential code

Figure 2. Parallelization speedup for proteins in group 2, considering 1 and 16 threads. The Mobcal execution time was used as reference, dashed line, for speedup calculation. [Color figure can be viewed at wileyonlinelibrary.com]

Table 2. Experimental and theoretical collision cross section given in Å² for a range of proteins (group 2), considering helium as the buffer gas.

Protein information				Collision cross section (Å ²)			
Protein name	PDB ID	MW (Da)	Charge	Exp. ^[a]	Mobcal ^[b] (%error) ^[c]	IMPACT ^[d] (%error) ^[c]	HPCCS (%error) ^[c]
Melittin	1MLT	2846	+3	544	627 (15.3)	522 (4.0)	622 (14.3)
Bovine pancreatic trypsin inhibitor	6PTI	6512	+4	770	922 (19.7)	750 (2.6)	917 (19.1)
Ubiquitin	1UBQ	8565	+4	1059	1054 (0.5)	855 (19.3)	1048 (1.0)
Cytochrome c	1HRC	12355	+3	1139	1323 (16.2)	1039 (8.8)	1328 (16.6)
			+5	1238	1324 (7.0)	1047 (15.4)	1328 (7.3)
Alpha-lactalbumin	1HFX	14178	+6	1342	1539 (14.7)	1219 (9.2)	1539 (14.7)
Lysozyme	1DPX	14305	+5	1313	1484 (13.0)	1190 (9.4)	1485 (13.1)
			+6	1333	1485 (11.4)	1193 (10.5)	1488 (11.6)
			+8	1487	1491 (0.3)	1187 (20.2)	1493 (0.4)
Apo-Calmodulin	1CFD	16700	+7	1900	2112 (11.2)	1688 (11.2)	2082 (9.6)
Apo-myoglobin	1VXG	17566	+8	1742	1738 (0.2)	1353 (22.3)	1721 (1.2)
Haemoglobin tetramer [α ₂ β ₂ H ₄]	1GZX	64447	+13	3051	2844 (6.8)	3200 (4.9)	2894 (5.2)
Average %error					9.7 ± 6	11.5 ± 6	9.5 ± 6

[a] Experimental results reported by Jurneczko and Barran et al.^[11] [b] Theoretical values calculated at the TM level using Mobcal.^[11] [c] Percentage errors computed relative to the experimental CCS values. [d] Theoretical values calculated with IMPACT, based on the PA algorithm.

annotation by means of program `pragmas`.^[29] The Intel `Vtune`^[35] profiler tool was used to find the most time-consuming parts of the code, pointing to the possible candidates for parallelization.

To further improve HPCCS performance, arrays were memory aligned, enabling vectorization of the code^[33,36] based on the SIMD execution model,^[29] which uses processor vector units to simultaneously compute the same instruction over a set of array elements. This approach turned out most effective for speeding up potential function calculations since all threads were allocated to run the collisions in parallel.

Results and Discussion

The test sets presented in this paper are divided into three groups: group 1 - small organic molecular ions, group 2 - isolated proteins, and group 3 - proteins complexes. All the calculations were executed on a cluster node consisting of Dual Intel Xeon E5 processor with 8 cores each and AVX2, running at 2.4 GHz. The turbo boost was disabled to have deterministic processor frequency, without any change during the experiments.^[37] The Intel Parallel Studio 2016^[35] compiler was used, which has OpenMP support. `Vtune` was used to profile HPCCS along the entire program development, until the most time-consuming parts of the code were identified. Using only physical cores revealed good scalability and efficiency. As found in many applications, `Hyperthread`^[38] was used to double the number of available threads, but speedups did not scale well.

Small organic molecules

Since ion-mobility coupled with mass spectrometry has long been used for the detection of dangerous substances, drugs and explosives, and to evidence the illegal use of chemical agents,^[1–6] we carried out CCSs calculations for a range of 20 small organic molecules using He as buffer gas. A comparison

between CCS experimental data and theoretical results obtained from HPCCS is presented Table 1, showing excellent agreement.

The Density Functional Theory (DFT), at M06/6-311++G(2d,p):M06-2X/6-31+G(2d,p) level,^[40] was applied on the study of the small molecules presented in Table 1.^[41–43] The Mulliken charges, derived from these calculations, were used to assign the individual atomic charges. For small molecules, the average standard deviation is only approximately 5%.

Proteins

The description of long-range interactions, through eq. (3), allows us to obtain accurate collision cross sections for any system. Since the interactions must be calculated over all atoms, the computational time for each collision experiment

Figure 3. Total time for serial execution, in minutes, for proteins in group 2, considering the HPCCS (1 thread) and Mobcal programs. The number of atoms for each protein is indicated. [Color figure can be viewed at www.onlinelibrary.com]

Figure 4. Mobcal, TM algorithm, versus HPCCS, CCS in Å², showing the excellent agreement for group 2 proteins. [Color figure can be viewed at wileyonlinelibrary.com]

increases linearly with the number of atoms. Therefore, the software parallelization capability is an important feature of HPCCS, which enables handling of large biomolecules within an attractive computational cost. Figure 2 shows the parallelization speedup for single proteins (group 2; further details are presented in Table 2).

According to Figure 2, there is no significant difference in speedup for proteins in group 2. Considering only optimizations applied to the sequential code, running with 1 thread, the average speedup was 3.2, which confirms the efficiency of our software even for serial execution. When 16 threads were used, the average speedup was 48.1, showing that HPCCS accelerates considerably CCS calculation. More details about the speedup and efficiency of HPCCS software can be found in the Supporting Information (SI).

Figure 3 shows the increase in the total execution time as it varies with the protein size. For the largest protein in group 2, containing 4392 atoms, the total execution time was ~6 and ~20 h using HPCCS (1 thread) and Mobcal, respectively. The above execution times show that before HPCCS it was impractical to extensively use TM to perform CCS calculation. HPCCS allows the user to accurately obtain structural information in a feasible time for large biomacromolecules.

A quantitative assessment of the HPCCS results is shown in Table 2 for proteins in group 2 in comparison to available experimental data and theoretical results obtained from other codes.

As can be observed in Table 2, the percentage deviation average obtained from the HPCCS package is somewhat smaller than other code's. The IMPACT package uses the most approximate algorithm (PA), which adequately describes the selected proteins. Isolated proteins, without substrate or other proteins complexed, are generally easy to describe by the approximations mentioned here.^[11]

Since HPCCS is based on a refactoring of Mobcal, the results from both codes should converge to the same CCS values. Figure 4 shows CCS data for group 2 proteins computed with Mobcal versus HPCCS. The agreement is excellent, as expected. Small deviations are mainly due to the replacement of the random number generator, as aforementioned. Further comparisons between HPCCS and Mobcal can be found in Supporting Information Table S1.13.

Until now, CCS computations for larger biomolecular systems such as protein complexes were prohibitively costly at the TM level under Mobcal. Reasonable results at low computational costs are obtained at the PA level using IMPACT. However, in addition to helium, gases such as N₂ and CO₂ are now being used as buffers in IM-MS experiments, resulting in different values for the collision cross sections. Since the PA method cannot differentiate buffer gases, HPCCS results as an attractive option capable of handling different buffer gases at the TM level with excellent performance. A comparison between CCS experimental results and theoretical predictions for protein complexes (group 3) using He and N₂ is shown in Table 3.

It can be observed in Table 3 that IMPACT underestimates the experimental values, while the errors from HPCCS, in average, slightly overestimate the CCS values, specially for N₂ gas. Further improvements in the force field parameters and/or minimization protocol could improve even more the accuracy of HPCCS. The PA algorithm ignores the long-range interactions and all details of the scattering process between the molecular ion and buffer gas, which is determinant to the conferred systematic error.

In the experiments, the buffer gas is changed to separate the ions according to the interaction ion-gas. Commonly, N₂

Table 3. Experimental and theoretical collision cross section given in Å² for a range of proteins complexes (group 3) in a specific buffer gas (He and N₂).

Protein information					Collision cross section (Å ²)				
					Exp. ^[a]		IMPACT ^[b] (%error)	HPCCS (%error)	
Protein name	PDB	Chains	MW (kDa)	Charge	He	N ₂		He	N ₂
Beta-lactoglobulin	2AKQ	1	18	+7	1660	1950	1397 (15.8)	1784 (7.5)	2177 (11.6)
		2	37	+11	2850	3230	2368 (16.9)	3069 (7.7)	3593 (11.2)
Transthyretin	3BSZ	4	56	+14	3410	3840	2887 (15.3)	3751 (10.0)	4346 (13.2)
Concanavalin A	1TEI	4	103	+20	5550	6080	4687 (15.6)	5255 (5.3)	7220 (18.8)
Serum amyloid P	4AYU	5	125	+22	7030	7630	5630 (19.9)	7514 (6.9)	8593 (12.6)
Pyruvate kinase	1AQF	4	237	+31	10300	11100	8788 (14.7)	11660 (13.2)	12907 (16.3)
Serum amyloid P	4AVT	10	250	+31	10400	11100	8539 (17.9)	11550 (11.0)	12379 (11.5)
					Average %error		16.6 ± 2	8.8 ± 3	13.6 ± 3

[a] Experimental results reported Bush et al.^[44] [b] The PA algorithm do not consider the buffer gas, for comparison it was used the He experimental data.

Figure 5. Experimental versus computed CCS data using HPCCS, in Å², showing excellent agreement from small organic molecules to large protein complexes. [Color figure can be viewed at wileyonlinelibrary.com]

and CO₂ are used for this purpose, as they can present an ion-induced dipole moment. Table 3 shows that CCS using He is smaller than with N₂, a consequence of the strongest ion–N₂ interaction. The relative standard deviation, 13.6%, higher than the one obtained with He (8.8%), is also due to the larger error associated with the experimental data when N₂ is considered.^[44]

Taken together, the results presented here indicates that HPCCS is capable of reproducing very well the experimental CCS values for a variety of molecular types, ranging from small organic molecules to protein complexes. Figure 5 depicts the experimental versus computed CCS data for all structures evaluated in this work, showing an excellent agreement between them. The high accuracy of our calculations and the relatively low computational costs indicate that HPCCS can be a valuable tool to aid the interpretation of experimental CCS data in the study of biomolecular structures.

Concluding Remarks

IM-MS techniques have become highly valued as a tool for (bio)chemical analysis and can be profitably used for both analyte separation and structural investigation of a wide range of sample types. Despite its enormous potential, IM-MS data interpretation is often challenging in a variety of different scenarios. For instance, when multiple molecular conformers influence the resulting collision cross section, when the surface roughness of the target molecule affects CCS (i.e., dependence on the buffer gas) or when the target is a large macromolecular system like protein complexes. In these cases, reliable and fast CCS estimates are needed to help in the interpretation of IM-MS data.

In this work, we present a novel software which allows for High Performance Collision Cross Section (HPCCS) calculations. The software is based on the well-known trajectory method, the most accurate algorithm to date to compute CCS. We performed several tests to show that HPCCS can compute CCSs for small molecules and large proteins complexes accurately

within feasible times. For the largest protein complex considered here (~32,000 atoms), 16 computing threads resulted in a total execution time of about 5 h, which was hitherto unimaginable at the TM level of description. To the best of our knowledge, HPCCS is the first open source parallel software that efficiently and accurately computes collision cross section for any set of atomic coordinates, being a promising tool to help interpreting IM-MS experimental data.

Associated Content

Supporting Information: Detailed information on HPCCS compiling, input file configuration, and running DFT calculations and energy minimization are described, as well as the speedup and efficiency of parallelization for single proteins in group 2.

Keywords: ion mobility • mass spectrometry • collision cross section • trajectory method • HPC

How to cite this article: L. Zanotto, G. Heerdt, P. C. T. Souza, G. Araujo, M. S. Skaf. *J. Comput. Chem.* **2018**, *39*, 1675–1681. DOI: 10.1002/jcc.25199

Additional Supporting Information may be found in the online version of this article.

- [1] J. R. Verkouteren, J. L. Staymates, *Forensic Sci. Int.* **2011**, *206*, 190.
- [2] C. Fuché, J. Deseille, *Actual. Chim.* **2010**, *342*, 91.
- [3] C. K. Hilton, C. A. Krueger, A. J. Midey, M. Osgood, J. Wu, C. Wu, *Int. J. Mass Spectrom.* **2010**, *298*, 64.
- [4] Z. Karpas Bull. *Isr. Chem. Soc.* **2009**, *24*, 26.
- [5] M. A. Mäkinen, O. A. Anttalainen, M. E. Sillanpää, *Anal. Chem.* **2010**, *82*, 9594.
- [6] L. V. Tose, N. A. Santos, R. R. T. Rodrigues, M. Murgu, A. F. Gomes, G. A. Vasconcelos, P. C. T. Souza, B. G. Vaz, W. Romão, *Int. J. Mass Spectrom.* **2017**, *418*, 112.
- [7] F. Lanucara, S. W. Holman, C. J. Gray, C. E. Eyers, *Nat. Chem.* **2014**, *6*, 281.
- [8] R. Cumeras, E. Figueras, C. E. Davis, J. I. Baumbach, I. Gràcia, *Analyst* **2015**, *140*, 1376.
- [9] J. C. May, J. A. McLean, *Anal. Chem.* **2015**, *87*, 1422.
- [10] A. B. Kanu, P. Dwivedi, M. Tam, L. Matz, H. H. Hill, Jr., *J. Mass Spectrom.* **2008**, *43*, 1.
- [11] E. Jurneczko, P. E. Barran, *Analyst* **2011**, *136*, 20.
- [12] V. D'Atri, M. Porrini, F. Rosu, V. Gabelica, *J. Mass Spectrom.* **2015**, *50*, 711.
- [13] A. A. Shvartsburg, M. F. Jarrold, *Chem. Phys. Lett.* **1996**, *261*, 86.
- [14] M. F. Mesleh, J. M. Hunter, A. A. Shvartsburg, G. C. Schatz, M. F. Jarrold, *J. Phys. Chem.* **1996**, *100*, 16082; Erratum, *J. Phys. Chem. A* **1997**, *101*, 968.
- [15] B. T. Ruotolo, J. L. P. Benesch, A. M. Sandercock, S. J. Hyung, C. V. Robinson, *Nat. Protoc.* **2008**, *3*, 1139.
- [16] A. A. Shvartsburg, G. C. Schatz, M. F. Jarrold, *J. Chem. Phys.* **1998**, *108*, 2416.
- [17] J. Freeke, C. V. Robinson, B. T. Ruotolo, *Int. J. Mass Spectrom.* **2010**, *298*, 91.
- [18] M. Fasciotti, P. M. Lalli, G. Heerdt, R. A. Steffen, Y. E. Corilo, G. F. de Sá, R. J. Daroda, F. A. M. Reis, N. H. Morgon, R. C. L. Pereira, M. N. Eberlin, C. F. Klitzke, *Int. J. Ion Mobil. Spectrom.* **2013**, *16*, 117.
- [19] R. S. Galaverna, G. A. Bataglion, G. Heerdt, G. F. de Sa, R. Daroda, V. S. Cunha, N. H. Morgon, M. N. Eberlin, *Eur. J. Org. Chem.* **2015**, *2015*, 2189.

- [20] G. A. Bataglion, G. H. M. F. Souza, G. Heerdt, N. H. Morgon, J. D. L. Dutra, R. O. Freire, M. N. Eberlin, A. Tata, *J. Mass Spectrom.* **2015**, *50*, 336.
- [21] T. Wyttenbach, G. von Helden, J. J. Batka, D. Carlat, M. T. Bowers, *J. Am. Soc. Mass Spectrom.* **1997**, *8*, 275.
- [22] A. A. Shvartsburg, R. D. Smith, *J. Am. Soc. Mass Spectrom.* **2008**, *19*, 1286.
- [23] I. Campuzano, M. F. Bush, C. V. Robinson, C. Beaumont, K. Richardson, H. Kim, H. I. Kim, *Anal. Chem.* **2012**, *84*, 1026.
- [24] C. Bleiholder, T. Wyttenbach, M. T. Bowers, *Int. J. Mass Spectrom.* **2011**, *308*, 1.
- [25] C. Bleiholder, S. Contreras, T. D. Do, M. T. Bowers, *Int. J. Mass Spectrom.* **2013**, *345–347*, 89.
- [26] C. Bleiholder, S. Contreras, M. T. Bowers, *Int. J. Mass Spectrom.* **2013**, *354–355*, 275.
- [27] T. Wyttenbach, C. Bleiholder, M. T. Bowers, *Anal. Chem.* **2013**, *85*, 2191.
- [28] E. G. Marklund, M. T. Degiacomi, C. V. Robinson, A. J. Baldwin, J. L. P. Benesch, *Structure* **2015**, *23*, 791.
- [29] P. Pacheco, *An Introduction to Parallel Programming*, 1st ed.; San Francisco, CA, USA, Morgan Kaufmann: MA, **2010**.
- [30] M. Matsumoto, T. Nishimura, *ACM Trans. Model. Comput. Simul.* **1998**, *8*, 3.
- [31] F. James, *Comput. Phys. Commun.* **1994**, *79*, 111.
- [32] R. Y. Rubinstein, D. P. Kroese, *Simulation and the Monte Carlo Method*, 2nd ed.; John Wiley & Sons, New York: **2007**.
- [33] M. Wolfe, *High Performance Compilers for Parallel Computing*; Addison-Wesley Publishing Co., New York, **1996**.
- [34] L. Dagum, R. Menon, *Comput. Sci. Eng.* **1998**, *5*, 46.
- [35] A. Marowka, International Conference on Parallel Computing Technologies PaCT 2011: Parallel Computing Technologies, Vol. 6873, **2011**; pp. 317–331.
- [36] J. Jeffers, J. Reinders, Intel Xeon Phi Coprocessor High Performance Programming; Morgan Kaufmann: Waltham, MA, **2013**.
- [37] E. Krieger, G. Vriend, *J. Comput. Chem.* **2015**, *36*, 996.
- [38] D. T. Marr, F. Binns, D. L. Hill, G. Hinton, D. A. Koufaty, J. A. Miller, M. Upton, *Intel Technol. J.* **2002**, *6*, 1.
- [39] C. Laphorn, T. J. Dines, B. Z. Chowdhry, G. L. Perkins, F. S. Pullen, *Rapid Commun. Mass Spectrom.* **2013**, *27*, 2399.
- [40] Y. Zhao, D. G. Truhlar, *Theor. Chem. Acc.* **2008**, *120*, 215.
- [41] G. Heerdt, N. H. Morgon, *Quim. Nova* **2011**, *34*, 868.
- [42] G. Heerdt, N. H. Morgon, *J. Braz. Chem. Soc.* **2012**, *23*, 1741.
- [43] G. Heerdt, D. H. Pereira, R. Custodio, N. H. Morgon, *Comput. Theor. Chem.* **2015**, *1067*, 84.
- [44] M. F. Bush, Z. Hall, K. Giles, J. Hoyes, C. V. Robinson, B. T. Ruotolo, *Anal. Chem.* **2010**, *82*, 9557.

Received: 13 December 2017

Revised: 31 January 2018

Accepted: 7 February 2018

Published online on 2 March 2018