PsiRESP: calculating RESP charges with Psi4

Lily Wang[1] **and Megan L. O'Mara**[1]¶

1 Research School of Chemistry, College of Science, Australian National University, Canberra, ACT, 2601, Australia **true** ¶ Corresponding author

Summary

Molecular dynamics (MD) simulations study the movement of particles over time, and have become a fundamental tool in biomedical and materials research. The accuracy of a MD simulation depends on its ability to model the physics of the chosen system, a capacity that arises from the parameters used to model the interactions between atoms (also known as a "force field"). The functional form of most force fields is an approximation of real word physics to allow for tractable simulation; one common simplification is to model the dynamic distribution of electrons around an atom as single, static, fixed partial charge. Electrostatic interactions between non-bonded atoms can then be simplified to pairwise interactions between the partial charges.

PsiRESP is a Python package that uses the Psi4 quantum chemistry engine to calculate atomic partial charges. It supports multiple methods, each based on the electrostatic potential experienced at particular grid points around the molecule. These partial charges can then be used in MD simulations.

Background

A number of methods have been developed to generate molecular partial charges, and results vary widely between each method. Approaches based on quantum mechanics (QM) calculations are particularly advantageous for novel compounds, as no knowledge beyond the molecular structure is required. One common approach is to find per-atom charges that reproduce the electrostatic potential outside the molecular surface. Methods following this design include the simply named "ESP" model, the restrained ESP ("RESP") model, and the "RESP2" model.

ESP

The electrostatic potential V is the potential generated by the nuclei and electrons in a molecule. At any point in space r, it is:

$$V(\vec{r}) = \sum_{n=1}^{nuclei} \frac{Z_n}{|\vec{r} - \vec{R}_n|} - \int \frac{\rho(\vec{r'})}{|\vec{r} - \vec{r'}|} d\vec{r}$$

The "ESP" method published by Singh and Kollman in 1984 (Singh & Kollman, 1984) evaluates the electrostatic potential on a grid of points around the molecule. This method canonically uses the HF/6-31G* level of theory in the gas phase. The approach is still used by the Automated Topology Builder (Malde et al., 2011) to derive charges for GROMOS, although the ATB method uses B3LYP/6-31G* in implicit solvent.

One flaw of the ESP method is that it is difficult to distinguish the contribution of buried atoms within the molecule to the electrostatic potential at the surface. As a result, the charges assigned to the buried atoms can vary substantially with minor changes in geometry.

RESP

The restrained electrostatic potential (RESP) ([Bayly et al., 1993](#); [Cieplak et al., 1995](#); [Cornell et al., 1993](#)) approach introduces a hyperbolic restraint towards 0, penalizing charges with high magnitudes. This is intended to reduce overfitting and hence the variation between charges, resulting in charges that are more easily transferable between molecules and less conformationally dependent. The function follows the form:

$$\chi_{penalty} = a \sum_{n=1}^{nuclei} ((q_n^2 + b^2)^{1/2} - b)$$

a defines the asymptotic limits of the penalty, while b controls its width, or its degree of curvature at the minimum. b is typically set to 0.1 e. In a two-stage fit, a is usually tightened (increased) in the second stage. The standard values are 0.0005 au for the first stage, and 0.001 for the second stage. Hydrogens are often excluded in the hyperbolic penalty. Two-stage RESP is the typical charge model employed by the AMBER and GLYCAM force fields.

As with the ESP method, RESP typically computes the electrostatic potential at HF/6-31G* in the gas phase.

RESP2

The HF/6-31G* level of theory commonly used for RESP charge derivation overestimates the gas-phase polarity of molecules ([Carlson et al., 1993](#)). The recently published RESP2 ([Schauperl et al., 2020](#)) addresses this issue by using a higher level of QM theory (PW6B95/aug-cc-pV(D + d)Z), as well as combining charges fitted to both gas- and aqueous-phase data. While optimising charge parameters alone without modifying Lennard-Jones parameters may not increase the accuracy of simulations ([Mobley et al., 2007](#)), both parameters can be co-optimised for greater accuracy.

Existing tools

Several tools already exist for fitting ESP-based charges. The canonical program is resp ([Bayly et al., 1993](#)), written in FORTRAN and distributed by the AmberTools package ([D. A. Case & Kollman, 2020](#)). However, using this with multiple molecules is a tedious and manual process. As the charge distribution of a molecule is highly dependent on its conformation, it is often necessary to examine multiple conformers and orientations of a molecule to obtain a general charge distribution that can be used to derive atomic partial charges ([Reynolds et al., 1992](#)). In order to use the resp program, different conformers and orientations must be manually generated, individual QM jobs generated for each, and then data converted into the RESP format. Constraints are manually specified on multiple rows, which requires significant effort to input manually and is difficult to read and verify.

The online R.E.D. Tools (RESP and ESP charge Derive) ([F.-Y. Dupradeau et al., 2007](#); [F.-Y. Dupradeau et al., 2010](#)) provide a much more accessible interface to the resp tool. It supports quantum chemistry jobs using both the Gaussian ([Frisch et al., 2016](#)) and GAMESS ([Barca et al., 2020](#)) QM engines, as well as both CHELPG ([Breneman & Wiberg, 1990](#)) and Connolly ([Connolly, 1983](#)) surfaces. Users can furthermore specify intramolecular and intermolecular charge constraints and charge equivalence constraints. The R.E.D. Tools use the canonical resp program for the actual charge fitting. However, conformer geometries must be provided by the user, and re-orientations of each conformer manually specified. Molecules must be provided in a specialized P2N format, and it is strongly recommended that a human user check each file before submission. Moreover, while standalone tools can be downloaded and run, full functionality is only available for jobs submitted to the server.

Finally, a Python RESP plugin for Psi4 also exists ([Alenaizan et al., 2020](#)). Also called resp, the plugin implements the RESP scheme but does not allow for intermolecular constraints.

82 Intermolecular constraints are necessary for parametrising the atomic partial charges of multiple
83 molecules (e.g. connected residues in a polymer) in tandem, so this poses a significant limitation
84 for such goals. As with the canonical resp and the R.E.D. tools, all geometries must be specified
85 by the user.

Statement of Need

87 PsiRESP is a Python package that can be used to calculate ESP and RESP charges, as
88 well as the next-generation RESP2 scheme. Both intra-molecular and inter-molecular charge
89 constraints and charge equivalence constraints are supported. Users may generate their own
90 conformers of a molecule, or conformers can be automatically generated. Multiple orientations
91 can also be specified or automatically generated for each conformer, to ensure a generally
92 applicable distribution of charges. Multiple atomic radii sets are supported, including the Bondi
93 and Merz-Singh-Kollman sets of radii.

94 The automated conformer generation enables computing more transferrable charges with
95 fewer resources. Conformers are generated following the Electrostatically Least-interacting
96 Functional groups method (ELF). Here, a pool of conformers is ranked by the electrostatic
97 interaction between functional groups; the least-interacting conformers are selected; and then
98 a user-specified number of final conformers is chosen from the secondary pool to maximise
99 structural diversity (by the root mean square deviation of coordinates). This allows even a
100 small number of conformers to cover a breadth of structural range.

101 *Intra*-molecular charge constraints, or charge constraints that only apply to one molecule, are
102 useful for enforcing equivalent charges between symmetric groups. For example, these are used
103 to symmetrize the charges of hydrogens around a methyl or methylene group. They also allow
104 for computing charges for a molecule that is compatible with an existing environment. One
105 use case is calculating the charges of a non-canonical amino acid (ncAA) in a protein where
106 the charges of canonical residues are already known; here, it is general practice to constrain
107 the charges on the backbone of the ncAA to equal the known charges of the backbone in other
108 residues.

109 *Inter*-molecular charge constraints, or charge constraints that apply to multiple molecules, are
110 particularly useful for computing the charges of multiple components of a single macromolecule.
111 For example, a co-polymer can be comprised of multiple monomer species, A and B. In order
112 to calculate suitable charges for each monomer, their surrounding environment (the adjacent
113 monomers) must be taken into account. However, the same monomer may be multiple different
114 local environments: AAA, AAB, or BAB. Without charge constraints, the charges calculated
115 for monomer A may differ between different environments, which lowers the transferability of
116 the charge set for creating new polymers. Inter-molecular charge constraints can be applied
117 to constrain the charges of a particular monomer across all environments to be equivalent,
118 ensuring that the derived charges can be used transferrably.

119 The ability to serialize and deserialize a job or molecule to or from JSON also enables
120 easy transferrability between machines, as well as documentation of the parameters used to
121 generate charges. The multiple converters to other libraries (currently supported outputs
122 are an MDAnalysis universe (Gowers et al., 2016; Michaud-Agrawal et al., 2011) and an
123 OpenForceField toolkit-compatible (Wagner et al., 2021) RDKit (Landrum et al., 2020)
124 molecule) allow for further work, or additional options to write to different formats.

Functionality

126 PsiRESP uses QCElemental (Smith, Lolinco, et al., 2021) and RDKit to parse molecular input
127 and output, meaning that it supports a wide array of formats, including XYZ and SMILES.
128 A strict limitation is that elements must be given; bond connectivity must also be present,

129 or inferrable by proximity. Conformers are embedded and generated using RDKit. Quantum
130 chemistry calculations are run in Psi4 (Turney et al., 2012).

131 PsiRESP can be used interactively in a single job in connection with a QCFractal (Smith,
132 Altarawy, et al., 2021) server. Alternatively, in acknowledgement of possible limits on time or
133 other computing resources, users can use PsiRESP in two stages: first to generate input files
134 for Psi4 that users can run manually or in parallel job submissions, and secondly to read the
135 data computed from each file to finish the charge calculations.

136 The project contains a thorough Continuous Integration test suite ensuring that charges are
137 comparable with those derived from other similar packages and previous versions. Code is
138 written in a modular style, allowing for easy extension of capabilities, e.g. including charges
139 computed from electric fields in the future, or using other QM engines. Core dependencies
140 have been kept as minimal as possible to ease installation, although users are encouraged to
141 make use of other packages (e.g. MDAnalysis or the OpenForceField toolkit for additional I/O
142 and functionality). PsiRESP can be installed via `pip` and `conda`. Releases follow Semantic
143 Versioning 2.0.

Acknowledgements

References

155 Alenaizan, A., Burns, L. A., & Sherrill, C. D. (2020). Python implementation of the restrained
156   electrostatic potential charge model. *International Journal of Quantum Chemistry*, *120*(2),
157   e26035. https://doi.org/10.1002/qua.26035

158 Barca, G. M. J., Bertoni, C., Carrington, L., Datta, D., De Silva, N., Deustua, J. E., Fedorov,
159   D. G., Gour, J. R., Gunina, A. O., Guidez, E., Harville, T., Irle, S., Ivanic, J., Kowalski, K.,
160   Leang, S. S., Li, H., Li, W., Lutz, J. J., Magoulas, I., … Gordon, M. S. (2020). Recent
161   developments in the general atomic and molecular electronic structure system. *The Journal*
162   *of Chemical Physics*, *152*(15), 154102. https://doi.org/10.1063/5.0005188

163 Bayly, C. I., Cieplak, P., Cornell, W., & Kollman, P. A. (1993). A well-behaved electrostatic
164   potential based method using charge restraints for deriving atomic charges: The RESP
165   model. *The Journal of Physical Chemistry*, *97*(40), 10269–10280. https://doi.org/10.
166   1021/j100142a004

167 Breneman, C. M., & Wiberg, K. B. (1990). Determining atom-centered monopoles from
168   molecular electrostatic potentials. The need for high sampling density in formamide
169   conformational analysis. *Journal of Computational Chemistry*, *11*(3), 361–373. https:
170   //doi.org/10.1002/jcc.540110311

171 Carlson, H. A., Nguyen, T. B., Orozco, M., & Jorgensen, W. L. (1993). Accuracy of free
172   energies of hydration for organic molecules from 6-31g*-derived partial charges. *Journal of*
173   *Computational Chemistry*, *14*(10), 1240–1249. https://doi.org/10.1002/jcc.540141013

174 Cieplak, P., Cornell, W. D., Bayly, C., & Kollman, P. A. (1995). Application of the multimolecule
175 and multiconformational RESP methodology to biopolymers: Charge derivation for DNA,
176 RNA, and proteins. *Journal of Computational Chemistry*, *16*(11), 1357–1377. https:
177 //doi.org/10.1002/jcc.540161106

178 Connolly, M. L. (1983). Analytical molecular surface calculation. *Journal of Applied Crystal-*
179 *lography*, *16*(5), 548–558. https://doi.org/10.1107/S0021889883010985

180 Cornell, W. D., Cieplak, P., Bayly, C. I., & Kollman, P. A. (1993). Application of RESP
181 charges to calculate conformational energies, hydrogen bond energies, and free energies
182 of solvation. *Journal of the American Chemical Society*, *115*(21), 9620–9631. https:
183 //doi.org/10.1021/ja00074a030

184 D. A. Case, I. Y. B.-S., K. Belfon, & Kollman, P. A. (2020). *Amber 2020*. University of
185 California, San Francisco.

186 Dupradeau, F.-Y., Cezard, C., Lelong, R., Stanislawiak, E., Pecher, J., Delepine, J. C.,
187 & Cieplak, P. (2007). R.E.DD.B.: A database for RESP and ESP atomic charges,
188 and force field libraries. *Nucleic Acids Research*, *36*(Database), D360–D367. https:
189 //doi.org/10.1093/nar/gkm887

190 Dupradeau, F.-Y., Pigache, A., Zaffran, T., Savineau, C., Lelong, R., Grivel, N., Lelong, D.,
191 Rosanski, W., & Cieplak, P. (2010). The R.E.D. Tools: Advances in RESP and ESP charge
192 derivation and force field library building. *Physical Chemistry Chemical Physics*, *12*(28),
193 7821. https://doi.org/10.1039/c0cp00111b

194 Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R.,
195 Scalmani, G., Barone, V., Petersson, G. A., Nakatsuji, H., Li, X., Caricato, M., Marenich,
196 A. V., Bloino, J., Janesko, B. G., Gomperts, R., Mennucci, B., Hratchian, H. P., Ortiz, J.
197 V., … Fox, D. J. (2016). *Gaussian 16 Revision C.01*.

198 Gowers, R. J., Linke, M., Barnoud, J., Reddy, T. J. E., Melo, M. N., Seyler, S. L., Domański, J.,
199 Dotson, D. L., Buchoux, S., Kenney, I. M., & Beckstein, O. (2016). MDAnalysis: A Python
200 Package for the Rapid Analysis of Molecular Dynamics Simulations. *Proceedings of the 15th*
201 *Python in Science Conference*, 98–105. https://doi.org/10.25080/Majora-629e541a-00e

202 Landrum, G., Tosco, P., Kelley, B., sriniker, gedeck, Ric, Vianello, R., NadineSchneider, Dalke,
203 A., N, D., Cole, B., Kawashima, E., Turk, S., Swain, M., AlexanderSavelyev, Cosgrove,
204 D., Vaucher, A., Wójcikowski, M., Probst, D., … Jensen, J. H. (2020). *Rdkit/rdkit:*
205 *2020_09_1 (Q3 2020) release* (Version Release_2020_09_1) [Computer software]. Zenodo.
206 https://doi.org/10.5281/zenodo.4107869

207 Malde, A. K., Zuo, L., Breeze, M., Stroet, M., Poger, D., Nair, P. C., Oostenbrink, C., &
208 Mark, A. E. (2011). An Automated Force Field Topology Builder (ATB) and Repository:
209 Version 1.0. *Journal of Chemical Theory and Computation*, *7*(12), 4026–4037. https:
210 //doi.org/10.1021/ct200196m

211 Michaud-Agrawal, N., Denning, E. J., Woolf, T. B., & Beckstein, O. (2011). MDAnalysis:
212 A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational*
213 *Chemistry*, *32*(10), 2319–2327. https://doi.org/10.1002/jcc.21787

214 Mobley, D. L., Dumont, É., Chodera, J. D., & Dill, K. A. (2007). Comparison of Charge
215 Models for Fixed-Charge Force Fields: Small-Molecule Hydration Free Energies in Explicit
216 Solvent. *The Journal of Physical Chemistry B*, *111*(9), 2242–2254. https://doi.org/10.
217 1021/jp0667442

218 Reynolds, C. A., Essex, J. W., & Richards, W. G. (1992). Atomic charges for variable
219 molecular conformations. *Journal of the American Chemical Society*, *114*(23), 9075–9079.
220 https://doi.org/10.1021/ja00049a045

Schauperl, M., Nerenberg, P. S., Jang, H., Wang, L.-P., Bayly, C. I., Mobley, D. L., & Gilson, M. K. (2020). Non-bonded force field model with advanced restrained electrostatic potential charges (RESP2). *Communications Chemistry*, *3*(1), 1–11. https://doi.org/10.1038/s42004-020-0291-4

Singh, U. C., & Kollman, P. A. (1984). An approach to computing electrostatic charges for molecules. *Journal of Computational Chemistry*, *5*(2), 129–145. https://doi.org/10.1002/jcc.540050204

Smith, D. G. A., Altarawy, D., Burns, L. A., Welborn, M., Naden, L. N., Ward, L., Ellis, S., Pritchard, B. P., & Crawford, T. D. (2021). The MolSSI QCArchive project: An open-source platform to compute, organize, and share quantum chemistry data. *WIREs Computational Molecular Science*, *11*(2), e1491. https://doi.org/10.1002/wcms.1491

Smith, D. G. A., Lolinco, A. T., Glick, Z. L., Lee, J., Alenaizan, A., Barnes, T. A., Borca, C. H., Di Remigio, R., Dotson, D. L., Ehlert, S., Heide, A. G., Herbst, M. F., Hermann, J., Hicks, C. B., Horton, J. T., Hurtado, A. G., Kraus, P., Kruse, H., Lee, S. J. R., … Burns, L. A. (2021). Quantum chemistry common driver and databases (QCDB) and quantum chemistry engine (QCEngine): Automation and interoperability among computational chemistry programs. *The Journal of Chemical Physics*, *155*(20), 204801. https://doi.org/10.1063/5.0059356

Turney, J. M., Simmonett, A. C., Parrish, R. M., Hohenstein, E. G., Evangelista, F. A., Fermann, J. T., Mintz, B. J., Burns, L. A., Wilke, J. J., Abrams, M. L., Russ, N. J., Leininger, M. L., Janssen, C. L., Seidl, E. T., Allen, W. D., Schaefer, H. F., King, R. A., Valeev, E. F., Sherrill, C. D., & Crawford, T. D. (2012). Psi4: An open-source ab initio electronic structure program. *WIREs Computational Molecular Science*, *2*(4), 556–565. https://doi.org/10.1002/wcms.93

Wagner, J., Thompson, M., Mobley, D. L., Chodera, J., Bannan, C., Rizzi, A., trevorgokey, Dotson, D., Rodríguez-Guerra, J., Camila, Behara, P., Mitchell, J. A., Bayly, C., JoshHorton, Lim, N. M., Lim, V., Sasmal, S., Wang, L., Dalke, A., … Zhao, Y. (2021). *openforcefield/openff-toolkit: 0.10.1 Minor feature and bugfix release* (Version 0.10.1) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.5601736