




aiida-aimall: A Python package for automating workflows for AIMAll software

Kevin M. Lefrancois-Gagnon ^{1*} and Robert C. Mawhinney ^{1*}

¹ Lakehead University, Thunder Bay, ON, Canada * These authors contributed equally.

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

Summary

Since its introduction by Richard Bader, the Quantum Theory of Atoms in Molecules (QTAIM) has become a useful tool for computational chemists. This Python package provides plugins for a common QTAIM software, AIMAll, for the AiiDA Python infrastructure. aiida-aimall is an essential tool for ensuring reproducible calculations, with full generation history. Workflows are also provided to interface AIMAll software with any quantum chemistry package that can be run through the command line, so long as it generates the input files required by AIMAll.

Statement of need

aiida-aimall is a Python package based on the AiiDA ([Talirz et al., 2020](#)) infrastructure designed to assist users with generating inputs for AIMAll software ([Keith, 2019](#)). The goal of the AiiDA infrastructure are, in part, to ensure data provenance and calculation reproducibility. While aiida-aimall has been developed primarily for interface with Gaussian software outputs ([Frisch et al., 2016](#)), a versatile workflow enabling interface with other quantum chemistry packages is also made available.

Through a variety of workflows that can start with an .xyz file, AiiDA StructureData, or even with a SMILES string of a molecule, aiida-aimall provides a variety of use cases for automating complex workflows. Additionally, tools to ensure that computers are not overloaded through too many simultaneous processes are made available through classes of FromGroupSubmissionControllers from aiida-submission-controller to limit active processes.

Features

aiida-aimall contains many different classes from aiida tailored to ensure ease of use of AIMAll calculations. Numerous features provided by aiida-aimall are described in full on the [documentation webpage hosted on ReadTheDocs](#). A brief description of main features is provided here.

Running Simple AIMAll Calculations

The simplest functionality provided by aiida-aimall is running AIMAll calculations. All AIMAll calculations utilize the AimqbParameters datatype provided by aiida-aimall. The AimqbParameters datatype is a validator for AIMAll command line input. Command line parameters are to be provided as a dictionary, then AimqbParameters ensures that the parameters match options available for AIMAll software as [defined on the software website](#), and that the correct data type is provided for each parameter. In this way, AimqbParameters verifies the provided input to AIMAll calculations prior to launch of the calculation. These parameters,

38 along with SinglefileData of a valid AIMAll input file, a Code object for AIMAll software,
39 and relevant metadata are provided to an AimqbCalculation.

40 This functionality in itself is an overcomplication of the simple process of running the software
41 normally. However, it does have some benefits. The output is already extracted and stored in
42 the database in a readily useable manner through the use of the AimqbBaseParser. It is now
43 simple to see the history of the calculation.

44 Substituent Properties

45 Some of the workflows in aiida-aimall automate calculation of substituent properties from
46 AIMAll output. These substituent properties have been developed by the authors.(Lefrancois-
47 Gagnon & Mawhinney, 2023) The SubstituentParameterWorkChain does this automatically,
48 and any routine AIMAll calculation can make use of this by using the AimqbGroupParser,
49 which can be provided in metadata input to AimqbCalculation as an entry in the metadata
50 dictionary: metadata.options.parser_name: 'aimall.group'. A detailed description of the
51 calculated substituent properties is available in a tutorial in the documentation. AIMAll
52 integrated and graph properties are obtained.

53 Integrations with Computational Chemistry Software

54 aiida-aimall's main draw is that it enables automation to link the outputs of standard
55 computational chemistry software directly to an AIMAll calculation. A list of provided
56 workflows is shown in Table 1. The software with the most robust implementation is Gaussian
57 software,(Frisch et al., 2016) as Gaussian already has an implemented aiida package. Other
58 computational chemistry software like ORCA can be run through the QMToAIMWorkchain, which
59 uses aiida-shell to run software than can be run through the command line. If .molden or
60 .cp2k.out output formats are available, one could alternatively use these to generate the needed
61 .wfx files for AIMAll, and automatically run AIMAll through the GenerateWFXToAIMWorkchain.

62 Table 1: Main workflows provided by aiida-aimall, their aiida entry points that can be used
63 to load them by aiida.plugins.WorkflowFactory, and a brief description. These workflows
64 all end with the output of an AimqbCalculation as their main output.

Workflow	Entry Point	Purpose
QMToAIMWorkchain	aimall.qmtoaim	Run a general computational chemistry software and link it to an AIMAll calculation
GenerateWFXToAIMWorkchain	aimall.wfxtoaim	Take non-standard AIMAll input files, and run AIMAll
GaussianToAIMWorkChain	aimall.g16toaim	Run a Gaussian calculation and automatically run an AIMAll calculation on its outputs
SubstituentParameterWorkChain	aimall.subparam	Compute substituent properties defined by the authors automatically

65 Controllers to limit computer burden when running large numbers of jobs

66 The last main contribution of aiida-aimall is through the definition of controllers from
67 the aiida-submission-controller package. These controllers limit active processes and
68 can be used together as demonstrated in a tutorial notebook to automate the entire

69 SubstituentParameterWorkchain. These use a number of Workchains developed just
70 for their use in these controllers. The process flows as SmilesToGaussianController ->
71 AIMAllReorController -> GaussianController -> AIMAllController. The latter two con-
72 trollers can also be seen and used as general use controllers wrapping GaussianCalculations
73 and AimqbCalculations

74 Acknowledgements

75 We acknowledge the National Science and Engineering Research Council of Canada (grant
76 RGPIN-2022-05060), and the Northern Ontario Heritage Fund Corporation (grant 7401278)
77 for financial support.

78 References

- 79 Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R.,
80 Scalmani, G., Barone, V., Petersson, G. A., Nakatsuji, H., Li, X., Caricato, M., Marenich,
81 A. V., Bloino, J., Janesko, B. G., Gomperts, R., Mennucci, B., Hratchian, H. P., Ortiz, J.
82 V., ... Fox, D. J. (2016). *Gaussian~16 Revision C.01*.
- 83 Keith, T. A. (2019). *AIMAll* (Version 19.10.12). TK Gristmill Software. aim.tkgristmill.com
- 84 Lefrancois-Gagnon, K. M., & Mawhinney, R. C. (2023). Toward Universal Substituent
85 Constants: Relating QTAIM Functional Group Descriptors to Substituent Effect Proxies.
86 *Journal of Chemical Information and Modeling*, 63. [https://doi.org/10.1021/acs.jcim.](https://doi.org/10.1021/acs.jcim.3c00987)
87 [3c00987](https://doi.org/10.1021/acs.jcim.3c00987)
- 88 Talirz, L., Kahle, L., Häuselmann, R., Gresch, D., Müller, T., Yakutovich, A. V., Andersen,
89 C. W., Ramirez, F. F., Adorf, C. S., Gargiulo, F., Kumbhar, S., Passaro, E., Johnston,
90 C., Merkys, A., Cepellotti, A., Mounet, N., Marzari, N., Kozinsky, B., & Pizzi, G. (2020).
91 AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows
92 and data provenance . *Scientific Data*, 7. <https://doi.org/10.1038/s41597-020-00638-4>