

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Published: unpublished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 is, 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. To the best of the authors' knowledge, no tool for automatically linking the output of quantum chemistry packages with AIMAll while tracking data provenance exists. Before this tool, individuals would manually provide output files from one software to the other. This tool significantly simplifies complex multi-step workflows such as substituent parameter generation into one simple step, easing the burden on the end user and allowing for high-throughput calculations.

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

38 `AimqbParameters` datatype is a validator for AIMAll command line input. Command line para-
 39 meters are to be provided as a dictionary, then `AimqbParameters` ensures that the parameters
 40 match options available for AIMAll software as [defined on the software website](#), and that the
 41 correct data type is provided for each parameter. In this way, `AimqbParameters` verifies the
 42 provided input to AIMAll calculations prior to launch of the calculation. These parameters,
 43 along with `SinglefileData` of a valid AIMAll input file, a Code object for AIMAll software,
 44 and relevant metadata are provided to an `AimqbCalculation`.

45 This functionality in itself is an overcomplication of the simple process of running the software
 46 normally. However, it does have some benefits. The output is already extracted and stored in
 47 the database in a readily usable manner through the use of the `AimqbBaseParser`. It is now
 48 simple to see the history of the calculation.

49 Substituent Properties

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

58 Integrations with Computational Chemistry Software

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

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

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

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

71 The last main contribution of aiida-aimall is through the definition of controllers from
72 the aiida-submission-controller package. These controllers limit active processes and
73 can be used together as demonstrated in [a tutorial notebook](#) to automate the entire
74 SubstituentParameterWorkchain. These use a number of Workchains developed just
75 for their use in these controllers. The process flows as SmilesToGaussianController ->
76 AIMAllReorController -> GaussianController -> AIMAllController. The latter two con-
77 trollers can also be seen and used as general use controllers wrapping GaussianCalculations
78 and AimqbCalculations

79 Acknowledgements

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

83 References

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