The Journal of Open Source Software

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

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

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 availabe 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,

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

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

Substituent Properties

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

Integrations with Computational Chemistry Software

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

Table 1: Main workflows provided by `aiida-aimall`, their `aiida` entry points that can be used to load them by `aiida.plugins.WorkflowFactory`, and a brief description. These workflows 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 AIMAlll input files, and run AIMAll
`GaussianToAIMWorkChain`	aimall.g16toaim	Run a Gaussian calculationl and automatically run an AIMAll calculation on its outputs
`SubstituentParameterWorkChain`	aimall.subparam	Compute substituent properties defined by the authors automatically

Controllers to limit computer burden when running large numbers of jobs

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

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

Acknowledgements

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

References

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

Keith, T. A. (2019). (Version 19.10.12). TK Gristmill Software. aim.tkgristmill.com

Lefrancois-Gagnon, K. M., & Mawhinney, R. C. (2023). Toward Universal Substituent Constants: Relating QTAIM Functional Group Descriptors to Substituent Effect Proxies. *Journal of Chemical Information and Modeling*, *63*. https://doi.org/10.1021/acs.jcim.3c00987

Talirz, L., Kahle, L., Häuselmann, R., Gresch, D., Müller, T., Yakutovich, A. V., Andersen, C. W., Ramirez, F. F., Adorf, C. S., Gargiulo, F., Kumbhar, S., Passaro, E., Johnston, C., Merkys, A., Cepellotti, A., Mounet, N., Marzari, N., Kozinsky, B., & Pizzi, G. (2020). AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance . *Scientific Data*, *7*. https://doi.org/10.1038/s41597-020-00638-4