

Responses to Reviewer 1

Acknowledgements and General Revision Statement

In the scope of the Journal of Open Source Software (JOSS), one can read: “*We built this journal because we believe that after you’ve done the hard work of writing great software, it shouldn’t take weeks and months to write a paper about your work*”. In writing our manuscript, we have firmly adhered to this journal’s scope statement. Consequently, to avoid redundancy and for the sake of focus, we have not provided definitions and details of physical concepts that are already explained at length in previous works authored by us [e.g., Materials Horizons 10, 1757 (2023) and The Journal of the American Chemical Society 146, 8269 (2024)] and other authors, all of which are listed in the Reference section of our article. Likewise, in our manuscript we have provided the general and most essential details of our IonDiff software implementation since those readers interested in the most technical details can always refer to our open-source Python codes.

Nevertheless, we found that some of the Reviewer’s points suggest adding redundant physical information and technical implementation details, which, in our humble opinion, is not necessary given the scope of the JOSS and the openness of our codes and documentation. On the other hand, we thank the Reviewer for their careful and comprehensive reading of our work and for their very useful and insightful comments that have allowed us to significantly improve the readability of our article and the user-friendliness of our IonDiff software.

Key improvements accompanying this revision are:

- 1.- Enrichment of the explanations about the code’s functionalities in the documentation of the code and the main text.
- 2.- Addition of our code to the Python Package Index (PyPI).
- 3.- Generation of documentation in ReadTheDocs.

1 Contribution and Authorship

Could the authors clarify the contributions of authors Riccardo Rurali and Claudio Cazorla? Are they sharing the git account IonRepo?

A: The GitHub repository is being shared among all authors, which actively participated in their implementation and discussions. Specifically, the authors contributions were as follows: C.C. and C.L. conceived the IonDiff software idea and planned the research, which was discussed in-depth with the rest of authors. C.L. was responsible for the practical implementation of the IonDiff software and construction of the GitHub repository, which is designed and shared among all the authors. The manuscript was written by all the authors.

2 Functionality

The authors must automatize the installation of the software. I could not install it by just following the procedure described.

A: In view of the Reviewer’s comment, the IonDiff code has been added to PyPI to automatize its installation via pip (see updated **README** file). Therefore, now IonDiff can be used both as a python library or a

command-line tool (illustrated in the *examples* folder).

The authors claimed that their implementation scales linearly with different factors, such as the length of the temporal window, the total duration of the simulation, and the number of mobile ions. Evidence for this claim must be provided.

A: The evidence for our claim, which refers to the estimation of the “self” and “distinct” parts of the ionic diffusion coefficient, is provided by Eqs. 1-4 in the manuscript and by the fact that our D implementation is matricial. Since the memory requirements for such a type of matricial calculation scale linearly, our implementation naturally scales linearly.

The authors mention vectorizing operations using a common library such as Numpy. However, could this be improved by using Numba? Is the code parallelized?

A: For the analysis of molecular dynamics simulations comprising a large number of particles and time steps, keeping the trajectory of each simulated particle in memory would require huge amounts of RAM. Consequently, conventional personal computers would not be able to efficiently parallelize such tasks. We found that vectorizing calculations for a given particle is the most interesting solution to this potential RAM problem. Having said this, we agree with the Reviewer that parallelizing our IonDiff code could be an interesting improvement for future releases.

The authors claim that their implementation outperforms standard evaluation algorithms; evidence for this claim should be provided.

A: This claim has been removed from the main text. The primary feature of our implementation is the unsupervised and parameter-free identification of ionic diffusion events and many-ion diffusing correlations from molecular dynamics simulation data, which, to the best of our knowledge, have not been previously implemented in any other code. Consequently, comparisons with other codes are not possible on this regard.

With respect to the evaluation of the diffusion coefficient, which can be actually achieved with other codes, our implementation is based on vectorization, as clearly specified in the manuscript. This approach is more efficient in terms of execution time compared to other methods presented in previous works. However, since this capability is not the central functionality of the IonDiff code, we do not find it necessary to include any quantitative evidence for this improved quality of our code.

3 Documentation

There is no clear documentation of the code’s functionalities. I had to read the source code to understand some of its key functionalities. The authors need to generate documentation for the API that is not embedded in the source code.

A: The documentation for the code’s functionalities has been created through ReadTheDocs and now can be found in the GitHub site and PyPI (<https://pypi.org/project/IonDiff/>). Additionally, in view of the Reviewer’s comment, the explanations regarding the code’s functionalities have been enriched in the main text.

Statement of need: It is unclear why the authors provide a new solution to this problem. Do the authors benchmark their work against more established libraries such as MDAnalysis, MDTraj or TRAVIS?

A: A “Statement of need” now appears written in the revised manuscript. Therein, it is clearly stated which are the problems that our IonDiff software is solving.

Although MDAnalysis, MDTraj and TRAVIS are well-established platforms used for handling and interpreting MD simulations, they do not target any of the problems we address with our code. In particular, our implementation has been designed to analyze, among other different aspects, the many-body correlations between diffusing ions as obtained from molecular dynamics simulations. This kind of functionality is not implemented in any of the codes mentioned by the Reviewer, thus no benchmark analysis can be presented on this regard.

Installation instructions: The installation does not work as described by the authors, as mentioned before.

A: As it was mentioned above, and in view of the Reviewer’s comment, in order to facilitate its usage, the code installation is now done through pip. However, if preferred, it can still be used as a command-line tool.

Example usage: The examples do not work as described by the authors. I have to modify the provided commands to make them work. Additionally, it is unclear what results are obtained; it would be better if the examples were described in more detail. On the other hand, the execution time of the examples is considerably long. This is not explicitly mentioned. The first example took around 13 minutes using 8 cores. The second (analyze_correlations) did not work. The third (analyze_descriptors) yields an AttributeError. Finally, the script run.sh refers to a local path, leading to confusion.

A: As we mentioned earlier, future IonDiff releases will contain parallelization. However, this feature does not represent real limitations for the execution of our code. The amount of time mentioned by the Reviewer is more than reasonable considering the large amount of operations that the IonDiff code has to perform in order to retrieve all the correlation analysis results.

Regarding the example that comes with IonDiff, although in fact the execution time of this could be reduced, we believe that since is a real-world functioning example (similar to what users may want to analyse in their own researches) is more interesting and illustrative than a more simplistic and artificially generated one. Nevertheless, in view of the Reviewer’s comment, we have added more information about the examples in the GitHub site.

It is also worth noting the following points, which rationalize in part the time involved in the execution of the IonDiff software for real-world functioning examples: (1) the *analyze – correlations* script requires to finish the extraction of the DIFUSSION file, otherwise it cannot extract the descriptors and (2) the *analyze – descriptors* script will not work unless more than one simulation are provided, given that this section compares descriptors from various simulations.

We thank the Reviewer for pointing out the presence of the **run.sh** file, which has been removed (since this was part of a closed branch).

4 Software Paper

The paper does not contain a statement of need. Additionally, it does not describe the current state of the art or how this new software can compare with well-established codes for analyzing MD trajectories.

A: We thank the Reviewer for having spotted this absence. A “Statement of need” section has been added to the revised software paper (this was somehow embedded in the original Introduction section). As we mentioned earlier, to the best of our knowledge, there are not other codes that analyze the ionic hoppings and correlations between mobile ions in the fashion that the IonDiff code does. The scope of the IonDiff code is not simply to visualize ionic diffusion trajectories and its scope is significantly different from those of other codes like MDTraj, MDAnalysis and TRAVIS. Hence, quantitative direct comparisons with other codes cannot be meaningfully provided on this regard.

Summary: The authors must specify their target audience. If I understand correctly, their code can only be used to analyse trajectories from ab-initio simulations, which excludes people who make classical simulations. Even among people who make ab initio simulations, it seems their code is targeted to be used with VASP, reducing the pool of possible users considerably.

A: There seems to be a misunderstanding on this Reviewer’s point. The IonDiff code can be perfectly used by any scientist performing either classical molecular dynamics simulations (classical MD) or ab initio molecular dynamics simulations (AIMD). In both types of simulations, atomic trajectories are generated and this is the main input information that the IonDiff code necessitates to perform its correlation and ionic hopping analysis. In other words, the IonDiff analysis does not depend on how the atomic forces are calculated in the undertaken molecular dynamics simulations, whether these are obtained through classical force fields or quantum mechanical methods (e.g., density functional theory). As far as the output trajectory files generated by any classical MD code can be converted to the output trajectory file format of VASP, which is very simple and intuitive, the IonDiff code can be purposely employed. There is already a myriad of open-source codes and scripts that can be

used for this end, namely, to convert a trajectory file generated by a classical MD code to the VASP format, like, for instance, LAVA (<https://github.com/lanl/LAVA>, to convert from LAMMPS format to VASP format) and the online website <https://www.xfroggie.com/index.cgi/strconv> (to convert from GROMACS format to VASP format). Therefore, there is not any factual limitation on the number of potential users of the IonDiff code. Likewise, there is neither a pressing need to replicate already existing codes and scripts able to perform MD files format conversions.

State-of-the-art: The authors must add appropriate citations to some basic concepts to develop the paper. For example, the definition of fast-ion conductors requires a reference to where it is taken. Additionally, citations to other codes that can be used for a similar task must be added if those exist.

A: In view of the Reviewer’s comment, we have added a couple of bibliography references regarding the technical definition of the physical concept “fast-ion conductor”. Nevertheless, it is worth noting that these physical concepts are already quite standard in the fields of condensed matter physics and materials science. Moreover, the physical concept “fast-ion conductor” appears properly defined in most of the supplied bibliography references specialised in Physics (e.g., JACS 146, 8269 -2024- and Mater. Horiz. 10, 1757 -2023-). In view of this fact, and also of the “Acknowledgements and General Revision Statement” section written at the beginning of this document, we do not think that is necessary to add more technical information regarding well-known physical concepts to the present article, which is on the description of new open-source software. Likewise, we do not think that is necessary to add references to other MD analysis codes which, as explained in previous points, have a different scope and suite of capabilities than IonDiff.

References: As mentioned, the authors must enrich their references to cover the state-of-the-art.

A: This point has already been addressed in previous answers.

In addition, I have the following questions:

The authors do not adequately specify the format of the MD trajectory that needs to be used in their code. This must be clarified because the code seems heavily dependent on only that specific format. Complementary, the specification of the trajectory format helps ensure the reliability of results obtained with other software.

A: As it appears clearly specified in our paper and previous answers, the format of the MD trajectory file that is read by IonDiff corresponds to the canonical VASP format. There is already a myriad of open-source codes and scripts that can be used to convert virtually any trajectory file generated by a classical and/or quantum MD code to the VASP format, since VASP is one of the most widely used DFT codes worldwide.

The authors mentioned that one of the code’s functionalities is analyze – descriptors. I think a more detailed description of which descriptors are obtained and how they are obtained is required.

A: Based on the scope of the journal “We built this journal because we believe that after you’ve done the hard work of writing great software, it shouldn’t take weeks and months to write a paper about your work” we believe that the description of the functionality mentioned by the Reviewer that appears in the paper is already appropriate. A more detailed description of the ion mobility descriptors analysis performed by IonDiff can be found in the reference JACS 146, 8269 (2024), thus there is not a pressing need to repeat that information here. In the revised version of the article, and in view of the Reviewer’s comment, we have specified that complementary explanations about the code functionalities can be found in the reference JACS 146, 8269 (2024).

From the technical side, the authors need to provide information about hyperparameters for the k-means algorithm or give a criterion, either empirically or physically based, for selecting those values that are known to be key to using the unsupervised method.

A: The value of the hyperparameters employed by IonDiff correspond to the default values of the Sklearn implementation. Nevertheless, in case of need, those can be varied at wish by the user. This information appears now better explained in the revised version of our article.

In the same line as above, I wonder if the authors consider using other unsupervised machine learning algorithms (see for examples, section 5 of Chem. Rev. 2021, 121, 16, 9722–9758) to do the clustering. It is

mentioned that it is possible to use spectral clustering. If that is the case, an explicit example should be added.

A: As it appears now explained in the revised article, in a previous work (Lopez et al. 2024b) it was found that the performance of k-means clustering in identifying ionic hops was generally superior to that of other clustering approaches such as spectral clustering for a database containing about 170 different AIMD simulations. Therefore, we do not consider as necessary to add any example for this particular case. Nevertheless, the user interested in these technical details may always conduct their own tests by simply changing few IonDiff execution commands.

Regarding the methods section, in equation 1, the authors mention that the ionic conductivity is a function of the number of spatial dimensions. I would like to know if the code allows the computation of the ionic conductivity in a specific dimension (i.e., in the x-axis only).

A: As suggested by the Reviewer, we have implemented this new functionality in IonDiff, which allows for the calculation of the diffusion coefficient in any specific dimension.

Authors need to provide information about the equation used to obtain the diffusion coefficient; equation 2 seems to be the Einstein relationship, while equation 4 seems like a discretized version of the Green-Kubo expression. To do this, I encourage the authors to review the following reference: Living J. Comput. Mol. Sci., 2019, 1, 6324.

A: Equation 4 in our manuscript follows directly from Equation 2 (there is neither a modification nor an assumption in between then). In turn, Equation 2, which is well known anyway, comes from the bibliography reference Sasaki et al., 2023 in our manuscript. Therefore, we do not consider as necessary to add more references on this regard.

The authors mention that they assumed that the non-diffusive particles have an isotropic behaviour in their fluctuations. What is the reason for that assumption? In which cases is this assumption fulfilled? Is the developed code still useful if the non-diffusive particles do not follow that assumption?

A: For most typical and technologically relevant fast-ion conductor materials, ionic diffusion is three-dimensional and pretty much isotropic. For nano-structured like compounds, for instance, layered quasi two-dimensional materials, this may not be the case. But in those situations the IonDiff program already allows for using other clustering schemes, as it has been mentioned earlier. In the revised version of the manuscript, we have clarified that in case the assumption of isotropic ionic diffusion did not hold, the user can always select other different and more appropriate clustering schemes.

The authors adopt an arbitrary threshold of 0.5. What is the justification for this definition? As they mentioned, the definition of this distance is critical for obtaining the k-centers. Then, what is the effect of this value? Could the author provide information on what happens if other threshold values are used?

A: The Reviewer’s claim that “the definition of this distance is critical for obtaining the k-centers” is not correct. The k-centers are obtained independently of the arbitrary threshold distance of 0.5 Angs (i.e., it does not influence the results obtained with the k-means clustering algorithm). This threshold distance is selected mostly for representation purposes of the ionic diffusion paths. The numerical value of 0.5 Angs is chosen according to typical values of atomic mean squared displacements in crystals and is an order of magnitude smaller than the typical length of diffusion paths. As explained in Lopez et al. (2024b), if one reasonably varies the value of this threshold distance (e.g., from 0.3 to 0.7 Angs), the outcomes of our ionic diffusion analysis essentially remain invariant. By increasing or decreasing this threshold distance value, you simply decrease or increase the number of points depicting the ionic diffusion paths, respectively. In view of the Reviewer’s point, in the revised version of our manuscript, we have added a clarification regarding the invariability of the results obtained with IonDiff under reasonable variations of this threshold distance parameter.

In the subsection regarding correlations between mobile ions, the procedure explained in the first paragraph can be illustrated as pseudocode to facilitate its comprehension.

A: We consider as unnecessary to add pseudocode to our manuscript for illustrating how the IonDiff code identifies many mobile ion correlations. This identification process is already explained in detail in the main text,

so adding pseudocode would be redundant. The scope of the present journal advocates for non-redundancy, or at least we interpret it as such, and we would like to adhere to this principle.

Regarding the examples, why do you always obtain one or two clusters? Is that what we should expect? Does that mean that only one diffusion path is observed in your simulations? What is the physical meaning of the clusters obtained by the code, if there is one?

A: The physical meaning of the clusters obtained by the IonDiff code is the equilibrium positions around which particles vibrate during molecular dynamics simulations. The points located between different ionic vibrational centers, i.e., different k-means clusters, are regarded as part of the ionic diffusion paths connecting them. IonDiff can certainly identify more than one or two k-means clusters (as shown, for instance, in Supplementary Figure 2 of JACS 146, 8269 -2024-). In the examples provided here, we present two-cluster identification cases because they are easier to visualize. In view of the Reviewer’s comment and to avoid possible misunderstandings, we have added further explanations regarding the physical interpretation of the identified k-means clusters in the revised version of our manuscript.

5 Suggested Improvements

Besides the points mentioned before, I think the author could make the following improvements:

Improve the code installation using a `setup.py` file or a `.toml` file. Also, a mention of specific requirements such as Python and package version will be very useful.

A: In view of the Reviewer’s comment and to facilitate the installation of the IonDiff software, we have added it to PyPI.

Create detailed tutorials for using the code with the aid of either scripts or Jupyter notebooks that allow the user to understand what the code is doing.

A: In view of the Reviewer’s comment and to improve the user-friendliness of the IonDiff software, Jupyter notebooks have been created and added to the `examples` folder, better exemplifying the usage of the code.

The authors must provide a better description of their code’s capabilities. Judging from their software’s reported use, I believe this code has multiple applications. However, it is unclear to me which capabilities and how they can be applied by the MD community.

A: We believe that the principal capabilities of the IonDiff software are already clearly explained in the present article and the corresponding GitHub site. These capabilities are: (1) extracting ionic migration paths with spatial and temporal resolution, (2) extracting atomic-scale descriptors of ionic diffusivity (e.g., residence times in metastable positions), and (3) analyzing possible correlations between many mobile ions. Additionally, the recently published work JACS 146, 8269 (2024) is entirely based on the capabilities and analysis performance of the IonDiff software. Therefore, there are already ample explanations and examples available in the literature for the MD community to understand and utilize our software. Nevertheless, in view of the Reviewer’s comment, we have made an effort to improve the readability of our manuscript in this revised version.

As mentioned before, documentation must be improved. I also suggest that the authors include a detailed section explaining the theory behind it. This should not be a problem because the authors explained it in their previous work. Last but not least, a section on the documentation on constructing the input files for the code must be added.

A: In view of the Reviewer’s comment, the explanations and online documentation regarding the usage and capabilities of the IonDiff software have been substantially increased. Nevertheless, as repeatedly pointed out throughout this revision and explained in the “Acknowledgements and General Revision Statement” section at the beginning of this document, there is no need to repeat or further expand on information that is already clearly and concisely explained and/or available from other sources. Regarding the software input files, their content is already clearly explained in the manuscript and software documentation. Users can find examples in

the *Examples* GitHub folder.

I suggest the authors add more details about generating the MD Trajectories. As these simulations are very sensitive to different factors, the user must be alerted that their results will depend on those.

A: The IonDiff software does not generate any MD trajectories. MD trajectories constitute the input files for IonDiff and may be generated with various classical and/or quantum simulation codes developed by other research groups and computer scientists. This paper is about the IonDiff code, which analyzes but does not generate MD trajectories. Thus, we cannot address the Reviewer’s query to “add more details about generating the MD trajectories” because such details pertain to specific software other than IonDiff.

6 Overall

I believe the presented work can be useful for analysing MD trajectories for ionic systems. However, I have several doubts about the software’s functionality and applications; therefore, the paper requires major revision to have a clear view of them. Also, the state-of-the-art software must be improved, and suitable comparisons with other available software must be made. Finally, I failed to see how this work can be combined with existing frameworks as it seems to be specialized in analyzing VASP trajectories and is incompatible with classical MD. Likewise, I failed to see how this code systematically determines exact ionic migration paths and the level of coordination between mobile particles in diffusive events. To conclude, the paper and the code must go through major revision before considering its acceptance.

A: Following the Reviewer’s assessment, our code, documentation, and paper have all undergone a major revision. We have addressed all the criticisms raised by the Reviewer point-by-point. Nevertheless, as we have reasonably argued in this response document, some of the Reviewer’s queries implied replicating information that was already available from other sources and/or providing additional details that were not necessary or did not pertain to the IonDiff software. In these latter instances, we did not implement the Reviewer’s suggested changes.

7 Other questions

In one of the previous works by the authors, the van Hove (VH) autocorrelation function is reported. Is it possible to obtain the VH function with the present code?

A: As we discussed in the work mentioned by the Reviewer, the van Hove correlation function is clearly outperformed by our unsupervised many-ion correlation approach, which provides atomistic information about diffusion instead of simple two-body and merely qualitative correlations. Therefore, we do not find it necessary to add that functionality to the IonDiff software.