

Response to comments from reviewer Wood

12/18/23. Reviewer comments in italics.

- *I failed to get this installed on MacOS. I suspect I would also fail on Windows. I was able to follow the installation instructions on Linux/aarch64 in a VM, and all of the unit tests passed. Could you update the requirements to say this requires Linux, or could you do some testing to get this working on Mac and Windows?*
 - CM++ now supports MacOS. Now, the software supports Mac and Linux (not Windows), and I have indicated that in the requirements section of the docs.
 - https://illinois-or-research-analytics.github.io/cm_pipeline/installation/
- Your requirements.txt file is very specific. Is there any way to relax the version requirements, e.g. to specify dependencies to only their minor version numbers rather than to their patch version numbers? I worry that this installation could be very brittle.
 - requirements.txt now includes <version>.<minor> and not patch numbers.
 - https://github.com/illinois-or-research-analytics/cm_pipeline/blob/main/requirements.txt
- Could you include example usage in the paper? Or in the documentation? Could you structure the documentation in docs / README so that it is more clear where I should start to learn how to use this program? The examples directory just contains json files, while the command to run in the first documentation page (python -m [hm01.cm](#) -i network.tsv -e clustering.tsv -c leiden -t 1log10 -n 32 -o output.tsv) gives this error;

Traceback (most recent call last):

```
File "/home/parallels/mambaforge/envs/review/lib/python3.10/runpy.py", line 196, in
_run_module_as_main
    return _run_code(code, main_globals, None,
```

```
File "/home/parallels/mambaforge/envs/review/lib/python3.10/runpy.py", line 86, in
_run_code
    exec(code, run_globals)
```

```
File "/home/parallels/review/cm_pipeline/hm01/cm.py", line 514, in <module>
    entry_point()
```

```
File "/home/parallels/review/cm_pipeline/hm01/cm.py", line 510, in entry_point
    typer.run(main)
```

```
File "/home/parallels/review/cm_pipeline/hm01/cm.py", line 422, in main
    assert resolution != -1, "Leiden requires resolution"
```

AssertionError: Leiden requires resolution

- Example commands have been modified accordingly and are included in the documentation and in the paper
- https://illinois-or-research-analytics.github.io/cm_pipeline/examples/

I recommend adding something like a "quick start" guide or similar, with complete input files for the examples, so that it is easier to get started straight away with the software. While you don't need as much, see [here](#) how in my software the user is taken on a journey from seeing if the software has the features they need, then a very simple install guide, then a quick start guide to quickly explore the software (before then more detailed guides and a tutorial). Something like this can really make a difference to a user's experience of the software.

- A quick start guide has been created in the documentation, which includes a basic tutorial with downloadable sample data and visualization.
- https://illinois-or-research-analytics.github.io/cm_pipeline/quickstart/
- Could you add some community guidelines as detailed in the checklist above? I am sorry if I missed them. Feel free to steal from [How to ask for help](#) and [Contributing](#).
 - The community guidelines are in the documentation
 - https://illinois-or-research-analytics.github.io/cm_pipeline/contributions/
- Could you provide the input files so that I can confirm the performance claims in figure 1? Also, could you change this into a scaling plot rather than a time to solve plot? (i.e. the speed relative to 1 core for each core count - ideally a 32-core job would be 32 times faster, but this would almost never be the case). This would give a better idea of how the code scales.
 - The paper now shows a speedup curve instead of a runtime curve to show scaling
 - To retrieve the input files:
 - Visit <https://databank.illinois.edu/datasets/IDB-0908742>
 - CEN: cen_pipeline.tar.gz