Published on *cisTEM* (https://cistem.org)

cannot export in cisTEM 2 --- "negative content height"

Wed, 11/22/2023 - 04:49

#1

mrhansenable2

cannot export in cisTEM 2 --- "negative content height"

hi,

I am testing out cisTEM 2 on our cluster. Seems to run jobs fine. However I tried testing the export stack function and got the error below

Negative content height -9 (allocation 1, extents 5x5) while allocating gadget (node button, owner GtkButton)

For testing I did the following.... I imported a small RELION stack, no issues. I tried to export that same stack (either RELION legacy or RELION 3.1) and I get the error above.

any help is appreciated

thanks

Jesse

jojoelfe

Hi Jesse,

Hi Jesse,

can you be more specific about which version you tried? In the most recent versions this will fail because cisTEM will try to output motion correction information to enable the bayesian polishing in relion. In your case this information does not exists.

I think some people went around this by opening the project in cisTEM 1 and exporting from there. Otherwise we hope to push out a fix for this soon and I'll reply here once this is up.

Best,

Johannes

Wed, 11/22/2023 - 14:34 (Reply to #2)

mrhansenable2

hi Johannes,

hi Johannes,

thanks for the reply. I am running:

cisTEM version : 2.0.0-alpha-174-e4b9038-dirty
Commit datetime: 2023-10-16 18:27:45 -0400

My real problem here is that I desperately need to import a particle stack into cisTEM then export that stack while being able to match up each particle to the original input stack. That way I can ultimately add back all the metadata to the star file and can continue my helical processing. As you know, cisTEM 1 appears to strip metadata. I was hoping cisTEM 2 fixed this.

I thought I had a clever solution for cisTEM 1. The defocus values apparently seem to be unchanged between import/export of a stack, so I wrote a script to scramble the decimal parts of the defoci.. creating a bit of a barcode for each particle. However I noticed that upon exporting the stack those defoci decimal parts were scrambled in cisTEM. Not sure why cisTEM modified defoci decimal points but it made it so I could no longer identify my particles.

I desperately need to be able to identify these particles, but it appears there's no way to do it. I saw a paper recently where they wrote a program to do it using numpy arrays (see below). Do you have any clever solutions or workaround for those trying to retrieve this information? thanks in advance

from doi: https://doi.org/10.1101/2023.06.15.545104

One limitation of segment selection after 2D classification in cisTEM was the program's inability to carry forward all particle metadata, such as the extraction coordinates and name of the original micrographs. We therefore mapped back the 4,983 segments of the initial helical reconstruction to the original segment particle stack from the combined two data sets. We did this by converting each image into a Python NumPy array[73], calculating an MD5 checksum, and mapping identical checksums between the two stacks using Python dictionaries.

Wed, 11/22/2023 - 14:45 (Reply to #3)

jojoelfe

Hi,

Hi,

I think these metadata would also not bet maintained in cisTEM2. What processing are you doing in cisTEM? 2D Classification and then selection of a subset of data?

Wed, 11/22/2023 - 14:54 (Reply to #4)

mrhansenable2

hi,

hi,

oh, that's sad to hear. I'd suggest you add that capacity to maintain the metadata! That way cisTEM can be better integrated with other packages, right now it ends up a bit of a black hole because once particles go in they can never come out.

I am processing helical data, so my particles file has lots of helical metadata. cisTEM is doing a great job at sorting out heterogeneity in my dataset, so then I'd like to separate out 2D classes which have the helical diameter that I like. So yes, it's 2D then subset selection. I've tested it a bit today, I notice that if I import a stack then immediately export it then it retains the defoci decimal parts. However when I run a 2D classification and create a new stack then for some reason the defoci decimal parts get lost.

thanks again

Jesse

jojoelfe

Hi Jesse,

Hi Jesse,

encoding information in the defocus estimate seems unwise to me. The scrambling probably occurs due to conversion between different float types either in refine2d or when writing to the database. The idea with the image checksums is neat, but probably overkill. cisTEM will maintain the order of the particles when importing and keeps track of the original particle number when creating a new refinement package. You should be able to use that information to filter your relion star file. You can get it using:

```
sqlite3 ${cistem .db file} "SELECT
ORIGINAL_PARTICLE_POSITION_ASSET_ID FROM
REFINEMENT_PACKAGE_CONTAINED_PARTICLES_${id}"
```

where id is the id of the new refinement package asset.

I would use Alister Burts excellent starfile python package to manipulate your relion star file.

Hope this helps you,

Johannes

Wed, 11/22/2023 - 15:57 (Reply to #6)

mrhansenable2

Hi Johannes,

Hi Johannes,

THANK YOU!!! I have been using cisTEM for so many years and only now did I discover this. Maybe it's elsewhere in this discussion group and I missed it.

For those that are less computer literate, here's how you'd type the command from your cisTEM folder:

```
sqlite3 myproject.db "SELECT
ORIGINAL_PARTICLE_POSITION_ASSET_ID FROM
REFINEMENT_PACKAGE_CONTAINED_PARTICLES_18"
```

in this example, #18 refers to the refinement package # in Assets > Refinement Packages.
Careful though that numbering starts at 0 in cisTEM but sqlite starts at 1. Thus, if you want to export refinement package #17 (cisTEM) you need to use a value of 18 in the sqlite command.

Thanks again Johannes, you have saved me much headache.

Jesse

Log in or register to post comments

Source URL:https://cistem.org/cannot-export-cistem-2-negative-content-height