



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

[Home](#) > parallelization of ctffind and unblur

---

# parallelization of ctffind and unblur

Wed, 10/03/2018 - 12:45

#1



dauidgenemorgan

parallelization of ctffind and unblur

Hi,

Older versions of ctffind (v3 and v4) and unblur were parallelized using openMP and the number of processors to be used could be set using the OMP\_NUM\_THREADS environmental variable. I did a bunch of tests on the ctffind programs and showed that on our cluster, I had parallel but worse performance if I did not set this variable. In other words, if I ran a job without setting this variable, it did use multiple cores if they were available, but without setting the number explicitly, there was some sort of system thrashing that actually impeded execution. It wasn't a huge performance hit, but I made sure that various scripts that ran ctffind (including my queue submission scripts for relion) explicitly set this variable.

The versions of ctffind and unblur in cisTEM appear to be different (and may not even be parallelized). Has anyone gotten parallel performance outside of the cisTEM GUI? I am wondering about (for example) running the cisTEM executables through relion, and how to tell it to use any available parallelization.

Any help would be appreciated.





alexis

Hi David,

Hi David,

In earlier versions, if you didn't set `OMP_NUM_THREADS`, I think it would automatically spawn as many threads as there were (virtual) cores detected, which is often not optimal, as you found out.

The versions distributed as part of cisTEM were re-written without threading, because we were mostly targeting the use case where you want to process, say, 10,000 movies, so parallel processing over many nodes made sense and threading was not crucial. Since then we have implemented some threading again. We now have a threaded version of `ctffind` undergoing testing, and I think we'll also soon have an `unblur` version with threading internally. So I think it's likely that the next release will include binaries with OpenMP threading. Sorry this doesn't help you today, but it's on its way.

Cheers,

Alexis



[Log in](#) or [register](#) to post comments

---

**Source URL:** <https://cistem.org/parallelization-ctffind-and-unblur>