In order to perform an analysis on the zcluster it is important to make sure that you have done, or checked the following:

1. Your data are located on the zcluster fast storage system (you home account or escratch NOT archival storage).
2. Make sure you have permission to read the data you wish to analyze
3. Write a shell (.sh) script to run your analysis.
4. Test the command line located in your .sh script on a VERY SMALL data set to make sure that all is working correctly via an interactive session (see below).
5. If all is OK, change permissions on your .sh script to make it executable
6. Select a queue that you would like to use and make note of its name.
7. Submit your job to the “scheduler”
8. Run qstat to see if your jobs are waiting or running
9. Check the output to see if there are any errors
10. Logout if all is OK. If not, begin trouble shooting

A note about “running” or “submitting” jobs

- **The “Head” node** - When you login, you are on the “login” or “head” node of the computer. DO NOT run jobs here. You can do small stuff like arrange files, edit files, etc but DO NOT run applications. They can, and probably will, crash the system or slow everyone down. Why? The Headnode is busy controlling all communications with the compute nodes.

- **An interactive session** - It is important to test our scripts before embarking on a big compute job. We do this with “interactive” nodes. These are computers that are especially set aside for you to test your code on. They are VERY easy to access. Just type “qlogin” at your prompt and you are there and so is your home directory. Everything looks the same except your prompt. Test away. You should think of this as a location where you can do things that run an hour or two. Remember to logout of your interactive session and then logout of your connection to the cluster when you are done (2 logouts/exit) commands.

- **Submitting a job to the scheduler** – the scheduler is for big jobs. It controls the “queues” (lists of different types of computers, or computers that can run your job for various lengths of time, like 30 days). The scheduler knows how to “play fair”. It controls how much access users have to machine so that no one person can “hog it all”. It knows how long everyone has been waiting and the type of computer they are waiting for. It is smart. Some analyses can be broken into “batches”, for example you could send 1000 sequences to a single processor for BLAST analysis, but things would go faster if you sent 100 sequences to 10 processors. Since the scheduler will handle them for you, the job gets quite easy. As a user, you are allowed at least 64 processors at a time. This is a lot of computer power.