LIGO practical intro doc

Some useful links for registering to LVK and test-runs of GWs events.

Step 1: Become a LIGO member

- Register for a LIGO membership here.

After your membership is accepted:

- Register for LIGO's Mattermost here.

  Many channels are open for joining → you can find them by clicking the + sign, next to `LIGO` at the top-left. This will open a list of available public channels (beware, the list continues for multiple pages - click `Next` at bottom-right, to check them all).
- Apply for a LIGO cluster account here.
- Create a GitLab LIGO account here.

Step 2: Background for GWs analysis

- The GWs Open Data Workshops (here) are very useful to understand the basics of GWs analysis. There is also a dedicated forum (here) with Q&A about the workshops, but also more general issues.
- The IGWN Public Alerts page (here) has a very useful website describing the whole process, but also many of the relevant terminology.

Step 3: LIGO Observing runs

- A useful compilation of FAQ can be found here.
- A quick tutorial, using `bilby` to analyse, i.e. do a parameter estimation (PE), of a GW event is here. Beware that there could be some typos on the event's naming, so try to be consistent with the original event's naming at the top, i.e. the command that creates the `bilby` configuration file.

- The general idea is the following:
  1. We create a directory for the PE run
  2. Activate relevant python environments
  3. Make all the necessary (physical) changes in the `filename_config.ini`
  4. Secure permission (If problems with permission occur, check (here) for alternative authorisation options. Sometimes the error log files of a failed job can have useful links.)
  5. Run the analysis: `bilby_pipe filename_config.ini --submit`
  6. Check analysis: `condor_q --nobatch`. More on `HTCondor` workload manager here.
- More details on `bilby` configuration options can be found here.

- For user-interface, i.e. options in the config file, check here.
- `Bilby` can be used to create a configuration `.ini` file based on *GraceDB* events. For a specific example (here) and more generally (here).
- **Note**: There is a dedicated channel in LIGO mattermost (*Bilby Help*) for issues with `bilby`.

- **Where to run?**

  To access the LIGO clusters, follow the ways here. For UNLV GWs group, we use the *CIT* site, with the following hosts:

Hostname	Description	Mem	CPU Model	Core	GPU
ldas-grid	Production submit	96G	AMD EPYC 7543 VM	16	
ldas-pcdev1	Production submit	64G	AMD EPYC 7543 VM	8	
ldas-pcdev2	GPU mixed	256G	AMD EPYC 7502P	32	4
ldas-pcdev3	GPU mixed	128G	Intel E5-2670	16	3
ldas-pcdev4	IGWN development repo	32G	AMD EPYC 7543 VM	4	
ldas-pcdev5	Post-processing	512G	Intel E5-2698 v4	40	
ldas-pcdev6	Large mem/post procs	1.5T	Intel Gold 6154	72	
ldas-pcdev8	IGWN testing repo	96G	Intel Gold 5115	10	1
ldas-pcdev10	EPEL 8 upstream repo	64G	Intel E3-1240 v5	4	
ldas-pcdev11	GPU mixed	128G	Intel E5-2630 v4	20	4
ldas-pcdev12	GPU 4 x A100-SXM4-80GB	512G	AMD EPYC 7763	128	4
ldas-pcdev13	GPU mixed	128G	Intel E5-2650 v4	24	4
ldas-pcdev14	Large mem benchmarking	256G	AMD Opteron 6376	16	
ldas-osg	IGWN OSG submit machine	192G	AMD EPYC 7543 VM	24	
ldas-osg2	IGWN OSG submit machine	128G	AMD EPYC 7543 VM	17	
condor-f1	Condor 10.x Feature Release	32G	AMD EPYC 7543 VM	32	
condor-f2	Condor 10.x Feature Update	16G	AMD EPYC 7543 VM	4	
condor-f3	Condor 10.x Release Candidate	8G	AMD EPYC 7543 VM	2	
cbc	Dedicated CBC	32G	AMD EPYC 7543 VM	8	
cwb	Dedicated cWB	96G	AMD EPYC 7543 VM	12	
detchar	Dedicated DetChar	256G	AMD EPYC 7543 VM	16	
emfollow	Dedicated EM Follow-up	64G	AMD EPYC 7543 VM	8	
gstlal	Dedicated GstLAL	64G	AMD EPYC 7543 VM	8	
lowlatency	Dedicated low-latency (SL7)	64G	Intel E5-2630 v3	16	
spiir	Dedicated SPIIR	32G	AMD EPYC 7543 VM	8	

LIGO has a number of workstations - the `HTCondor` system will distribute the run accordingly. The `ldas-grid` and `ldas-pcdev*` are good for hosting your files and analyses (check the system configuration, in case you are interested in a job with specific characteristics, like high memory etc).

Step 4: Checking your `bilby` runs' results

Of course, the files could be downloaded and inspected locally. But there is also the possibility to examine some of the plots online:

- Follow the links here. Either `Jupytel Lab` or the `public_html` page organise the produced plots.
- To create a summary webpage (shown in `public_html` above), the following options must be selected in the configuration file:

```
create-summary = True
email = albert.einstein@ligo.org
webdir = /home/albert.einstein/public_html/project
```

- The `dag_name.submit` file and `bash_name.sh` file have useful information on the order of jobs submissions, but also on the commands used at each step that initialise the specific part of the run (these can be useful, if you want to check)
- The different `.log` files show details (duration, memory etc) about the runs.