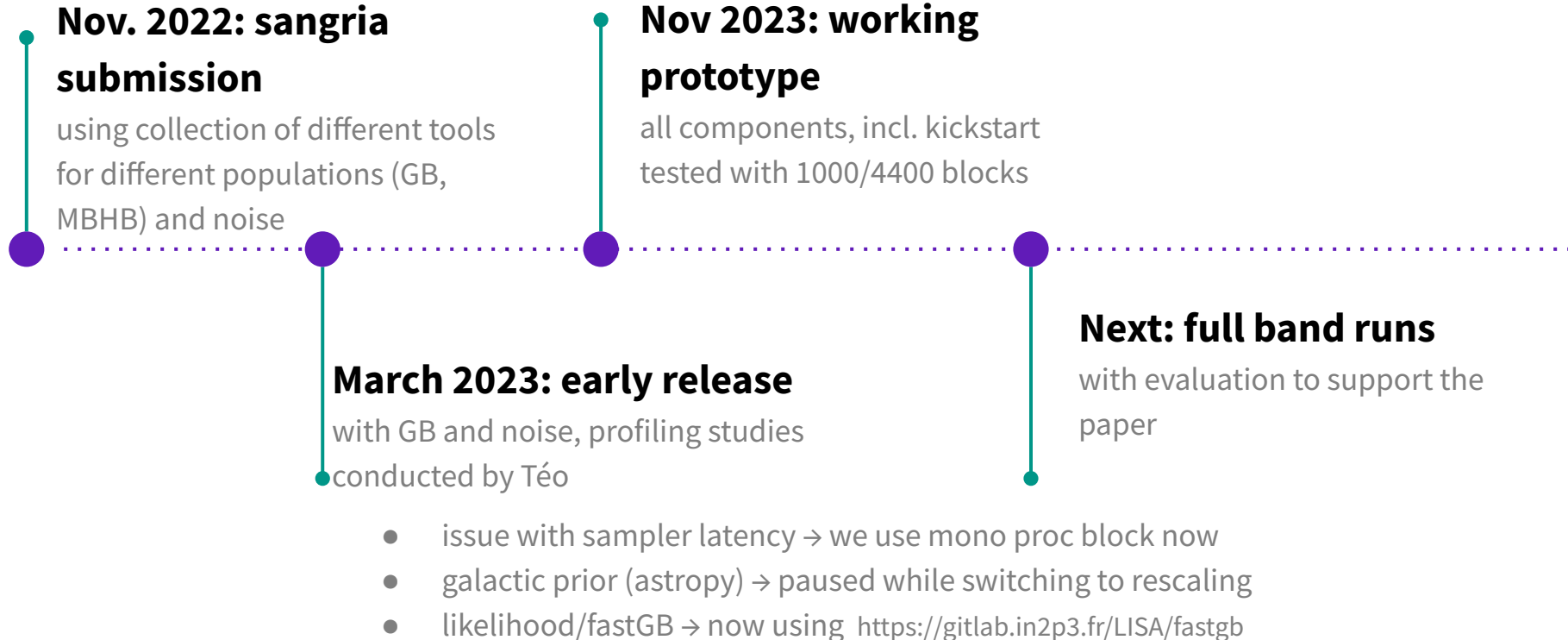


Global-fit introduction

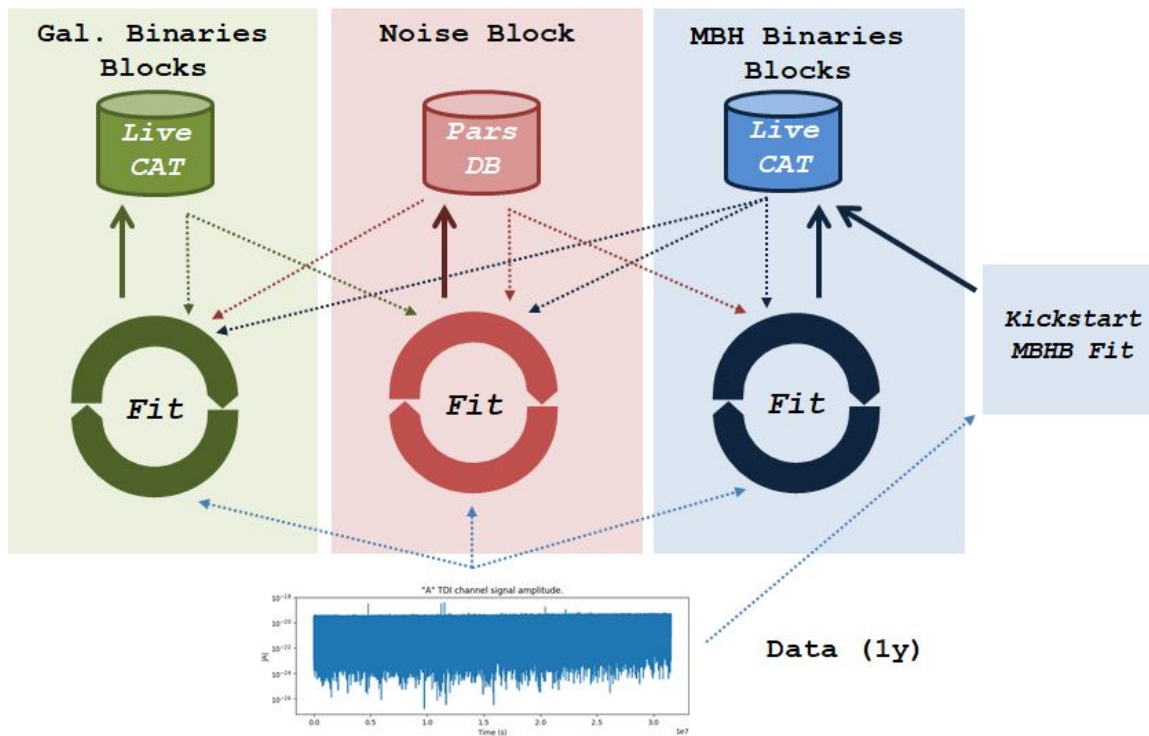
M. Le Jeune, A. Sartirana

Where we are



Overall structure

- Blocks iteration (see below for details)
 - subtraction
 - detection and PE
 - update catalog.
- 1k-4k jobs/iteration
- need to refine sync and tune duration;
- MCMC Chains, plots, logs and debug info stored locally;
- currently "live catalogs/dbs" are hdf5 file
 - concurrent access on shared FS is problematic;
 - plan to move to a DB server (see below).



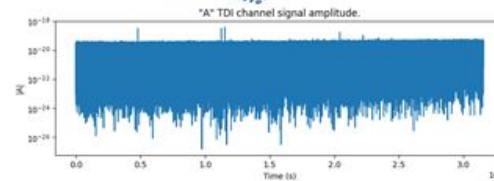
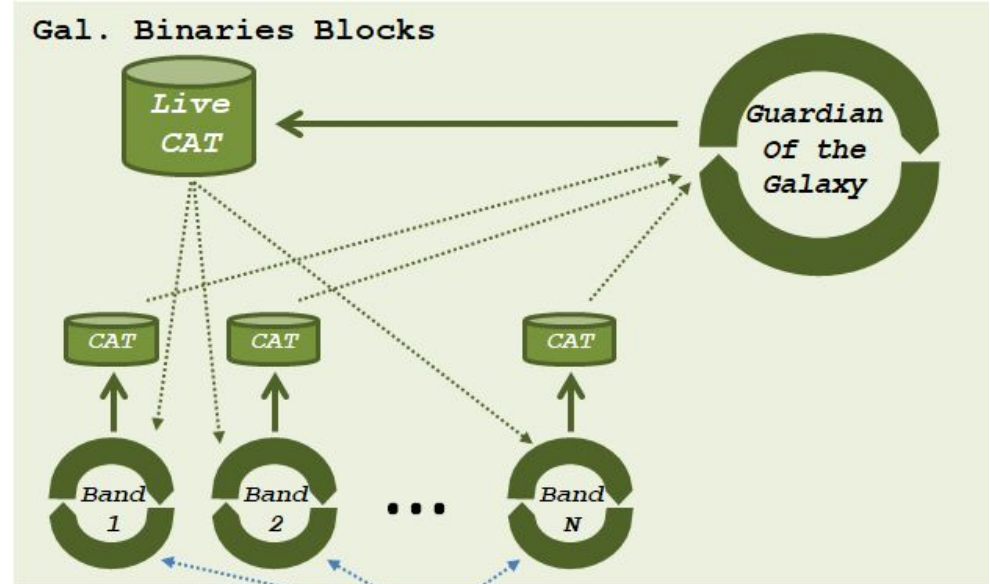
Overall structure: GBs fit

1 GB Block for each frequency band

- ~4.4k blocks (ran with 1k so far);
- one HTC job per block;
- each job submits the job for its next iteration:
 - configurable, but seems to be the best option for the GB blocks.

Shepard block (GoC) to coordinate the GBs activity and deal with I/O

- each GB block writes to local cat, the GoC updates the global cat
 - this avoids parallel writing issues on the Trex/HAL SPS;
 - for production we plan to use a DB server;
- follows, kills, recreates the bands blocks.



Data (1y)

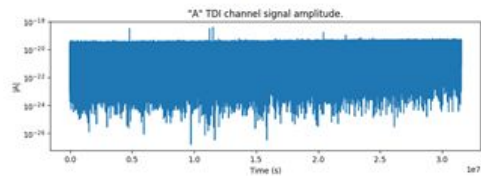
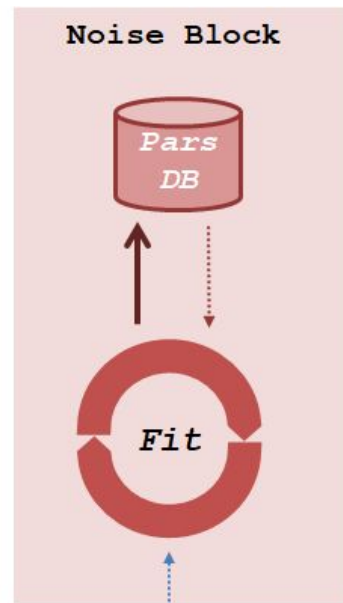
Overall structure: Noise fit

Single block (job) fitting noise parameters

- works on a set (14) of frequency bands in which the noise is roughly considered constant;
- subtract MBHB sources and outband GBs from the live catalogs;
- inband GBs are included in the likelihood and (partially) fit with the noise;
- runs MCMC to make an estimation of the noise model parameters;
- updates the noise parameters DB (that will be used by other blocks).

Designed so to easily switch likelihood definition and/or noise model

- currently we run with a 7 pars noise model and likelihood includes inband GBs with amplitudes analytically marginalized and other parameters fixed;
- tested (and will test) different likelihoods;
- plan to move to a more general model based on splines.



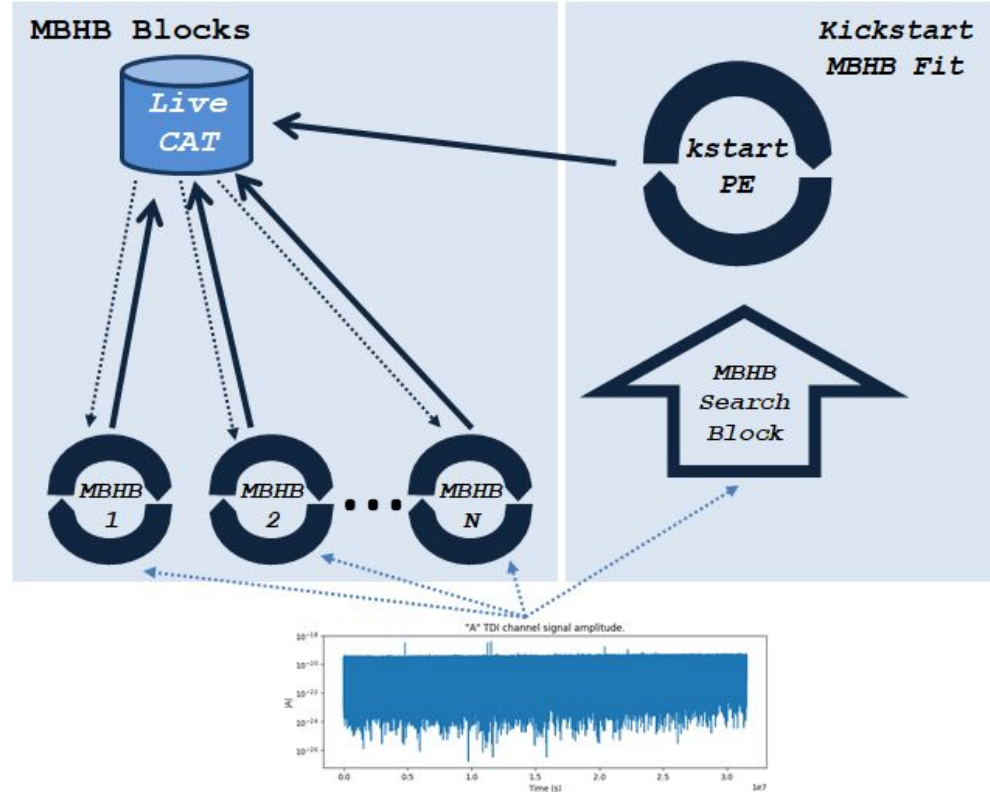
Overall structure: MBHB fit

Kickstart blocks

- runs before the GF loop;
- the search blocks looks for MBHB merger candidates;
- the kickstart PE block makes a first rough PE of the MBHB (w/o subtracting other sources).

MBHB Blocks

- 1 for each identified MBHB;
- refines PE subtracting other sources with updated parameters from the live catalogue.



Overall structure: central DB (draft)

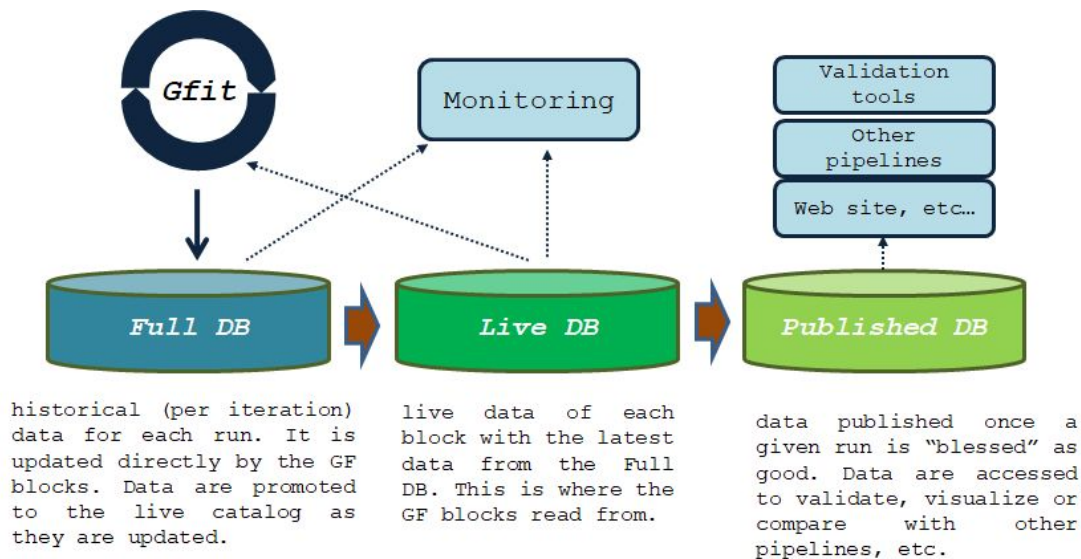
We currently use hdf5 to store the live data and store historical data in local directories.

This is not optimal:

- concurrent writing to the live catalog on the shared filesystem is problematic;
 - worked around making the GoC block deal with I/O;
- historical data are spread and not easy to aggregate for monitoring/analysis.

Use a central DBS instead:

- CNES and CC provide such services;
- here is a draft of the arch.



These can be different DBs on the same server or different sets of tables in the same DB.

Latest results

simulation*

start from some default noise model and some default catalog

1. subtract 1 source / block within the band as soon as $\text{snr} > 10$
2. update central catalog within the band
3. update noise model
4. iterate on 1-2-3

run #1

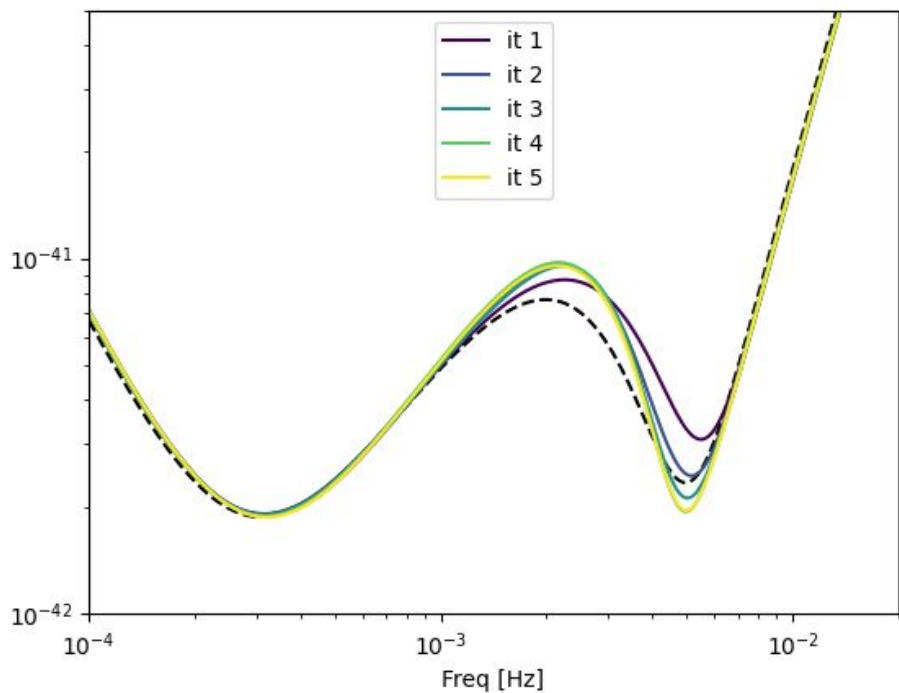
- 1000 GB bands between 4 and 5.3 mHz (use true catalog with $\text{snr} > 10$ elsewhere)
- all 14 MBHBs, kickstart from short powell
- noise model with 7 parameters only, marginalized over GB amplitudes
- All types of blocks are synchronous, restricted to GB blocks matching noise bands
- GBs stop as soon as the number of selected sources is the same for last 3 iterations
- 5 iterations

Results here: `/work/LISA/lejeune/globalfit/gfrun8`

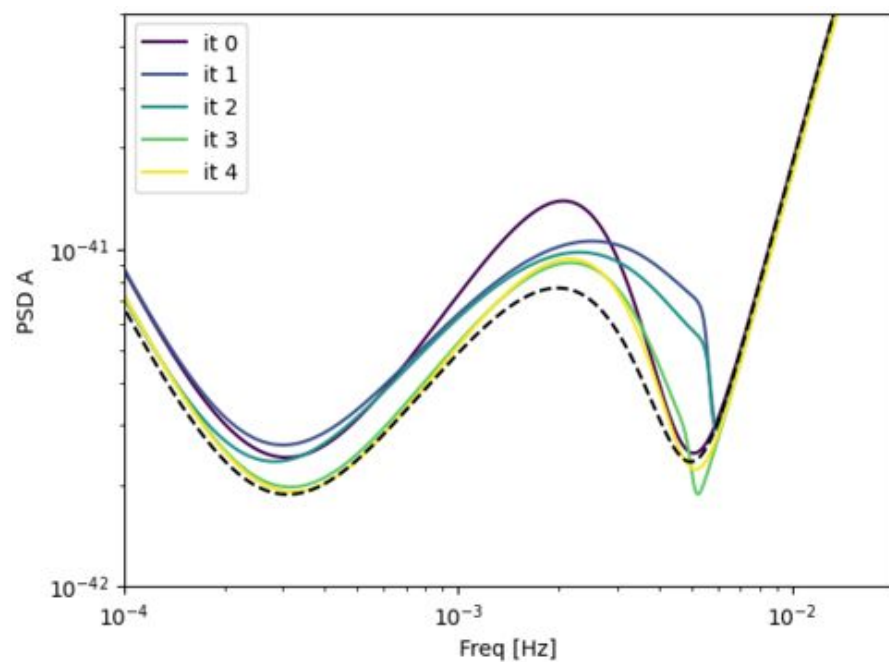
*MBHB are perfectly subtracted here

Latest results

simulation*



run #1



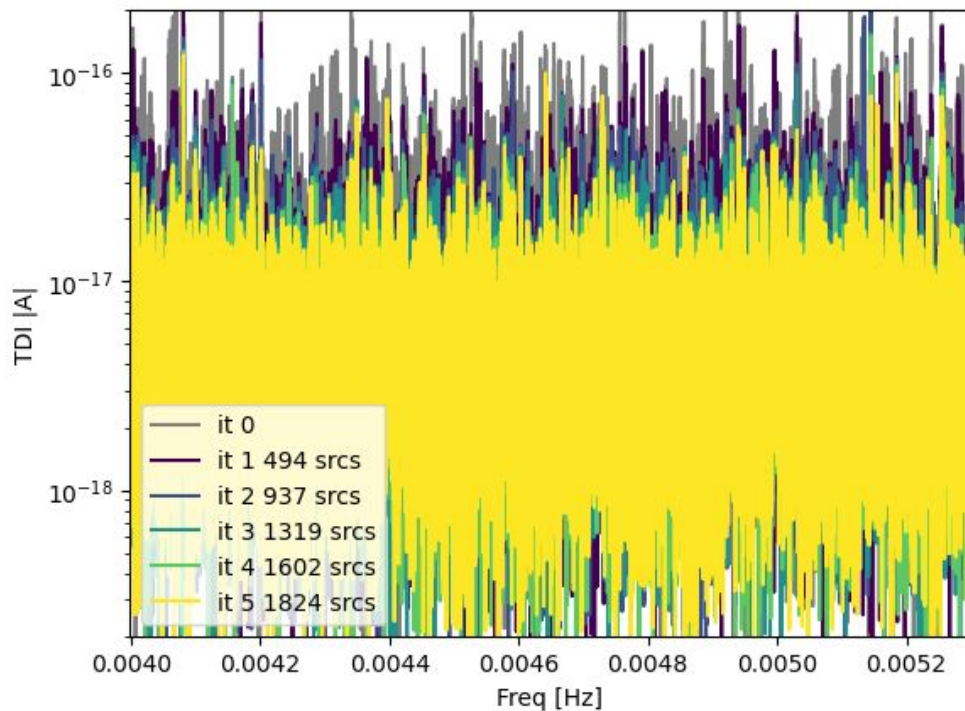
Latest results

simulation*

run #1

Number of expected GB

	snr 8	snr 10
it 1	604	598
it 2	1089	1006
it 3	1510	1396
it 4	1832	1663
it 5	2055	1835



TD

- installation

- one has to be member of the globalfit gitlab repo
- `git clone git@gitlab.in2p3.fr:lisa-apc/global-fit.git`
- see [README](#)
- get the data: `python install_data.py -d /work/data/globalfit`

- local run

- describe the option
- 3 GB, no noise, no MBHB, start from precomputed kickstart data

https://apc.u-paris.fr/~sartirana/LISA/GlobalFit/data/kickstart_short_powell/

- run @ trex

- additional options

in the following

- `/work/data/globalfit` is the location for data common to all runs
- `/work/data/globalfit/gfrun` is the location for this specific test run

```
usage: run_gfit.py [-h] [-i IDS] [-o OFFSET] [--bindb BINDB] [--mbhb-index MBHB_INDEX] [--noise-pars NOISE_PARS] [-d DATA_DIR] [-b BASE_DIR] [--script-dir SCRIPT_DIR]
[--root-dir ROOT_DIR] [--lsbt-inf-dir LSBT_INF_DIR] [-c SUB_CMD] [-s SUB_TPL] [-e ENV_FILE] [--nmonth NMONTH] [--niter NITER] [--iter ITER] [--gb-iter-per-job
GB_ITER_PER_JOB] [--no-noise] [--no-gbs] [--clean-central] [--no-mbhbs] [--mock-run] [--fake-gb] [--sync SYNC] [-n] [--debug]
```

options:

```
-h, --help                show this help message and exit
-i IDS, --ids IDS         GB bin identifier. Default: 1-5
-o OFFSET, --offset OFFSET
                        Bin identifier offset. Default: 1644 (around 4mHz)
--bindb BINDB             GB bins database. Default /work/soft/global-fit/test/bindb.npy
--mbhb-index MBHB_INDEX
                        MBHB merger index. Default: '1-3'
--noise-pars NOISE_PARS
                        File .npy with the initial noise parameters. Default: use default noise parameters
-d DATA_DIR, --data-dir DATA_DIR
                        Input data location. Default: ./data
-b BASE_DIR, --base-dir BASE_DIR
                        Output data location. Default: ./gfrun
--script-dir SCRIPT_DIR
                        Blocks execution scripts location. Default: /work/soft/global-fit/globalfit
--root-dir ROOT_DIR      Root Blocks outputs directory. Default BASE_DIR/data.
--lsbt-inf-dir LSBT_INF_DIR
                        Lisabeta inference directory (default from the current installation)
-c SUB_CMD, --sub-cmd SUB_CMD
                        Submission command. Default: qsub -keo SUBSCRIPT
-s SUB_TPL, --sub-tpl SUB_TPL
                        Submission script template. Default: /work/soft/global-fit/test/run_gfit.d/qsub.sh
-e ENV_FILE, --env-file ENV_FILE
                        File setting up the local environment. Default: /work/soft/global-fit/test/run_gfit.d/env_cnes.sh. Set to None to deactivate.
--nmonth NMONTH          Number of months considered for the analysis. Default: 12
--niter NITER            Number of steps for discovering new sources. Default: 6
--iter ITER              First step index. Default: 1
--gb-iter-per-job GB_ITER_PER_JOB
                        Iterations for each GB job. Default: all iterations
--no-noise               Do not run noise fit.
--no-gbs                 Do not run GBs fit.
--clean-central          Start from a clean central GB catalog.
--no-mbhbs               Do not run MBHBs fit.
--mock-run               Quick mock run to test the workflow.
--fake-gb                Use fake fitting for GBs.
--sync SYNC              Comma separated sequence of synchronous blocks. E.g noise,gb means that the gb blocks will wait for the last iteration of the noise block to
                        start a new iteration. Default None (all async)
-n, --nosub              Do not submit jobs.
--debug                  Debug verbosity.
```

Local run

```
copy kickstart data into /work/data/globalfit/gfrun/data
python test/run_gfit.py -d /work/data/globalfit
                        -b /work/data/globalfit/gfrun
                        -c 'chmod u+x SUBSCRIPT;./SUBSCRIPT &'
                        -s test/run_gfit.d/local.sh
                        -e None
                        -i 0,1,2 --clean-central
                        --no-mbhb
                        --no-noise
                        --niter 5
```

Run @ trex

```
python test/run_gfit.py -d /work/data/globalfit
                        -b /work/data/globalfit/gfrun
                        -c 'sbatch --account=lisa SUBSCRIPT '
                        -s test/run_gfit.d/slurm_sub.sh
                        -e test/run_gfit.d/env_trex.sh
                        -i 0,1,2 --clean-central
                        --gb-iter-per-job 1
                        --no-mbhb
                        --no-noise
                        --niter 5
```