KAGALI PE development mid-term report

Lee Hyung Won, Inje University

15 January, 2018

Contents

- Codes Developed
- How to build
- Some results
- Future works

KGLInferenceReadData.c

- Read frame data file
  - Resample data file for time series
  - Evaluate PSD from frame data using Welch method(MedianMean)
- Inject a template wave form
  - Injection file read
  - Noise realization

Options implemented

- $./inferenceMCMC --help shows all options

```
#define USAGE  "\
    ==== Options for KGLInferenceReadData related to MCMC ========\n\
    --srate SRATE : sampling rate, we assume all detector has the same value [4096, Hz] \n\
    --seglen SEGLEN : segment length, we assume all detector has the same value [16, s] \n\
    --ifo H1,K1 : comma separated list of IFOs \n\
    --flow f1,f2 : comma separated list of low frequencies[defaut:10Hz], integration range in FD is flow~fhigh\n\
    --fhigh f1,f2 : comma separated list of high frequencies[default:srate/2 - 1/seglen]\n\
    --strain file1,file2 : comma separated list of strain frame files names\n\
                                file1.gwf, file2.gwf for real strain frame files, file should exist in the current directory\n\
                                approx1, approx2 for injection waveforms, usually these are the same\n\
    --channel channel1,channel2 : comma separated list of channel names when given real strain frame data files\n\
    --psd psd1,psd2 : comma separated list of psd options[default: use strain frame files]\n\
                        sim1, sim2 for simulated PSD data\n\
                            possible values are SimIniLIGO        : initial LIGO analytic function\n\
                                                SimAdvLIGO        : advanced LIGO analytic function\n\
                                                SimAdvVIRGO       : advanced VIRGO analytic function\n\
                                                SimGEO            : GEO analytic function\n\
                                                SimEGO            : EGO analytic function\n\
                                                SimTAMA           : TAMA analytic function\n\
                                                SimETB            : Einstein B analytic function\n\
                                                DesigniKAGRA      : Design sensitivity PSD file for iKAGRA(unknwon)\n\
                                                DesignbKAGRA-vrsed : Design sensitivity PSD file for bKAGRA(BW2009_VRSED.dat)\n\
                                                DesignbKAGRA-vrseb : Design sensitivity PSD file for bKAGRA(BW2009_VRSEB.dat)\n\
                                                DesignaLIGO-bhbh   : Design sensitivity PSD file for adv. LIGO(AdvLIGO_BHBH_20deg.dat)\n\
                                                DesignaLIGO-hf     : Design sensitivity PSD file for adv. LIGO(AdvLIGO_High_freq.dat)\n\
                                                DesignaLIGO-no-srm : Design sensitivity PSD file for adv. LIGO(AdvLIGO_NO_SRM.dat)\n\
                                                DesignaLIGO-nsns   : Design sensitivity PSD file for adv. LIGO(AdvLIGO_NSNS_Opt.dat)\n\
                                                DesignaLIGO-zdhp   : Design sensitivity PSD file for adv. LIGO(AdvLIGO_ZERO_DET_high_P.dat)\n\
                                                DesignaLIGO-zdlp   : Design sensitivity PSD file for adv. LIGO(AdvLIGO_ZERO_DET_low_P.dat)\n\
                        file1.dat, file2.dat for known or private PSD data files\n\
                        file1.gwf, file2.gwf for real strain frame files from which calculate PSD(ex. average, etc.)\n\
    --psdchannel channel1,channel2 : comma separated list of channel names when given real strain frame data files for psd[default:use channel option]\n\
    --psdstart gps1,gps2 : comma separated list of gps times for psd start[default:start of frame file]\n\
    --psdlength len1,len2 : comma separated list of length in seconds for psd[default:GPSTrig - seglen + TRIG_MARGIN - psdStart]\n\
    --psdresample : use resampled frame data for PSD calculation[default:no]\n\
    --window hann,rectangular : comma separated list of window function for each IFOs[default:hann]\n\
    --trigtime trigger : time in 113847.5747 style in geocenter \n\
    --inj inj_file : injection data file name, this is not xml file it is plain text file similar to *.ini file.\n\
                        injection file data override --strain option for approximants\n\
    --noise n1,n2 : comma separated noise realization for IFOs, -1:random seed, 0:no-noise, #:given seed[defaut:0]\n\
    --inj_id n : injection id in the given injection file.\n\
    --dumptime : dump time series data[default:no]\n\
    --dumpfreq : dump frequency series data[default:no]\n\
    --dumppsd : dump psd and asd data[default:no]\n\
    ========================================================\n\
"
```

--ifo option

- --ifo H1,K1
- comma separated list of IFOs

- Example
- --ifo H1,L1,V1,K1

--strain option

- --strain file1,file2
- comma separated list of strain frame files names
- file1.gwf, file2.gwf for real strain frame files, file should exist in the current directory
- approx1, approx2 for injection waveforms, usually these are the same

- Example
- --strain H-H1_HOFT_C00_T1700401_v1-1187005000-4096.gwf,L-L1_HOFT_C00_T1700401_v1-1187005000-4096.gwf,V-V1O2Repro1A-1187005000-5000.gwf,TaylorF2

--channel option

- --channel channel1,channel2
- comma separated list of channel names when given real strain frame data files

- Example
- --channel H1:GDS-CALIB_STRAIN_T1700401_v1,L1:GDS-CALIB_STRAIN_T1700401_v1,V1:Hrec_hoft_V1O2Repro1A_16384Hz,V1:Hrec_hoft_V1O2Repro1A_16384Hz

--psd option

- --psd psd1,psd2
- comma separated list of psd options[default: use strain frame files]

Other options

- Other option can be understandable or figure out from help message

ReadData()

- if --help is given then print help and stop
- Get number of IFOs(--ifo)
- Set trigger time (--trigtime) it is mendatory
- Read injection table if option(--inj) given and show
- For each IFO do
  - Set IFO epoch as trigger time
  - Set IFO Data
  - Update network SNR
- Print network SNR
- Destroy Injection table

SetIFOData()

- Check if the prefix exist
- Set IFO frequency range (--flow, --fhigh)
- Set window function for IFO (--window)
- Set strain data (--strain)
- Set PSD data (--psd)
- Apply time shift for IFO
- Set white data for IFO only for real frame data
- Set noise realization for IFO only for injection data
- Dump data

SetIFODataFrequency()

- If --flow is given then set ifo->fLow as that value
- Else set ifo->fLow as DEFAULT_FLOW(10.0Hz)

- If --fhigh is given then set ifo->fHigh as that value
- Else set ifo->fHigh as srate – 1.0/seglen

- fLow and fHigh is actual integration range for snr and likelihood calculation

SetIFODataWindowFunction()

- If --window option is given set ifo->window as that value
- Else set DEFAULT_WINDOW(Hann)

- This window function will be used for PSD calculation and windowedTimeSeries data

SetIFOStrainData()

- If --strain option is not given abort the program
- If approximant name is given
  - Check exist injection table
  - Set ifo->srate as --srate option value or default value(4097Hz)
  - Set ifo->seglen as --seglen option value or default value(16s)
  - Set appropriate frequency range
  - Call template function
  - Set response and time shift
  - Set type as Injected data
- Else *.gwf is given
  - Read frame data(next slide)
- Else abort the prorgam

Read frame data

- Check channel name is given (--channel)
- Check trigger time is within the frame data
- Set srate and seglen
- Resample frame data
- Set time data
- Set frequency data
- Set windowed time data
- Set windowed frequency data
- Set type as frame data

SetIFOPsdData()

- If --psd option is given
  - Set psd data
- Else if --strain option is given
  - If frame file is given
    - Estimate psd using frame data
  - Else if approximant is not gven
    - Set psd using "aLIGO-anl"
  - Else
    - Abort the program
- Else
  - Abort the program

SetIFOWhiteData()

- If type is inject data, then return
- Calculate white frequency data from frequency data
- Calculate reverse FFT to get white time data

ApplyIFOTimeShift()

- Apply time shift in frequency domain
- Calculate SNR for this IFO

SetIFONoiseData()

- If type is frame data, then return
- If --noise option is not given or -1 then
  - Set noise seed as random
- Else
  - Set noise seed as the given value
- Add Gaussian noise in frequency data

Dump data

- If --dumptime option is given
  - Dump time data (file : IFO_time_series_check.dat, IFO is H1, K1)
- If --dumpfreq option is given
  - Dump frequency data (file : IFO_freq_series_check.dat, IFO is H1, K1)
- If --dumppsd option is given
  - Dump power spectral data (file : IFO_psdasd_series_check.dat, IFO is H1, K1)

Directory structure
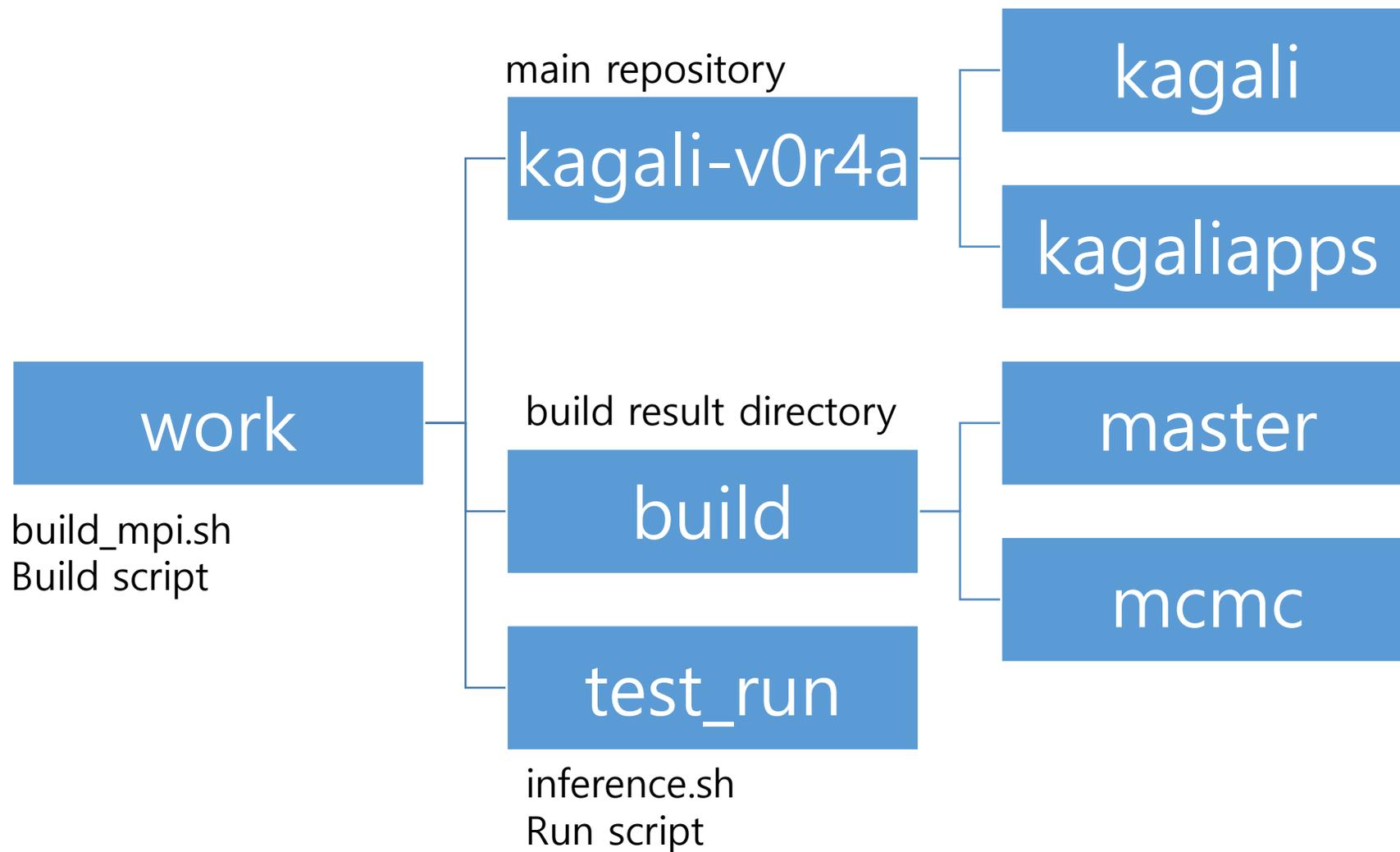
Preparing source

- Make a working directory, like work
- Clone kagali
- $cd work
- $git clone https://vt001.resceu.s.u-tokyo.ac.jp/git/kagali-v0r4a

- Check out mcmc branch
- $git fetch origin
- $git checkout –b mcmc origin/mcmc

Build application

- $cd work
- $nohup ./build_mpi.sh &
- $tail –f nohup.out
- Sample build result

```
rm -f config.h stamp-h1
rm -f libtool config.lt
rm -f TAGS ID GTAGS GRTAGS GSYMS GPATH tags
rm -f cscope.out cscope.in.out cscope.po.out cscope.files
This command is intended for maintainers to use
it deletes files that may require special tools to rebuild.
make[1]: Leaving directory '/home/hwlee/projects/KGL/kagali/kagali-v0r4a/kagaliapps'
rm -f config.status config.cache config.log configure.lineno config.status.lineno
rm -rf ./autom4te.cache
rm -f Makefile




==========*********************
 built for branch mcmc
 Do command source /home/hwlee/projects/KGL/kagali/build/mcmc/etc/kagali-user-env.[c]sh to use this software
2018. 01. 15. (월) 11:41:29 KST
==========*********************
```

Run application

- $cd test_run
- $./inference.sh
- Sample result

```
xArm Altitude    :         3.141400e-03[rad]
xArm Azimuth     :         1.054113e+00[rad]
yArm Altitude    :        -3.627000e-03[rad]
yArm Azimuth     :        -5.166798e-01[rad]
xArm Midpoint    :         1.513254e+03[m]
yArm Midpoint    :         1.511611e+03[m]
=====================================================
ifoData = 0x14ade20
cmdLine = 0x14aa610
RunState = 0x14a6fb0
RunState = 0x14a6fb0
RunState = 0x14a6fb0
 ==== This is thread 0 of 1 ====
 ======= run MCMC sampler ======= for thread 0 of 1
KGLInferenceOneStep called
KGLInferenceOneStep called
KGLInferenceOneStep called
KGLInferenceOneStep called
KGLInferenceOneStep called
KGLInferenceOneStep called
KGLInferenceOneStep called
KGLInferenceOneStep called
KGLInferenceOneStep called
KGLInferenceOneStep called
KGLInferenceOneStep called
Thread 0 has 11 effective samples. Stopping...
 ========== main(): finished. ==========
```

Some graphs(using gnuplot)
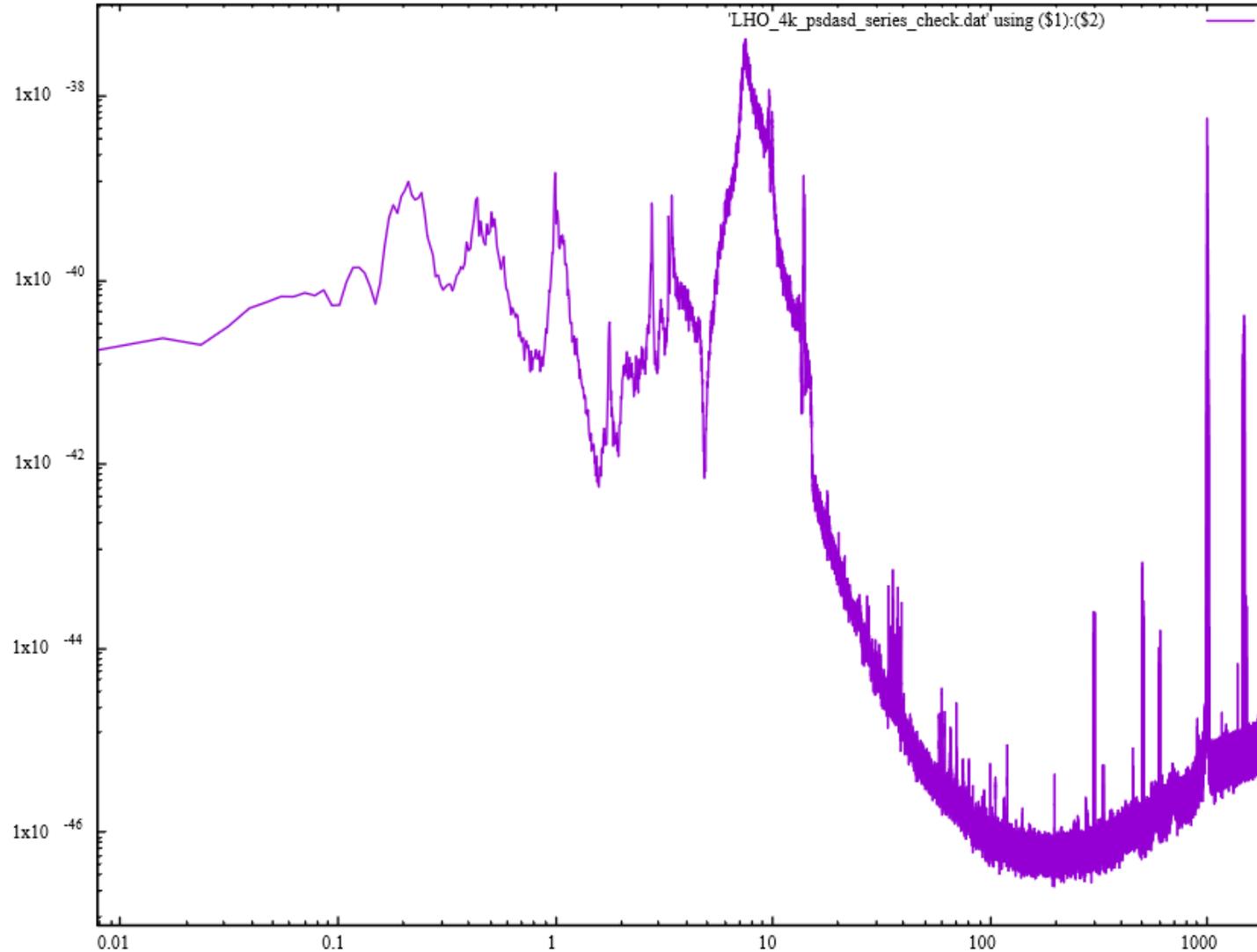
- PSDs

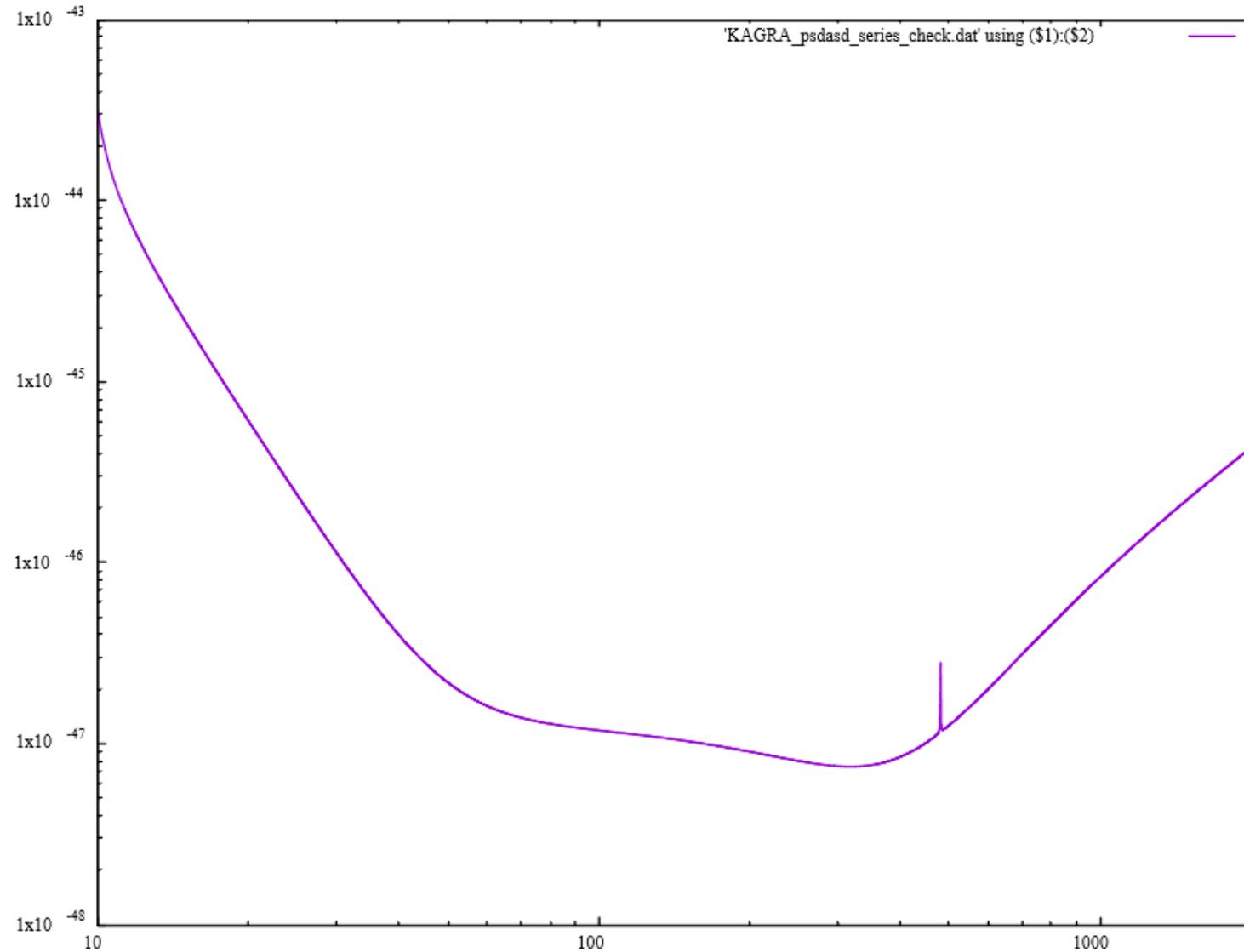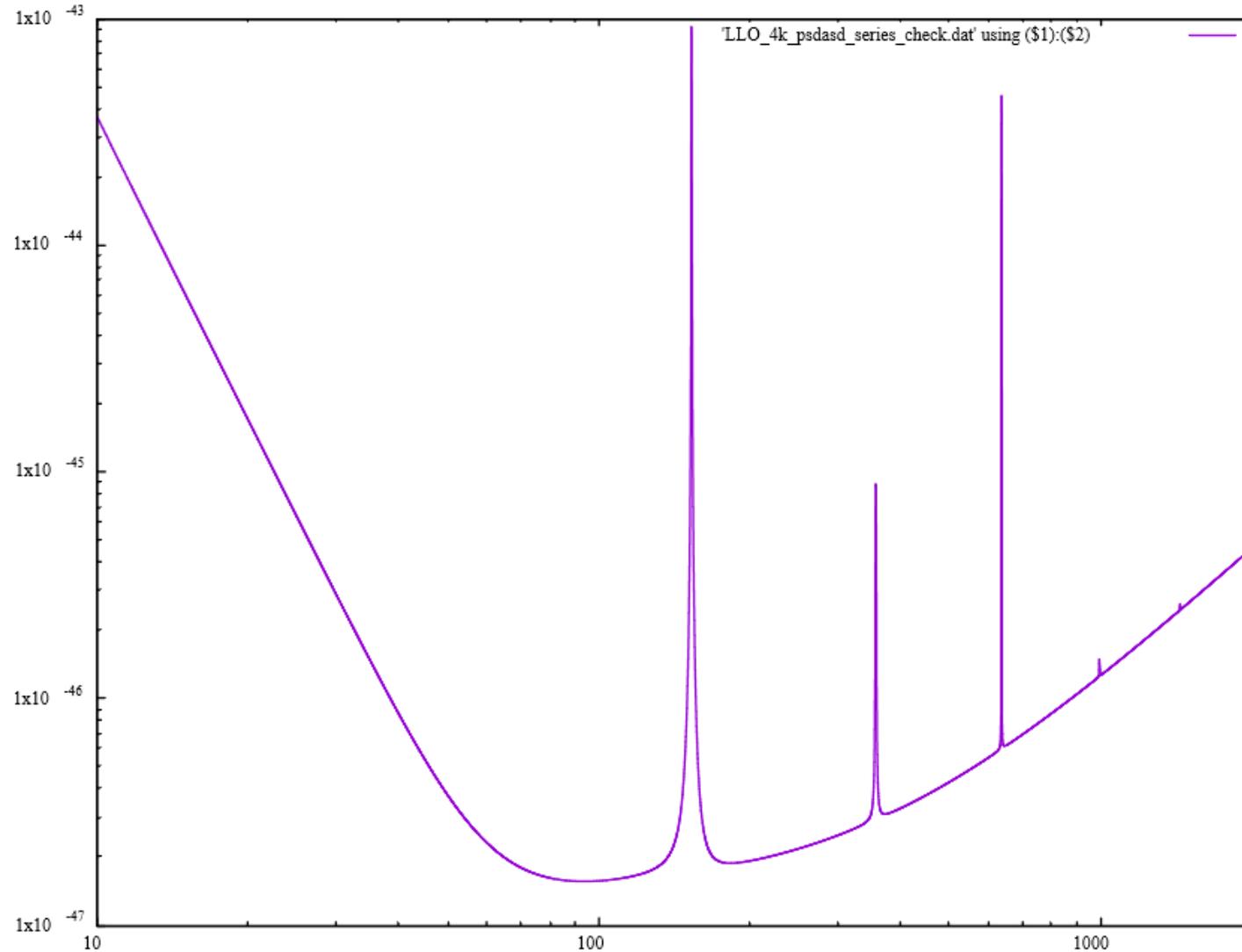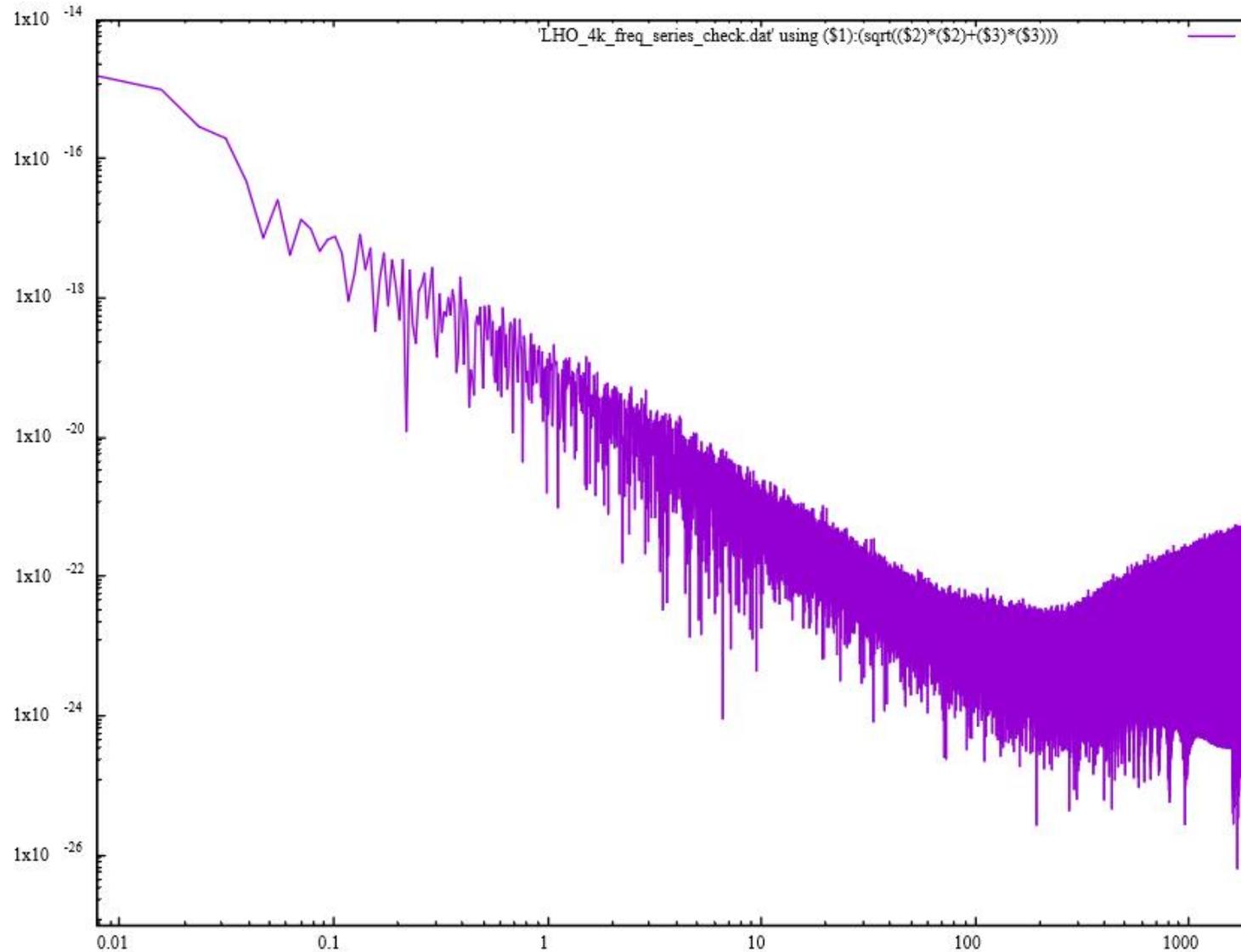# Some graphs(using gnuplot)

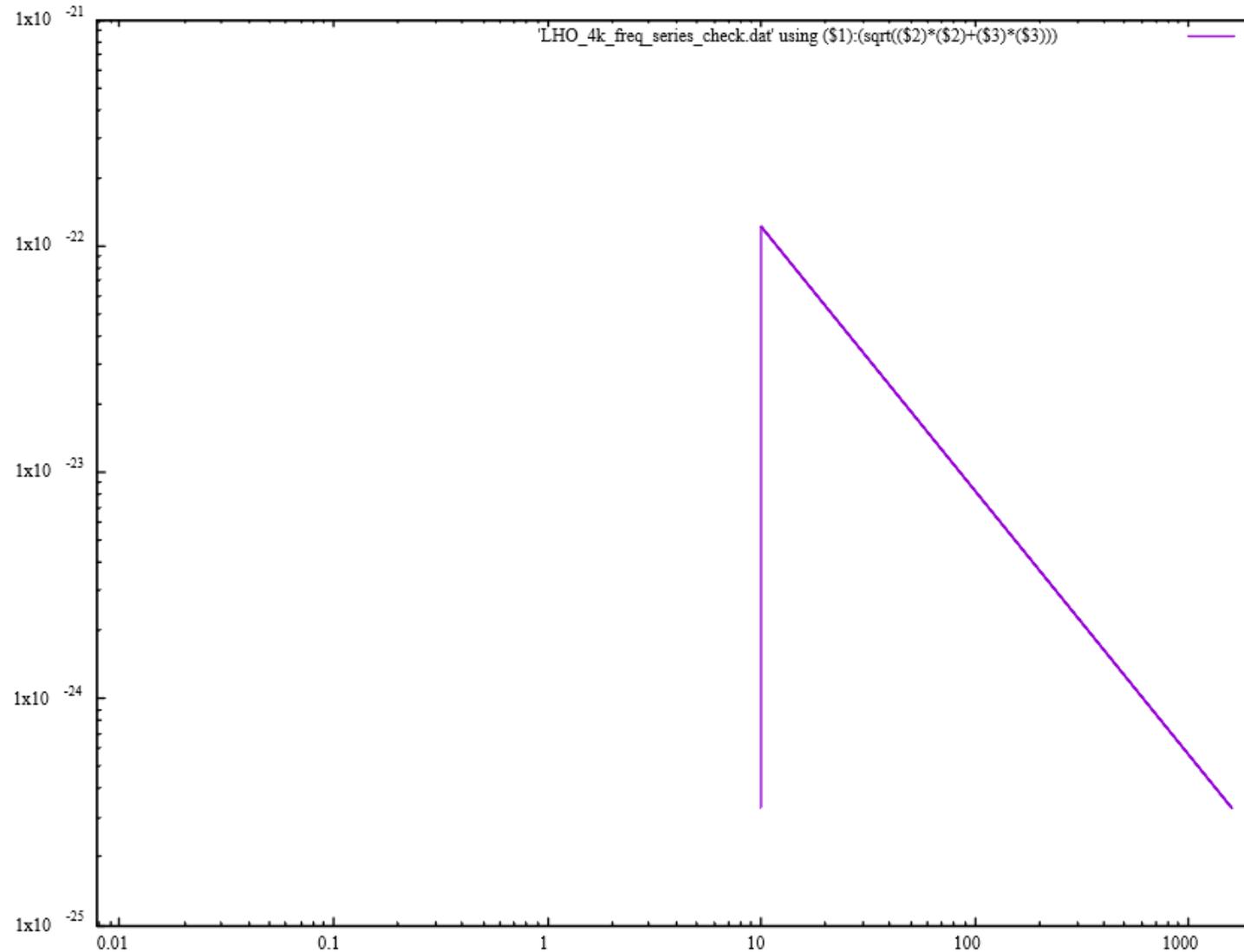• PSDs

Some graphs(using gnuplot)

- PSDs

Some graphs(using gnuplot)

- PSDs

Some graphs(using gnuplot)

- PSDs

Future works

- InitCBC implementation
  - Likelihood function implementation
  - Prior function implementation
  - Proposal function implementation
- Parallel tempering implementation
- Main MCMC implementation

- Postprocessing implementation