BBlack:
Bayesian analysis for astrophysics

- 22th January 2024 -

Carole Périgois

BBlack : Motivation and structure

Aims: - Comparing astrophysical models with gravitational wave observations
- Multichannel analysis

Steps for an analysis

1) Preparation of the models
2) Compare models and observations with bayesian quantities
3) Compute likelihood of each models
4) Compute multichannel analysis

run.py

```python
import json
import Run.getting_started as GS # Initiate the file Params.json
import Run.advanced_params as AP
import astrotools.AstroModel as AM
import bayesiantools.bayes_model as BA
import bayesiantools.bayesian_computation as BC


if __name__ == '__main__':
    params = json.load(open('Run/Params.json', 'r'))
    for m in params['astro_model_list'].keys():
        astromodel = AM.AstroModel(name=m)
        astromodel.generate_samples()
        BA.process_bayes_model(astromodel)
        BC.compute_likelihood(astromodel)
    mc = params['compute_multi_channel']
    for key in mc.keys():
        BC.multichannel_analysis(name=key)
    AP.clean()
```

Prepare astromodels : from CosmoRate outputs to catalog samples

```
(base) perigois@scighera:~/Stefano_GC/field$ ls
catalogs               MRD_spread_9Z_40_No_MandF2017_0.3_No_No_0.dat
field_vcm_1_50.dat  Sampled_parameters_9Z_40_No_MandF2017_0.3_No_No_0.dat
field_vcm_2_50.dat  Zperc_z_9Z_40_No_MandF2017_0.3_No_No_50_0.dat
```

```python
astromodel_1 = {'name': "Example1",
                'path_to_catalogs': "./../BBHs_m01/",
                'path_to_MRD': './../BBHs_m01/MRD_spread_9Z_40_No_MandF2017_0.3_No_No_0.dat'}
```

- Read and organize original files
- Compute new variables
- Build catalogs
- Resample the catalogs

```python
import Run.getting_started as GS # Initiate the file Params.json

if __name__ == '__main__':
    params = json.load(open('Run/Params.json', 'r'))
    for m in params['astro_model_list'].keys():
        astromodel = AM.AstroModel(name=m)
        astromodel.generate_samples()
        BA.process_bayes_model(astromodel)
        BC.compute_likelihood(astromodel)
    mc = params['compute_multi_channel']
    for key in mc.keys():
        BC.multichannel_analysis(name=key)
    AP.clean()
```

Process bayes model : comparison with GW data

For each GW event k :

$$\mathcal{I}^k = \int \mathcal{L}^k(h^k|\theta)\, p(\theta|\lambda)\, \mathrm{d}\theta \approx \frac{1}{N_s^k} \sum_{i=1}^{N_s^k} \frac{p(\theta_i^k|\lambda)}{\pi^k(\theta_i^k)},$$

For each model compute the efficiency

$$\beta(\lambda) = \frac{\mu_\lambda}{N_\lambda}$$

run.py

```
import json
import Run.getting_started as GS # Initiate the file Params.json
import Run.advanced_params as AP
import astrotools.AstroModel as AM
import bayesiantools.bayes_model as BA
import bayesiantools.bayesian_computation as BC


if __name__ == '__main__':
    params = json.load(open('Run/Params.json', 'r'))
    for m in params['astro_model_list'].keys():
        astromodel = AM.AstroModel(name=m)
        astromodel.generate_samples()
        BA.process_bayes_model(astromodel)
        BC.compute_likelihood(astromodel)
    mc = params['compute_multi_channel']
    for key in mc.keys():
        BC.multichannel_analysis(name=key)
    AP.clean()
```

Likelihood computation and multichannel analysis

$$\mathcal{L}(\lambda|\mathcal{H}) = \prod_{k=1}^{N_{\text{obs}}} \frac{\mathcal{I}^k}{\beta(\lambda)} N(\lambda),$$

$$p(\lambda|\mathcal{H}) \sim \pi(\lambda) \prod_{k=1}^{N_{\text{obs}}} \frac{\mathcal{I}^k}{\beta(\lambda)},$$

run.py

```python
import json
import Run.getting_started as GS # Initiate the file Params.json
import Run.advanced_params as AP
import astrotools.AstroModel as AM
import bayesiantools.bayes_model as BA
import bayesiantools.bayesian_computation as BC


if __name__ == '__main__':
    params = json.load(open('Run/Params.json', 'r'))
    for m in params['astro_model_list'].keys():
        astromodel = AM.AstroModel(name=m)
        astromodel.generate_samples()
        BA.process_bayes_model(astromodel)
        BC.compute_likelihood(astromodel)
    mc = params['compute_multi_channel']
    for key in mc.keys():
        BC.multichannel_analysis(name=key)
    AP.clean()
```

Structure and example

```
BBlack
├── astrotools
│   ├── __init__.py
│   ├── AstroModel.py
│   ├── auxiliary_cosmorate.py
│   └── utility_function.py
├── AuxiliaryFiles
│   ├── LVC_data
│   │   ├── Posterior
│   │   │   └── GW190403_post.dat
│   │   └── Prior
│   │       └── GW190403_prior.dat
│   ├── observing_runs_info
│   │   └── O1_events.csv
│   ├── PSDs
│   │   └── Livingston_O1.dat
│   └── Pw_single.dat
├── bayesiantools
│   ├── __init__.py
│   ├── bayes_model.py
│   ├── bayesian_computation.py
│   └── process_bayes_model.py
├── GWtools
│   ├── __ini__.py
│   ├── detector.py
│   ├── gw_event.py
│   └── observing_run.py
├── Run
│   ├── advanced_params.py
│   └── getting_started.py
├── venv
└── run.py
```