

CosmoBit tutorial B – solutions

November 4, 2020 – TOOLS 2020

Patrick Stöcker (stoecker@physik.rwth-aachen.de)

- Create a copy of `yaml_files/CosmoBit_quickstart.yaml` and call it `yaml_files/CosmoBit_tutorial_B.yaml`. This is straight forward.

- All files of the scan should be redirected to `runs/CosmoBit_tutorial_B`. Can you please change this?

- So far, there is no ΔN_{eff} assumed. I would like you to include a model that represents some dark radiation content that is the same at BBN and at recombination. It should be scanned with a flat prior in the range `[0.0, 2.0]`.

- So far, we assume that the CMB temperature is constant. Can you please change it such that it is scanned over with a Gaussian prior according to the FIRAS measurement $2.72548 \text{ K} \pm 0.00057 \text{ K}$.

- Now I want you to include the Planck likelihoods. In particular I want you to include the `highl_TTTEEE`, `lowl_TTEE`, and `lensing` likelihood of the 2018 data release. Keep in mind that the Planck likelihoods come with nuisance parameters that need to be scanned with appropriate priors. In the folder `yaml_files/include/` you will find helpful files.

```

# Lyman alpha data:
eBOSS_DR14_Lya_combined: default

# A breakdown of each likelihood component in the above total lnL,
# plus any others that you might like to compute but not include in the
# likelihood.
- purpose: Observable
  capability: MP_LogLikelihoods
  type: map_str_dbl

# Add Omega0_m to output data file
- purpose: Observable
  capability: Omega0_m
  function: get_Omega0_m_classy

#----- 7) Rule entries -----
Rules:
# ----- 7.a) BBN Likelihoods, AlterBBN interface -----
# (CosmoBit Module paper, Appendix C.1)

# error calculation precision settings for AlterBBN
# (use failsafe 1 and 0 to speed up Helium abundance calculation for testing)
# (For more precise calculations, use 3 or 7)
# Here, we choose err: 0, as the error calculation is dictated by the rule below
- capability: AlterBBN_Input
  options:

# Add Omega0_m to output data file
- purpose: Observable
  capability: Omega0_m
  function: get_Omega0_m_classy

# ----- 6.c) Planck Likelihoods -----
- purpose: LogLike
  capability: Planck_nuisance_prior_loglike

- purpose: LogLike
  capability: Planck_sz_prior_loglike

- purpose: LogLike
  capability: Planck_highl_loglike
  function: function_Planck_highl_TTTEEE_2018_loglike

- purpose: LogLike
  capability: Planck_lensing_loglike
  function: function_Planck_lensing_2018_loglike

- purpose: LogLike
  capability: Planck_lowl_loglike
  function: function_Planck_lowl_TTEE_2018_loglike

#----- 7) Rule entries -----
Rules:

```

- I would like that the result of MontePython's sh0es likelihood is included as an observable but should not be used to drive the scan. Can you help me here?

```

# A breakdown of each likelihood component in the above total lnL,
# plus any others that you might like to compute but not include in the
# likelihood.
- purpose: Observable
  capability: MP_LogLikelihoods
  type: map_str_dbl

# Add Omega0_m to output data file
- purpose: Observable
  capability: Omega0_m
  function: get_Omega0_m_classy

#----- 7) Rule entries -----
Rules:

# A breakdown of each likelihood component in the above total lnL,
# plus any others that you might like to compute but not include in the
# likelihood.
- purpose: Observable
  capability: MP_LogLikelihoods
  type: map_str_dbl
  sub_capabilities:
    sh0es: default

# Add Omega0_m to output data file
- purpose: Observable
  capability: Omega0_m
  function: get_Omega0_m_classy

#----- 7) Rule entries -----
Rules:

```

- Switch to the MultiNest scanner and reduce the setting to 500 live points and a tolerance of 0.1

```

#----- 5) Scanner setup -----
# different scanner settings, for details check chapter 6.7 of the GAMBIT manual (https://arxiv.org/abs/1705.07959)
# comparison of different sampling algorithms in https://arxiv.org/abs/1705.07959

Scanner:
# use random just for testing.
use_scanner: random

scanners:
multinest:
  plugin: multinest
  like: LogLike
  nlive: 5000
  tol: 0.01

random:
  plugin: random
  point_number: 2
  like: LogLike

#----- 5) Scanner setup -----
# different scanner settings, for details check chapter 6.7 of the GAMBIT manual (https://arxiv.org/abs/1705.07959)
# comparison of different sampling algorithms in https://arxiv.org/abs/1705.07959

Scanner:
# use random just for testing.
use_scanner: multinest

scanners:
multinest:
  plugin: multinest
  like: LogLike
  nlive: 500
  tol: 0.1

random:
  plugin: random
  point_number: 2
  like: LogLike

```

- Now that everything is prepared, disable the debug messages

```

#----- 9) Generic Name/Value Section -----
KeyValues:
# if set to true, the log files will contain more information, e.g.
# the complete input dictionary passed to CLASS for each parameter point
debug: true

# where to save the output

#----- 9) Generic Name/Value Section -----
KeyValues:
# if set to true, the log files will contain more information, e.g.
# the complete input dictionary passed to CLASS for each parameter point
debug: false

# where to save the output

```