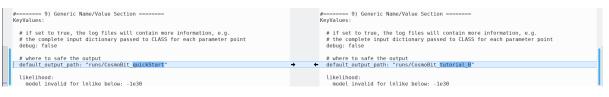
CosmoBit tutorial B – solutions November 4, 2020 – TOOLS 2020

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 $\Delta N_{\rm eff}$ 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.

# unu1_tho: # prior_type: flat # range: [-1.0,1.0]	unuwns: prior_type: flat range: [0.0,2.0]	
#=====================================	 # nuisance parameters in Planck likelihood cosmo nuisance Planck_TITEEE: !import include/Planck_2018/cosmo_nuisance_Planck_TITEEE_flat.yaml 	
# you can also have an extra section specifying priors. Or just do it a: # for details check chapter 6.3 of the GAMBIT manual (https://arxiv.org, Priors:	#====== 3) Prior setup ======== # you can also have an extra section specifying priors. Or just do it as above	

<pre>#U_man alpha data: eBOSS_DR14_Lya_combined: default</pre>	# Add Omega@_m to output data file - purpose: Observable canability: Omega@_m	
# A breakdown of each likelihood component in the above total lnL,	function: get Omega0 m classy	
<pre># plus any others that you might like to compute but not include in the # likelihood. - purpose: Observable</pre>	<pre># 6.c) Planck Likelihoods</pre>	
- purpose: Observable capability: MP_LogLikes type: map str dbl	 purpose: LogLike capability: Planck nuisance prior loglike 	-
	capability. Flanck_nuisance_prior_cogrike	
<pre># Add Omega0_m to output data file - purpose: Observable</pre>	 purpose: LogLike capability: Planck sz prior loglike 	
capability: Omega0_m function: get_Omega0_m_classy	- purpose: LogLike	
	capability: Planck_highl_loglike function: function Planck_highl TTTEEE 2018 loglike	
#====== 7) Rule entries =======		
Rules:	- purpose: LogLike	
	capability: Planck_lensing_loglike	
# 7.a) BBN Likelihoods, AlterBBN interface # (CosmoBit Module paper, Appendix C.1)	function: function_Planck_lensing_2018_loglike	
# error calculation precision settings for AltterBBN	 purpose: LogLike capability: Planck lowl loglike 	
# (use failsafe 1 and θ to speed up Helium abundance calculation for testing)	function: function Planck low1 TTEE 2018 loglike	
# (For more precise calculations, use 3 or 7) # Here, we choose err: θ , as the error calculation is dictated by the rule below		1
- capability: AlterBBN Input	#======= 7) Rule entries =======	
options:	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?

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

<pre>#====================================</pre>		<pre>#====================================</pre>	
Scanner:		Scanner:	
# use random just for testing.		# use random just for testing.	
use_scanner: random	→ ←	use_scanner: multinest	
scanners: multinest: plugin: multinest like: Loglike nlive: 5000 tol: 0.01	→ ←	scanners: multinest: plugin:multinest like: LogLike nlive: 500 tol: 0.1	i
random: plugin: random point_number: 2 like: LogLike		random: plugin: random point_number: 2 like: Loglike	

• Now that everything is prepared, disable the debug messages

#=====================================		#====================================	
# 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	+	# 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: faise	
# where to safe the output		# where to safe the output	1