CosmoGen: A Genetic Algorithm Framework for Exploration of Dark Energy Dynamics

Diogo Castelão[1], Ismael Tereno[1]

1- Instituto de Astrofísica e Ciências do Espaço, Universidade de Lisboa

Abstract

We introduce CosmoGen, a computational framework developed in Python, that implements genetic programing (GP) and genetic algorithms (GA) from the Distributed Evolutionary Algorithm for Search and Optimization (DEAP) library, to generate and evaluate candidate cosmological models with varying dark energy components. The framework integrates the Boltzmann code CLASS and Bayesian inference (MontePython) to evaluate the physical validity of the candidates. We present a case study addressing cosmological tensions. Our approach provides a new method to explore the vast space of potential dark energy models and identify viable candidates based on their dynamical properties.

CosmoGen in a nutshell

Initial Population

- Dark energy component generated by a base set of operations: (e.g. Add, Mul, Sin, Inv).

Output:
- Symbolic expressions for the evolution of the dark energy component, with parameters Ω_{DT} and additional free parameters.

Pre-testing

- Simplify symbolic expression (Sympy simplification, Permutation check, reparametrization of constants)
- Check for duplicate equation.
- Calculate first and second derivatives.
- For different values of the free parameters (e.g. close to zero, close to unity, large value):
  - Check for complex values.
  - Check if values of equation of state over scale factor are restricted in the desired range.
  - Check if values of adiabatic sound speed over scale factor are restricted in the desired range.

Output:
- Viable candidate for dark energy density equation, first and second derivative.
- Select the best value for the free parameters.

Evaluate function

- CLASS modification:
  - Convert equations for CLASS notation
  - Modify background module (implement new fluid component by writing ρ(a), ρ'(a), ρ''(a). From these equations, the equation of state and adiabatic speed of sound are calculated).
  - Modify input module (replace Cosmological constant and calculate the Ω_{DT} value today).
  - Compile new version of CLASS.
  - Check compilation and the DE model using single test run with free parameters values of pre-testing.
- MontePython MCMC chain:
  - Write new parameter file setting the new parameters of the model, the standard cosmological parameters and the likelihoods to be used.
  - Run small MCMC chain for evaluating the DE candidate.

Output:
- best χ2 value as fitness value for the GP evolutionary method.

Next Population

- Candidates with high fitness value are selected.
- Apply evolutionary strategy for selected population (Mate and Mutate).

Output:
- Next Generation of candidates.

Final Population

- All individuals that passed the CLASS test, from all generations, are stored to form the final list of models.

Output: List of models sorted by fitness value.

Case Study

Ask CosmoGen to generate cosmological models that can alleviate the S_8 and H_0 tensions (and restricted to the case of unperturbed dark energy fluids). To this goal we set-up the following conditions, where a crucial aspect is to set the likelihood used in the procedure to the one of CMB (Planck 2018) multiplied by a H_0 prior (SH0ES) and a S_8 prior (DES)

Population:
- Size of Initial population: 2048
  - Generated by a base set of operations: Add, Sub, Mul, Pow, safe_Div, safe_Inv, Exp, Ln, Neg.
  - Parameters: Ω_{DT}, D.
- Number of generations: 8
- Number of selected candiates by generation (mu): 128
- Number of generated candidates for next generation (lambda): 512
- Mate probability: 0.5
- Mutate probability: 0.5

Pre-testing:
- Values of D for testing: (0.05, 0.95, 1000).
- EoS accepted range: (-1.5, 0).

Evaluate function:
- CLASS modification:
  - DE candidate replacing Cosmological constant: $H^2(a) = H_0^2(\Omega_m a^{-3} + \Omega_r a^{-4} + \Omega_{DT}(a))$
  - DE component is considered homogeneous.
- MontePython MCMC chain:
  - We use the combined likelihoods: CMB Planck 2018 data: Planck_high_l_TTTEEE_lite + Planck_low_l_EE + Planck_low_l_TT + H_0 prior (SH0ES value) + S_8 prior (DES value).
  - Free parameters: h, D, ω_{cdm}.
  - Fixed parameters: bestfit value of Planck 2018 for ΛCDM.
  - MCMC chain with 1200 steps.

Selected Population and Ouput:
- The method used was the (mu + lambda)-ES: A version of evolution strategy where children and parents together will define the population for the next iteration.

Results

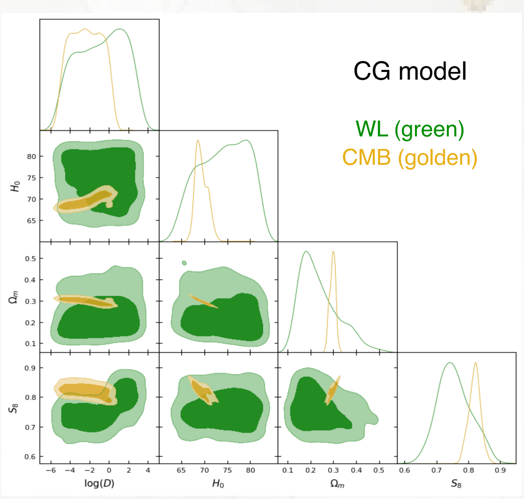
Top 5 models:

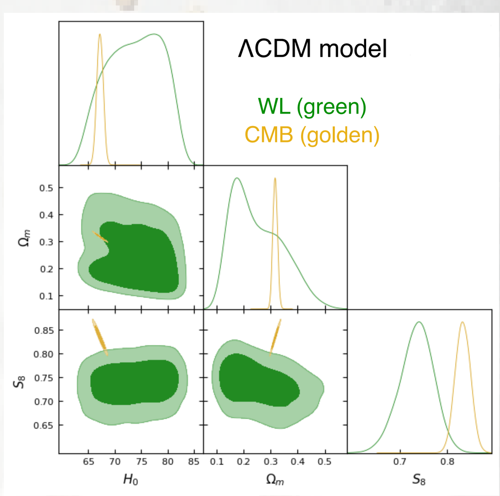
- Model 1: $\Omega_{DT}(a) = \Omega_{DT,0} / (D\,a - \ln(D\,a))$
- Model 2: $\Omega_{DT}(a) = \Omega_{DT,0} / (D\,a^{a} - a)$
- Model 3: $\Omega_{DT}(a) = \Omega_{DT,0} / (D^3 a^3 - \ln(a))$
- Model 4: $\Omega_{DT}(a) = \Omega_{DT,0} / (D\,a^{(\frac{D}{a})} - a)$
- Model 5: $\Omega_{DT}(a) = \Omega_{DT,0} / (D^3 a^{(a+2)} - \ln(a))$

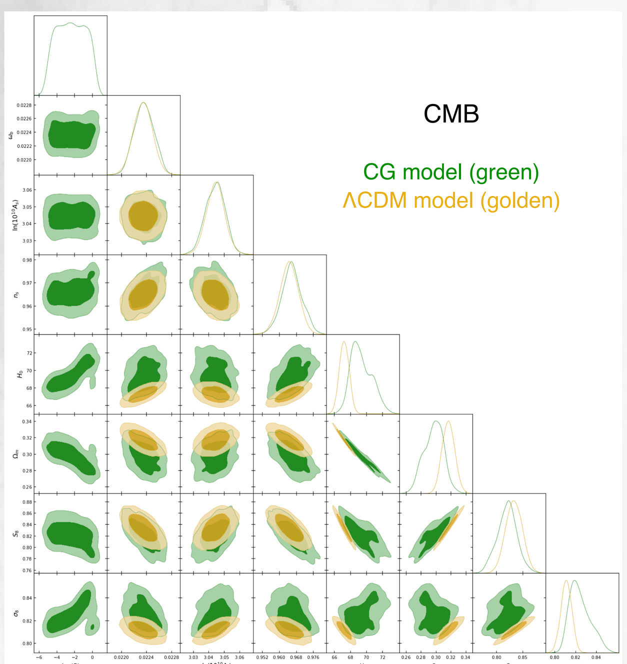
Analysis of the CosmoGen (CG) model: We select model 1 as the "CG model" to further explore. We computed its structure formation properties and tested them against data, verifying if the model actually has an impact on the H_0 and S_8 tensions as requested to CosmoGen.

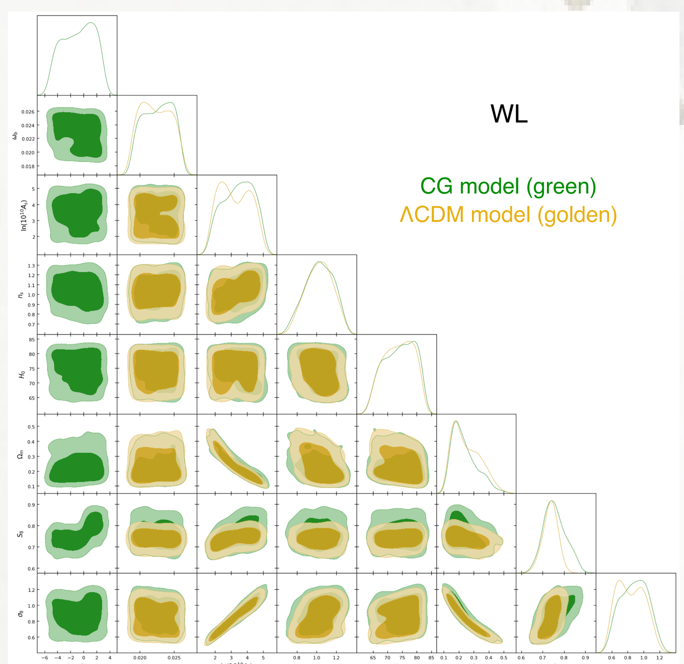
We performed a Nested Sampling analysis using PolyChords with the following free parameters: (parameter_D, h, ω_b, ω_{cdm}, A_s, n_s). All other parameters were fixed with the Planck 2018 results for ΛCDM. We tested the model against two observables:
- CMB Planck 2018 data: Planck_high_l_TTTEEE_lite + Planck_low_l_EE + Planck_low_l_TT .
- Weak lensing (WL) KiDS+VIKING-450 data.

The exact same analysis was made for the ΛCDM model for the sake of model comparison.









	Planck		KiDS	
Parameter	CG Model	ΛCDM	CG Model	ΛCDM
$\log(D)$	-2.3 ± 1.5	n.a.	$-0.8^{+3.2}_{-2.5}$	n.a.
ω_b	0.02236 ± 0.00015	0.04876 ± 0.00069	$0.0226^{+0.0028}_{-0.0020}$	$0.0401^{+0.0047}_{-0.0081}$
Ω_c	$0.249^{+0.013}_{-0.010}$	0.2582 ± 0.0076	$0.196^{+0.044}_{-0.11}$	$0.207^{+0.085}_{-0.11}$
H_0	$69.4^{+1.3}_{-1.7}$	67.88 ± 0.62	$74.2^{+7.1}_{-4.3}$	$73.8^{+6.4}_{-4.6}$
n_s	$0.9660^{+0.0045}_{-0.0050}$	0.9702 ± 0.0043	$1.03^{+0.17}_{-0.13}$	1.02 ± 0.13
$\ln 10^{10} A_s$	3.0443 ± 0.0061	3.1217 ± 0.0058	$3.46^{+1.3}_{-0.84}$	3.24 ± 0.93
Ω_m	$0.297^{+0.015}_{-0.012}$	0.3084 ± 0.0083	$0.239^{+0.046}_{-0.11}$	$0.253^{+0.066}_{-0.12}$
σ_8	$0.8246^{+0.0069}_{-0.013}$	0.8399 ± 0.0053	0.89 ± 0.16	$0.85^{+0.17}_{-0.22}$
S_8	$0.820^{+0.020}_{-0.018}$	0.852 ± 0.016	$0.757^{+0.045}_{-0.062}$	$0.737^{+0.038}_{-0.031}$
$\ln B_{CG,\Lambda}$	-1.27		-0.99	

Table 1: Mean and 68% uncertainty estimates of the 6 basis parameters and 3 derived parameters for the CG model and ΛCDM models by the 2 data sets. The Bayes factor used for model comparison $B_{CG,\Lambda}$ = Z(CG Model)/Z(ΛCDM) is also shown.

Conclusions: The model generated by CG tries indeed to solve the H_0 and S_8 tensions.
Figs.1 and 2 show a better agreement between the CMB and WL constraints of the CG model in comparison to the constraints of the ΛCDM model. Fig.4 shows this is caused by the CG model allowing higher values of S_8 for lower values of Ω_m (as shown in the S_8-Ω_m contour). In Fig.3 we see a shift in the H_0 distribution towards higher values. This behaviour allows for a better agreement with H_0 estimates from background observations.

Acknowledgements

The authors gratefully acknowledge financial support from IDPASC (FCT, Portugal) through PD/BD/150489/2019.

Ciências ULisboa

ia instituto de astrofísica e ciências do espaço

FCT Fundação para a Ciência e a Tecnologia

IDPASC