

---

# Coding for Precision Cosmology

---

Luciano Barosi  
UAF/CCT/UFCG  
@BINGO Collaboration

18 de Agosto de 2019

## Resumo

A pesquisa em cosmologia no s c. XXI   uma cosmologia baseada em uma grande quantidade de dados dispon veis e sua manipula  o por c digos dedicados. O objetivo deste curso   promover uma experi ncia *hands on* para a obten  o e an lise dos dados cosmol gicos dispon veis utilizando principalmente **python** e sua interface com os pacotes bem estabelecidos.

## 1 Objetivos

### 1.1 Objetivos Gerais

Permitir ao aluno uma compreens o geral das principais ferramentas utilizadas na pesquisa em cosmologia observacional.

### 1.2 Objetivos Espec ficos

- Indicar ferramentas de gerenciamento de sistemas
- Mostrar onde   poss vel obter ajuda para desenvolvimento e utiliza  o de c digos.
- Apresentar as principais bibliotecas Python: Numpy, Matplotlib, SciPy e Pandas
- Apresentar um panorama resumido de f sica de CMB
- Apresentar c digos Boltzmann
- Apresentar c digos de amostragem MonteCarlo

- Apresentar elementos de machine learning
- Apresentar mapmaking e pipelines para o tratamento de dados de radiotelescópios.

## 2 Conteúdo Programático

1. Ferramentas úteis: Git, Docker, Bash, Markdown,  $\LaTeX$ , Jupyter. Instalando o sistema de maneira agnóstica.
2. Conceitos básicos de python: filosofia, PEP8 [15], funções, classes, módulos.
3. Principais bibliotecas python: NUMPY [14], MATPLOTLIB [8], PANDAS [13], SCIPY [10], ASTROPY [16].
4. Dados cosmológicos disponíveis, WMAP [4], PLANCK [3], LIGO [1], SDSS [2], JLA [5].
5. Healpy [17].
6. Boltzmann codes: CAMB [12]
7. Boltzmann codes: CLASS [6]
8. Boltzmann codes: COSMOSIS [18]
9. Métodos de Montecarlo: COSMOMC [11], MontePython [7], Multi-nest. GetDist
10. Simulação de dados cosmológicos: MapMaking, efeitos atmosféricos, shot noise.
11. Análise de Séries Temporais (TOD): FFT, DFFT, efeitos de sampling.
12. Elementos de Machine Learning para Cosmologia [9].

## 3 Metodologia

As apresentações envolvem o conteúdo teórico mínimo necessário para a utilização dos códigos e o entendimento das operações, com a indicação clara da literatura necessária para o aprofundamento.

Todas as aulas serão realizadas com códigos ao vivo, em modelo *hands on*, no qual os alunos tem oportunidade de rodar os códigos em notebooks jupyter e tirar dúvidas e realizar pequenas modificações.

O curso é ministrado em uma semana corrida, 4h por dia, com intervalo de uma semana, durante três semanas. Ao final de cada semana de aula os alunos recebem uma tarefa que devem apresentar na próxima semana de aula, motivo pelo qual é fundamental que eles tenham de fato uma semana livre logo em seguida para poderem se dedicar as atividades.

### 3.1 Recursos Computacionais

É fundamental que o aluno tenha um computador a disposição para realizar as atividades, durante as aulas. As aulas podem ser realizadas em laboratório de informática ou os alunos podem utilizar seus próprios dispositivos.

Os requisitos de sistema são: Um computador com qualquer sistema operacional, processador de pelo menos 4 núcleos, mínimo de 4Gb de memória RAM e 15Gb de espaço disponível.

Para a utilização de laboratórios de informática, o usuário deve ter permissão para criação de uma imagem Docker.

É necessário acesso a internet.

## 4 Avaliação

Cada aluno receberá três notas. A média final corresponde a média aritmética das notas.

1. Frequência nas aulas (normalizada entre 0 - 10)
2. tarefa da primeira semana
3. tarefa da segunda semana

Todas as tarefas devem ser entregues como um módulo em python e como um notebook jupyter que rode o módulo com explicações. O módulo deve rodar também standalone. As tarefas devem estar disponíveis no bitbucket segundo as boas práticas de criação de módulo PEP8.

As tarefas devem ser apresentadas em sala de aula.

## Referências

- [1] B. P. Abbott et al. «Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914». Em: *Classical and Quantum Gravity* 33.13, 134001 (jul. de 2016), p. 134001. DOI: 10.1088/0264-9381/33/13/134001. arXiv: 1602.03844 [gr-qc].

- [2] Bela Abolfathi et al. «The Fourteenth Data Release of the Sloan Digital Sky Survey: First Spectroscopic Data from the Extended Baryon Oscillation Spectroscopic Survey and from the Second Phase of the Apache Point Observatory Galactic Evolution Experiment». Em: *ApJS* 235.2, 42 (abr. de 2018), p. 42. DOI: 10.3847/1538-4365/aa9e8a. arXiv: 1707.09322 [astro-ph.GA].
- [3] Y. Akrami et al. «Planck 2018 results. I. Overview and the cosmological legacy of Planck». Em: (2018). arXiv: 1807.06205 [astro-ph.CO].
- [4] C. L. Bennett et al. «Nine-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Final Maps and Results». Em: *ApJS* 208.2, 20 (out. de 2013), p. 20. DOI: 10.1088/0067-0049/208/2/20. arXiv: 1212.5225 [astro-ph.CO].
- [5] M. Betoule et al. «Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples». Em: *A&A* 568, A22 (ago. de 2014), A22. DOI: 10.1051/0004-6361/201423413. arXiv: 1401.4064 [astro-ph.CO].
- [6] Diego Blas, Julien Lesgourgues e Thomas Tram. «The Cosmic Linear Anisotropy Solving System (CLASS). Part II: Approximation schemes». Em: *J. Cosmology Astropart. Phys.* 2011.7, 034 (jul. de 2011), p. 034. DOI: 10.1088/1475-7516/2011/07/034. arXiv: 1104.2933 [astro-ph.CO].
- [7] Thejs Brinckmann e Julien Lesgourgues. «MontePython 3: boosted MCMC sampler and other features». Em: *arXiv e-prints*, arXiv:1804.07261 (abr. de 2018), arXiv:1804.07261. arXiv: 1804.07261 [astro-ph.CO].
- [8] J. D. Hunter. «Matplotlib: A 2D graphics environment». Em: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.
- [9] Ž. Ivezić et al. *Statistics, Data Mining and Machine Learning in Astronomy*. Princeton, NJ: Princeton University Press, 2014.
- [10] Eric Jones, Travis Oliphant, Pearu Peterson et al. *SciPy: Open source scientific tools for Python*. [Online; accessed ]. 2001–. URL: <http://www.scipy.org/>.
- [11] Antony Lewis e Sarah Bridle. «Cosmological parameters from CMB and other data: A Monte Carlo approach». Em: *Phys. Rev. D* 66 (2002), p. 103511. DOI: 10.1103/PhysRevD.66.103511. arXiv: astro-ph/0205436 [astro-ph].
- [12] Antony Lewis, Anthony Challinor e Anthony Lasenby. «Efficient computation of CMB anisotropies in closed FRW models». Em: *ApJ* 538 (2000), pp. 473–476. DOI: 10.1086/309179. arXiv: astro-ph/9911177 [astro-ph].

- [13] Wes McKinney et al. «Data structures for statistical computing in python». Em: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56.
- [14] Travis Oliphant. *NumPy: A guide to NumPy*. USA: Trelgol Publishing. [Online; accessed <today>]. 2006–. URL: <http://www.numpy.org/>.
- [15] *PEP 8 – Style Guide for Python Code*. URL: <https://www.python.org/dev/peps/pep-0008/>.
- [16] A. M. Price-Whelan et al. «The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package». Em: *AJ* 156, 123 (set. de 2018), p. 123. DOI: 10.3847/1538-3881/aabc4f.
- [17] Andrea Zonca et al. «healpy: equal area pixelization and spherical harmonics transforms for data on the sphere in Python». Em: *Journal of Open Source Software* 4.35 (27 de mar. de 2019), p. 1298. ISSN: 2475-9066. DOI: 10.21105/joss.01298. URL: <http://dx.doi.org/10.21105/joss.01298>.
- [18] J. Zuntz et al. «CosmoSIS: Modular cosmological parameter estimation». Em: *Astronomy and Computing* 12 (set. de 2015), pp. 45–59. DOI: 10.1016/j.ascom.2015.05.005. arXiv: 1409.3409 [astro-ph.CO].