Group 3 : Nico, Ansh, Minshan, Rodrigo

Emails:

Nico: nemezare at ucdavis dot edu
Ansh: physrivastava@ucdavis.edu
Rodrigo: rbecerra@ucdavis.edu
Minshan: mszheng@ucdavis.edu

GitHub Repo: https://github.com/anshsrivastava/H0llyM0lly

Write here any comments:

Plan :

Green:Done

- Create repo and add data. Add collaborators. DONE
- Understand the context and models used: Understand distance modulus - redshift data and their relationship with the cosmological parameters, etc. Mostly done. | To finish this week. DONE BY APR28
- Understand MCMC algorithm. - Done
- Write MCMC code first and test in simple cases (HW as test).
    1. Nico will write general code to implement MCMC for any given posterior and generating function. DONE
    2. Ansh worksout the function for likelihood, priors as well as generators. By May 3.
    3. Nico works code to make banana plots/ contours. By May 3.
    4. Rodrigo writes a function to obtain luminosity distance By May 3.

- Minshan: Fit the data with MCMC. Check chain “sanity”. By May 5
- Nico : Do plots. By May 7
- Prepare the presentation. Starting May 5 and finish by May 8
    a. Nico : Intro Cosmology part
    b. Minshan : Bayesian+MCMC part
    c. Ansh : Method to solve the problem. Sanity checks.
    d. Rodrigo : Results/Conclusions.

- Everyone : Practice/Review of presentation. May 9.
- TBD : Present May 10th
- Everyone : Win.

Structure of Repository: (https://github.com/anshsrivastava/H0llyM0lly)

- Branches to separate our work and avoid conflicts during development.
- Main branch will contain the latest stable code.
- We will create a new branch everytime we want to add a new feature/functionality
- Once the feature is polished and ready to be merged into main, we will have a code review and when at least 2 people review the code, it will be accepted into the main branch. That way we can get good experience with Git, code reviews, and other good development practices.

Code Outline:

MCMC(seed, N_steps) -> dependent of likelihood/priors, and generator
RETURNS CHAIN

Generator(seed)
Posterior(parameters)
lum_distance(z,parameters)

Code Flow:

1. Read Data
2. Choose seeds/walkers
3. Iterate to get MCMC chain
4. Chain = [$(\Omega_{M1}, \Omega_{\Lambda 1}, \Omega_{K1})$,...]
5. Marginalize over Ω_k
    a. Flat Prior - means ignore the parameter Ω_k
6. Do contours