Next generation cosmological analysis with nested sampling

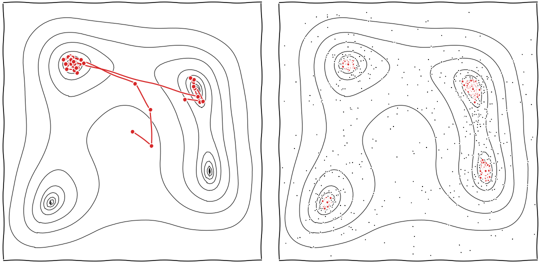
Will Handley
<wh260@cam.ac.uk>

Royal Society University Research Fellow & Turing Fellow
Astrophysics Group, Cavendish Laboratory, University of Cambridge
Kavli Institute for Cosmology, Cambridge
Gonville & Caius College
github.com/williamjameshandley/talks
willhandley.co.uk

8th September 2022

Overview

- DiRAC 2020 RAC allocation of 30MCPUh
- Main goal: Planck Legacy Archive equivalent
  - Parameter estimation $\to$ Model comparison
  - MCMC $\to$ Nested sampling
  - Planck $\to$ $\{$Planck, DESY1, BAO, $\ldots\}$
  - Pairwise combinations
- Suite of tools for processing these
  - `unimpeded` 1.0
  - `margarine` 1.0
  - `anesthetic` 2.0
  - `zenodo` archive
- MCMC chains also available.
- Work in progress, but α-testers requested (email wh260@cam.ac.uk)



The three pillars of Bayesian inference

Parameter estimation

What do the data tell us about the parameters of a model?

e.g. the size or age of a ΛCDM universe

$$P(\theta|D,M) = \frac{P(D|\theta,M)P(\theta|M)}{P(D|M)},$$

$$\mathcal{P} = \frac{\mathcal{L}\times\pi}{\mathcal{Z}},$$

$$\text{Posterior} = \frac{\text{Likelihood}\times\text{Prior}}{\text{Evidence}}.$$

Model comparison

How much does the data support a particular model?

e.g. ΛCDM vs a dynamic dark energy cosmology

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)},$$

$$\frac{\mathcal{Z}_\mathcal{M}\Pi_\mathcal{M}}{\sum_m Z_m\Pi_m},$$

$$\text{Posterior} = \frac{\text{Evidence}\times\text{Prior}}{\text{Normalisation}}.$$

Tension quantification

Do different datasets make consistent predictions from the same model?

e.g. CMB vs Type IA supernovae data

$$\mathcal{R} = \frac{\mathcal{Z}_{AB}}{\mathcal{Z}_A\mathcal{Z}_B},$$

$$\log\mathcal{S} = \langle\log\mathcal{L}_{AB}\rangle_{\mathcal{P}_{AB}} - \langle\log\mathcal{L}_A\rangle_{\mathcal{P}_A} - \langle\log\mathcal{L}_B\rangle_{\mathcal{P}_B}$$

Occam's Razor [2102.11511]

- Bayesian inference quantifies Occam's Razor:
  - *"Entities are not to be multiplied without necessity"* — William of Occam
  - *"Everything should be kept as simple as possible, but not simpler"* — "Albert Einstein"
- Properties of the evidence: rearrange Bayes' theorem for parameter estimation

$$\mathcal{P}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}} \qquad \Rightarrow \qquad \log \mathcal{Z} = \log \mathcal{L}(\theta) - \log \frac{\mathcal{P}(\theta)}{\pi(\theta)}$$

- Evidence is composed of a "goodness of fit" term and "Occam Penalty"

- RHS true for all θ. Take max likelihood value θ_*:

$$\log \mathcal{Z} = -\chi^2_{\mathrm{min}} - \text{Mackay penalty}$$

- Be more Bayesian and take posterior average to get the "Occam's razor equation"

$$\boxed{\log \mathcal{Z} = \langle \log \mathcal{L} \rangle_{\mathcal{P}} - \mathcal{D}_{\mathrm{KL}}}$$

- Natural regularisation which penalises models with too many parameters.

Kullback Liebler divergence

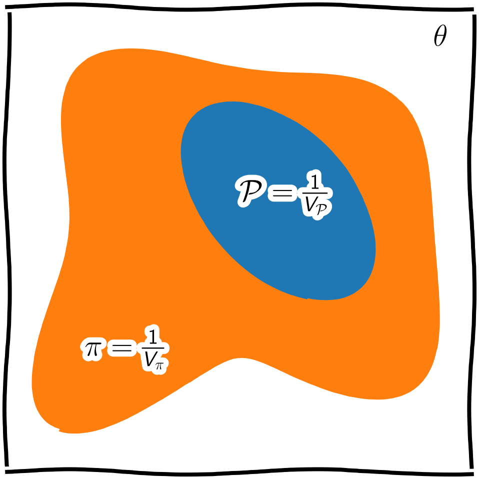
- The KL divergence between prior π and posterior $\mathcal{P}$ is defined as:

$$\mathcal{D}_{\mathrm{KL}} = \left\langle \log \frac{\mathcal{P}}{\pi} \right\rangle_{\mathcal{P}} = \int \mathcal{P}(\theta) \log \frac{\mathcal{P}(\theta)}{\pi(\theta)} d\theta.$$

- Whilst not a distance, $\mathcal{D} = 0$ when $\mathcal{P} = \pi$.
- Occurs in the context of machine learning as an objective function for training functions.
- In Bayesian inference it can be understood as a log-ratio of "volumes":

$$\mathcal{D}_{\mathrm{KL}} \approx \log \frac{V_\pi}{V_{\mathcal{P}}}.$$

(this is exact for top-hat distributions).

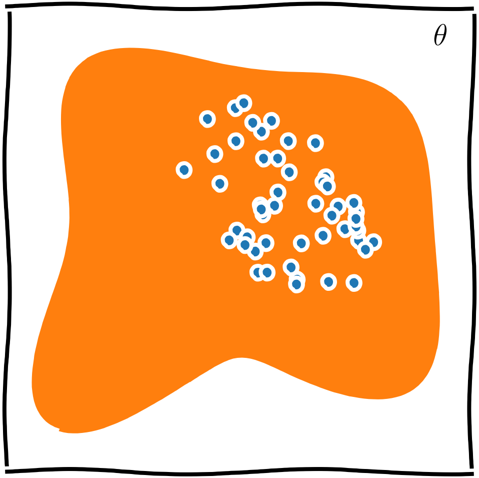


Why do sampling?

- The cornerstone of numerical Bayesian inference is working with **samples**.
- Generate a set of representative parameters drawn in proportion to the posterior $\theta \sim \mathcal{P}$.
- The magic of marginalisation $\Rightarrow$ perform usual analysis on each sample in turn.
- The golden rule is stay in samples until the last moment before computing summary statistics/triangle plots because

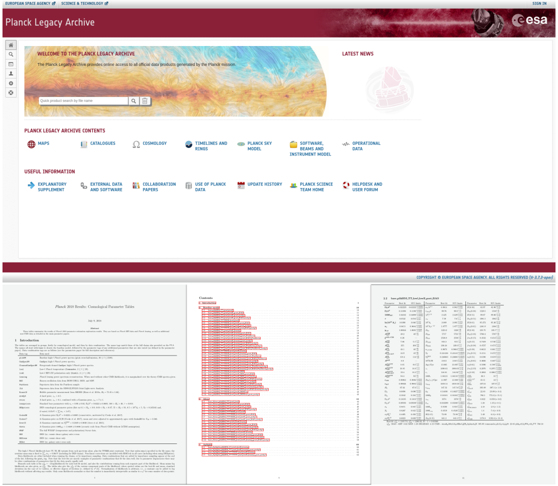
$$\boxed{f(\,\langle X\rangle\,) \ne \langle\, f(X)\,\rangle}$$

- Generally need $\sim \mathcal{O}(12)$ independent samples to compute a value and error bar.

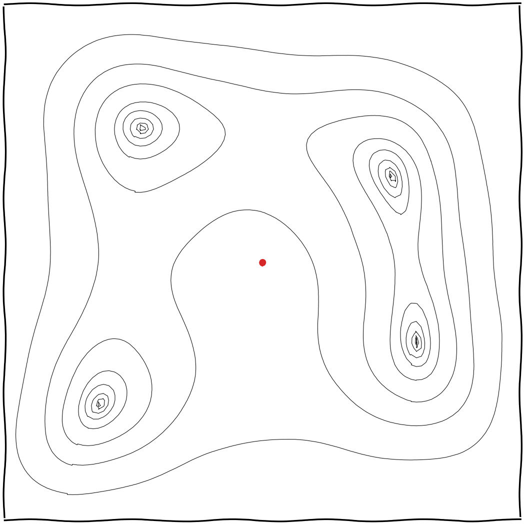


The Planck legacy archive

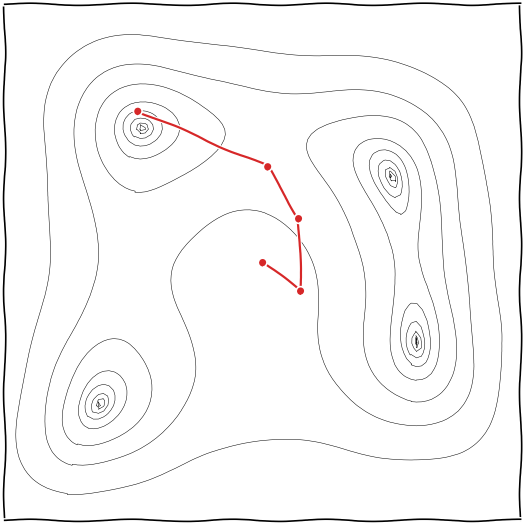
- *Planck* collaboration science products
- Distributed cosmology inference results as MCMC chains
- Across a grid of:
  - subsets/combinations of *Planck* data
    - TT, lowl, lowE, lensing
  - ΛCDM extensions
    - base, mnu, nrun, omegak, r
- importance sampling across some other likelihoods (BAO, JLA,. . .)
- Cannot compute evidences in high dimensions from MCMC chains
  - Only parameter estimation
  - no model comparison



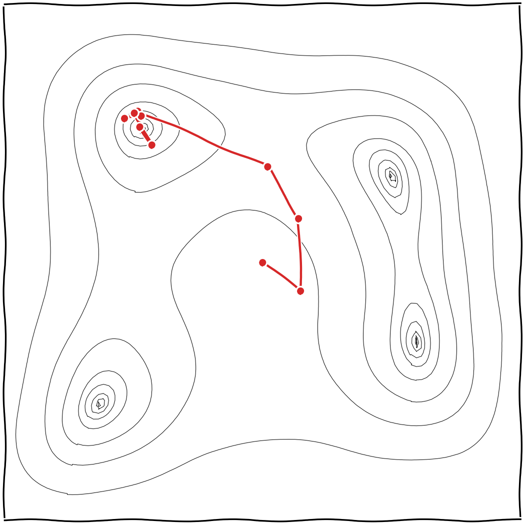
MCMC



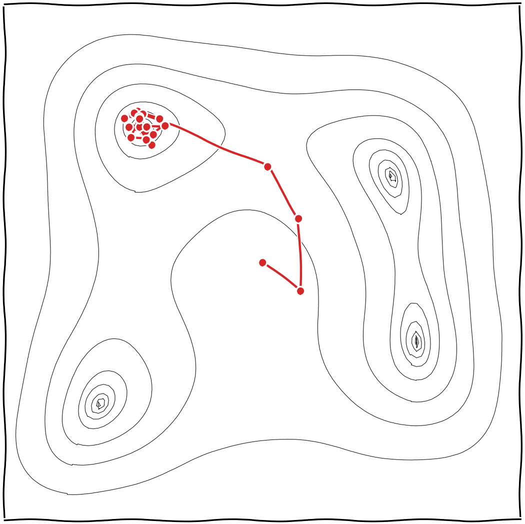
MCMC



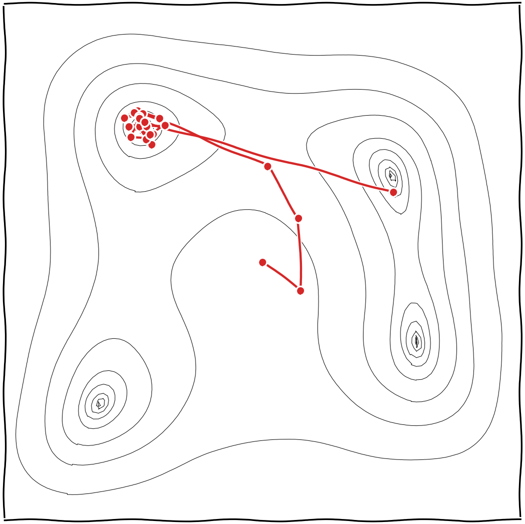
MCMC



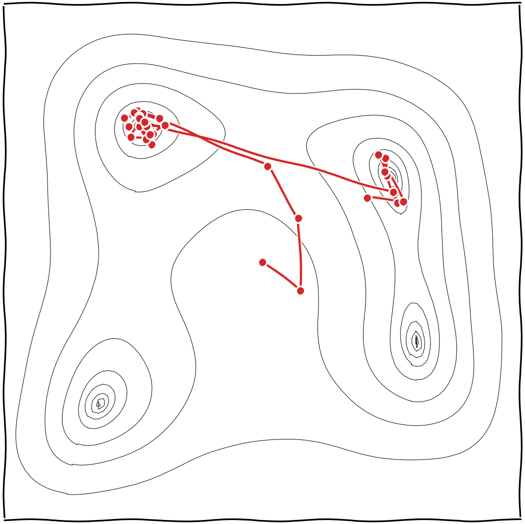
MCMC



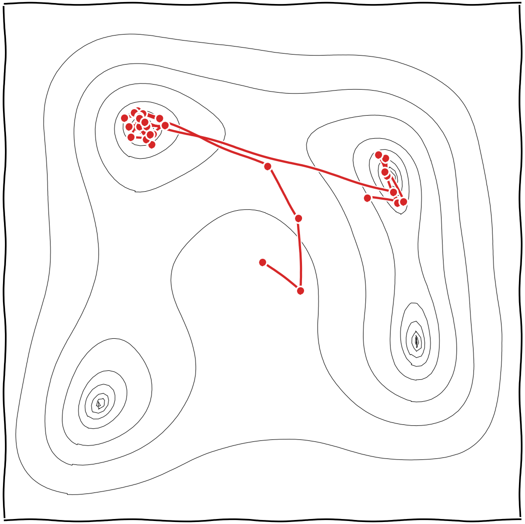
MCMC



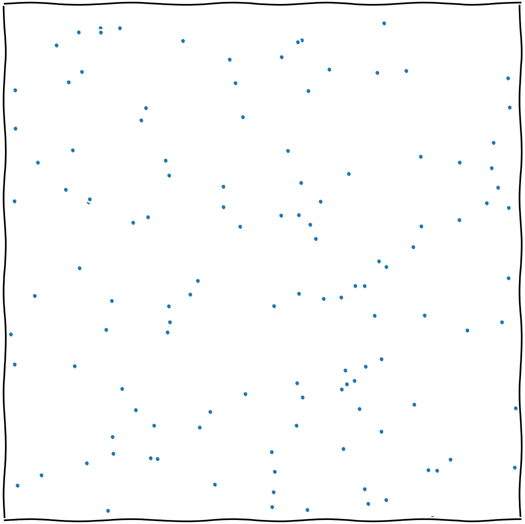
MCMC



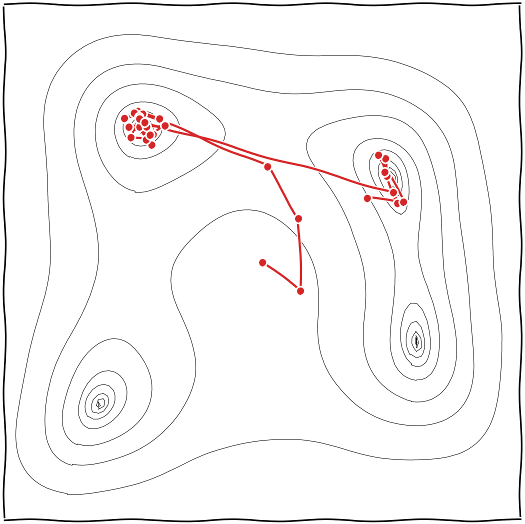
MCMC



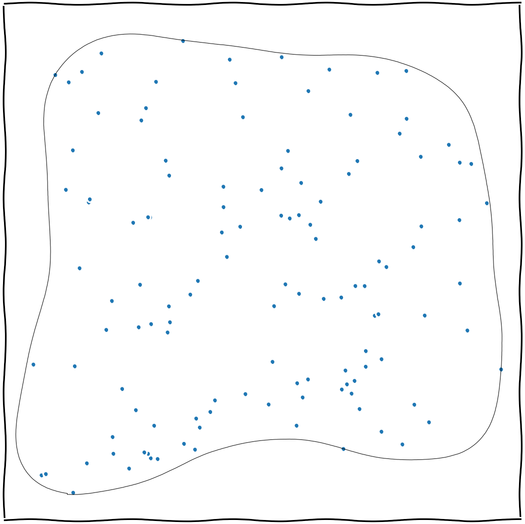
Nested sampling



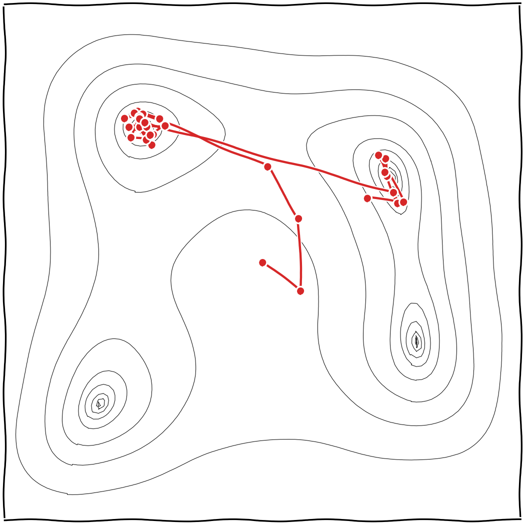
MCMC



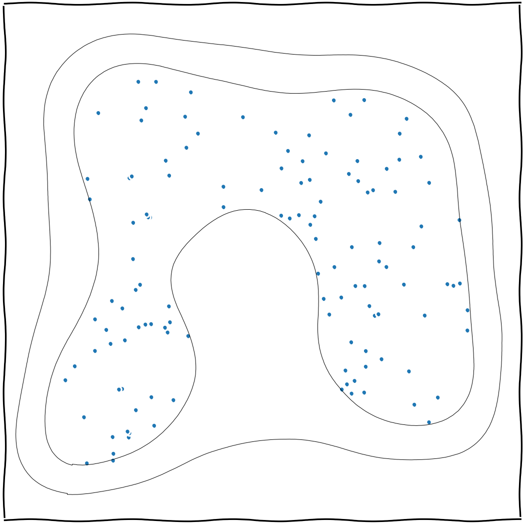
Nested sampling



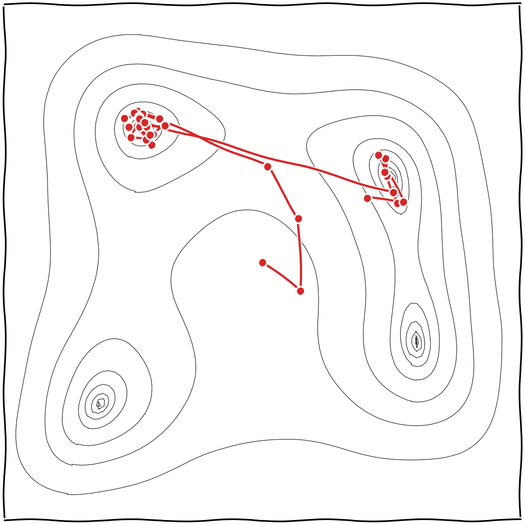
MCMC



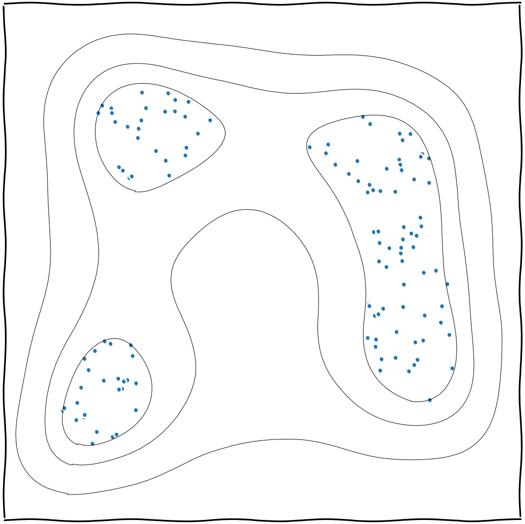
Nested sampling



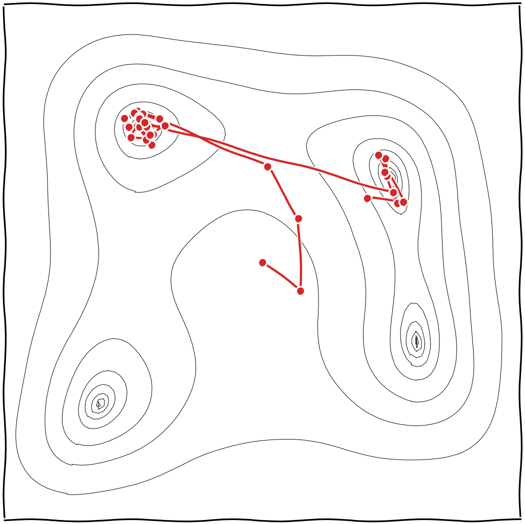
MCMC



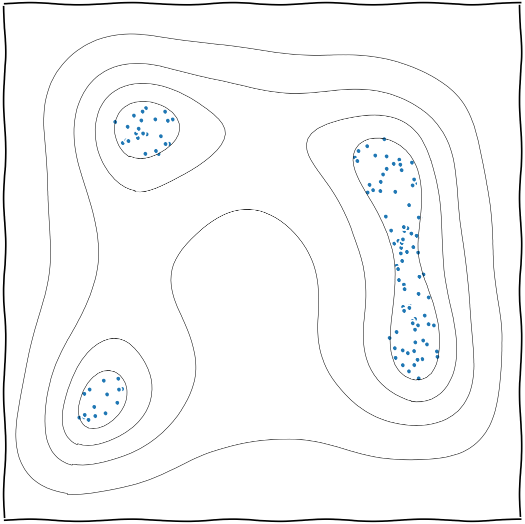
Nested sampling



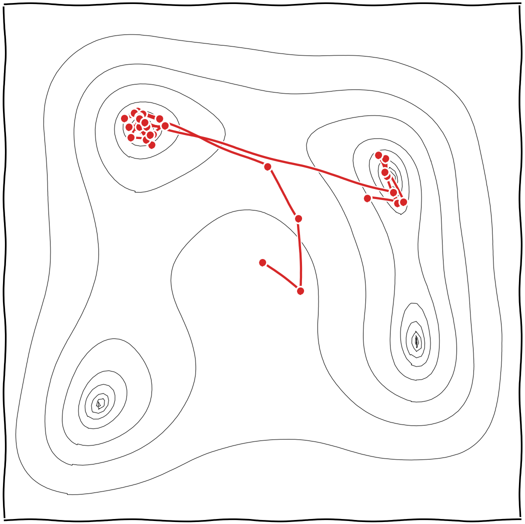
MCMC



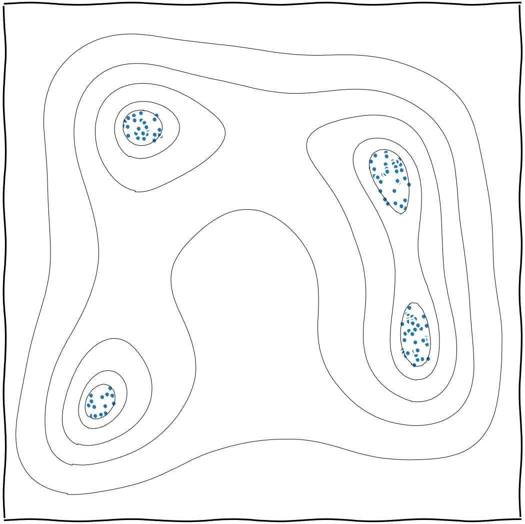
Nested sampling



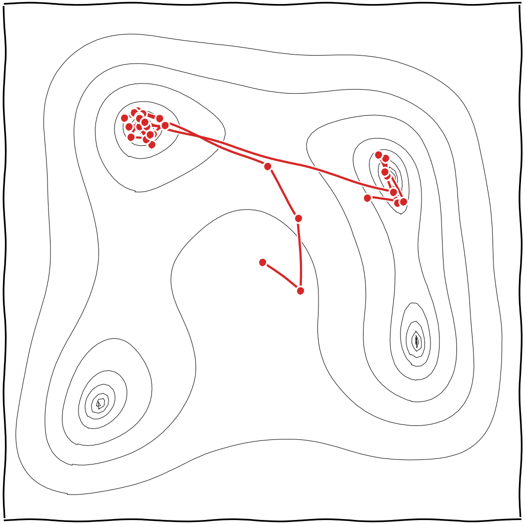
MCMC



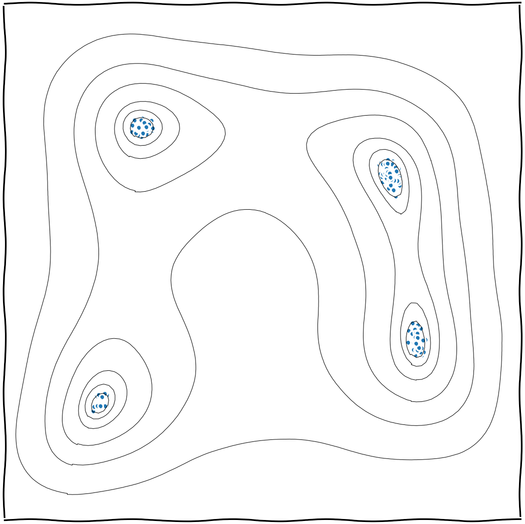
Nested sampling



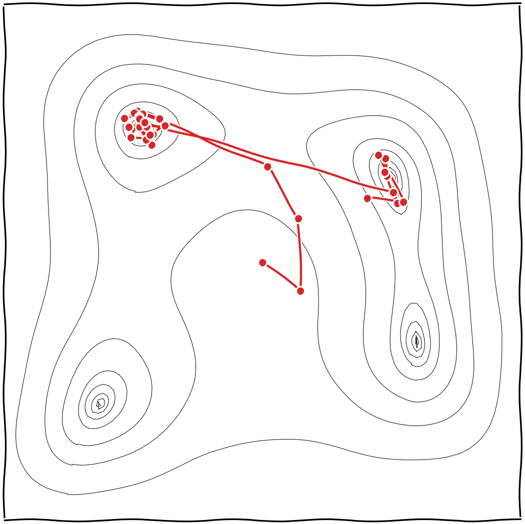
MCMC



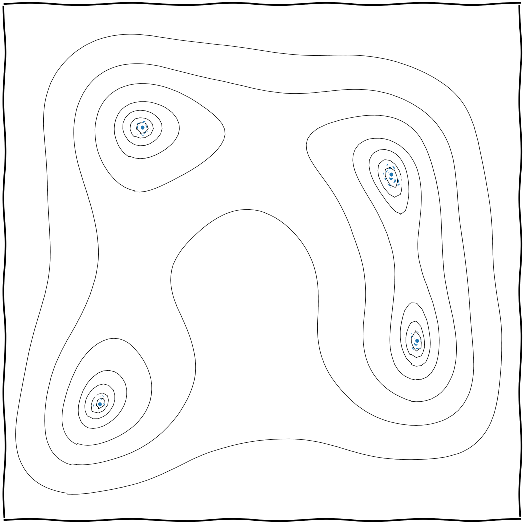
Nested sampling



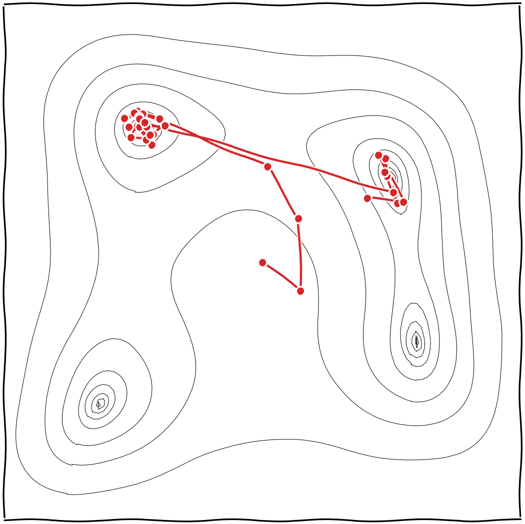
MCMC



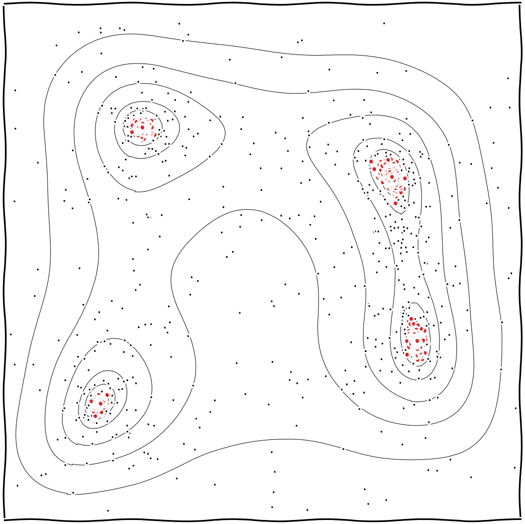
Nested sampling



MCMC

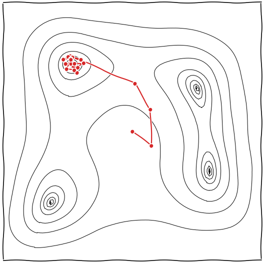


Nested sampling



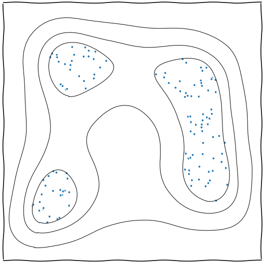
MCMC

- Single "walker"
- Explores posterior
- Fast, if proposal matrix is tuned
- Parameter estimation, suspiciousness calculation
- Channel capacity optimised for generating posterior samples



Nested sampling

- Ensemble of "live points"
- Scans from prior to peak of likelihood
- Slower, no tuning required
- Parameter estimation, model comparison, tension quantification
- Channel capacity optimised for computing partition function



The grid (so far)

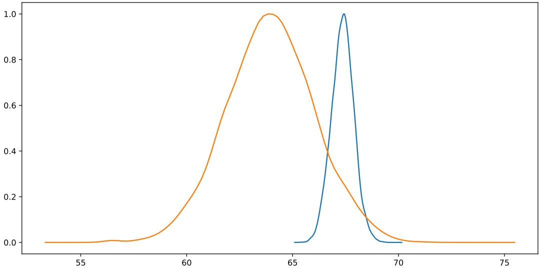
- Models: $[\Lambda\mathrm{CDM}, \Omega_K, \nu, r, w, w(a)]$
- Data: [`plik`, `camspec`, `DESY1`, `bicep+keck`, `BAO(DR16)`, `pantheon`]
- Pairwise combinations of datasets
- Breakdown of Planck & BAO data
- Samplers: [Metropolis Hastings MCMC, Nested Sampling]
- These exhaust what is currently available by default in `cobaya`
- Wide priors to allow for importance readjustment as desired
- roughly halfway through computational allocation.
- Feedback desirable as to what extensions to the grid would be of community interest (email wh260@cam.ac.uk) (`Pantheon+`, SH_0ES, `NPIPE`, `DESY3`,. . .).
- Further checking needed before first release by end of this year.

unimpeded

Universal Model comparison and Parameter Estimation Distributed over Every Dataset

- Python tool for seamlessly downloading, uploading and cacheing of chains
- Data stored on zenodo
- hdf5 storage for fast & reliable storage
- anesthetic compatible for processing of chains [1905.04768]
- α-testers wanted! (email wh260@cam.ac.uk)
- End goal – community library which everyone contributes to so expensive inference products are reusable and reused.

```python
from unimpeded import Unimpeded
store = Unimpeded(cache='data.hdf5')
samps = store('planck')
samps.H0.plot.kde_1d()
samps = store('planck', model='klcdm')
samps.H0.plot.kde_1d()
```



margarine: machine learning-enhanced Bayesian inference

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (margarine)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$
- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

margarine: machine learning-enhanced Bayesian inference

$\boxed{\mathcal{L}(\theta,\alpha)}$

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (`margarine`)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$

- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

`margarine`: machine learning-enhanced Bayesian inference

$\boxed{\mathcal{L}(\theta,\alpha)}$ $\boxed{\pi(\theta,\alpha)}$

Harry Bevins
[2205.12841]
[2207.11457]

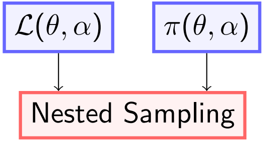
- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (`margarine`)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$

- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

margarine: machine learning-enhanced Bayesian inference

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (`margarine`)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$

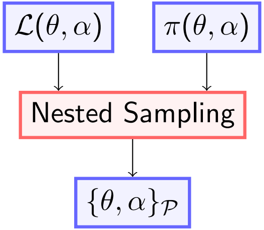


- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

`margarine`: machine learning-enhanced Bayesian inference

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (`margarine`)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$

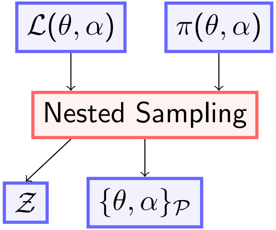


- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

`margarine`: machine learning-enhanced Bayesian inference

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (`margarine`)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$

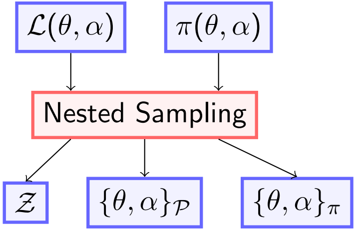


- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

margarine: machine learning-enhanced Bayesian inference

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (margarine)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$

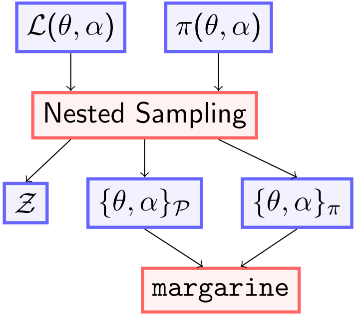


- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

margarine: machine learning-enhanced Bayesian inference

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (`margarine`)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$

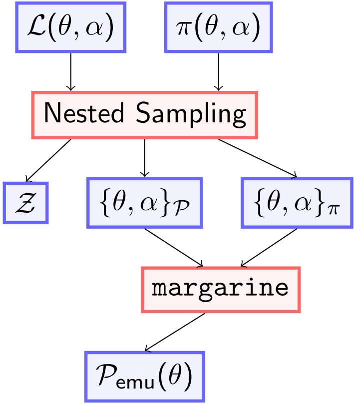


- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

`margarine`: machine learning-enhanced Bayesian inference

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (`margarine`)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$

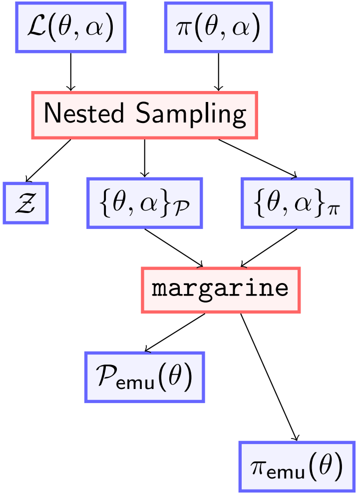


- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

margarine: machine learning-enhanced Bayesian inference

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (`margarine`)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$

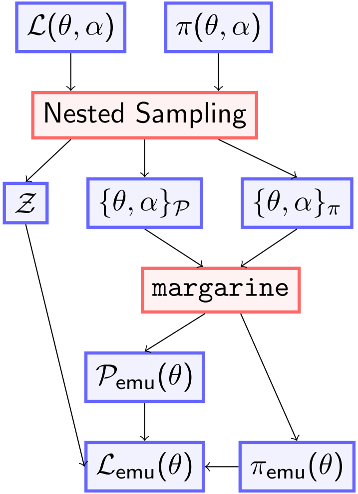


- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

`margarine`: machine learning-enhanced Bayesian inference

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (`margarine`)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$



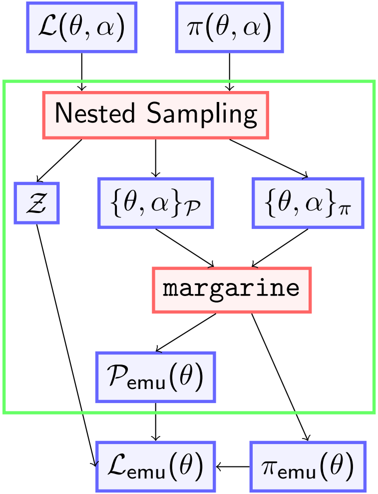
- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.

`margarine`: machine learning-enhanced Bayesian inference

Harry Bevins
[2205.12841]
[2207.11457]

- Can use machine learning + grid to dramatically speed up inference
- Emulate the marginal posterior and prior with masked autoregressive flows (`margarine`)
- Use nested sampling evidences to compute nuisance marginalised likelihood $\mathcal{L}(\theta) = \mathcal{P}(\theta)\mathcal{Z}/\pi(\theta)$

- Library of pre-trained bijectors to be used as priors/emulators/nuisance marginalised likelihoods
- e.g. easy to apply a *Planck*/DES/HERA/JWST prior or likelihood to your existing MCMC chains without needing to install the whole cosmology machinery.



Conclusions

- DiRAC RAC allocation for building a legacy grid of
  - MCMC & Nested sampling chains
  - gridded over (pairwise) up-to-date datasets
  - gridded over extensions to ΛCDM
  - Bijectors & emulators for fast re-use
  - Importance sampling toolkit via `anesthetic` for (re)processing
  - Long-term goal: community repository of chains to share model comparison compute resource
- Looking for:
  - α-testers for `unimpeded`
  - Suggestions for more datasets (and their incorporation into `cobaya`)