Marie Skłodowska-Curie Actions

Max-Planck-Institut für Physik (Werner-Heisenberg-Institut)

BAT.jl Bayesian Analysis Toolkit

BAT. jl — A Julia-based tool for Bayesian inference

Vasyl Hafych
on behalf of the BAT team

https://github.com/bat/BAT.jl

May 19, 2021
@ vCHEP 2021

The Bayesian Analysis Toolkit (BAT)

- A software package for Bayesian inference. Typical tasks:
  - Estimate model parameters

$$P(\boldsymbol{\lambda}|D) = \frac{P(D|\boldsymbol{\lambda})P(\boldsymbol{\lambda})}{\int P(D|\boldsymbol{\lambda})P(\boldsymbol{\lambda})d\boldsymbol{\lambda}}$$

  - Estimate model evidence (Bayes factors)

$$Z = \int P(D|\boldsymbol{\lambda})P(\boldsymbol{\lambda})d\boldsymbol{\lambda}$$

  - Quickly report and plot results
- Original BAT C++ (~2009): Very successful over the years, > 320 citations (INSPIRE):

  Caldwell, Allen, Daniel Kollar, and Kevin Kröninger. "BAT–The Bayesian analysis toolkit." Computer Physics Communications 180.11 (2009): 2197-2209.
- Upgraded to BAT.jl in Julia (started in 2017, released v2.0 in December 2020):

  Schulz, Oliver, et al. "BAT. jl: A Julia-Based Tool for Bayesian Inference." *SN Computer Science* 2.3 (2021): 1-17.

BAT.jl, the new BAT

- Core philosophy of the package
  - User provides likelihood (typically expensive, high data volumes, etc.) BAT does the rest
  - Designed for any scientific field with complex models, not just physics
  - Easy to use with defaults, but allow for detailed fine-tuning
- Functionalities of BAT.jl
  - Posterior space exploration via Markov chain Monte-Carlo (Metropolis-Hastings, Hamiltonian Monte Carlo) Nested Sampling, Sobol and Importance sampling
  - Sampling with space partition
  - Parallel execution of chains
  - Integration of non-normalized posteriors (AHMI and Cuba algorithms)
  - Automatic space transformations to convert target density into space suitable for algorithm
  - Report, visualize, save results

BAT.jl, the new BAT

- Additional functionalities from Julia
  - Excellent auto-differentiation (mode-finding, HMC)
  - Deep support for parallel operation (multithreaded and distributed)
  - Excellent package management
  - Easy calls to other programming languages

- Team
  - Max Planck Institute for Physics: A. Caldwell, O. Schulz (project lead), V. Hafych, S. Hayashi, L. Shtembari
  - TU-Dortmund: K. Kröninger, C. Grunwald, S. La Cagnina
  - ORIGINS Data Science Lab : F. Capel, P. Eller, J. Knollmüller
  - Master and Bachelor students

Example: Evidence Estimation

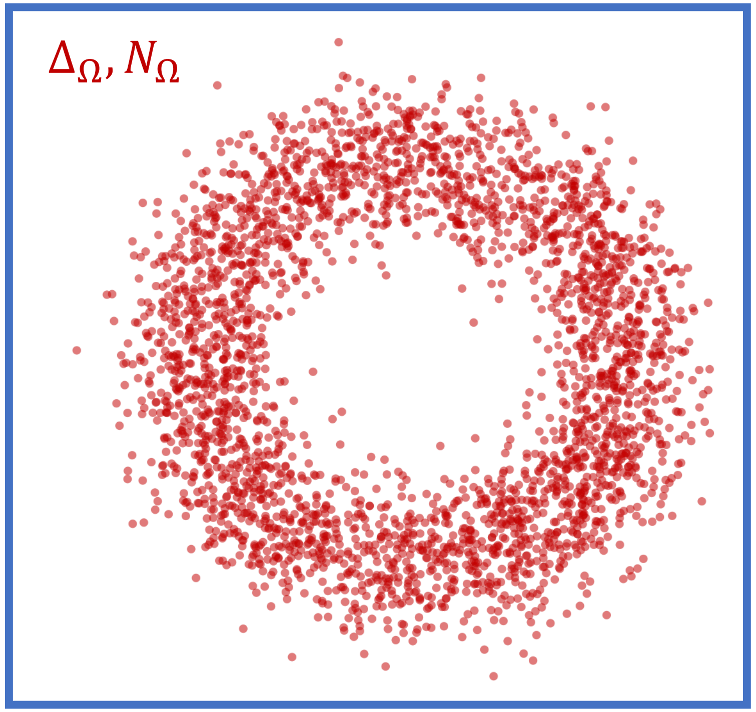
- Approximate Bayes' factor given samples $\Lambda \sim f(\boldsymbol{\lambda})$

$$Z = \int f(\boldsymbol{\lambda}) d\boldsymbol{\lambda}$$

- Harmonic Mean Estimate

$$\hat{Z} \approx \frac{\mathrm{N}_\Omega V_\Omega}{\sum_\Lambda \frac{1}{f(\Lambda)}}$$

- Not stable at $f(\Lambda) \to 0$



Example: Evidence Estimation

- Adaptive (AHMI) subvolumes Δ_k with well-behaved $f(\Lambda)$

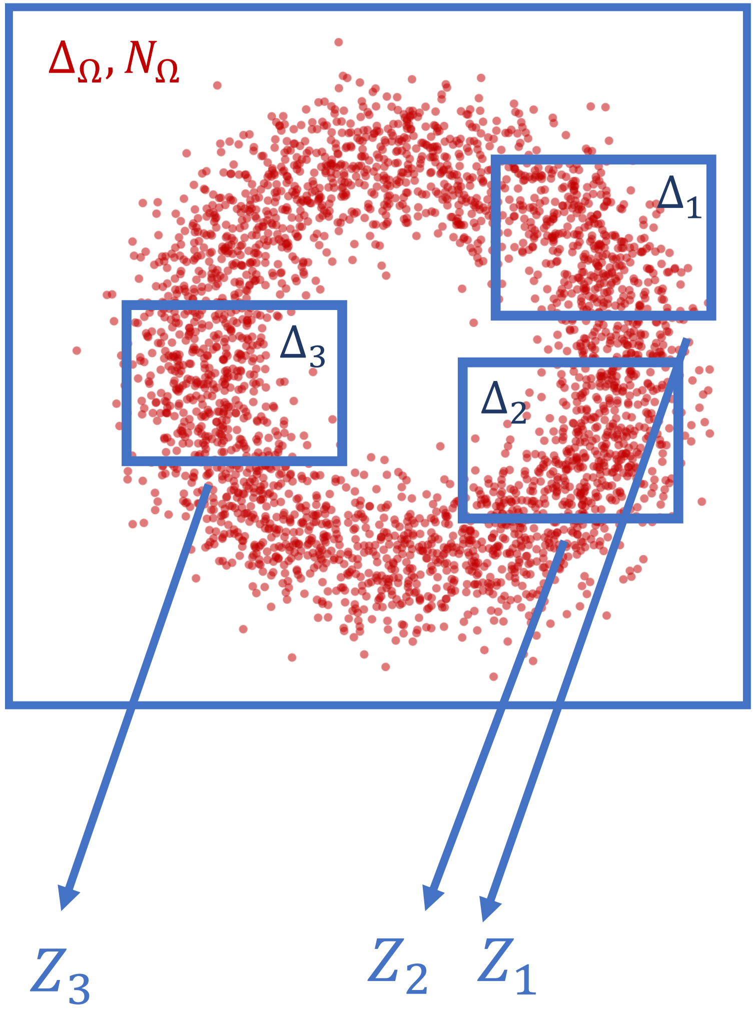
$$\hat{Z}_k \approx \frac{\mathrm{N}_\Omega V_{\Delta_k}}{\sum_{\Lambda \epsilon \Delta_k} \frac{1}{f(\Lambda)}}$$

- Use robust and unbiased estimator to combine the final result

$$\hat{Z} = \sum_k \omega_k Z_k$$

$$\omega_k = \frac{1}{\sigma_k^2} \Big/ \sum_{i \in N_{subv}} 1/\sigma_i^2$$

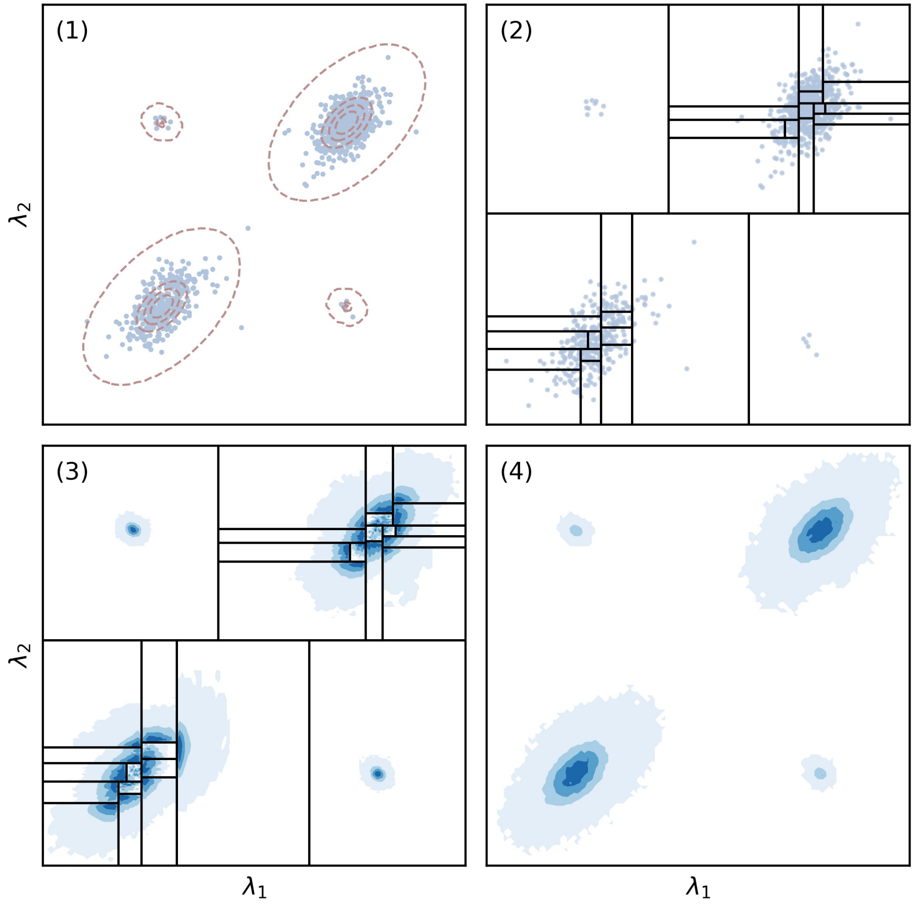
- Currently applicable to problems with <20 dimensions
- Upgrade to spherical volumes is in development



See more: Caldwell, Allen, et al. "Integration with an adaptive harmonic mean algorithm." *International Journal of Modern Physics A* 35.24 (2020): 2050142.

Example: Sampling with Space Partitioning

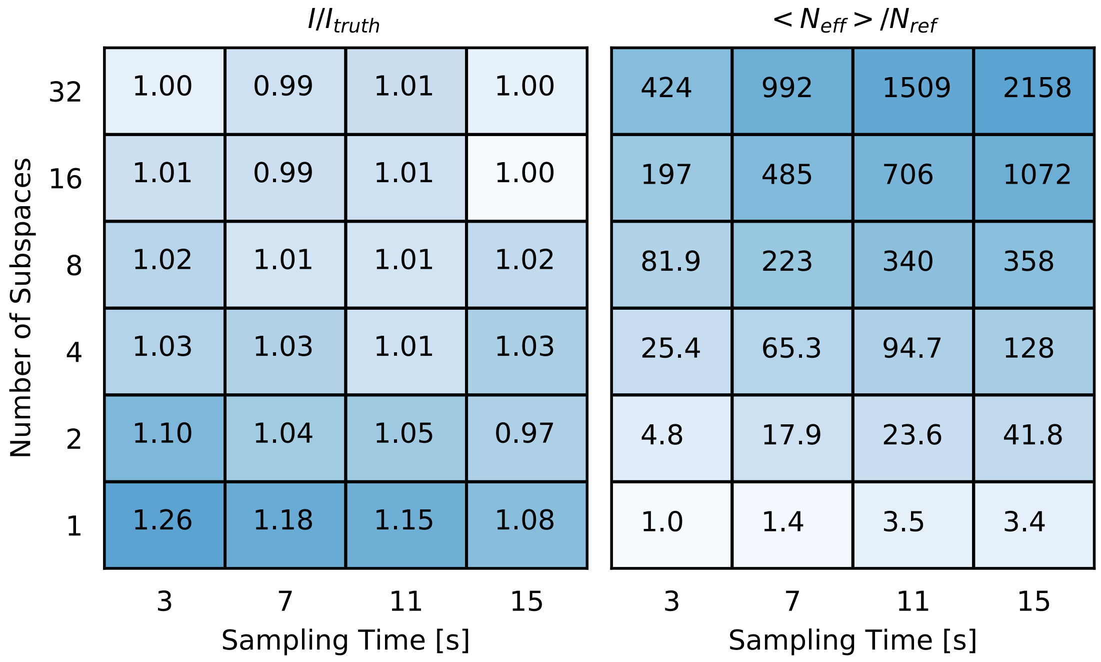
- Goal
  - Sample multimodal densities
  - Utilize distributed computing
- Idea
  - Quick space exploration
  - Divide space by multiple (simpler) subspaces
  - Sample them independently
  - Reweight (using AHMI) and stitch samples together



Example: Sampling with Space Partitioning

- Results
    - Improved tuning and convergence
    - Less correlated samples
    - Provides Bayes factor
    - Currently limited to problems with <20 dimensions

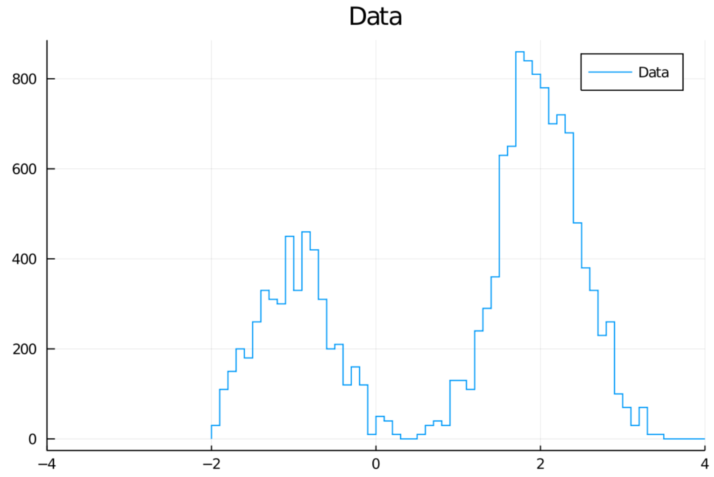
See more: Hafych, Vasyl, et al. "Parallelizing MCMC Sampling via Space Partitioning." arXiv preprint arXiv:2008.03098 (2020).

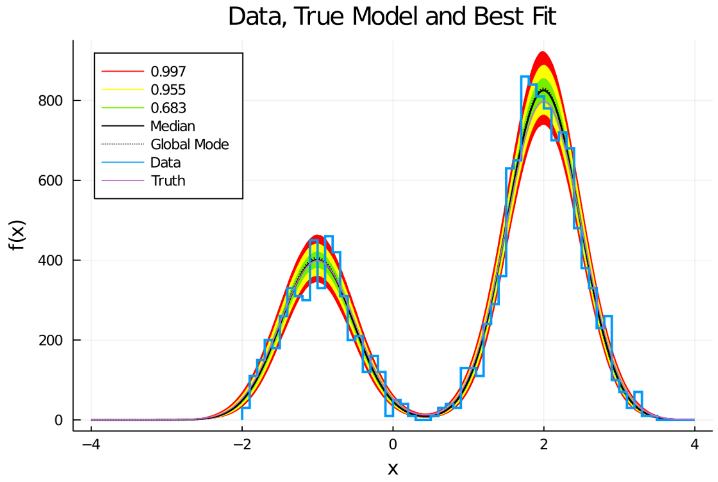


Example Run

- Generate synthetic data
- Define likelihood & sampler
- Sample
- Report and visualize results

Try out our tutorial:
https://bat.github.io/BAT.jl/





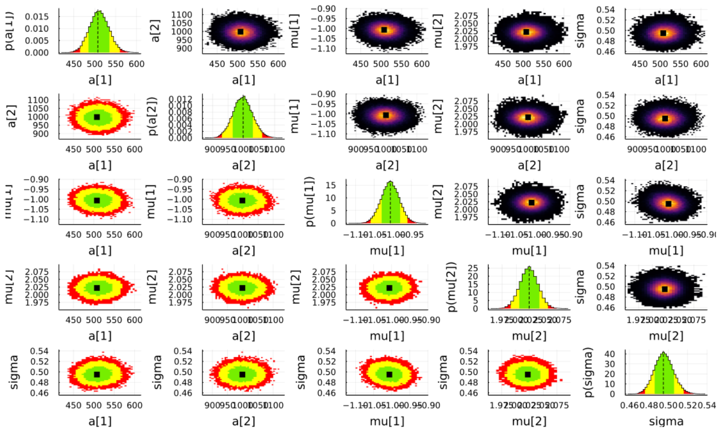

```
likelihood = let h = hist, f = fit_function
    # Histogram counts for each bin as an array:
    observed_counts = h.weights

    # Histogram binning:
    bin_edges = h.edges[1]
    bin_edges_left = bin_edges[1:end-1]
    bin_edges_right = bin_edges[2:end]
    bin_widths = bin_edges_right - bin_edges_left
    bin_centers = (bin_edges_right + bin_edges_left) / 2

    params -> begin
        # Log-likelihood for a single bin:
        function bin_log_likelihood(i)
            # Simple mid-point rule integration of fit function `f` over bin:
            expected_counts = bin_widths[i] * f(params, bin_centers[i])
            logpdf(Poisson(expected_counts), observed_counts[i])
        end

        # Sum log-likelihood over bins:
        idxs = eachindex(observed_counts)
        ll_value = bin_log_likelihood(idxs[1])
        for i in idxs[2:end]
            ll_value += bin_log_likelihood(i)
        end

        # Wrap `ll_value` in `LogDVal` so BAT knows it's a log density-value.
        return LogDVal(ll_value)
    end
end
```



Conclusions

- BAT concept: User brings domain knowledge and likelihood, BAT provides robust sampling, integration, and visualization
- Many algorithms are already implemented (MH, HMC, AHMI, Nested Sampling, etc.)
- Julia's benefits: Easy to write code, enriches functionality by auto differentiation, parallelization, and packages infrastructure
- Many more to come: AHMI with spherical volumes, bridge sampling, integrate MINUIT.jl, more algorithms, and lot's of ideas

Thank you for your attention!

This project has received funding from the European Union's Framework Programme for Research and Innovation Horizon 2020 (2014-2020) under the Marie Sklodowska-Curie Grant Agreement No.765710.