BAT.jl, the Bayesian Analysis Toolkit in Julia

Oliver Schulz on behalf of the BAT team

MAX-PLANCK-GESELLSCHAFT

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

oschulz@mpp.mpg.de

CHEP 2024, Krakow, October 23rd 2024

Statistical inference in Julia

- ROOT user: What's Julia's RooFit?
- Python user: What's Julia's pyHF?
- Julia user: Ah, theres lot's of statistics packages for Julia . . .
- ROOT and Python user: But which one should I use for my fit?

Statistical inference in Julia

- ROOT user: What's Julia's RooFit?
- Python user: What's Julia's pyHF?
- Julia user: Ah, theres lot's of statistics packages for Julia . . .
- ROOT and Python user: But which one should I use for my fit?
- This is not that talk . . .

Statistical inference in Julia

- ROOT user: What's Julia's RooFit?
- Python user: What's Julia's pyHF?
- Julia user: Ah, theres lot's of statistics packages for Julia ...
- ROOT and Python user: But which one should I use for my fit?
- This is not that talk ...
- ... but partially, it is.

The Julia Bayesian statistics ecosystem

- Several Bayesian sampler implementations in the Turing project.
- Rxinfer.jl has interesting approach via Bayesian graphs, but not applicable to all problems.
- Quite a few other sampler packages like ZigZagBoomerang.jl, AdaptiveMCMC.jl, MGVI.jl (in v4.0) and so on.
- Any function that maps parameters to data distributions (equivalent to a Markov kernel) can be a (forward) model in Julia.
- MeasureBase.Likelihood(v -> datadist, data), automatically builds likelihood functions from forward models and data.
- BAT.jl (this talk) aims to a be a common-API wrapper for existing samplers, plus some BAT-native samplers.

The Bayesian Analysis Toolkit (BAT)

- The Bayesian Analysis Toolkit (BAT):
  A software package for Bayesian inference
- Typical tasks: Given a set of data and prior knowledge
  - estimate parameters
  - compare models (Bayes factors)

The Bayesian Analysis Toolkit (BAT)

- The Bayesian Analysis Toolkit (BAT):
  A software package for Bayesian inference
- Typical tasks: Given a set of data and prior knowledge
  - estimate parameters
  - compare models (Bayes factors)
- Functionalities
  - Multi-method posterior space exploration
  - Integration of non-normalized posterior
    (i.e. evidence calculation)
  - User-friendly plotting and reporting

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

BAT.jl, the successor of BAT-C++

- Original: BAT-C++, developed at MPP
  [DOI: 10.1016/j.cpc.2009.06.026 (2009).]
    - Very successful over the years, > 250 citations (INSPIRE)
    - Written in C++, based on CERN ROOT
    - Gained wide user base, esp. HEP/nuclear/astro-physics
    - Had reached limit of original software design, needed a complete re-write.

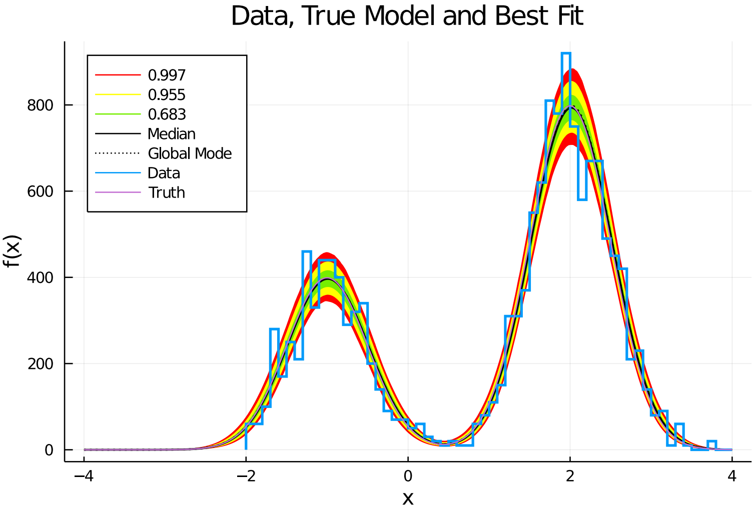
BAT.jl, the successor of BAT-C++

- Original: BAT-C++, developed at MPP
  [DOI: 10.1016/j.cpc.2009.06.026 (2009).]
  - Very successful over the years, > 250 citations (INSPIRE)
  - Written in C++, based on CERN ROOT
  - Gained wide user base, esp. HEP/nuclear/astro-physics
  - Had reached limit of original software design, needed a complete re-write.
- Successor: BAT.jl, written in Julia.
  [DOI: 10.1007/s42979-021-00626-4 (2021).]
  - MPP (A. Caldwell): O. Schulz (lead), A. Butorev, M. Dudkowiak
  - TU-Dortmund (K. Kröninger): C. Grunwald, S. Lacagnina,
  - ORIGINS ODSL: F. Capel, P. Eller, J. Knollmüller
  - . . . and many contributions from past students (thank you!)

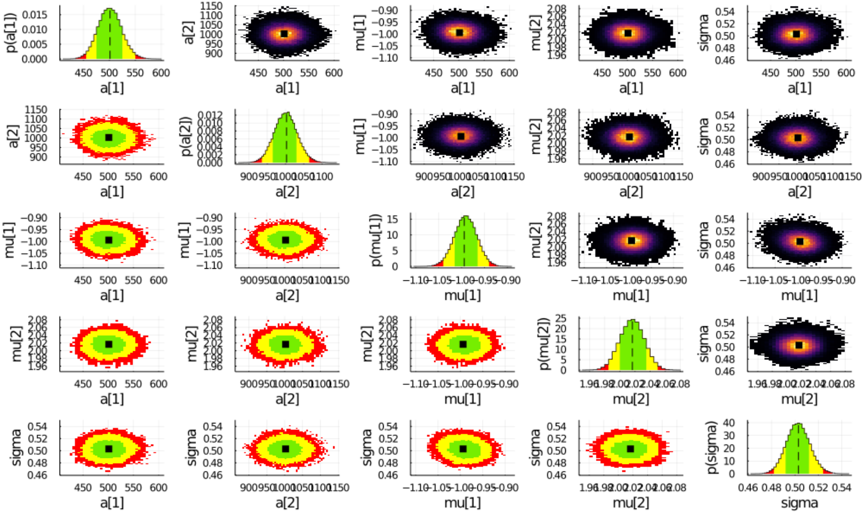
BAT.jl Features

- MCMC sampling via Metropolis-Hastings, Hamiltonian Monte Carlo, MGVI, Sobol and importance sampling, more soon.
- Posterior integration with nested sampling, bridge sampling, or Cuba (we'll add SciML Integrals.jl).
- Automatic space transformations cast target density into space suitable for algorithm.
- Over last year, changed much of BAT's terminology from densities to measures.
- Upcoming version v4.0 uses transformations/pushforwards as central paradigm (instead of proposal distributions).
  Basis for incorporation of normalizing flows into samplers and more.
- Current version BAT.jl v3.3 (v4.0 release in the next days).

Simple BAT.jl example: Histogram Fit



BAT.jl plotting: Posterior projections



BAT.jl usage

Models are just functions from parameters to data distributions:

```
f_model(params) = distprod(Poisson.(expected_counts(params)))
prior = distprod(a = Normal(), b = Exponential(), ...)
mock_truth = rand(prior)
mock_data = rand(f_model(truth))
likelihood = Likelihood(f_model, mock_data)
```

Now we can sample the posterior:

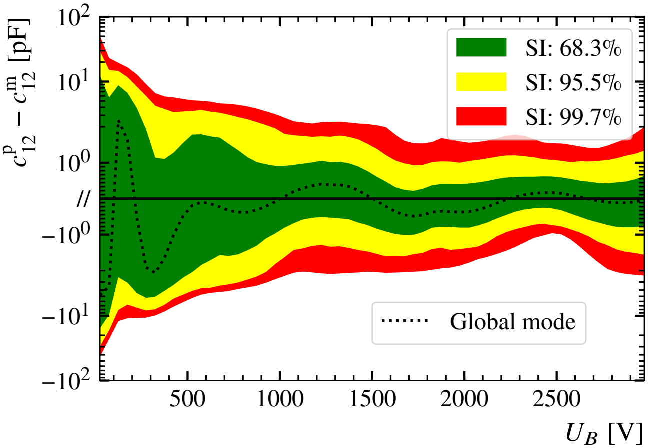
```
pstr = PosteriorMeasure(likelihood, prior)
smpls, _ = bat_sample(pstr)
plot(samples)
```

Can also use black-box likelihood functions.

Some BAT.jl use cases . . .

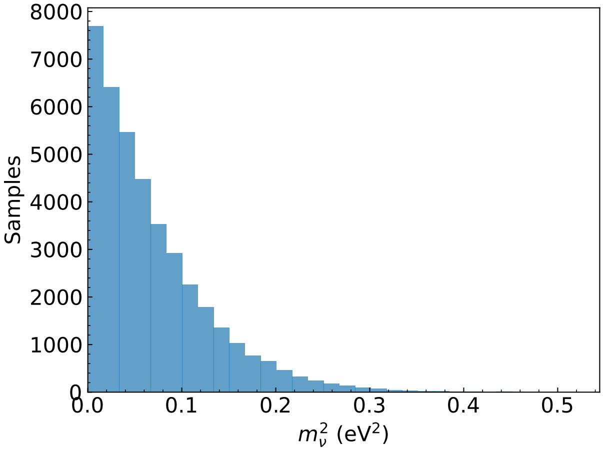
Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)

HPGe-Detector impurity profile inference



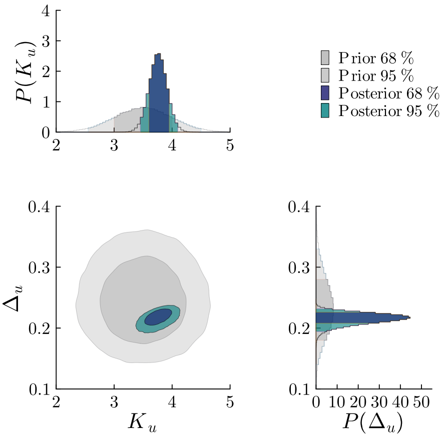
Cap./vol.-curves measured and simulated, ML surrogate, complex prior [Eur. Phys. J. C 83, 352 (2023)], Metropolis-Hastings

KATRIN m_ν^2 posterior, simulated data



NETRIUM DNN model [Eur. Phys. J. C 82, 439 (2022)] ported to Julia
Sampled with AdvandedHMC backend using Zygote-AD.

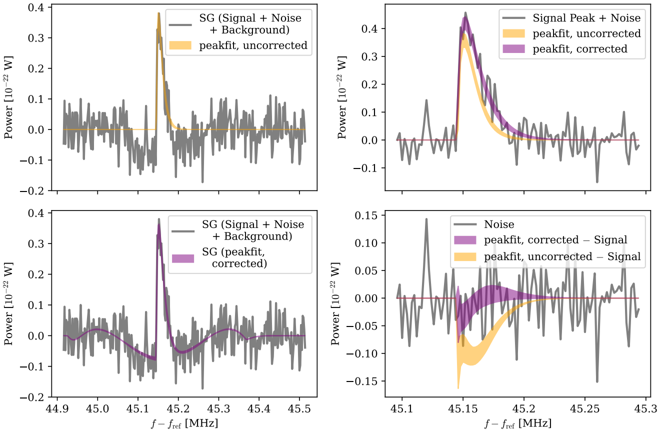
ZEUS ep-collision parton PDF fit



QCDNUM (Fortran) wrapped in Julia [PRL.130.141901]

Sampled with adaptive Metropolis-Hastings backend.

MADMAX bias-free bump hunt

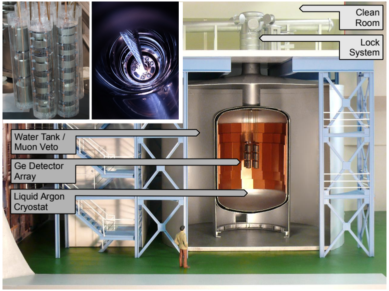


Sampled with Ultranest backend (simulated data)

[arXiv 2306.17667]

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

Final Results of GERDA

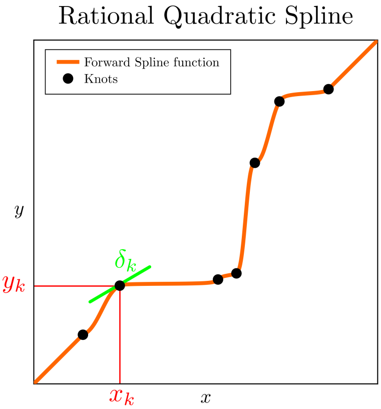


- $T^{0\nu}_{1/2} > 1.4 \times 10^{26}$ yr (90% CI) (equiprobable signal strengths)
- $T^{0\nu}_{1/2} > 2.3 \times 10^{26}$ yr (90% CI) (equiprobable Majorana neutrino masses)

Hierarchical prior,
sampled with adaptive Metropolis-Hastings backend.

[PRL 125, 252502 (2020)]

Monotonic Rational-Quadratic Splines



K Segments

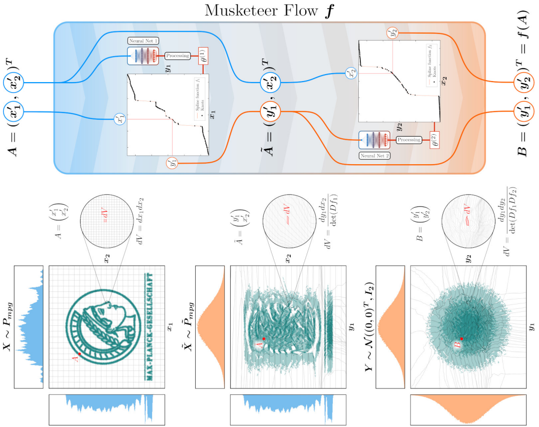
Characterized by

$\{\boldsymbol{x_k}, \boldsymbol{y_k}\}$, $\{\boldsymbol{\delta_k}\}$

[Conor Durkan et al. *Neural Spline Flows*]

MonoticSplines.jl: Based on "Neural Spline Flows"
high-performance CPU+GPU via KernelAbstractions.jl.

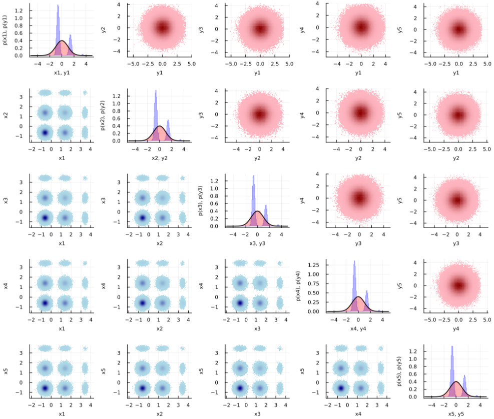
Prototype: Spline flows for low-dim marginals



Trying to turns this into an automated tool to pass marginal posteriors around (once trained, math is quite simple).

Challenge: Machine learning is hard to fully automatize.

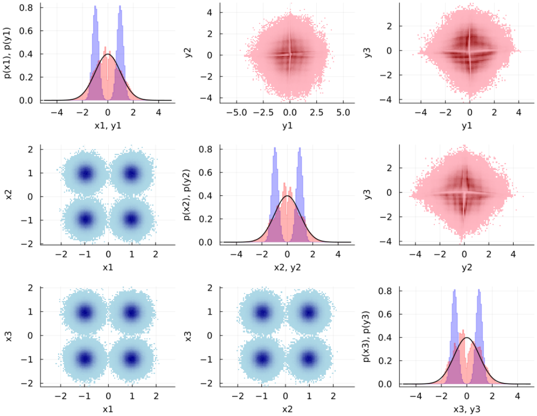
Prototype: Normalizing flow MCMC



[W. Weber]

Continuously adapt space transformation,
by machine-learning autogeressive flow using MALA multi-walking MCMC.
Prototype stage: again - machine learning is hard to fully automatize.

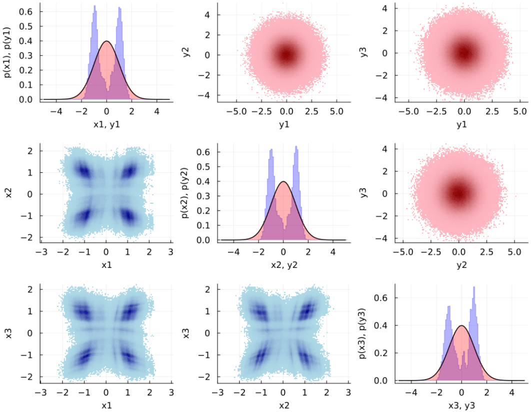
Normalizing flow MCMC, imperfect flow



[W. Weber]

Normalizing flow often imperfect, resulting in non-Gaussian latent space.

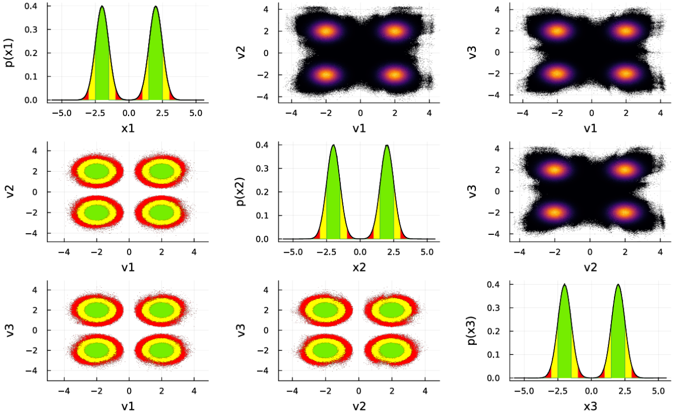
Normalizing flow MCMC, imperfect flow-samples



[W. Weber]

IID samples from imperfect flows often not of acceptable quality

Normalizing flow MCMC with importane sampling



[W. Weber]

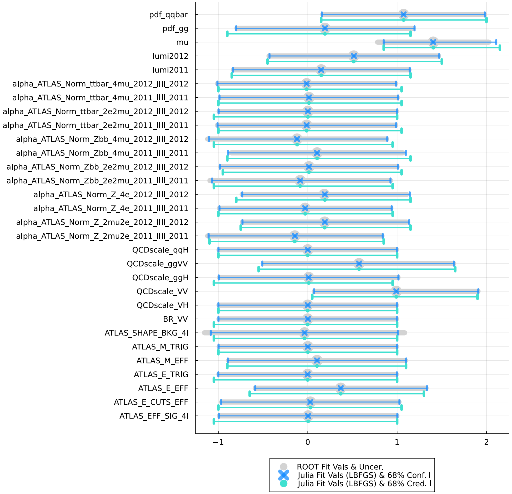
Can use importance sampling to correct for imperfectioins in flow.

HS^3 - HEP Statistics Serialization Standard

- Upcoming standard for representing (and publishing) statistical models in JSON
- Current state of the art: pyhf ("stacked histograms only")
- HS^3 is full superset of phhf, but much more general
- Cleaner terminology (less "community slang") than RootFit, yet bi-directionally convertible
- Standard being finalized, current prototype already implemented in ROOT
- Prototype Julia implementation using code generation,
  BAT tooling, importance sampling, and other stuff. **[R. Pelkner, J. Ling, O. Schulz]**

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

Julia HS3 Higg Parameter Estimates

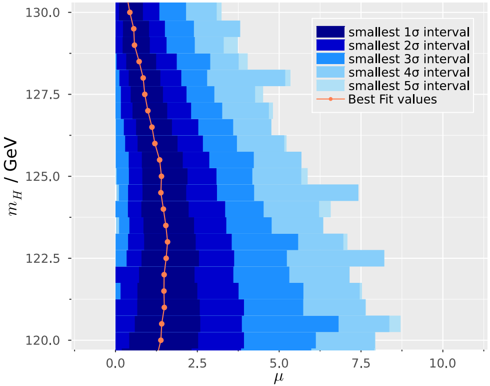


- Parameter estimate comparison RootFit vs. Julia HS3 prototype
- $H \rightarrow ZZ^* \rightarrow 4l$
- RooFit with Minuit2+Minor vs. ProfileLikelihood.jl with LBFGS (with some BAT.jl/ValueShapes.jl tools)

[Master thesis Robin Pelkner, TU Dortmund]

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

Julia HS3 Higgs Bayesian Posteriors



- Bayesian posteriors of μ for different m_H
- $H \rightarrow ZZ^* \rightarrow 4l$
- Julia HS3 prototype + BAT.jl MCMC

[Master thesis Robin Pelkner, TU Dortmund]

Bayesian Guided Maximum Likelihood (BGML)

- Maximum likelihood optimization often not easy to get to converge
- Typical solution: Transform to different space - but which one?
- Approach: Choose a prior that doesn't fully exclude any physically possible parameters
- BAT.jl automatically generates space transformation f from multivariate normal to prior
- Run optimizer on $\mathcal{L} \circ f$ in unconstrained space:
  unbiased, only excludes impossible parameter values,
  but optimizer has shorter path to favored values.
- Used in production for fitting calibrations in LEGEND Julia stack.
- We'll add this as a push-button tool to BAT.jl.

Conclusions and Outlook

- For Bayesians: BAT.jl tries to make Baysian inference easy,
  across multiple backends
  now also useful for some frequentist stuff
- In general:
  We try to integrate with statistic packages
  across the ecosystem, instead of building "HEP-stats-island".
- More and more: ~~either~~ Baysian ~~or~~ and Frequentist
- No full "RooFit" and "pyHF" equivalent in Julia yet, but . . .
- . . . many of the pieces in places.
- Julia implementation of upcoming HS3-standard can get us full RooFit compatibility,
  work ongoing.
  BAT.jl will play an important role, as will ProfileLikelihood.jl and others.