Bayesian and General Statistics in Julia

Oliver Schulz on behalf of the BAT team

MAX-PLANCK-GESELLSCHAFT

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

BAYESIAN ANALYSIS TOOLKIT

oschulz@mpp.mpg.de

JuliaHEP2024, CERN, October 1st 2024

Statistical inference in Julia

- HEP person: What's Julia's RooFit?
- Julia user: Ah, well, we have lot's of statistics packages, . . .
- But which ones, and where are we?

Distributions

- Distributions.jl provides extensive and high-quality implementation of common probability distributions.
- One of Julia's oldest packages, but has evolved well without breaking too much code.
- Missing HEP-specific distributions like Crystal Ball and so on.
- ToDo: Get together and create a package HEPDistributions.jl or similar.
- Limitations: Distributions.jl has little GPU support, no support for struct-like variates, not support for variable-length variates.

(Probability) Measures

- MeasureBase.jl/MeasureTheory.jl aim to augments Distributions.jl with measure-theory concepts.
- Clean way to deal with "non-normalized distributions" (often can't normalize Bayesian posteriors)
- Will give us clean way to represent things like nonhomogeneous Poisson point processses (in HEP often called extended distribution or extended PDF).
- Bayes theorem for continuous distributions actually based on measure theory:

$$\alpha_b(A) = \int_A \frac{\mathrm{d}\,\beta_a}{\mathrm{d}\,\bar{\beta}}(b)\mathrm{d}\,\bar{\alpha}(a)$$

- Currently undergoing major revision to use Pkg extension for Distributions.jl support and hierarchical measures.

Optimizers

- A lot of inference is solving optimization problems: Maximum likelihood, profile likelihood, maximum a-posteriori (MAP), etc.
- Well established Optim.jl provides Nelder-Mead, LBFGS and others
- SciML package Optimization.jl wraps almost all Julia optimization packages under common interface, support for many automatic differentiation (AD) backends as well.
- In general excellent tooling, only . . .
- . . . sometime people want to see a comparison with Minuit (also sometime Minuit is really good).
- ToDo: Wrap standalone Minuit for Julia.

Stochastic models

- 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.
- Turing and RxInfer (see later) both come with "fancy" model DSLs with Bayeian focus. Also interest in building bridges between them, but technically not so easy.

Frequentist inference

- Maximum likelihood is easy, due to Julia's composability and wide choice of optimizers.
- We don't have "the one" package, but ProfileLikelihood.jl that seems to be closest to the HEP-typical approach.

Bayesian inference

- 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 and so on.
- BAT.jl (led by your's truly, see later) 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

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++

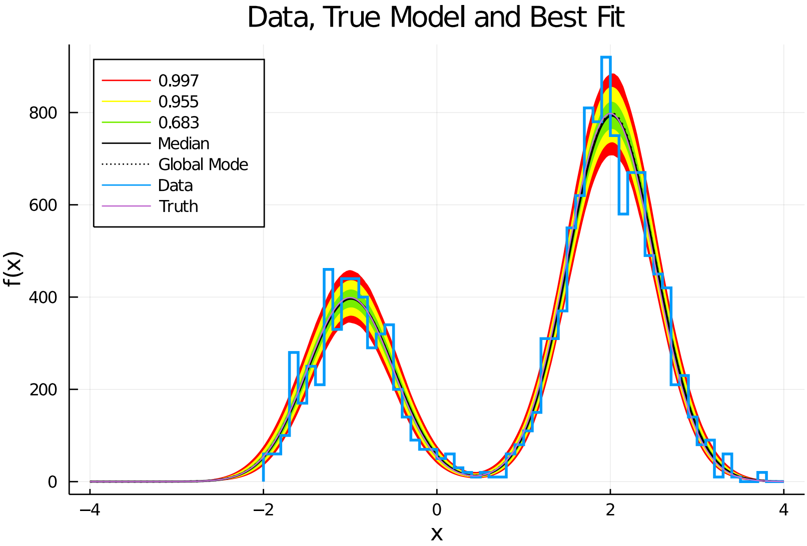
- Original: BAT-C++, developed at MPP
  [DOI: 10.1016/j.cpc.2009.06.026 (2009).]
  - Very successful over the years, > 250 citations (INSPIRE)
  - Written in C++, based on CERN ROOT
  - Gained wide user base, esp. HEP/nuclear/astro-physics
  - Had reached limit of original software design, needed a complete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

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

BAT.jl Features

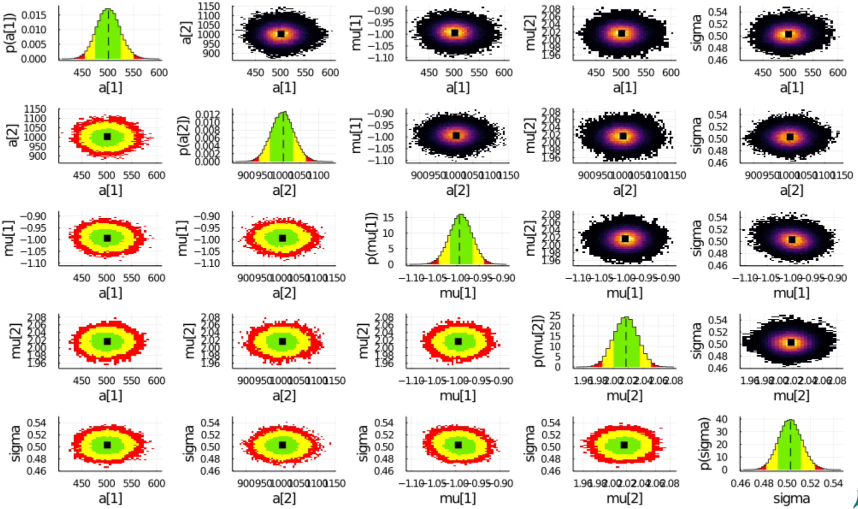
- MCMC sampling via Metropolis-Hastings, Hamiltonian Monte Carlo, Sobol and importance sampling, more soon.
- Posterior integration with nested sampling, bridge sampling, or Cuba. Will add SciML Integrals.jl.
- Automatic space transformations cast target density into space suitable for algorithm.
- Over last year, changed much of BAT's terminology from densities to measures. Next breaking new version of MeasureBase will allow for completing this transition.
- Current development focus: Move from proposal distributions and mass matrices and similar to space transformations. Will allow incorporation of normalizing flows into samplers and more.
- New sampler under development (still): Adaptive space transformations via RQS normalizing flows during algorithm tuning

Simple BAT.jl example: Histogram Fit



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

BAT.jl plotting: Posterior projections



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

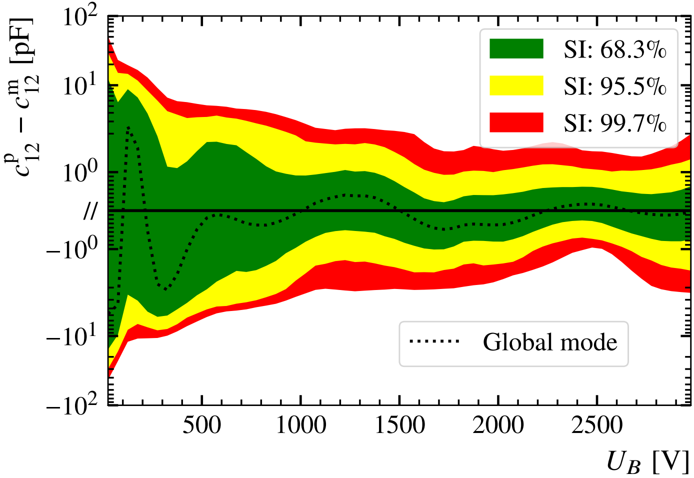
Bayesian Guided Maximum Likelihood (BGML)

- 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.
- We'll add this as a push-button tool to BAT.jl.

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

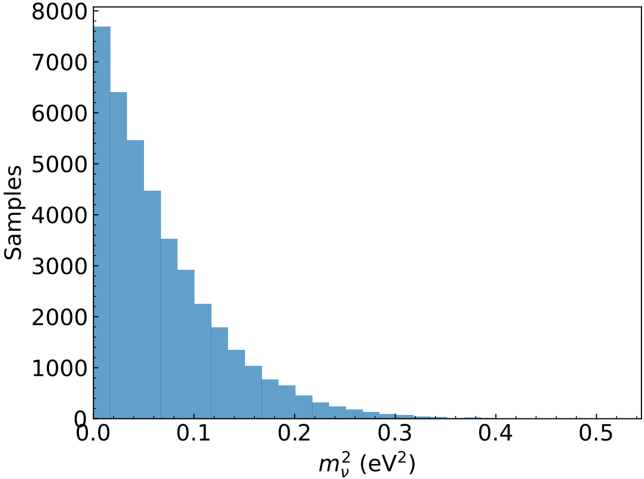
Some BAT.jl use cases . . .

HPGe-Detector impurity profile inference



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

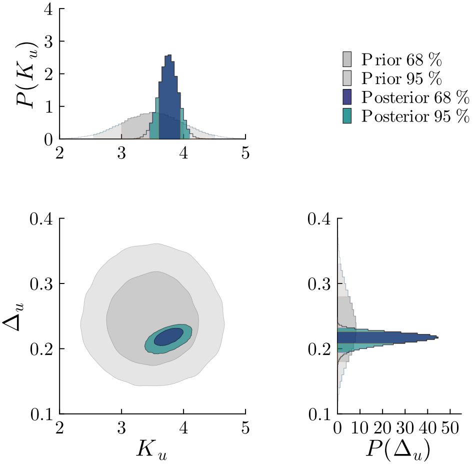
KATRIN m_ν^2 posterior, simulated data



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

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

ZEUS ep-collision parton PDF fit

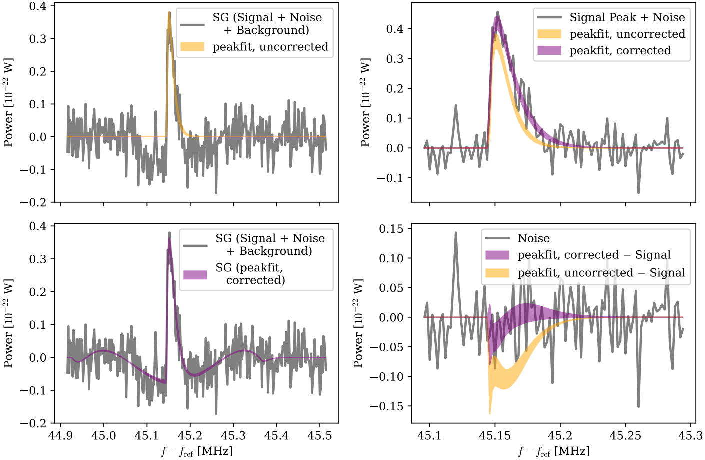


QCDNUM (Fortran) wrapped in Julia [PRL.130.141901]

Sampled with adaptive Metropolis-Hastings backend.

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

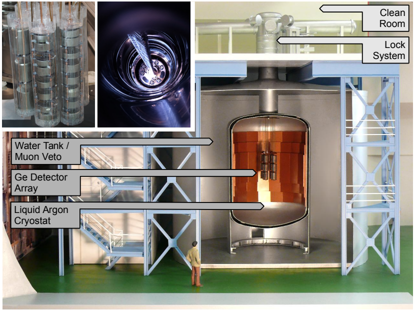
MADMAX simulated peak BG



Sampled with Ultranest backend

[arXiv 2306.17667]

Final Results of GERDA

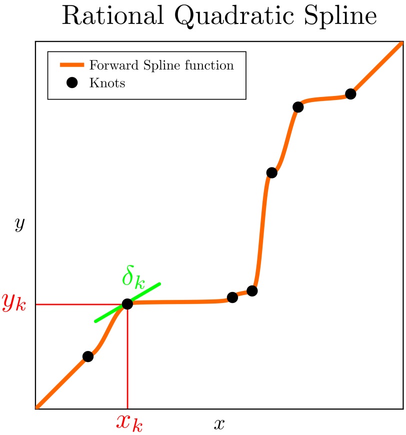


- $T_{1/2}^{0\nu} > 1.4 \times 10^{26}$ yr (90% CI) (equiprobable signal strengths)
- $T_{1/2}^{0\nu} > 2.3 \times 10^{26}$ yr (90% CI) (equiprobable Majorana neutrino masses)

Hierarchical prior,
sampled with adaptive Metropolis-Hastings backend.

[PRL 125, 252502 (2020)]

Monotone Rational-Quadratic Splines



K Segments

Characterized by

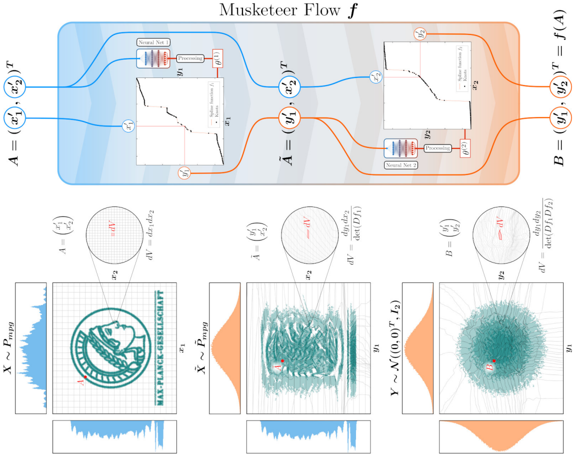
$\{x_k, y_k\},\ \{\delta_k\}$

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

MonoticSplines.jl: Based on "Neural Spline Flows" [NeurIPS 2019], high-performance CPU+GPU via KernelAbstractions.jl.

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

Spline flows for low-dim marginals


Musketeer Flow f
$A = (x'_1, x'_2)^T$
Neural Net 1
Processing
$\theta^{(1)}$
Spline function f_1
Knots
$\tilde{A} = (y'_1, x'_2)^T$
Neural Net 2
Processing
$\theta^{(2)}$
Spline function f_2
Knots
$B = (y'_1, y'_2)^T = f(A)$
$A = \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix}$
$dV = dx_1 dx_2$
$\tilde{A} = \begin{pmatrix} y'_1 \\ x'_2 \end{pmatrix}$
$dV = \frac{dy_1 dx_2}{\det(Df_1)}$
$B = \begin{pmatrix} y'_1 \\ y'_2 \end{pmatrix}$
$dV = \frac{dy_1 dy_2}{\det(Df_1 Df_2)}$
$X \sim P_{mpg}$
$\tilde{X} \sim \tilde{P}_{mpg}$
$Y \sim \mathcal{N}((0,0)^T, I_2)$
MAX-PLANCK-GESELLSCHAFT

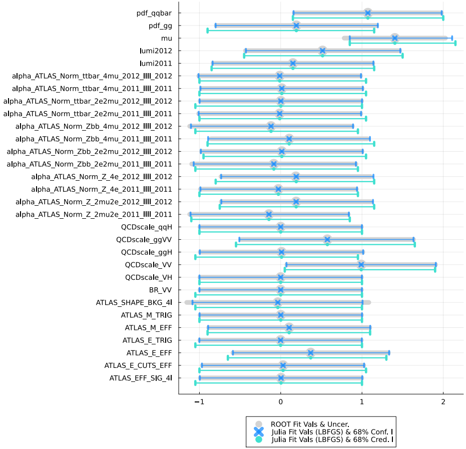
Trying to turns this into an automated tool to pass marginal posteriors around (once trained, math is quite simple).

Challenge: Machine learning is hard to automatize.

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, some BAT tooling and other stuff, needs to be packaged properly.

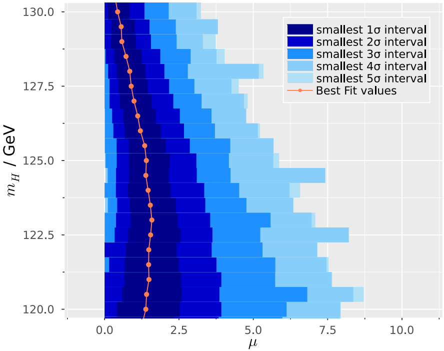
Julia HS3 Higg Parameter Estimates



[Master thesis Robin Pelkner, TU Dortmund]

- 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)

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]

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

Conclusions and Outlook

- Still no "RooFit", but . . .
- . . . many of the pieces in places.
- For Bayesians: BAT.jl tries to make Baysian inference easy, across multiple backends
  now also useful for some frequentist stuff
- Julia implementation of upcoming HS3-standard can get us full RooFit compatibility, but needs more work.
- In general:
  We should try to integrate with statistic packages
  across the ecosystem, instead of building "HEP-stats-island".