Turing.jl: what's new?

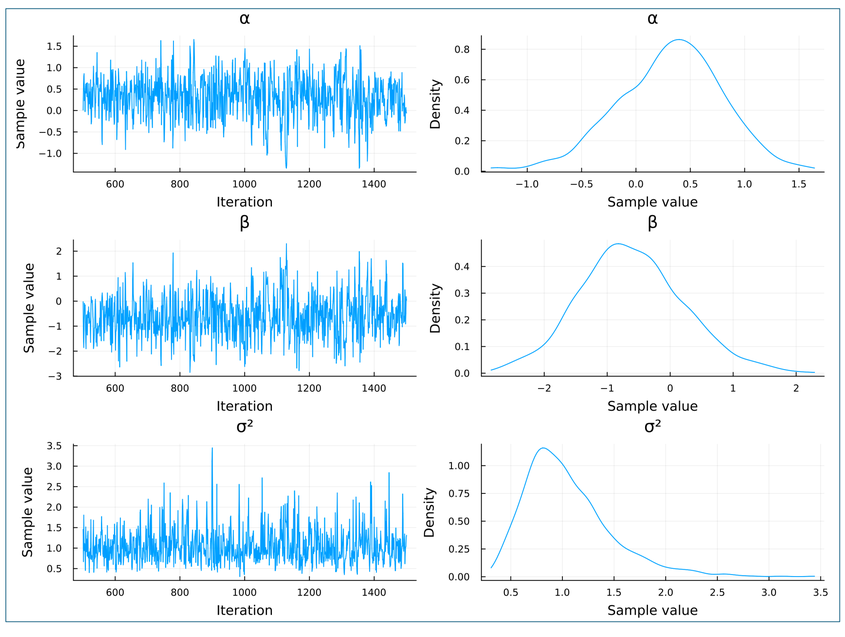
Markus Hauru, Penelope Yong
The Alan Turing Institute, United Kingdom

UNIVERSITY OF CAMBRIDGE
The Alan Turing Institute

What is Turing.jl?

Turing.jl is a package that makes Bayesian statistical inference easy. The user writes a statistical model using simple, intuitive syntax, and gives it observed data. With a single function call they can access a variety of inference methods, such as Markov chain Monte Carlo sampling from the posterior distribution.

```julia
using Turing, StatsPlots

@model function linear_regression(x)
    # Priors
    α ~ Normal(0, 1)
    β ~ Normal(0, 1)
    σ² ~ truncated(Normal(0, 1), lower=0)

    # Likelihood
    μ = α .+ β .* x
    y ~ MvNormal(μ, σ² * I)
end

x = rand(10)
y = 0.4 .* x .- 0.4 .+ randn(10)

m = linear_regression(x)
m_conditioned = m | (; y=y)

chain = sample(m_conditioned, NUTS(), 1000)
plot(chain)
```



> 9 years of development
> 2000 GitHub stars
> 500 citations

Applications

- Astrophysics, drug research, COVID modelling, cosmology, etc.
- Industry, academia, and public sector.

Turing models can have

- Continuous and discrete parameters
- Custom distributions
- Normal Julia control flow (`if`, `for`, `while`, etc.), including stochastic control flow
- Submodels within models
- Calls to (almost) any external libraries

Inference methods

- Markov chain Monte Carlo (Hamiltonian, particle methods, Gibbs sampling, etc.)
- External samplers
- Variational inference
- Maximum likelihood & maximum a posteriori

Works with

- DifferentialEquations.jl
- ForwardDiff, ReverseDiff, Enzyme & Mooncake
- Pigeons.jl
- Pathfinder.jl
- Lux.jl
- HiddenMarkovModels.jl
- Gaussian processes

Our development priorities

Pay off technical debt, simplify internals

Improve documentation

Fix a stable v1.0 interface

turinglang.org

Code examples

GitHub

Done

Submodels

Allow you to use one Turing.jl model within another.

- New syntax: `x ~ to_submodel(m)` instead of `@submodel(m)`
- Variables in submodels can now be conditioned or fixed to given values.

Table comparing automatic differentiation backends

Model name \\ AD type	EnzymeForward	EnzymeReverse	FiniteDifferences	ForwardDiff	MooncakeForward	MooncakeReverse	ReverseDiff	ReverseDiffCompiled	Zygote
broadcast_macro	3.016	2.826	35.892	1.434	25.138	6.426	29.891	2.907	error
dot_assume	2.916	1.902	76.795	1.426	29.447	5.064	20.883	1.882	error
dot_observe	2.972	2.648	27.654	1.606	19.247	7.219	56.512	5.333	error
dynamic_constraint	2.733	2.432	35.363	1.343	26.817	6.227	32.890	2.648	1735.748
multiple_constraints_same_var	3.056	2.227	59.541	1.299	27.281	12.725	29.531	2.343	error
observe_index	2.964	2.684	27.803	1.484	17.022	6.959	56.474	5.235	error
observe_literal	2.954	2.466	28.934	1.454	18.641	6.989	35.872	3.735	2922.312
observe_multivariate	4.063	2.600	52.210	1.338	22.467	5.379	24.795	2.238	error
observe_submodel	2.553	2.152	25.499	1.400	13.016	5.464	26.133	2.398	3253.258

`turinglang.org/ADTests`

Particle methods overhaul

Sequential Monte Carlo and Particle Gibbs samplers are now approximately 50–400% faster.

Variational inference improvements

- Full-rank Gaussian variational approximations, not just mean-field approximations
- Parameter-free stochastic optimization algorithms.

And more ...

- More granular selection of variables for the Gibbs sampler
- Mode estimation improvements
- Log probabilities now in chains
- Parallel progress bars!

In progress

v1.0 release

A stable user-facing interface: `@model` syntax, submodels, analysing chains.

Automatic differentiation

More extensive testing against, and support for, the next-generation autodiff packages Mooncake and Enzyme.

A new MCMC chain type

Faster; better support for heterogeneous data; a more coherent interface.

Variational inference improvements

- Data point subsampling (i.e. doubly stochastic variational inference).
- More modular and easily extendable to the benefit of algorithm developers and researchers.

And more ...

- Simpler evaluation contexts and tilde-pipeline
- Interoperability with JuliaBUGS.jl
- Better data structures for model traces

Future plans

New syntax for declaring variables

Would allow users to list names, types, and dimensions of all variables in a model. This would simplify Turing internals, make nonparametric methods more reliable to implement, and make it easier to implement robust samplers.

Submodels

Make submodels more first-class citizens, interchangeable with distributions.

Full thread-safety

Within a Turing.jl model, currently one can parallelise observations, but not creating parameters. We would like to lift this limitation.

And more ...

- brms-like interface for generalised linear models
- GPU support
- Educational materials on Bayesian inference