The Journal of Open Source Software

bayes_spec: A Bayesian Spectral Line Modeling Framework for Astrophysics

Trey V. Wenger [1]

1 NSF Astronomy & Astrophysics Postdoctoral Fellow, University of Wisconsin-Madison, USA

DOI: 10.21105/joss.07201

Software

- Review
- Repository
- Archive

Editor: Ivelina Momcheva

Reviewers:

- @ConorMacBride
- @kbwestfall
- @larryshamalama

Submitted: 15 August 2024
Published: 01 November 2024

Summary

The study of the interstellar medium (ISM)—the matter between the stars—relies heavily on the tools of spectroscopy. Spectral line observations of atoms, ions, and molecules in the ISM reveal the physical conditions and kinematics of the emitting gas. Robust and efficient numerical techniques are thus necessary for inferring the physical conditions of the ISM from observed spectral line data.

bayes_spec is a Bayesian spectral line modeling framework for astrophysics. Given a user-defined model and a spectral line dataset, bayes_spec enables inference of the model parameters through different numerical techniques, such as Monte Carlo Markov Chain (MCMC) methods, implemented in the PyMC probabilistic programming library (Oriol et al., 2023). The API for bayes_spec is designed to support astrophysical researchers who wish to "fit" arbitrary, user-defined models, such as simple spectral line profile models or complicated physical models that include a full physical treatment of radiative transfer. These models are "cloud-based", meaning that the spectral line data are decomposed into a series of discrete clouds with parameters defined by the user's model. Importantly, bayes_spec provides algorithms to determine the optimal number of clouds for a given model and dataset.

Statement of need

Bayesian models of spectral line observations are rare in astrophysics. Physical inference is typically achieved through inverse modeling: the spectral line data are decomposed into Gaussian components, and then the physical parameters are inferred from the fitted Gaussian parameters under numerous assumptions. For example, this is the strategy of `gausspy` (Lindner et al., 2015), `ROHSA` (Marchal et al., 2019), `pyspeckit` (Ginsburg et al., 2022), and `MWYDYN` (Rigby et al., 2024). This strategy suffers from two problems: (1) the degeneracies of Gaussian decomposition and (2) the assumptions of the inverse model. Bayesian forward models, like those enabled by bayes_spec, can overcome both of these limitations because (1) prior knowledge about the physical conditions can reduce the space of possible solutions, and (2) all assumptions are explicitly built into the model rather than being applied *a priori*.

bayes_spec is inspired by AMOEBA (Petzler et al., 2021), an MCMC-based Bayesian model for interstellar hydroxide observations. `McFine` (Williams & Watkins, 2024) is a new MCMC-based Bayesian model for hyperfine spectroscopy similar in spirit to bayes_spec. With bayes_spec, we aim to provide a user-friendly, general-purpose Bayesian modeling framework for *any* astrophysical spectral line observation.

Usage

Here we demonstrate how to use bayes_spec to fit a simple Gaussian line profile model to a synthetic spectrum. For more details, see the documentation and tutorials.

```
# Generate data structure
import numpy as np
from bayes_spec import SpecData

velocity_axis = np.linspace(-250.0, 250.0, 501) # km s-1
noise = 1.0 # K
brightness_data = noise * np.random.randn(len(velocity_axis)) # K
observation = SpecData(
    velocity_axis, brightness_data, noise,
    ylabel=r"Brightness Temperature $T_B$ (K)",
    xlabel=r"LSR Velocity $V_{\rm LSR}$ (km s$^{-1})$",
)
data = {"observation": observation}

# Prepare a three cloud GaussLine model with polynomial baseline degree = 2
from bayes_spec.models import GaussModel

model = GaussModel(data, n_clouds=3, baseline_degree=2)
model.add_priors()
model.add_likelihood()

# Evaluate the model for a given set of parameters to generate a
# synthetic "observation"
sim_brightness = model.model.observation.eval({
    "fwhm": [25.0, 40.0, 35.0], # FWHM line width (km/s)
    "line_area": [250.0, 125.0, 175.0], # line area (K km/s)
    "velocity": [-35.0, 10.0, 55.0], # velocity (km/s)
    # normalized baseline coefficients
    "baseline_observation_norm": [-0.5, -2.0, 3.0],
})

# Pack data structure with synthetic "observation"
observation = SpecData(
    velocity_axis, sim_brightness, noise,
    ylabel=r"Brightness Temperature $T_B$ (K)",
    xlabel=r"LSR Velocity $V_{\rm LSR}$ (km s$^{-1})$",
)
data = {"observation": observation}

# Initialize the model with the synthetic observation
model = GaussModel(data, n_clouds=3, baseline_degree=2)
model.add_priors()
model.add_likelihood()

# Draw posterior samples via MCMC
model.sample()

# Solve labeling degeneracy
model.solve()
```

```python
# Draw posterior predictive samples
from bayes_spec.plots import plot_predictive

posterior = model.sample_posterior_predictive(thin=100)
axes = plot_predictive(model.data, posterior.posterior_predictive)
axes.ravel()[0].figure.show()

# visualize posterior distribution
from bayes_spec.plots import plot_pair

axes = plot_pair(
    model.trace.solution_0,
    model.cloud_deterministics,
    labeller=model.labeller,
)
axes.ravel()[0].figure.show()
```

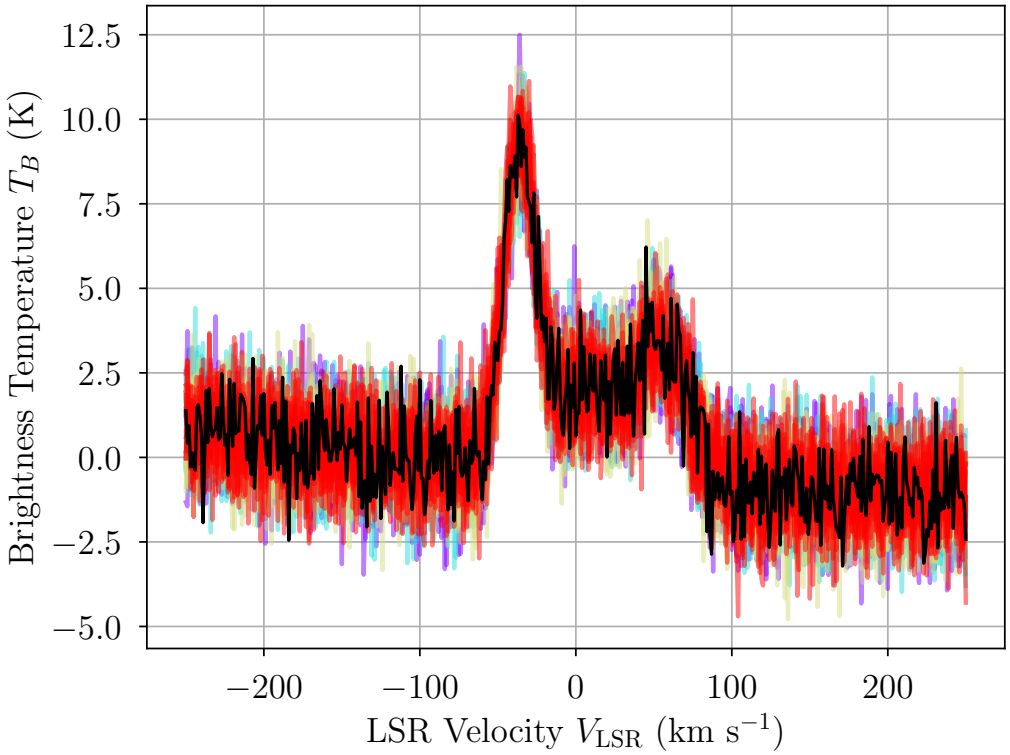

Figure 1: Posterior predictive samples for a three-cloud `GaussLine` model fit to a synthetic spectrum. The black line represents the synthetic spectrum, and each colored line is one posterior predictive sample.

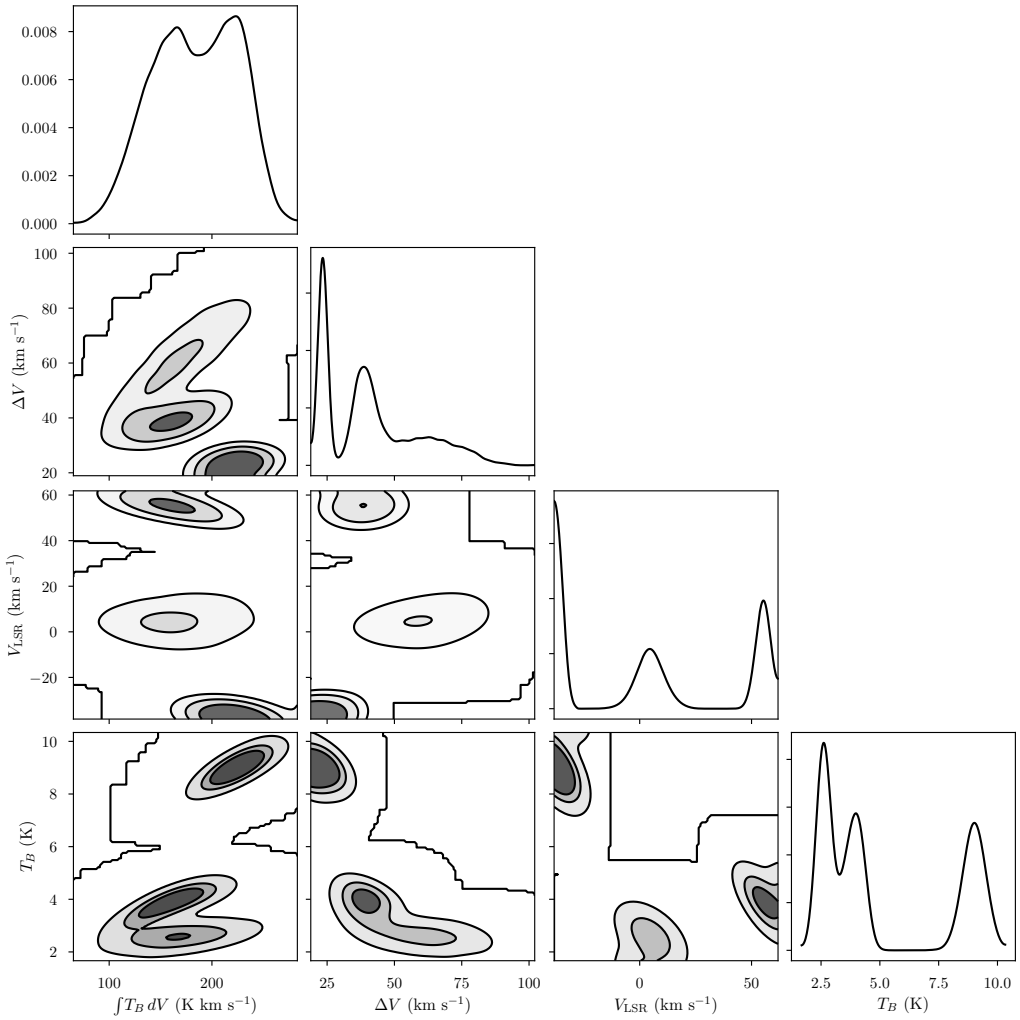

Figure 2: Projections of the posterior distribution for a three-cloud `GaussLine` model fit to a synthetic spectrum. The free model parameters are the integrated line area, $\int T_B dV$, the full-width at half-maximum line width, ΔV, and the line-center velocity, $V_{\rm LSR}$. The line amplitude, T_B, is a derived quantity. The three posterior modes correspond to the three clouds in this model.

References

Ginsburg, A., Sokolov, V., de Val-Borro, M., Rosolowsky, E., Pineda, J. E., Sipőcz, B. M., & Henshaw, J. D. (2022). Pyspeckit: A spectroscopic analysis and plotting package. *The Astronomical Journal*, *163*(6), 291. https://doi.org/10.3847/1538-3881/ac695a

Lindner, R. R., Vera-Ciro, C., Murray, C. E., Stanimirović, S., Babler, B., Heiles, C., Hennebelle, P., Goss, W. M., & Dickey, J. (2015). Autonomous Gaussian decomposition. *The Astronomical Journal*, *149*(4), 138. https://doi.org/10.1088/0004-6256/149/4/138

Marchal, A., Miville-Deschênes, M.-A., Orieux, F., Gac, N., Soussen, C., Lesot, M.-J., d'Allonnes, A. R., & Salomé, Q. (2019). ROHSA: Regularized Optimization for Hyper-Spectral Analysis. Application to phase separation of 21 cm data. *Astronomy & Astrophysics*, *626*, A101. https://doi.org/10.1051/0004-6361/201935335

Oriol, A.-P., Virgile, A., Colin, C., Larry, D., J., F. C., Maxim, K., Ravin, K., Jupeng, L., C., L. C., A., M. O., Michael, O., Ricardo, V., Thomas, W., & Robert, Z. (2023). PyMC:

A modern and comprehensive probabilistic programming framework in Python. *PeerJ Computer Science*, *9*, e1516. https://doi.org/10.7717/peerj-cs.1516

Petzler, A., Dawson, J. R., & Wardle, M. (2021). Amoeba: Automated Molecular Excitation Bayesian Line-fitting Algorithm. *The Astrophysical Journal*, *923*(2), 261. https://doi.org/10.3847/1538-4357/ac2f42

Rigby, A. J., Peretto, N., Anderson, M., Ragan, S. E., Priestley, F. D., Fuller, G. A., Thompson, M. A., Traficante, A., Watkins, E. J., & Williams, G. M. (2024). The dynamic centres of infrared-dark clouds and the formation of cores. *Monthly Notices of the Royal Astronomical Society*, *528*(2), 1172–1197. https://doi.org/10.1093/mnras/stae030

Williams, T. G., & Watkins, E. J. (2024). McFine: PYTHON-based Monte Carlo multicomponent hyperfine structure fitting. *Monthly Notices of the Royal Astronomical Society*, *534*(2), 1150–1165. https://doi.org/10.1093/mnras/stae2130