RIFT tutorial

Richard O'Shaughnessy

[with much assistance from James Clark for containers for today]

2020-11-19 ICERM workshop

https://github.com/oshaughn/RIFT_tutorials : 20201119-ICERM

Three tutorials

- Tutorial 1: RIFT principles
  - RIFT overview
  - Tutorial 1 [notebook]

- Tutorial 2: RIFT simple example: Binary black hole
  - RIFT and GW data analysis: Why RIFT?
  - Tutorial 2 [docker container]

- Tutorial 3: RIFT on real data

RIFT: Rapid Iterative Fitting (for inference)

- Problem to solve:

$$p(x) = \frac{\mathcal{L}(x) p_{prior}(x)}{\int \mathcal{L}(x) p_{prior}(x) dx}$$

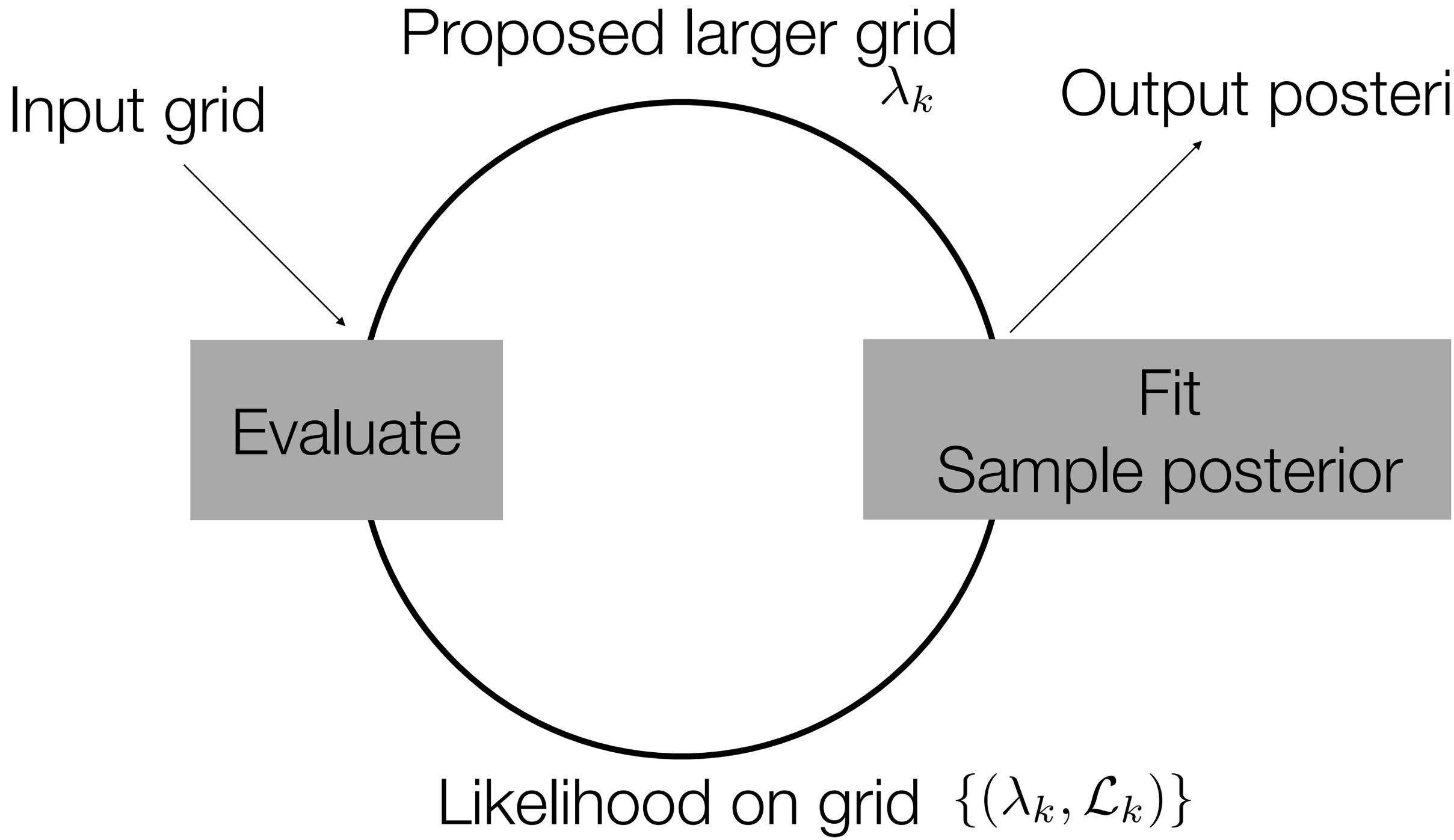
  - but $\mathcal{L}(x)$ is very expensive

- Method: Evaluate $\mathcal{L}(x)$ and train an approximation ("surrogate") $\hat{\mathcal{L}}(x)$
  - Iteratively refine
  - Target reliable posterior in 'bulk' of support (e.g.,, 99.9…% probability)

- Fit method: Anything flexible (e.g. random forests, …). We usually use gaussian processes in modest-sized problems
- Posterior generation: Monte Carlo integration. (Preferably adaptive)

RIFT: Rapid Iterative Fitting (for inference)

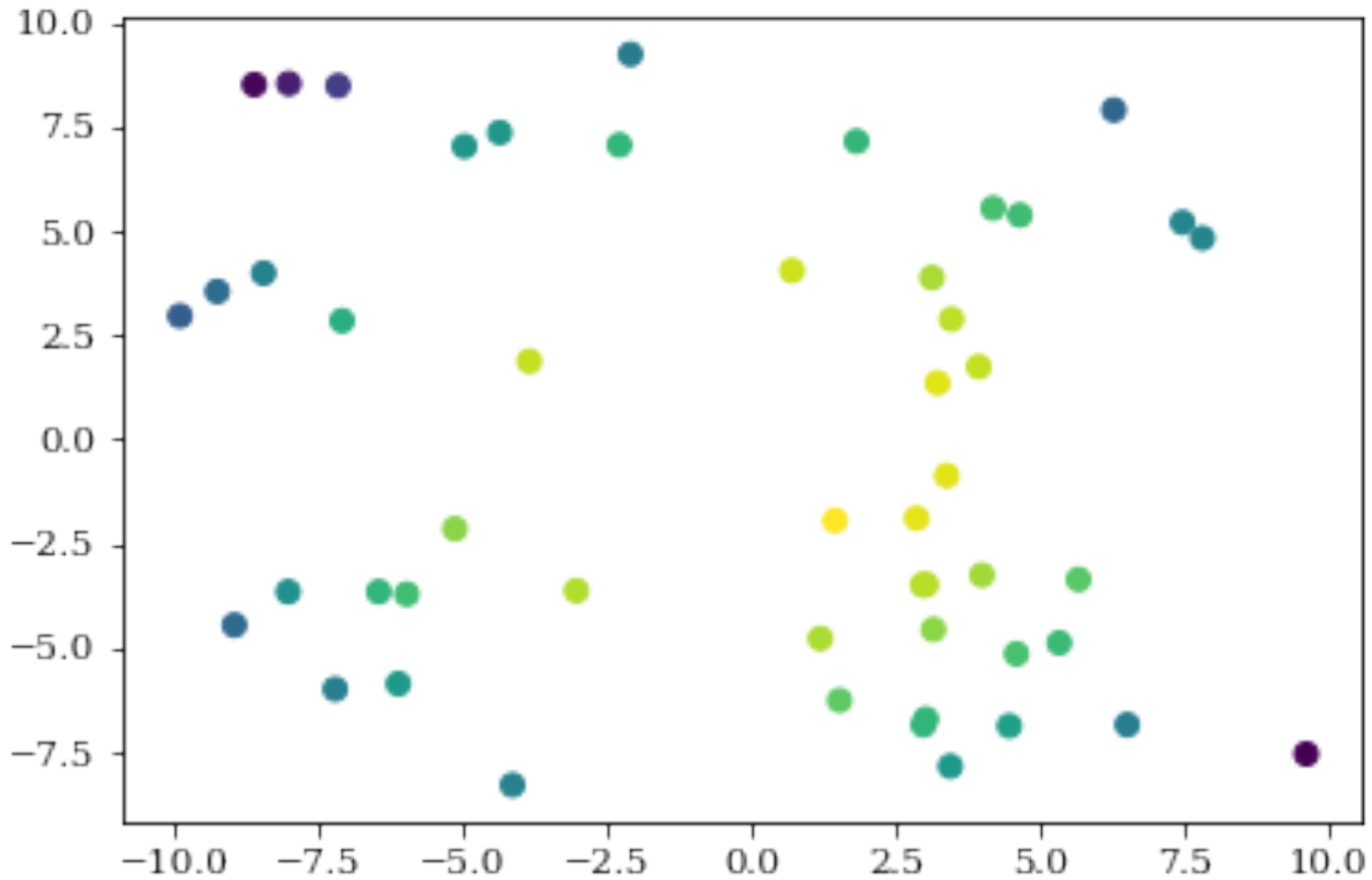
- **Repeat until stable**



Abbott et al 2016, PRD 94 4035

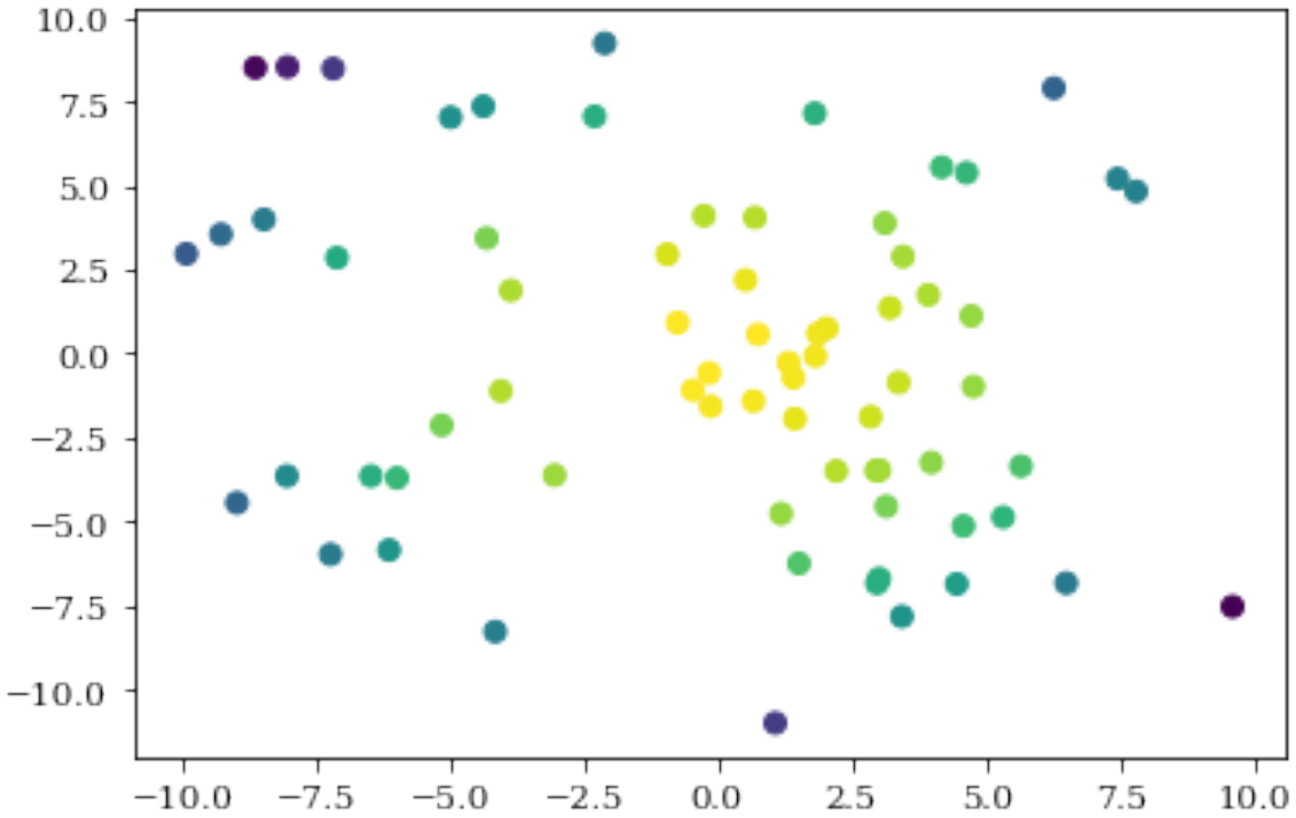
Tutorial 1: RIFT principles

- Toy problem : Gaussian
  - Iteration 0



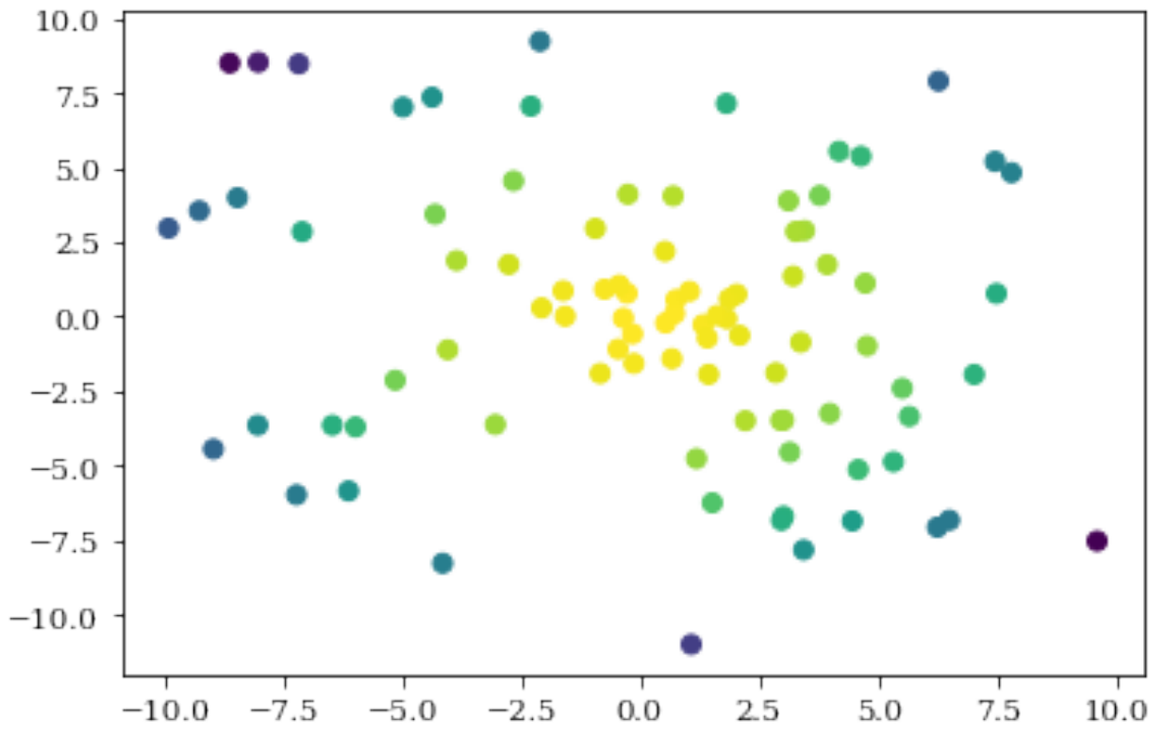
Tutorial 1: RIFT principles

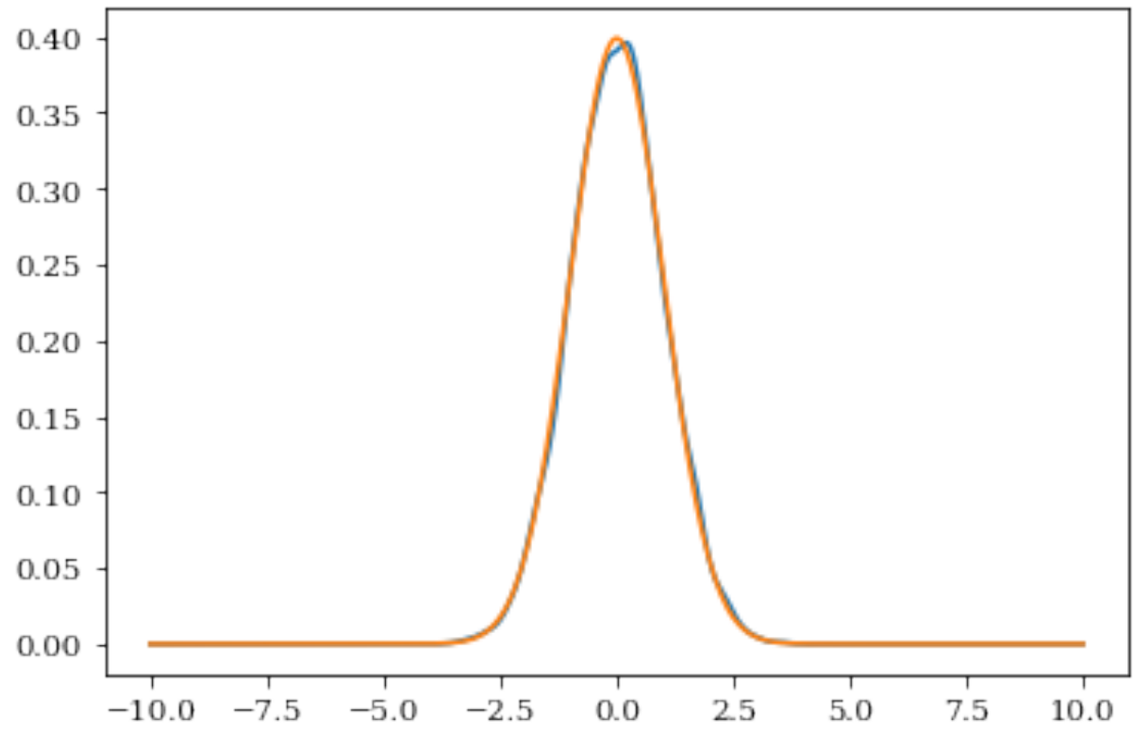
- Toy problem : Gaussian
  - Iteration 1



Tutorial 1: RIFT principles

- Toy problem : Gaussian
  - Iteration 2





Tutorial 1: Gaussian

- Toy problem
  - Notebook: link
  - Open notebook in colab.google.com

Background 2: RIFT and GW data

- RIFT works with GW data
  - Assumes likelihood is gaussian noise, with signal h, in detectors k

$$\ln \mathcal{L}(\lambda;\theta) = -\frac{1}{2}\sum_k \langle h_k(\lambda,\theta) - d_k | h_k(\lambda,\theta) - d_k \rangle_k - \langle d_k | d_k \rangle_k$$

  - **Very** efficient likelihood calculation (re-uses many things) **when "intrinsic" parameters (λ) fixed** (i.e., same binary seen different ways)

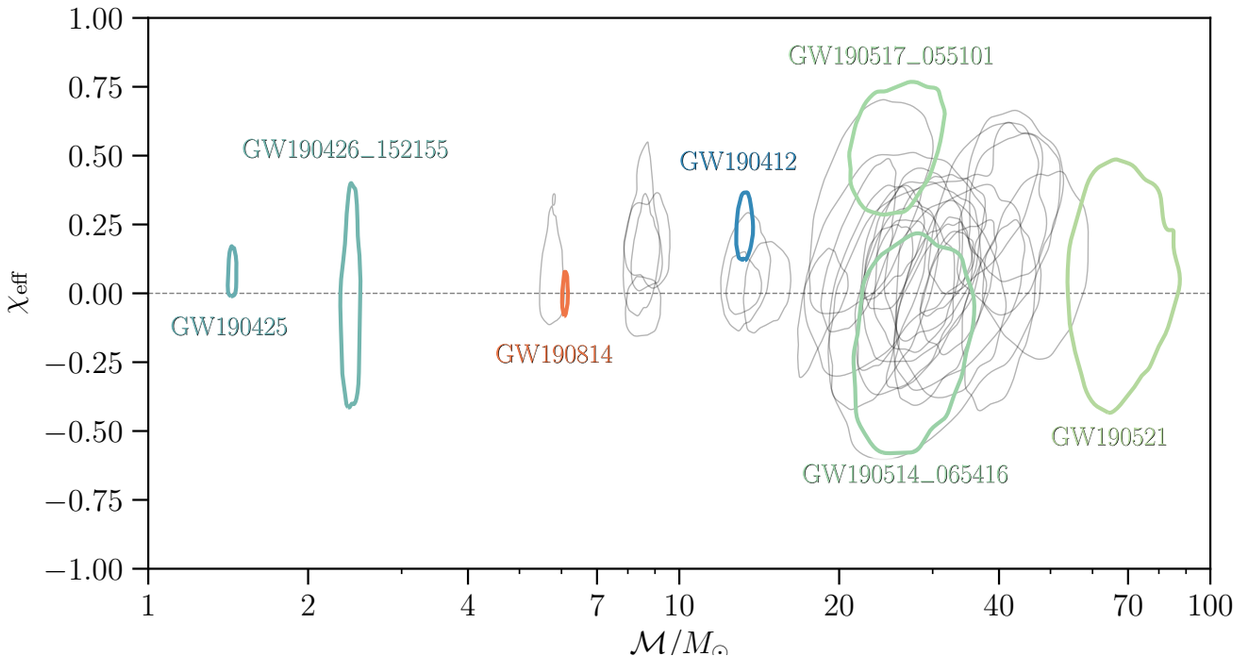
- RIFT splits the calculation into two parts: intrinsic and extrinsic

$$\mathcal{L}_{\rm marg}(\lambda) \equiv \int \mathcal{L}(\lambda,\theta) p(\theta) d\theta$$

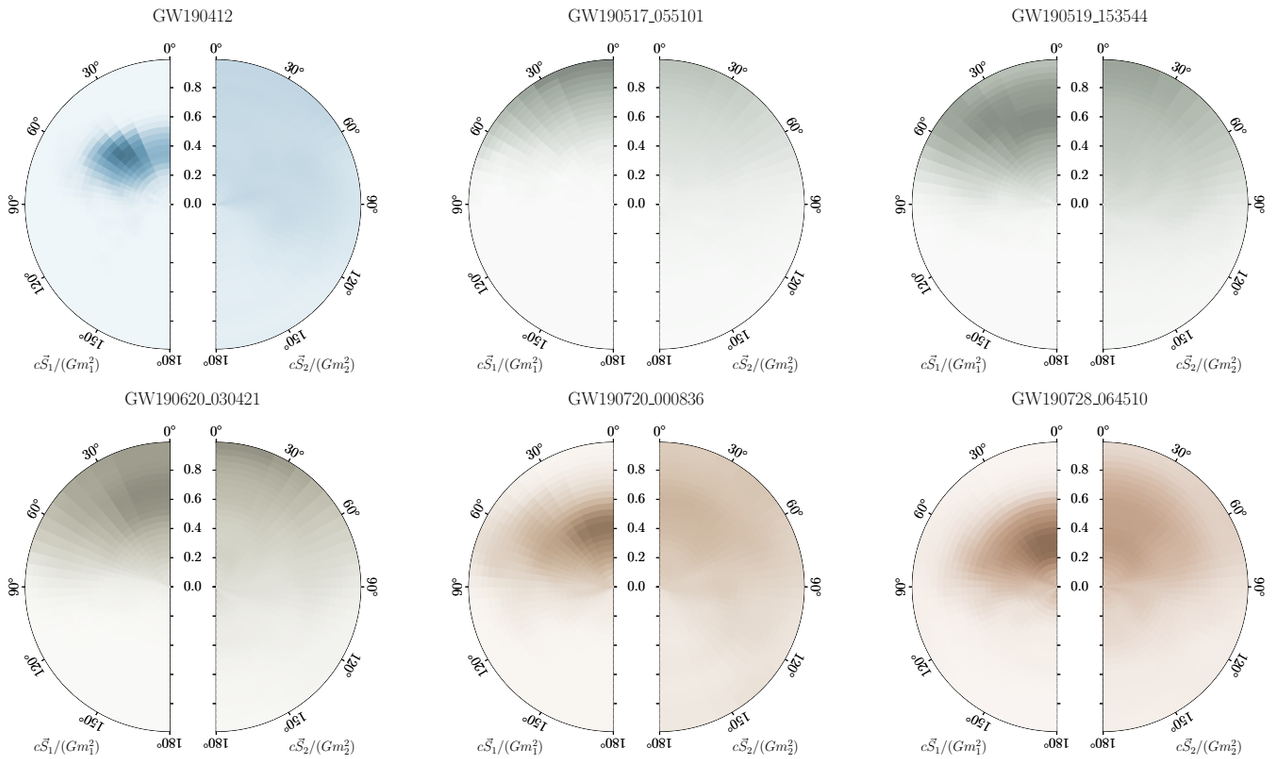
  - Iterative fitting applied to **marginal** likelihood

Why use RIFT? Very fast, cheap

- Optimized and low cost — good for expensive models
  - Example: LIGO GWTC-2 : Most HM calculations with RIFT

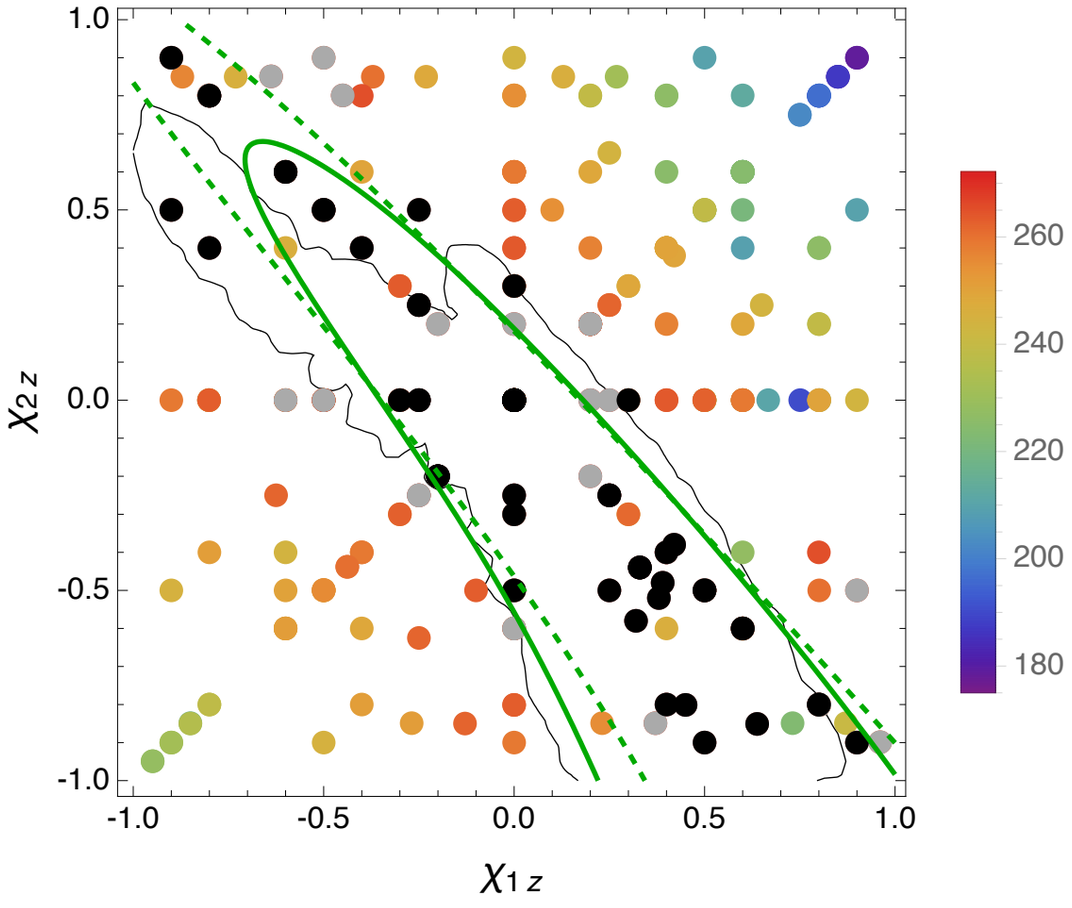


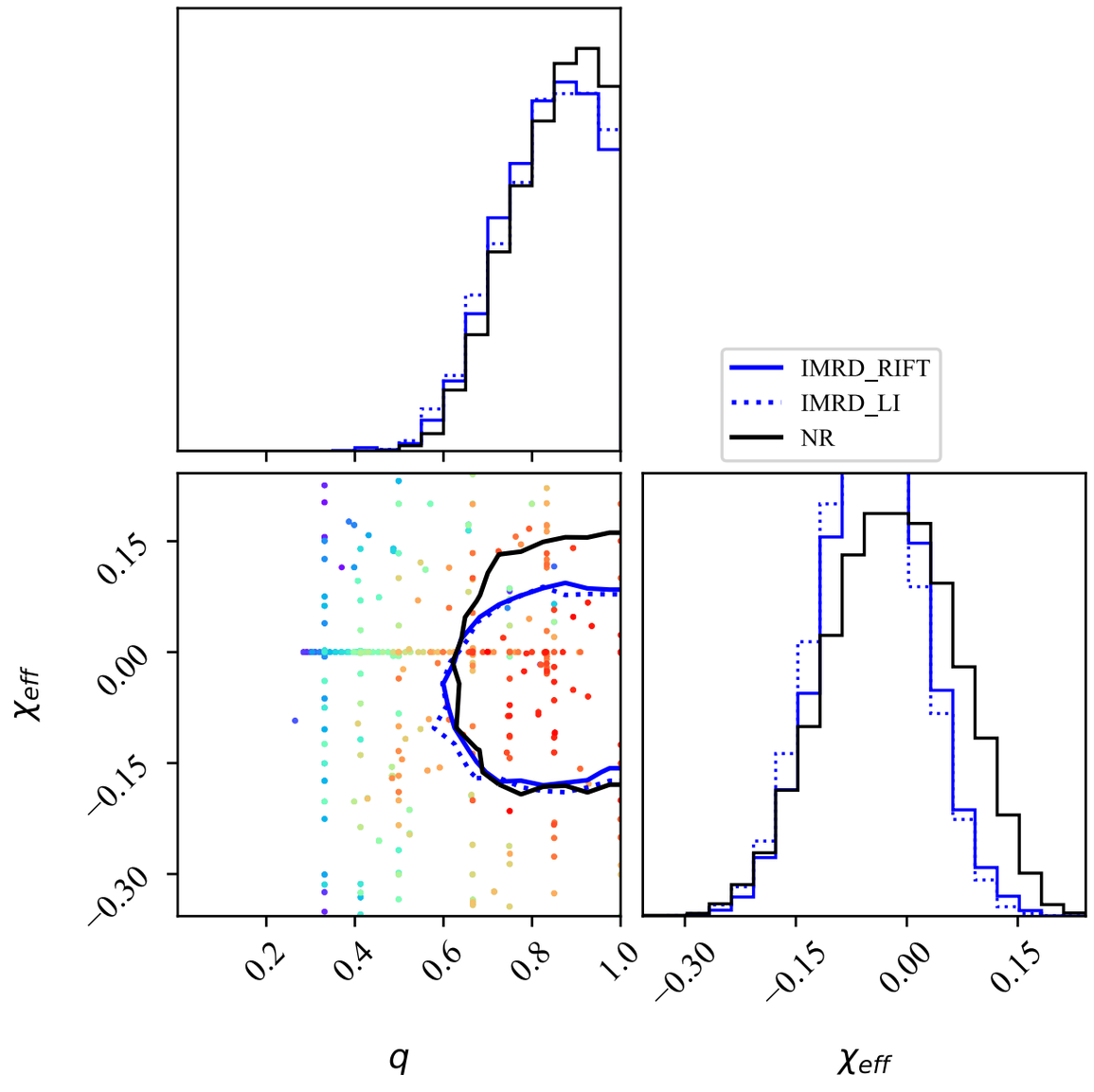
GWTC-2



Why use RIFT? Models not continuously available

- Some good models (NR) only available on discrete sets



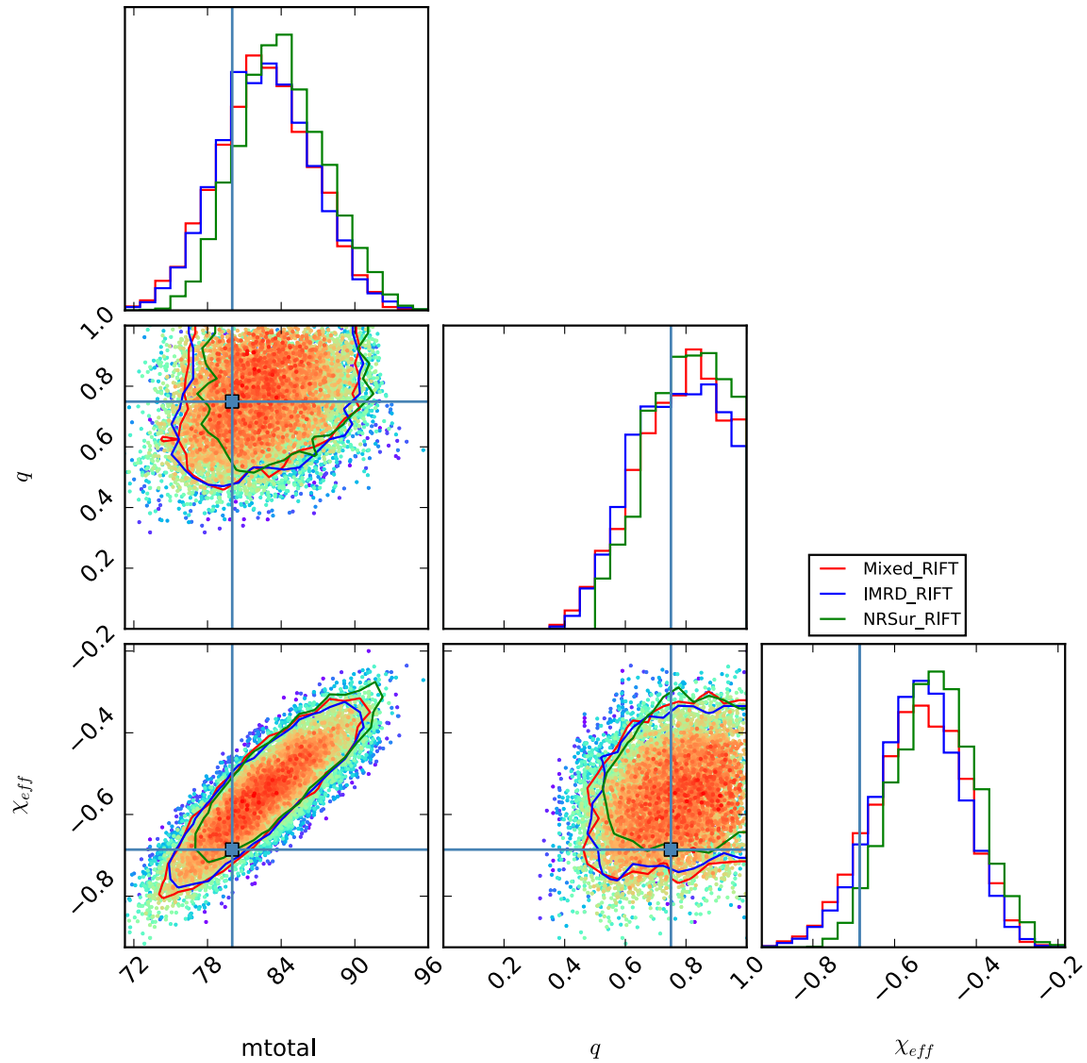


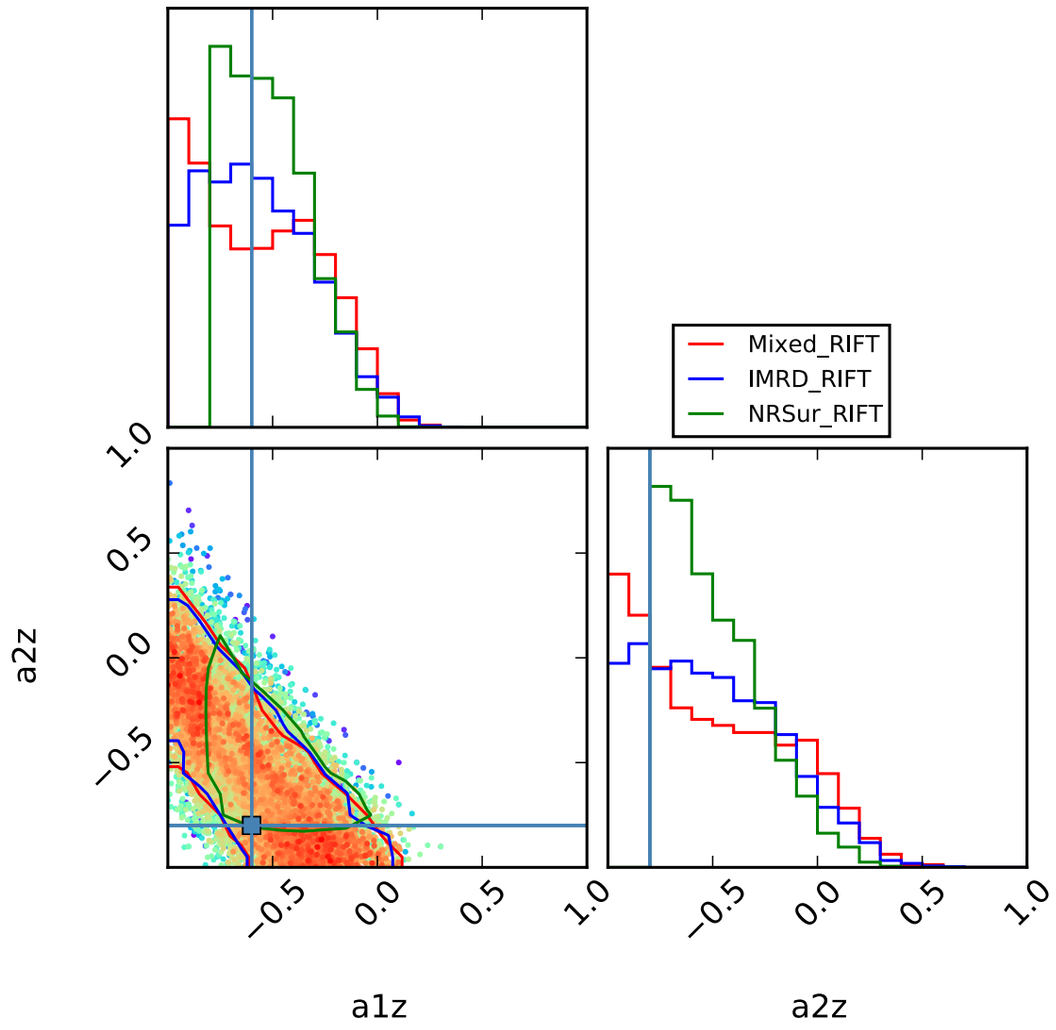
Abbott et al 2016, PRD 94 4035

see also Healy et al arxiv:2010.00108

Why use RIFT? Mixing models

- Some good models (NR-calibrated surrogates) have limited validity: mass ratio
  - Fill in the gaps with other models (PN; NR; …)



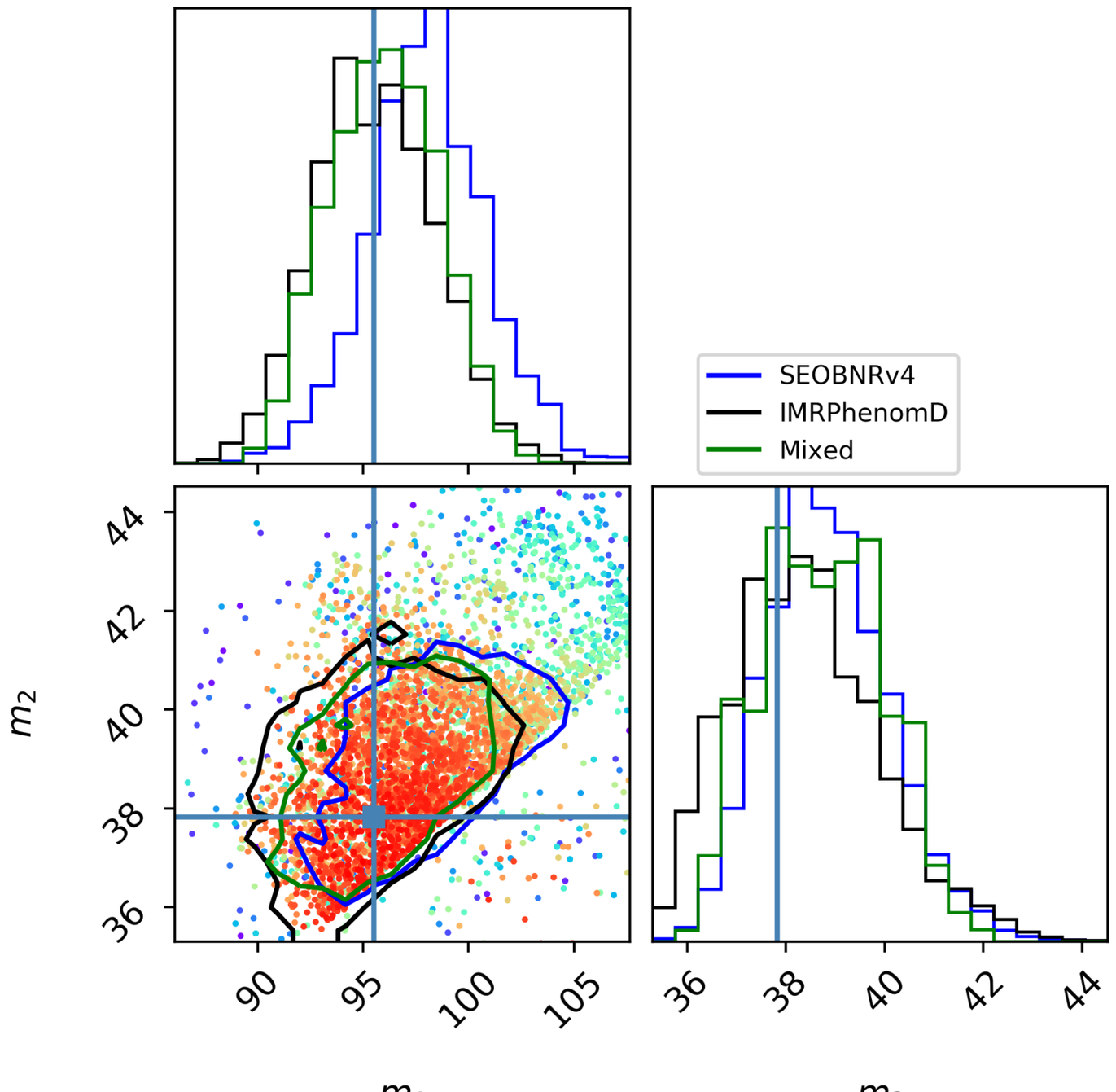


Lange et al arxiv:1805.06442

Why use RIFT? Model marginalization

- Natural technique to average over model differences

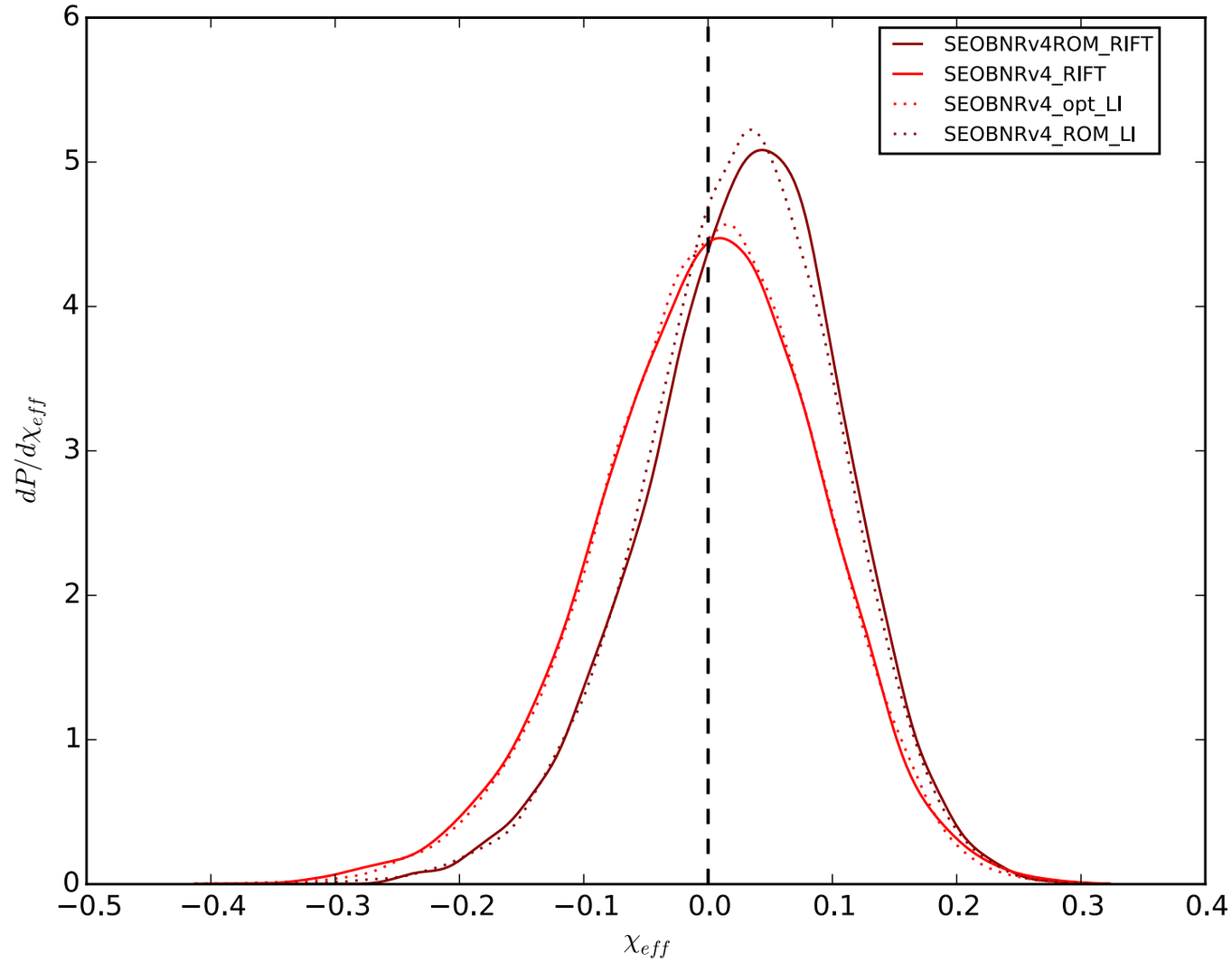
$$\mathcal{L}_1 p_1 + \mathcal{L}_2 p_2 + \ldots$$



Jan et al arXiv:2011.03571

Why use RIFT? Useful investigations easier

- Example of waveform systematics: GW150914-like synthetic source
  - One model designed as a fast approximation to another (match >0.997)



q=1,SNR~20

$\chi_{1,z} = -\chi_{2,z} = 0.25$

$M_z = 80 M_\odot$

Lange, LIGO-G1801107

- **Other examples:** 170729 paper; ROQ bias (mchirp for 170729);…

Getting oriented: a live walkthrough, v1

- Binary black hole with zero spin https://arxiv.org/abs/1902.04934
  - See tutorial link for instructions

Getting oriented: a live walkthrough, v2

- Same example, but if you have a cluster with condor

```
# install the code (e.g., in a virtual environment or as user)
pip install —user RIFT==0.0.15.4rc5    # or:   pip -m venv RIFT_demo; source RIFT_demo/bin/activate; pip install RIFT==0.0.15.4rc5


# Install and run the example
git  clone https://github.com/oshaughn/ILE-GPU-Paper.git
cd ILE-GPU-Paper/demos/
make test_workflow_batch_gpu_lowlatency
cd test_workflow_batch_gpu_lowlatency;   # good time to pause: do ‘./command-single.sh’ to confirm it is working!
condor_submit_dag marginalize_intrinsic_parameters_BasicIterationWorkflow.dag

# Monitor
watch -n 10 condor_q

# plot
```

Tutorial 3: RIFT on real data

- Automation to enable efficient setup and use