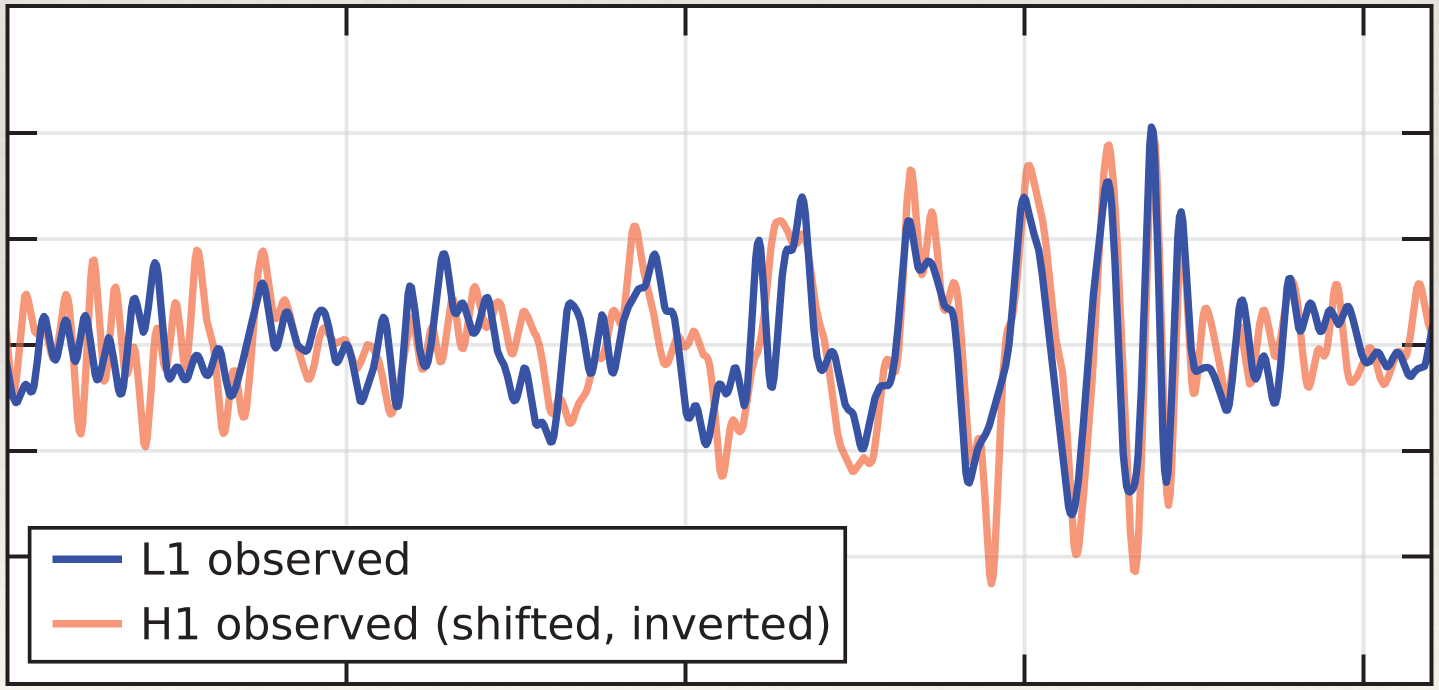
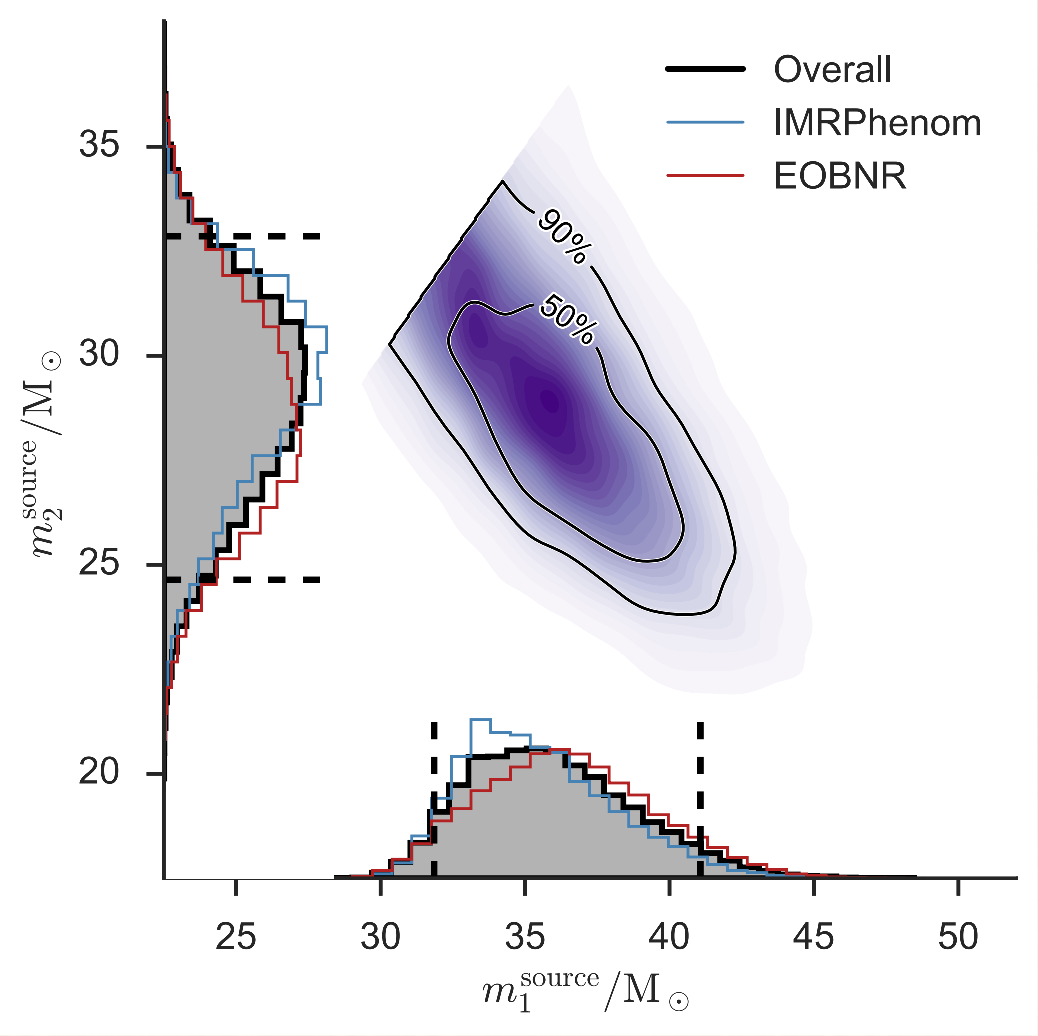
Gravitational Waves: Data Analysis Techniques

Jonathan Gair, University of Edinburgh

j.gair@ed.ac.uk





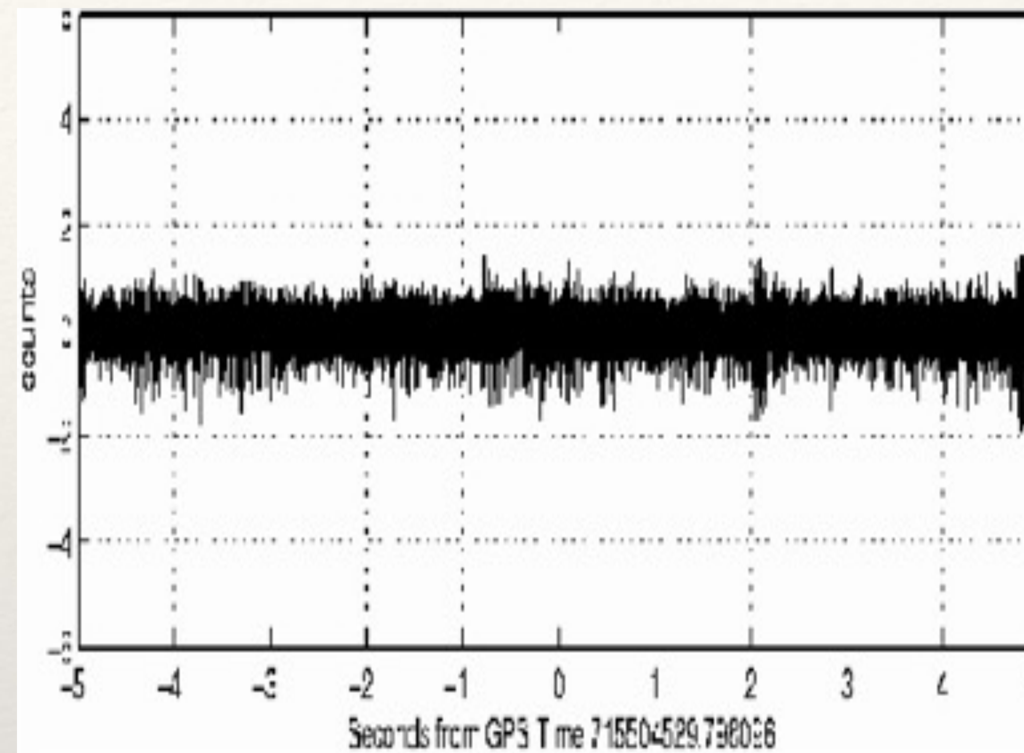
Searching for signals: Matched Filtering

Matched Filtering

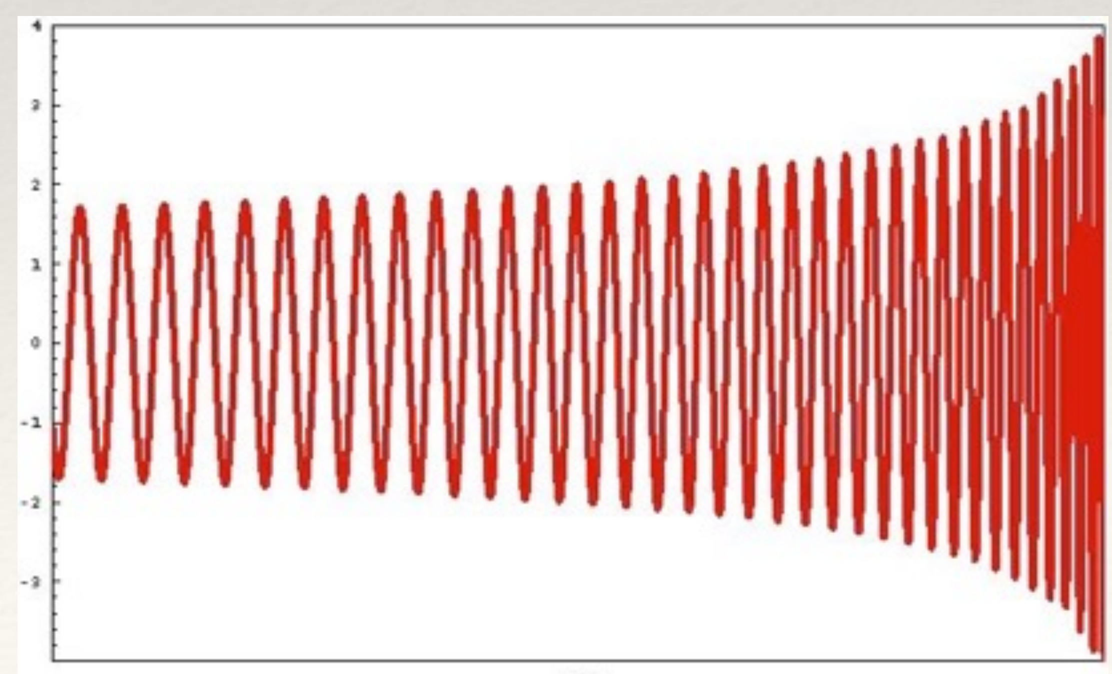
- Many searches are based on the concept of *matched filtering*.
- Recall from lecture 1 that the *optimal filter* for a known signal is one that matches the signal in the Fourier domain, weighted by the noise PSD.

$$\tilde{K}(f) = \frac{\tilde{h}(f)}{S_n(f)}$$

- In practice, signal is not known, so use a *template bank* of possible waveforms.
- If a template in the bank matches a signal in the data, we can pull it out of the noise



+

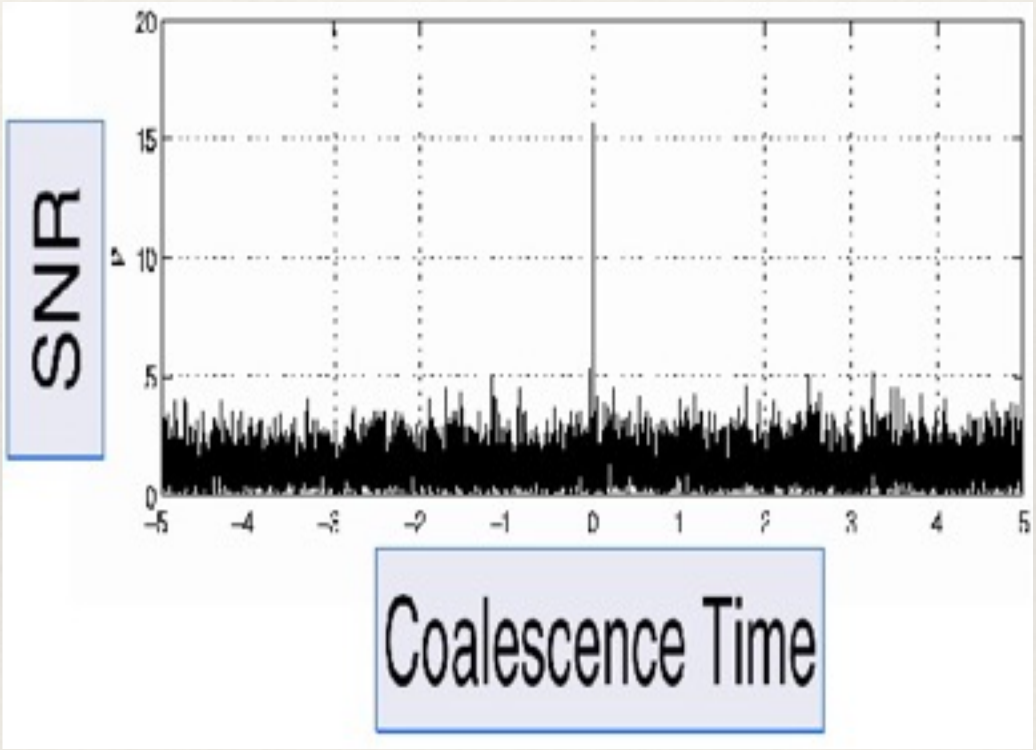


Matched Filtering

- Many searches are based on the concept of *matched filtering*.
- Recall from lecture 1 that the *optimal filter* for a known signal is one that matches the signal in the Fourier domain, weighted by the noise PSD.

$$\tilde{K}(f) = \frac{\tilde{h}(f)}{S_n(f)}$$

- In practice, signal is not known, so use a *template bank* of possible waveforms.
- If a template in the bank matches a signal in the data, we can pull it out of the noise



Likelihood

- Recall from first lecture our model for the detector output

$$s(t) = n(t) + h(t; \vec{\lambda})$$

- For stationary noise we have

$$\langle \tilde{n}^*(f)\tilde{n}(f')\rangle = S_n(f)\delta(f - f')$$

- If we additionally assume the noise is Gaussian then we can write down a probability distribution for *s(t)*

$$p(s|\vec{\lambda}) = p(n(t) = s(t) - h(t; \vec{\lambda})) \propto \exp\left[-\frac{1}{2}(s - h(\vec{\lambda})|s - h(\vec{\lambda}))\right]$$

- where

$$(a|b) = \int_{-\infty}^{\infty} \frac{\tilde{a}^*(f)\tilde{b}(f) + \tilde{a}(f)\tilde{b}^*(f)}{S_n(f)}\, \mathrm{d}f$$

- For normalised templates, maximum likelihood correspond to matched filter.

How well can we do: Fisher Matrix?

- If we write

$$s(t) = n(t) + h(t;\vec{\lambda}_0)$$

- and expand (this is the *linear signal approximation*)

$$\vec{\lambda} = \vec{\lambda}_0 + \Delta\vec{\lambda} \qquad h(t;\vec{\lambda}) = h(t;\vec{\lambda}_0) + \partial_i h(t;\vec{\lambda}_0)\Delta\lambda^i$$

- we find

$$p(s|\lambda) \propto \exp\left[-\frac{1}{2}\left(\Delta\lambda^i - (\Gamma^{-1})_{ik}(n|\partial_k h(t;\lambda_0)\right)\Gamma_{ij}\left(\Delta\lambda^j - (\Gamma^{-1})_{jl}(n|\partial_0 h(t;\lambda_0)\right)\right]$$

- where

$$\Gamma_{ij} = \left(\frac{\partial h}{\partial\lambda_i}\middle|\frac{\partial h}{\partial\lambda_j}\right)$$

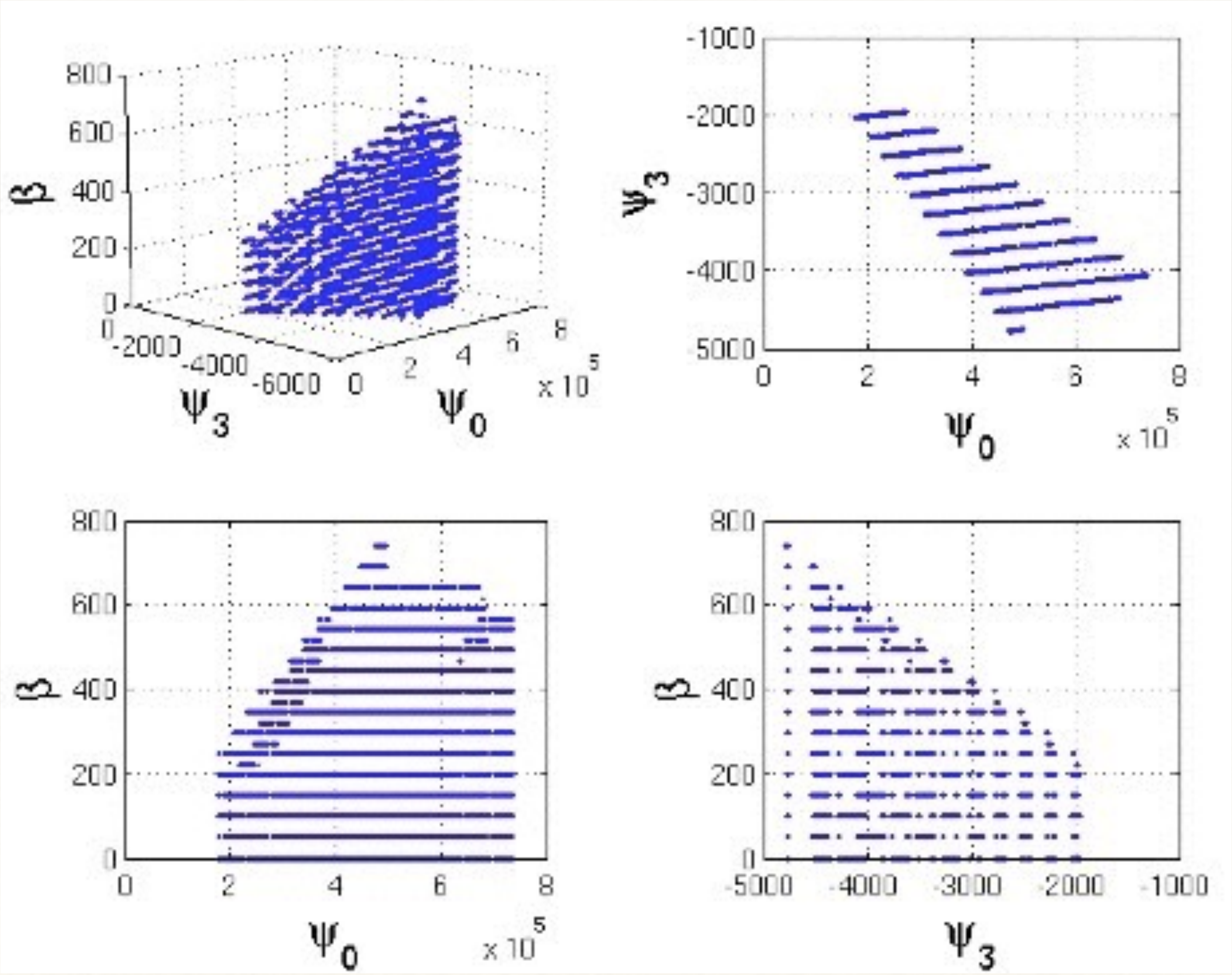
- is the *Fisher Information Matrix*.

Template Bank Construction

- Fisher matrix used to estimate precision of parameter estimation.
- Can also be used as a metric to construct a template bank satisfying a *minimal match* criterion

$$\min_{h_{\rm true}} \max_{h_{\rm temp}} \left(h_{\rm true}|h_{\rm temp}\right) \gtrsim 1 - {\rm MM}$$

- Fisher Matrix metric not easy to use in higher dimensional parameter spaces. Now common to use *stochastic banks*.
- Can also do *stochastic searches* (MCMC) that generate templates on the fly.



Waveform Consistency

- If we subtract the correct template the residual at each frequency should be Normally distributed.

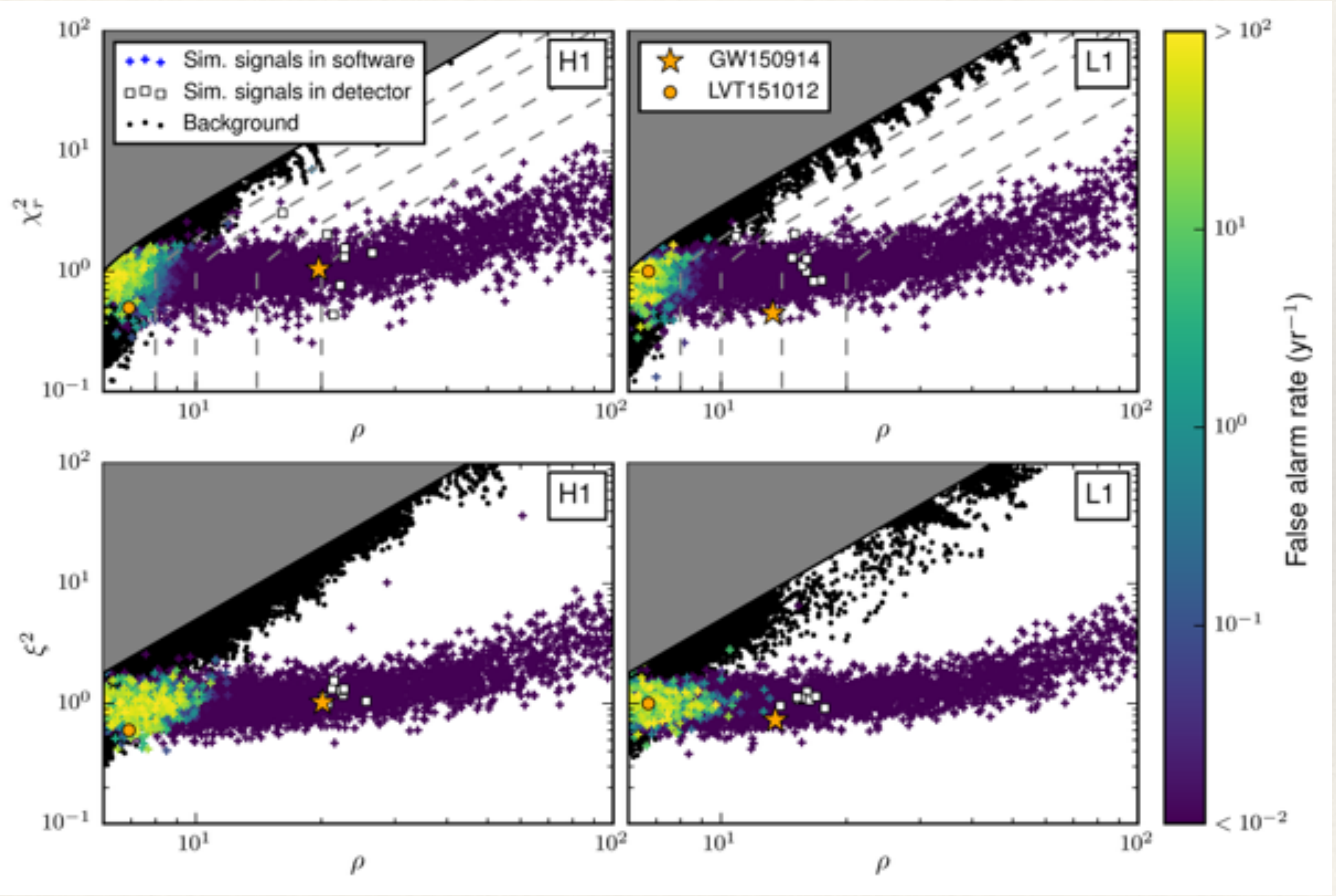
- Hence the quantity

$$\chi^2 = \sum_{k=1}^{N} \frac{|\hat{s}_k - \hat{h}_k|^2}{S_n(f_k)}$$

- follows a chi-squared distribution.

- Construct an *effective SNR* that penalises lack of fit

$$\hat{\rho} = \frac{\rho}{(1 + (\chi^2/N)^3)^{\frac{1}{6}}}$$



LVC, *Phys. Rev.* D **93**, 122003 (2016)

PSD Estimation

- Matched filter is noise-weighted. OK if you know the noise PSD, but in general we will not. For LIGO, estimate this using off-source data.

$s_{-3} = n_{-3}$	$s_{-2} = n_{-2}$	$s_{-1} = n_{-1}$	$s_0 = h + n_0$	$s_1 = n_1$	$s_2 = n_2$	$s_3 = n_3$

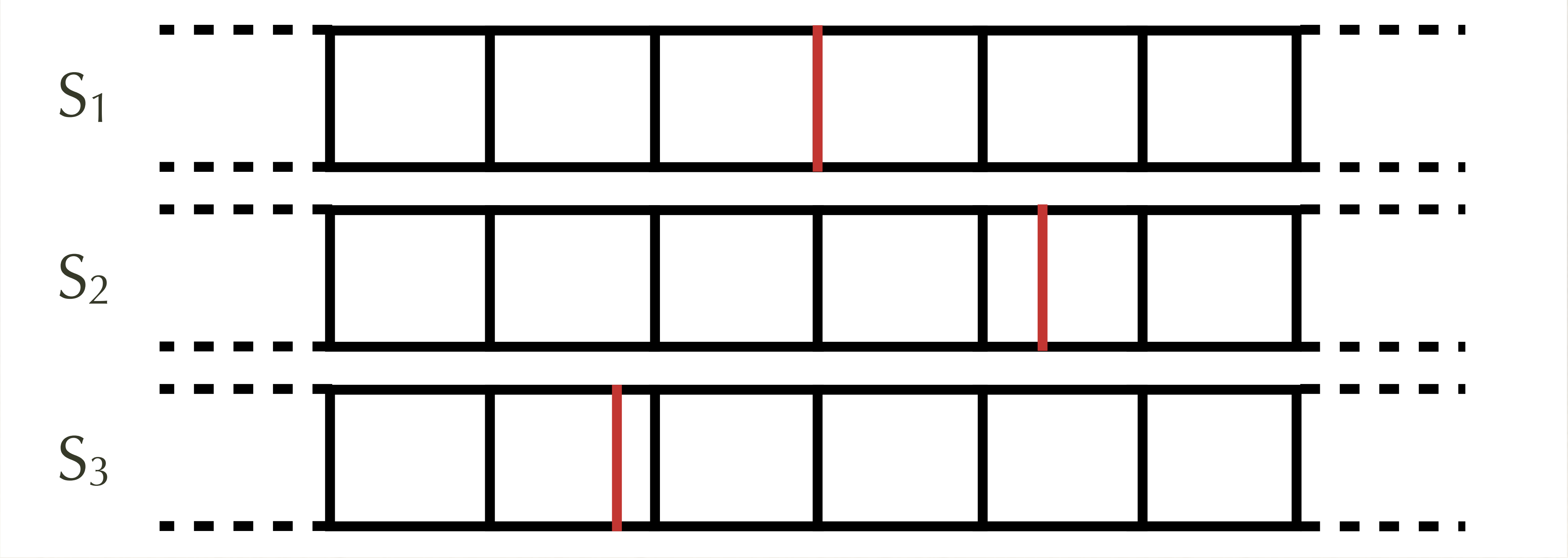
$$\sigma_0^2 \approx \frac{1}{2N} \sum_{k=1}^{N} \left(s_k^2 + s_{-k}^2 \right)$$

- In practice, use median of noise estimates, rather than the average. This is less sensitive to non-stationarities.

- No off-source data for LISA. Make progress by fitting noise and signal properties simultaneously - need reasonable noise model.

Background Estimation

- Noise is not stationary or Gaussian and contains glitches, lines etc.
- Use frequentist techniques to characterise noise background properties
  - process data in a way that eliminates signal but not noise
  - for LIGO - time slide data from different detectors



  - noise + signal coincidences are not background
  - significance of events in tail, i.e., sources, is hard to estimate

Phase and Time Parameters

- Certain parameters can be maximised over cheaply, e.g., unknown phase

$$h(t; A, f_0, t_c, \phi_0) = A\cos(2\pi f_0(t - t_c) + \phi_0)$$

$$\max_{\phi_0}(s|h)^2 = A^2\left((s|h(t; A, f_0, t_c, 0))^2 + (s|h(t; A, f_0, t_c, -\pi/2))^2\right)$$

- and unknown coalescence time

$$\tilde{h}(f; A, f_0, t_c, \phi_0) = \tilde{h}(f; A, f_0, 0, \phi_0)\exp(-2\pi i f t_c)$$

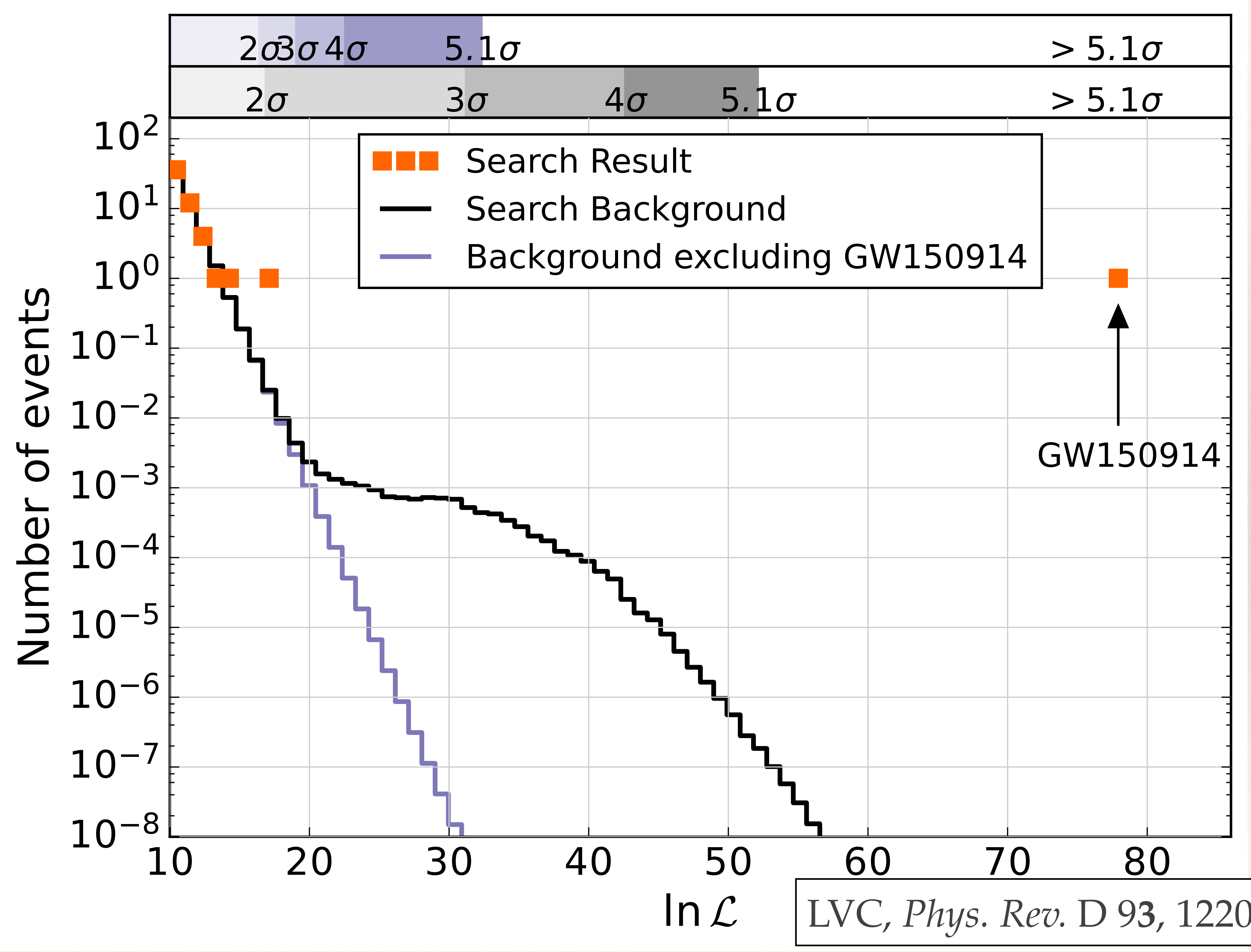
$$(s|h(t; A, f_0, t_c, \phi_0)) = 2\mathrm{Re}\int_{-\infty}^{\infty}\frac{\tilde{s}^*(f)\tilde{h}(f; A, f_0, 0, \phi_0)}{S_n(f)}\exp\left(-2\pi i f t_c\right)\,\mathrm{d}f$$

- This is the inverse Fourier transform of $\tilde{s}^*(f)\tilde{h}(f; A, f_0, 0, \phi_0)/S_n(f)$. Obtain overlap for all time offsets cheaply using a Fast Fourier Transform.

LIGO Pipelines

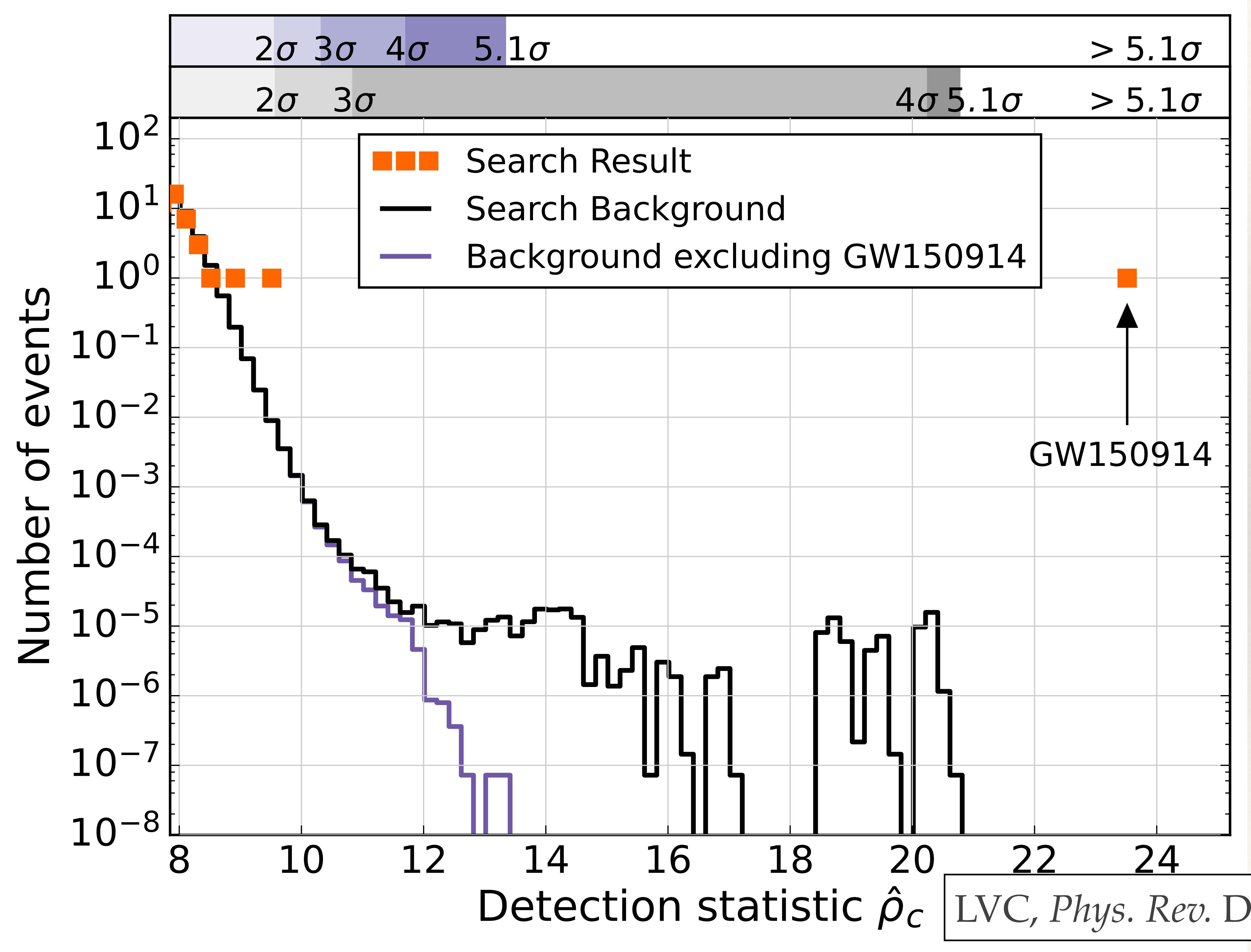
- Two main matched filtering pipelines used in LIGO for compact binary coalescence searches.
  - *pycbc*: constructs template bank of waveforms; computes chi-squared test for fit; uses effective SNR as a ranking statistic; background computed using time slides.
  - *gstLAL*: constructs template bank of waveforms, then does SVD decomposition to form a signal basis; detection statistic is likelihood ratio for signal versus noise; waveform consistency assessed by comparing SNR time series to theory; time slides again used to assess background.

LIGO Pipelines



LVC, *Phys. Rev.* D **93**, 122003 (2016)

LIGO Pipelines

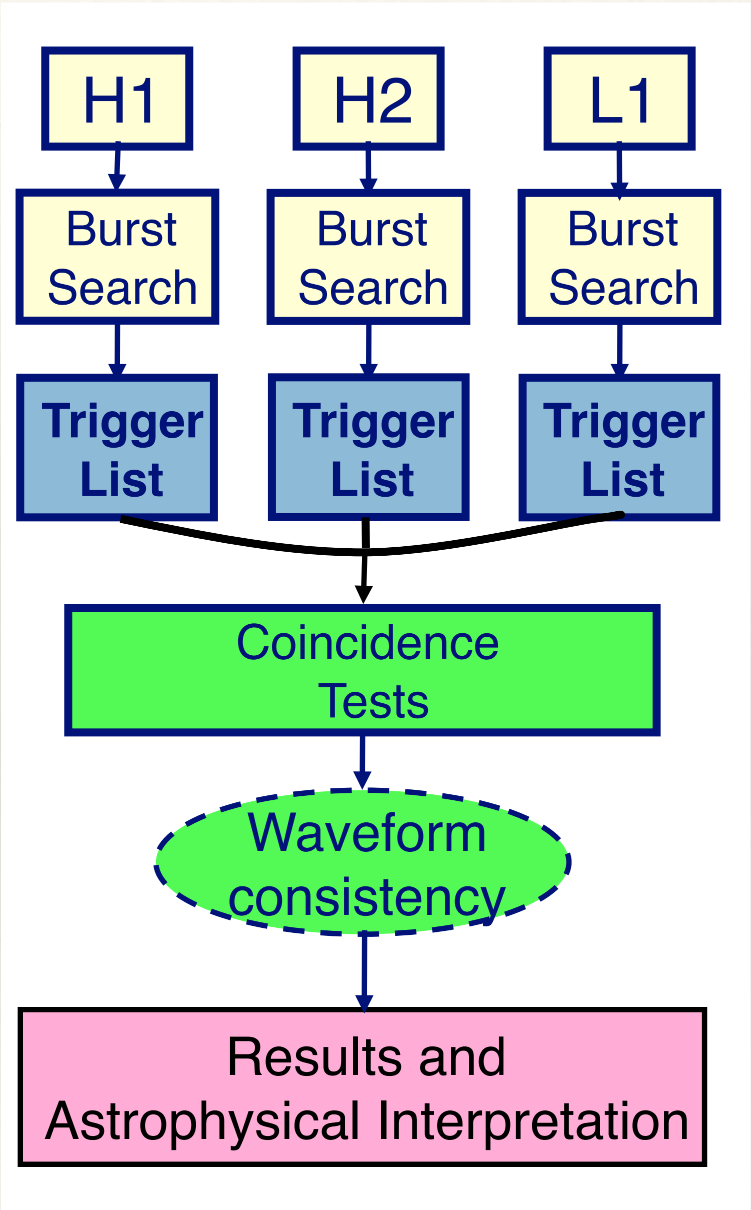


LVC, *Phys. Rev.* D **93**, 122003 (2016)

Searching for signals:
Unmodelled/Excess power searches

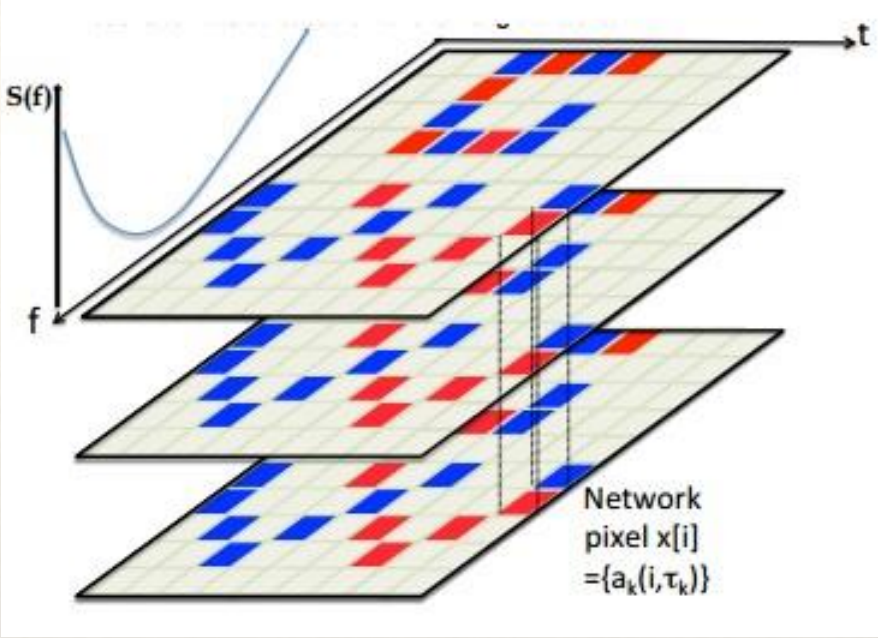
Unmodelled Searches

- Detection of gravitational wave bursts relies on two techniques
  - *Coincidence analysis*. As for stochastic background, combine data from multiple detectors. Likelihood of an instrumental artefact in two detectors simultaneously is small.
  - *Time-frequency analysis*. Look for changes in spectral properties over time, e.g., excess power in a set of connected pixels.
- Basic idea: construct *time-frequency* spectrograms of the data, i.e., estimate power at each frequency and time. Use spectrograms at multiple resolutions to give sensitivity to different burst morphologies.
- Look for clusters of pixels coincident between instruments.

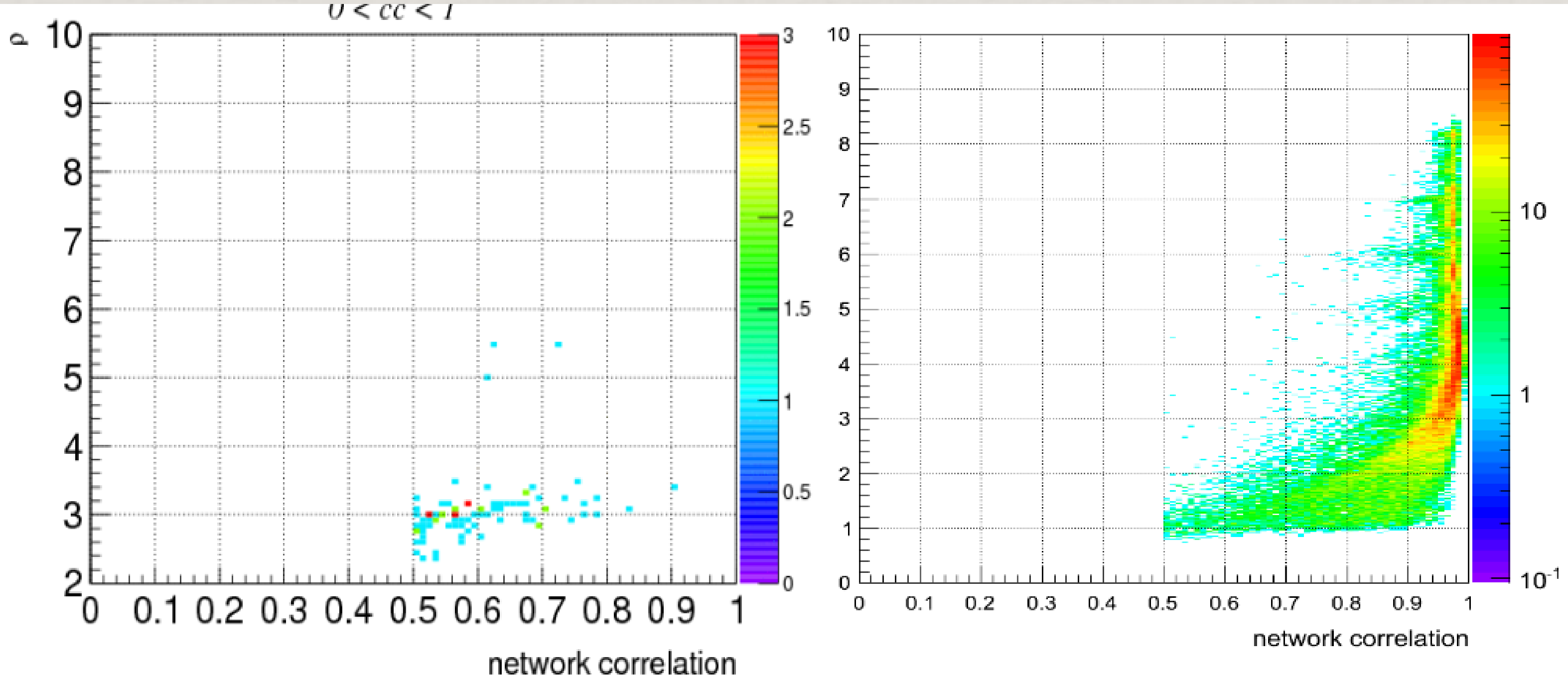


Unmodelled Searches: Coherent Wave Burst

- Combines spectrograms at multiple resolutions. Identifies pixel clusters.
- Uses various derived quantities to distinguish signals from noise artefacts, e.g., coherent and residual noise energies.

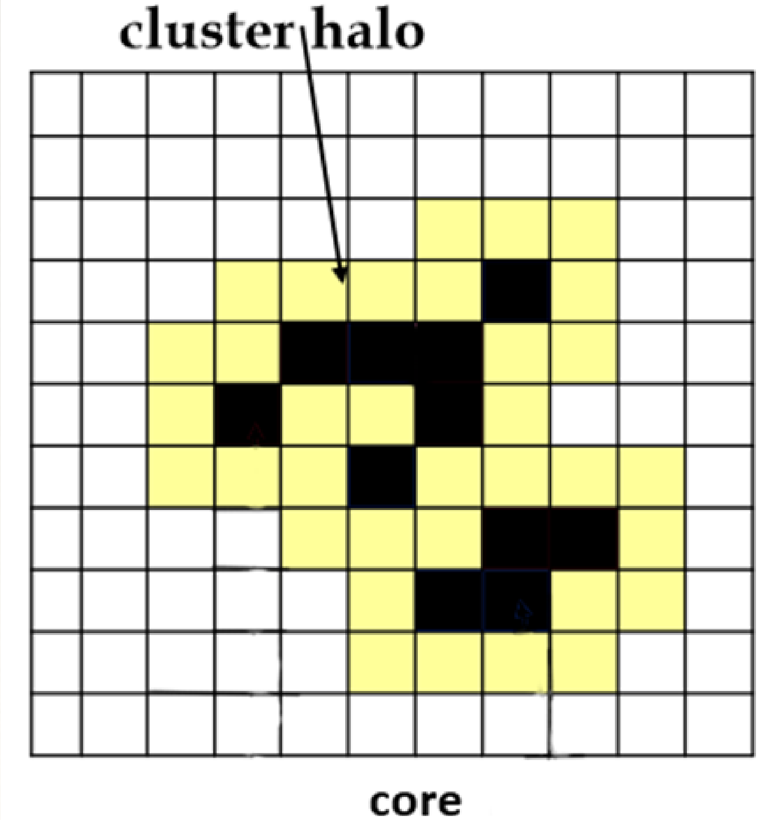


$$E_c = \sum_{m \neq n} L_{mn} \qquad N = |X - \xi_\sigma|^2 \qquad cc = \frac{E_c}{N + E_c}$$



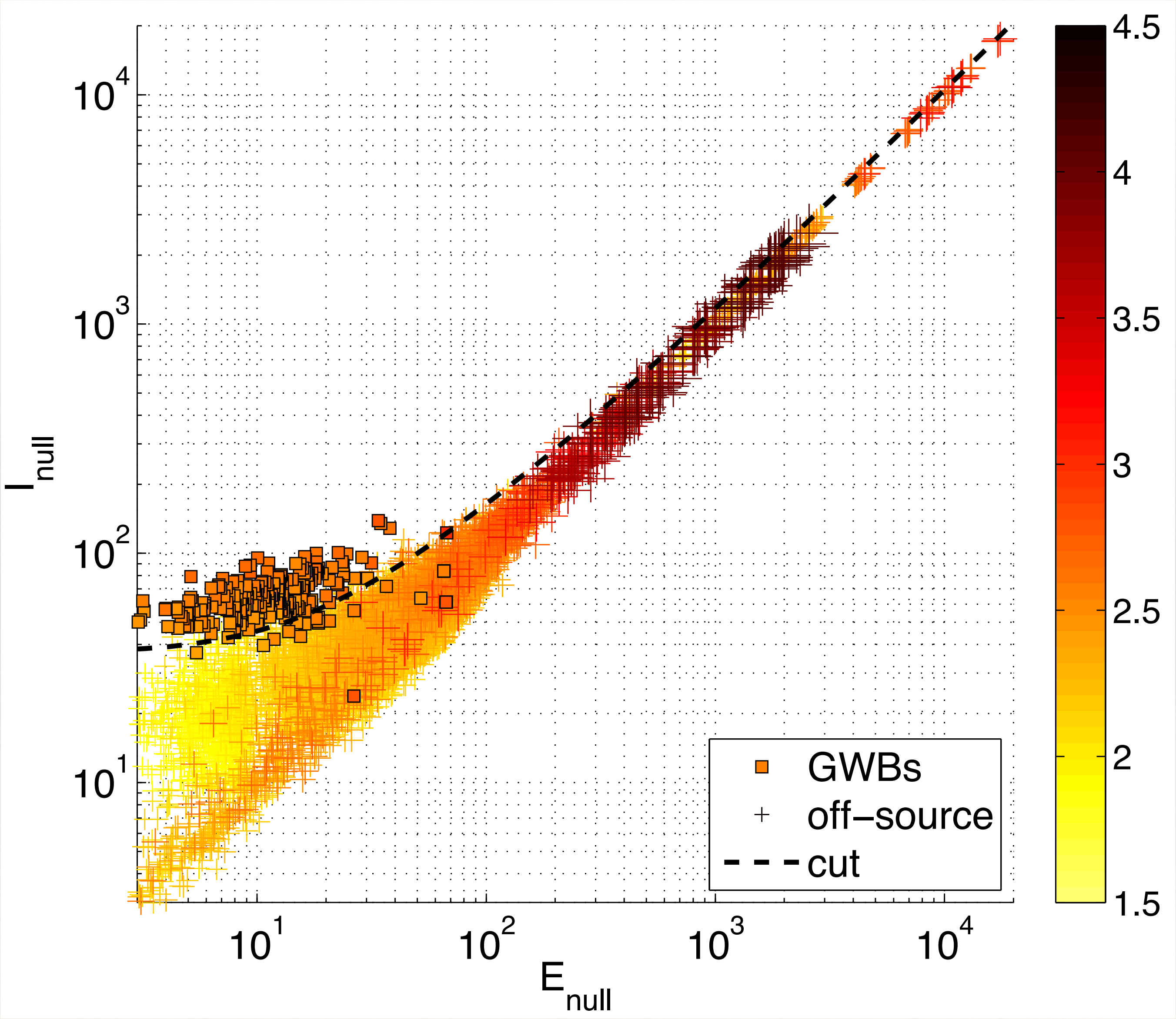
Background

Simulation

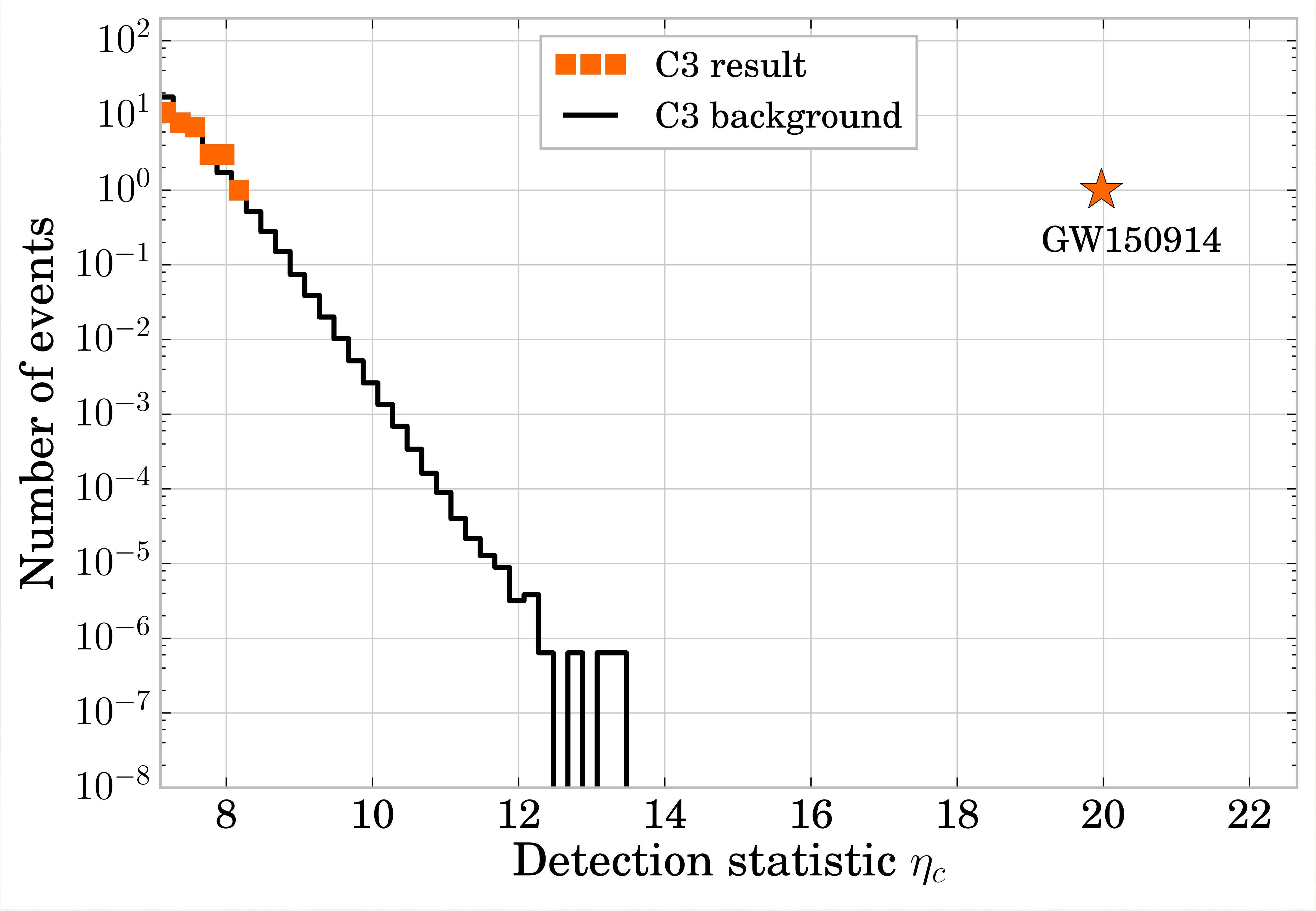


Unmodelled Searches: X-pipeline

- X-pipeline uses similar methods to CWB. Analysis is in two stages. Trigger generation, as described above, then post processing.
- Post processing involves rejecting background events based on event properties, and assessment of search efficiency.
- Rejection uses different combinations of energy measures, based on randomly selected training set of injections and time slides.



Unmodelled Searches: Coherent Wave Burst



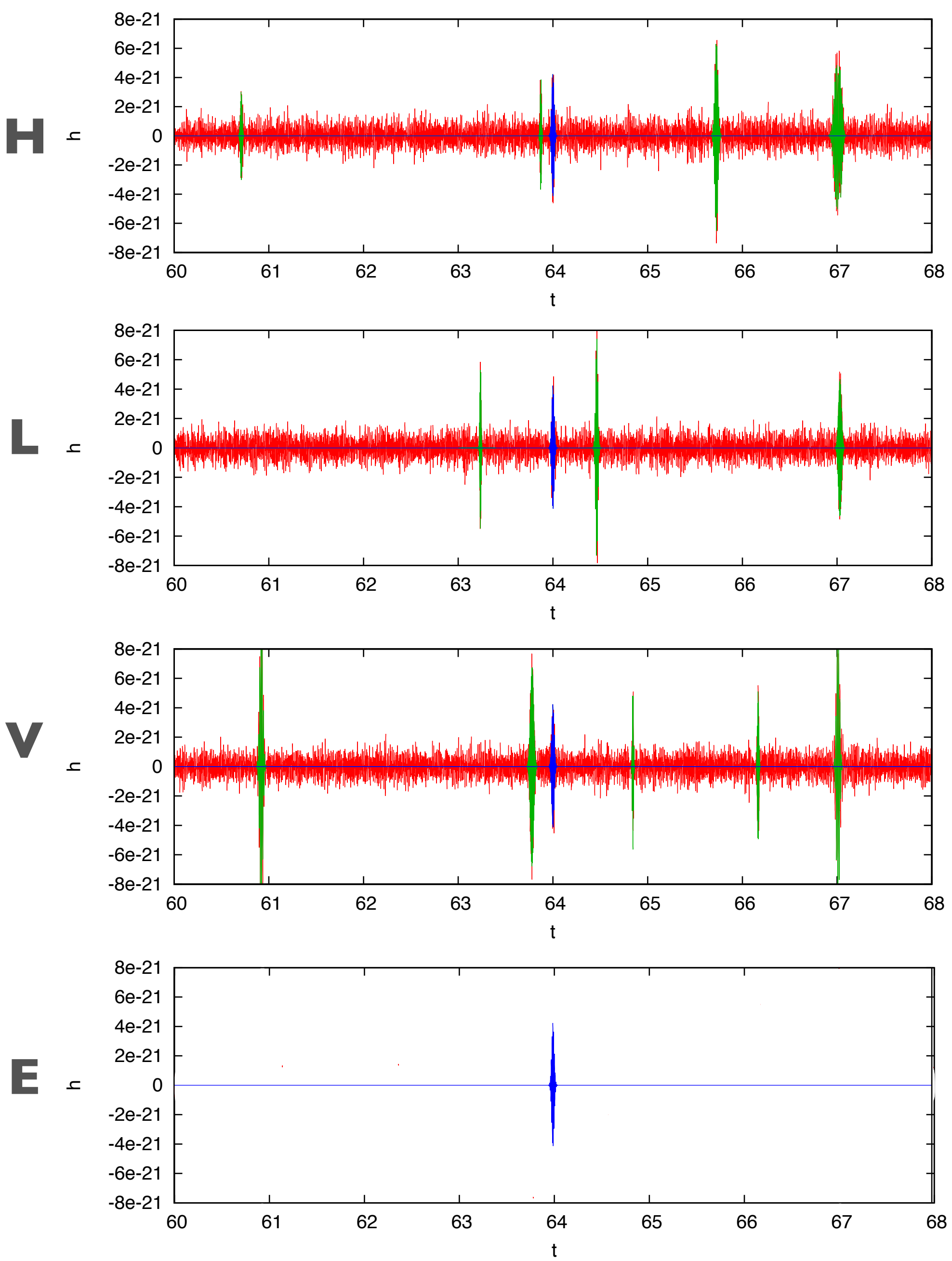
Unmodelled Searches: BayesWave

- The *BayesWave* pipeline takes a slightly different approach to modelling the noise and signal components.
- The smooth noise PSD component is modelled using a cubic spline.
- Lines in the instrumental noise are modelled using Lorentzian functions.

$$p(x; b, m) = \frac{1}{\pi} \frac{b}{(x-m)^2 + b^2}$$

- The remaining components of the data are modelled using *wavelets*, which resolve time series at particular times and frequencies. *BayesWave* uses the Morley-Gabor basis.
- There is a coherent wavelet component for sources and incoherent components to represent glitches.

Unmodelled Searches: BayesWave



Independent | Coherent

$$d^H = n^H + \sum_i^{N_G^H} \psi(\vec{\gamma}_i^H) + h^H(N_{\rm GW}^{\oplus}, \vec{\gamma}^{\oplus}, \vec{\lambda}\,)$$

$$d^L = n^L + \sum_i^{N_G^L} \psi(\vec{\gamma}_i^L) + h^L(N_{\rm GW}^{\oplus}, \vec{\gamma}^{\oplus}, \vec{\lambda}\,)$$

$$d^V = n^V + \sum_i^{N_G^V} \psi(\vec{\gamma}_i^V) + h^V(N_{\rm GW}^{\oplus}, \vec{\gamma}^{\oplus}, \vec{\lambda}\,)$$

$$h^{\rm IFO} = \mathcal{R}^{IFO}(\vec{\lambda}) * \sum_i^{N_{\rm GW}^{\oplus}} \psi(\vec{\gamma}_i^{\oplus})$$

Extrinsic

$$\{\vec{\Omega}, \delta\phi_{\oplus}, \delta t_{\oplus}, \delta\mathcal{A}, \psi, e\}$$

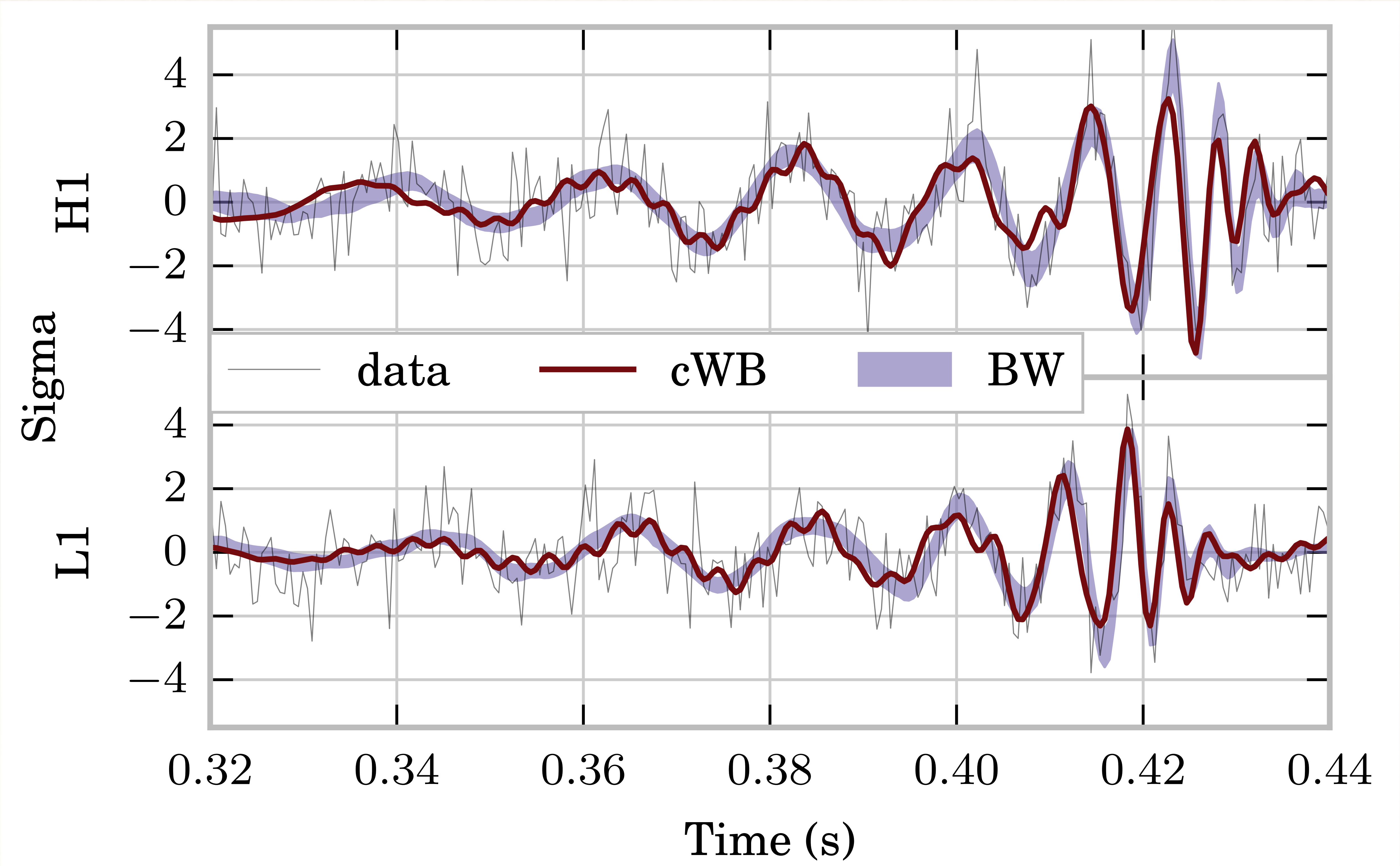
sky location, orientation, etc.

Intrinsic

$$\{t_0, f_0, \mathcal{A}, \text{etc.}\}$$

morphological params.

Unmodelled Searches: BayesWave



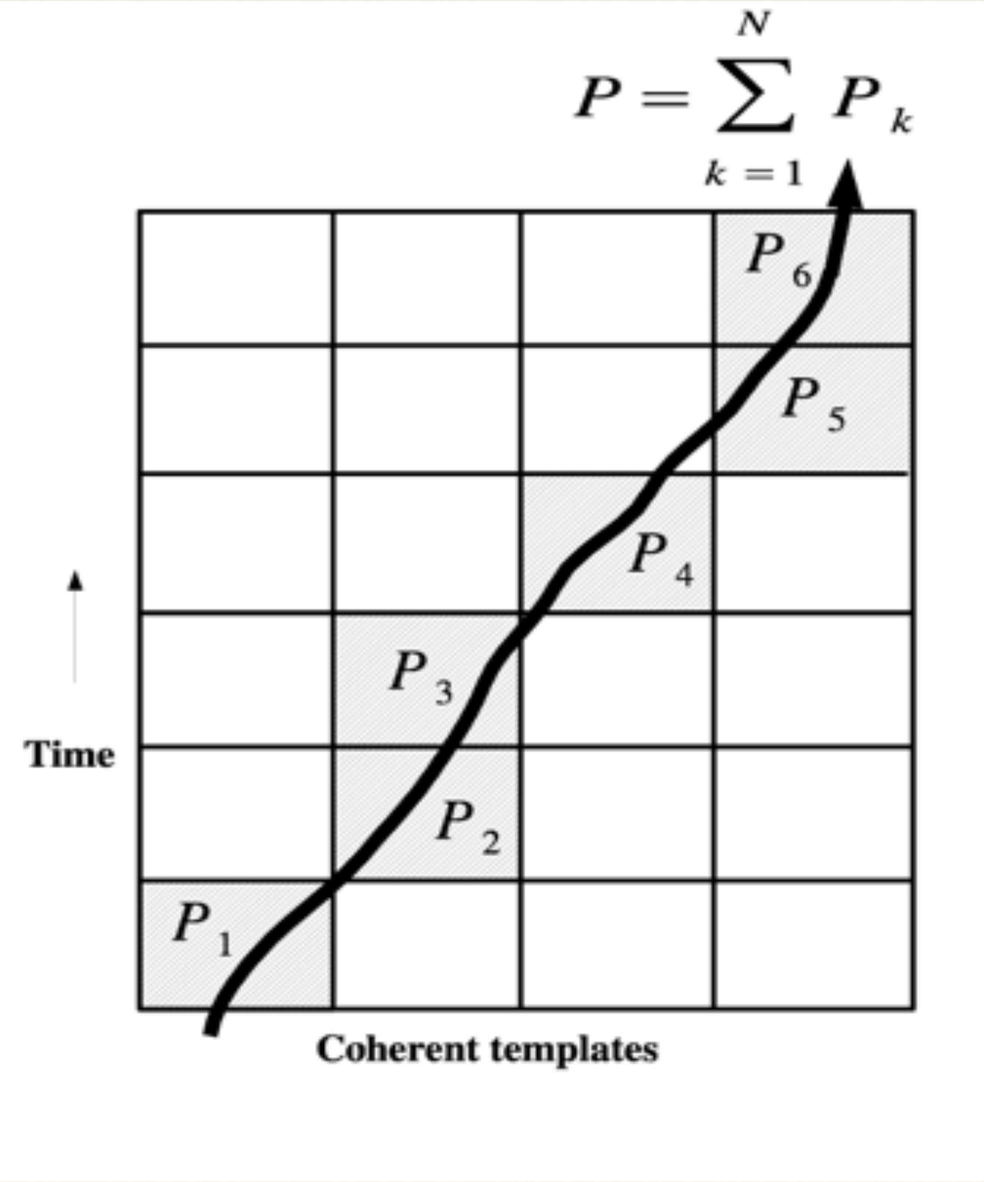
Searching for signals: Semi-coherent methods

Semi-coherent searches

- First stage is coherent matched filtering of shorter (~few week) waveform segments. Segment length set by computational limits.
- Second stage involves incoherent summation of maximized power along trajectories through the segments.

$$\rho^2 = \sum_{\alpha=I}^{II} \sum_{i=1}^{5} \langle h_i(\lambda_I), s_\alpha \rangle^2$$

where $$\langle a, b \rangle = 4\,\Re\left[\int_0^\infty \frac{\tilde{a}^*(f)\,\tilde{b}(f)}{S_b(f)}\,\mathrm{d}f\right]$$



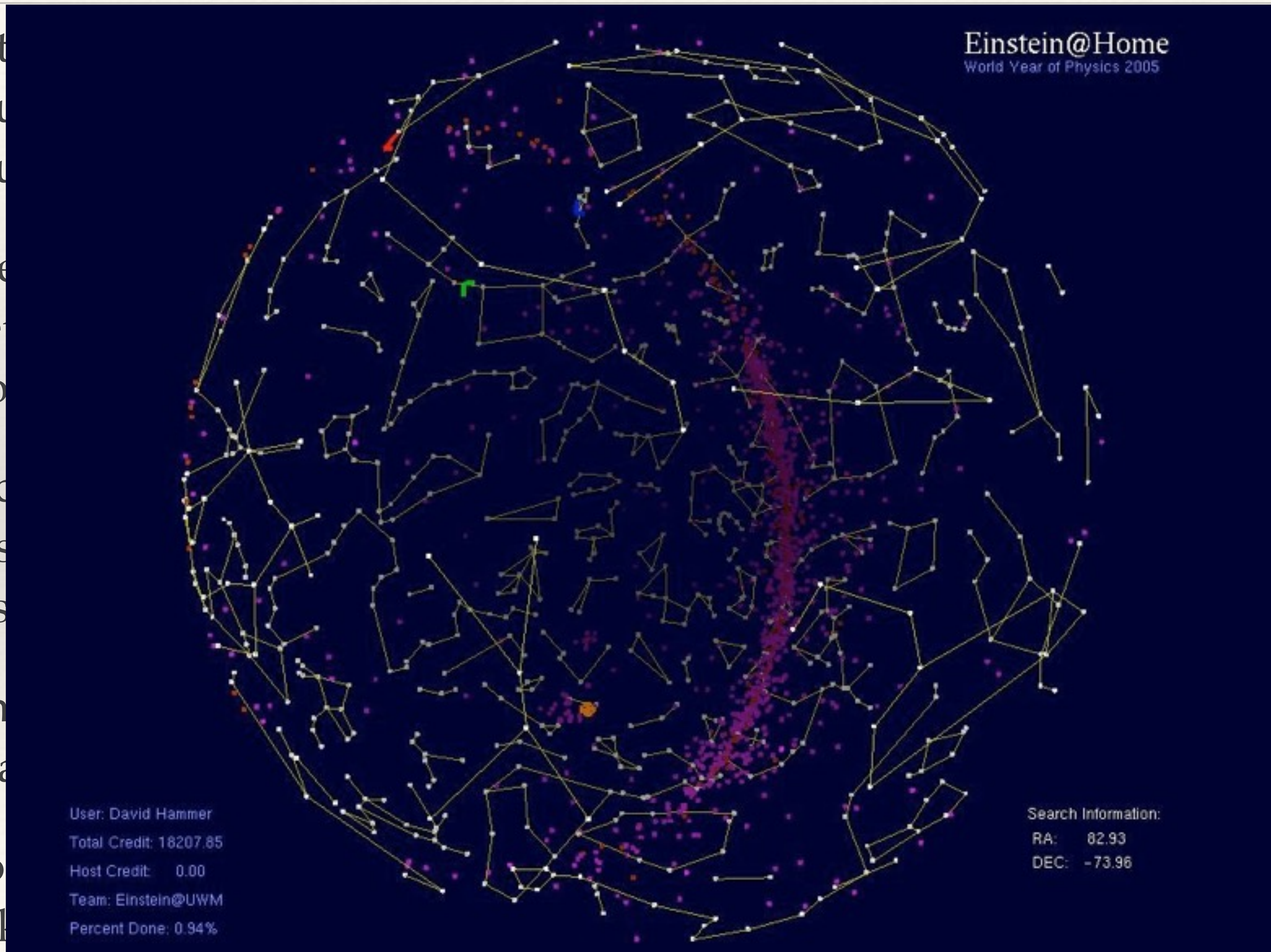
Semi-coherent searches

- First stage is coherent matched filtering of shorter (~few week) waveform segments. Segment length set by computational limits.
- Second stage involves incoherent summation of maximized power along trajectories through the segments.
- Performance analysed theoretically to derive estimated EMRI event rates. Computational cost has prevented practical implementation.

Semi-coherent searches: pulsars

- LIGO unknown pulsar search uses semi-coherent techniques.
- *Stack-Slide* algorithm as described above.
- Also *Hough Transform* method. This applies the Hough Transform, a well-established technique for detecting simple shapes (edges) in an image, to the output of the coherent stage of the search.
- Requires a huge amount of computer power - Einstein@home.

- In the spirit of *Seti@home*, *Einstein@Home* is an attempt to use idle cpu hours to analyse LIGO data and assist with the unknown pulsar search. You can sign up at http://einstein.phys.uwm.edu/ !
- The program is built on BOINC (Berkeley Open Infrastructure for Network Computing) and was released in 2005 to coincide with the World Year of Physics.
- Each computer analyses a different segment of data for a particular sky position. Each data segment is farmed out to at least two nodes to ensure accuracy.
- Einstein@Home currently has approximately 500,000 active users and a total of 5GFLOPs computing power.
- No gravitational waves discovered from pulsars, but has identified unknown pulsars in other data sets.



- In t … pu
  hou … arch.
  You
- The
  Ne … e
  Wo
- Eac … r sky
  pos
  ens
- Ein … nd a
  tota
- No … l
  unl

Searching for signals: Backgrounds

Stochastic Gravitational Wave Fore/Backgrounds

- Stochastic backgrounds are potentially present in all frequency bands, and could therefore be seen by any of our gravitational wave detectors.
- The Polarisation of the Cosmic Microwave Background is a direct probe of cosmological gravitational waves.
- In interferometers, search for an isotropic background using cross-correlation between multiple detectors to identify common noise.

$$
\begin{aligned}
Y_Q &= \int_0^T \mathrm{d}t_1 \int_0^T \mathrm{d}t_2 \, h_1(t_1) Q(t_1 - t_2) h_2(t_2) \\
&= \int_{-\infty}^{\infty} \mathrm{d}f \int_{-\infty}^{\infty} \mathrm{d}f' \delta_T(f - f') \tilde{h}_1^*(f) Q(f') \tilde{h}_2(f')
\end{aligned}
$$

Stochastic Gravitational Wave Fore/Backgrounds

- In the preceding equation, $\delta_T(f)$ denotes a finite time approximation to the Dirac delta function

$$\delta_T(f) \equiv \int_{-T/2}^{T/2} \mathrm{e}^{-2\pi i f t} \mathrm{d}t = \sin(\pi f T)/\pi f$$

- and $Q(t)$ denotes the cross-correlation filter. If the noise in the detectors is uncorrelated, the expectation value of depends only on the cross-correlated stochastic signal

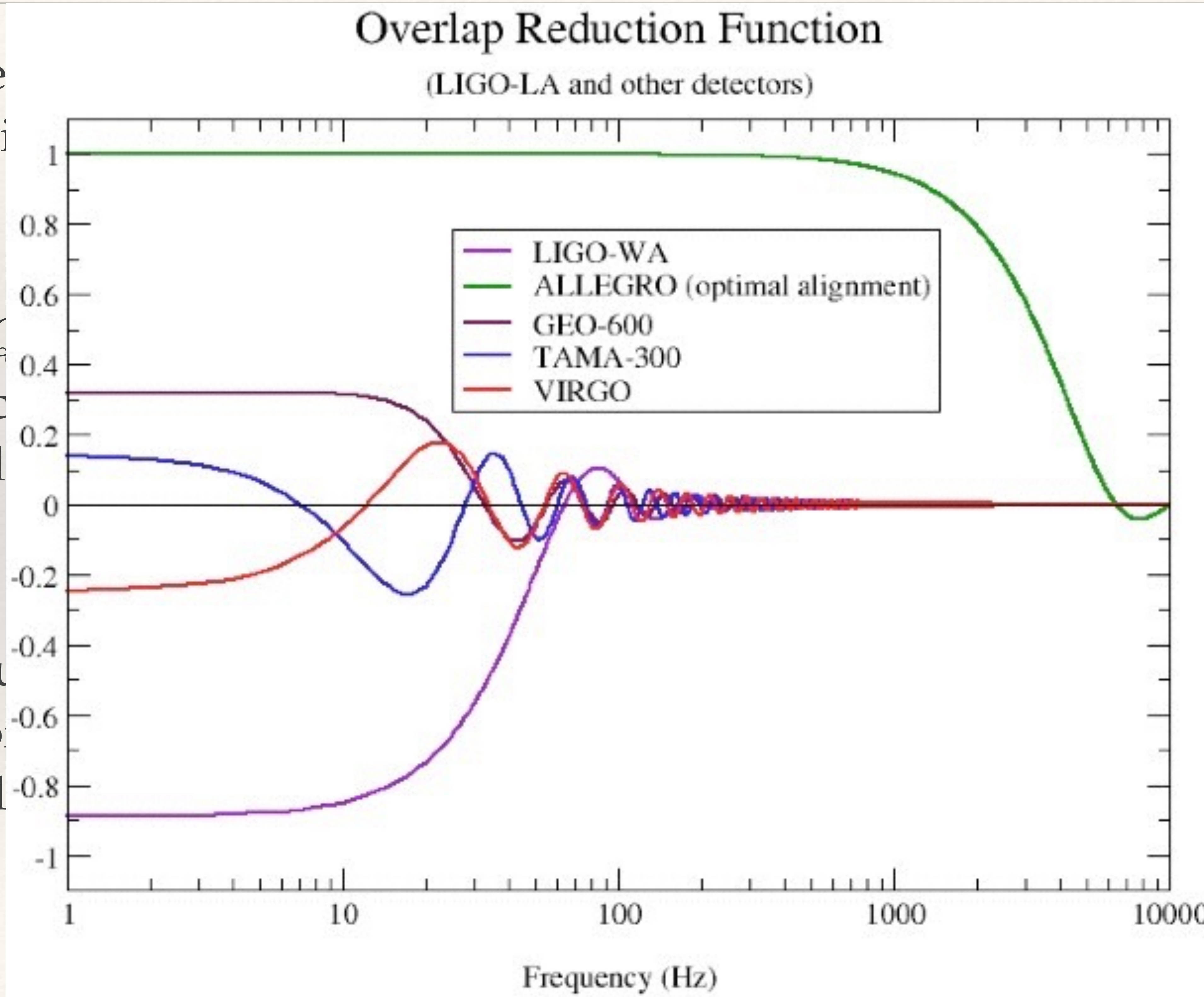
$$\langle Y_Q \rangle = \mu = \frac{T}{2} \int_{-\infty}^{\infty} \gamma(|f|) S_{\mathrm{gw}}(|f|) \tilde{Q}(f) \, \mathrm{d}f$$

- The function $\gamma(|f|)$ is the *overlap reduction function*, which measures the loss of sensitivity due to the separation and relative orientation of the two detectors. The SNR is maximized by using the optimal filter

$$\tilde{Q}(f) \propto \frac{\gamma(|f|) S_{\mathrm{gw}}(|f|)}{S_1(|f|) S_2(|f|)} \propto \frac{\gamma(|f|) \Omega \mathrm{gw}(|f|)}{|f|^3 S_1(|f|) S_2(|f|)}$$

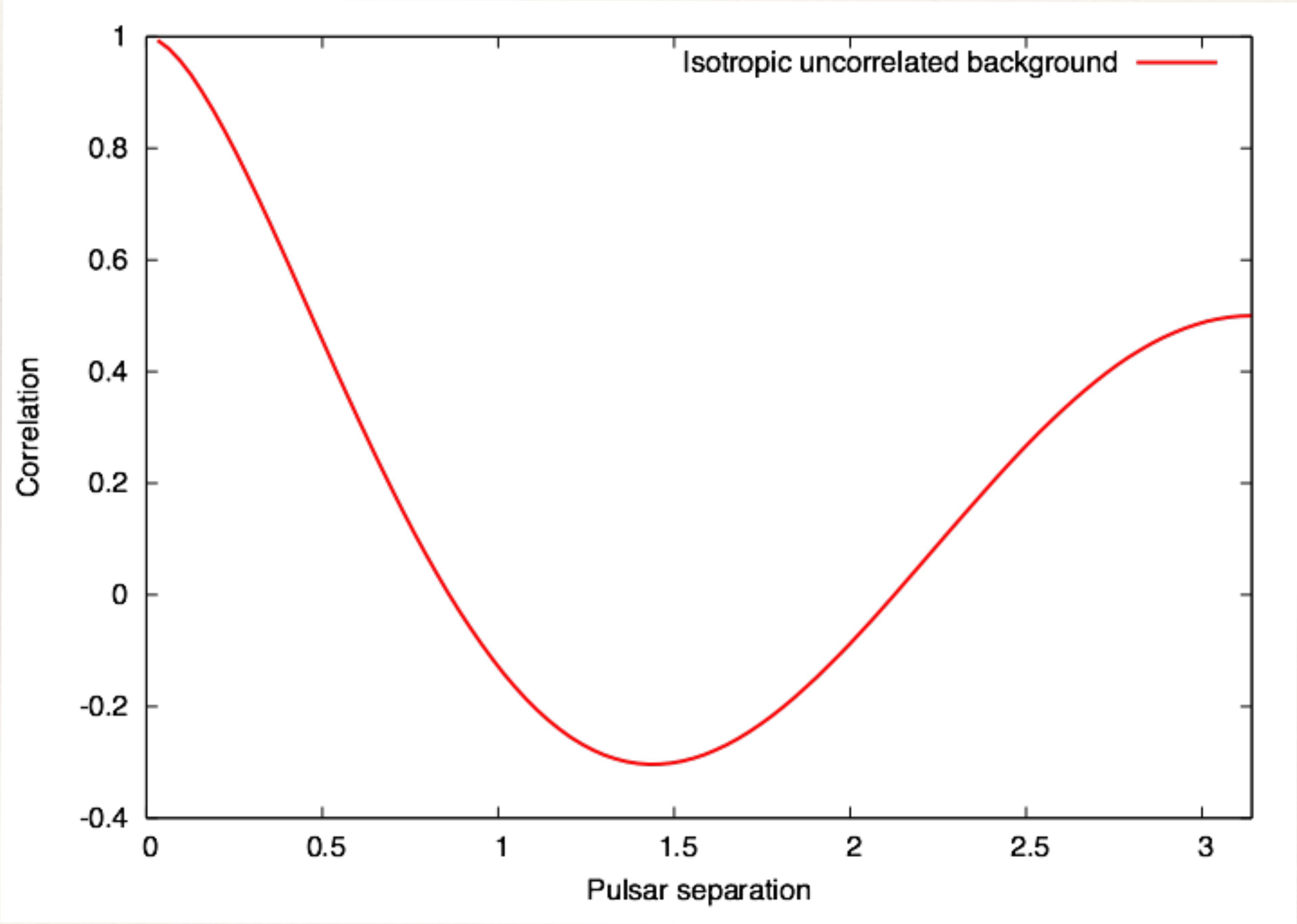
Stochastic Gravitational Wave Fore/Backgrounds

- In the ... ation to
  the Di...
- and Q ... detectors
  is unc ... ross-
  correl...
- The fu ... ures the
  loss o ... of the
  two d...



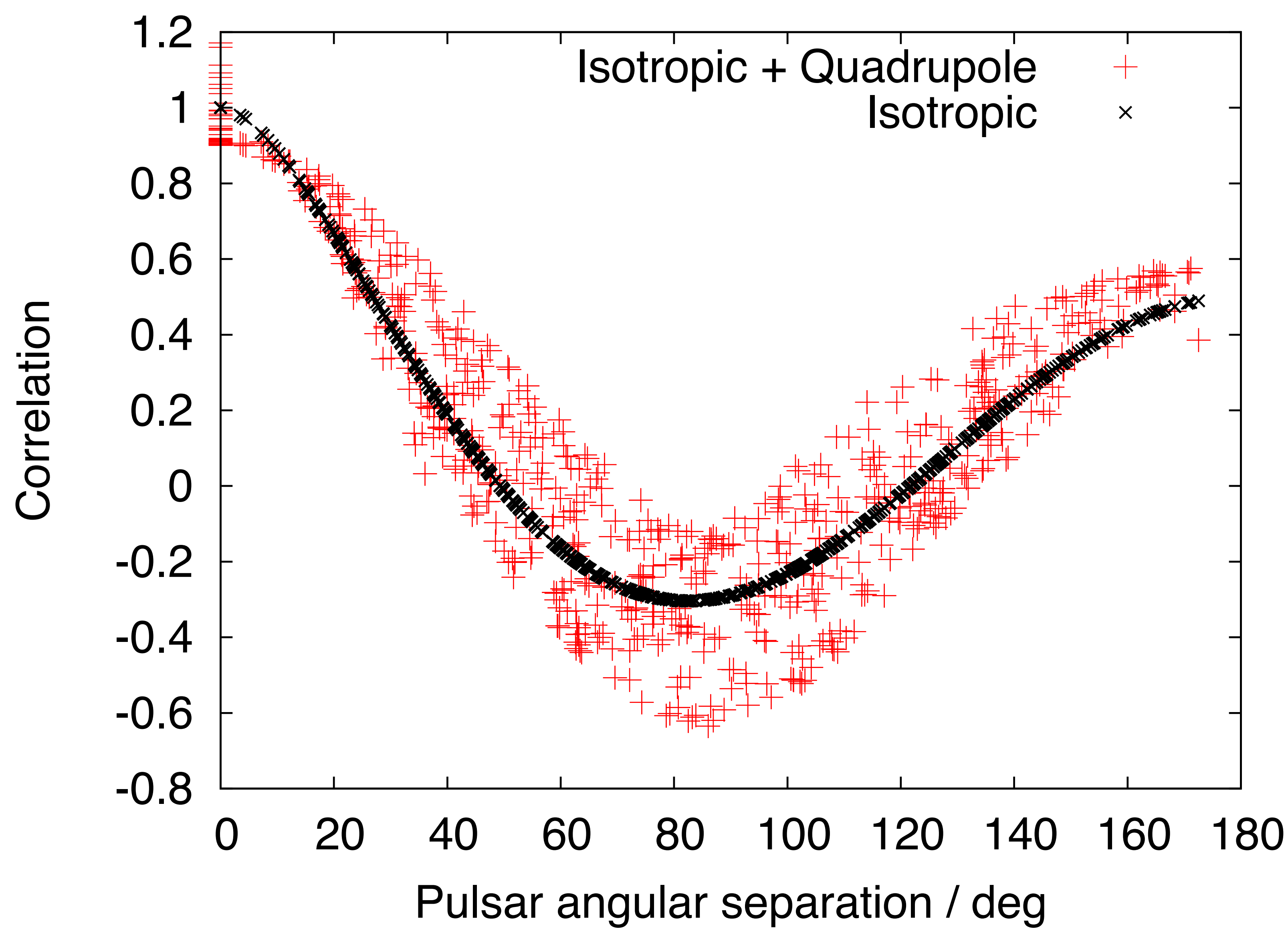
Stochastic Gravitational Wave Fore/Backgrounds

- For pulsar timing, the overlap reduction function for an isotropic background is the Hellings and Downs curve.



Stochastic Gravitational Wave Fore/Backgrounds

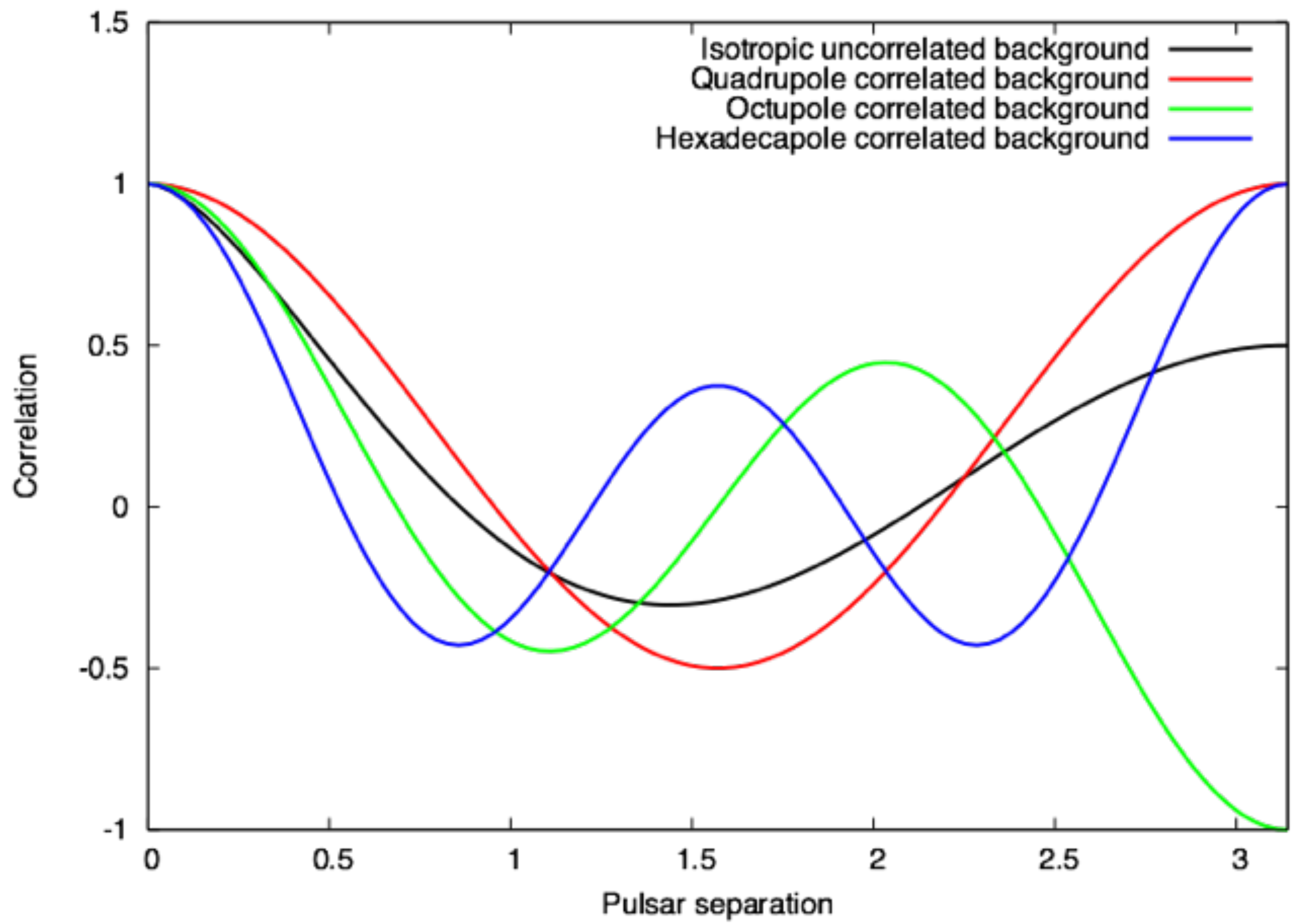
- Uncorrelated anisotropic and correlated backgrounds have different correlation functions.



Correlation function for uncorrelated quadrupole background

Stochastic Gravitational Wave Fore/Backgrounds

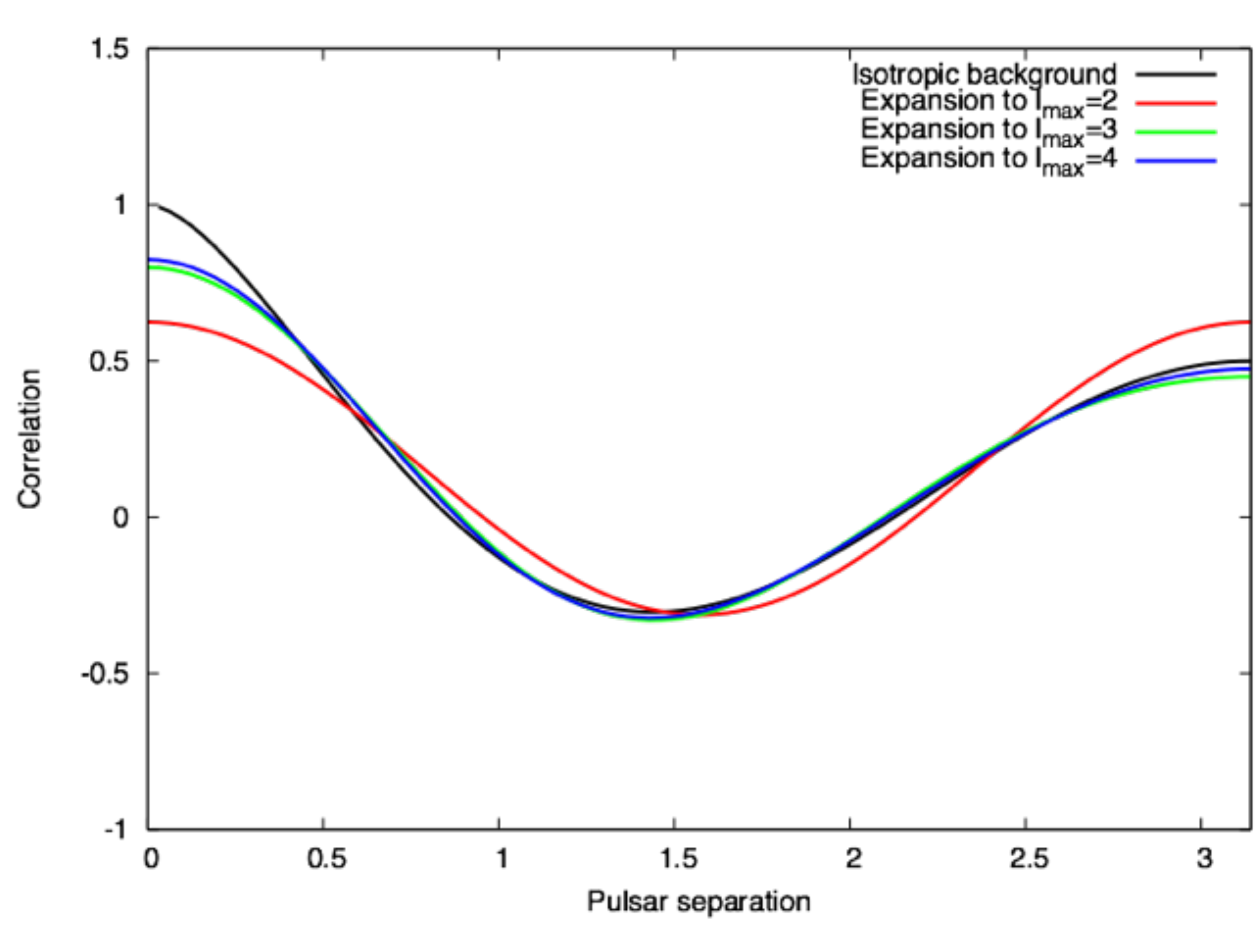
- Uncorrelated anisotropic and correlated backgrounds have different correlation functions.



Correlation function for correlated backgrounds

Stochastic Gravitational Wave Fore/Backgrounds

- Uncorrelated anisotropic and correlated backgrounds have different correlation functions.



Decomposition of isotropic uncorrelated background as a superposition of correlated backgrounds

Parameter Estimation

Bayes Theorem

- Recall definition of conditional probability:

$$p(A|B) = \frac{p(A \cap B)}{p(B)}$$

- Rearranging, we obtain Bayes' Theorem:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

- This is mathematically exact, but can be used in an approximate way for inference
  - p(A) — prior belief about state of the Universe, "A";
  - p(B | A) — likelihood of seeing data "B" if the state is "A";
  - p(A | B) — posterior belief on the state of the Universe after collecting data;
  - p(B) — "evidence" for your model (a normalising constant).

Sampling posterior distributions

- Typically, "A" will be a statement about the parameters of some model, M; "B" will be the observed data. The statement of Bayes theorem then becomes

$$p(\vec{\theta}|d, M) = \frac{p(d|\vec{\theta}, M)p(\vec{\theta}|M)}{p(d|M)}$$

- We want to compute the posterior distribution, $p(\vec{\theta}|d, M)$, for the model parameters based on the observed data.
- **Simplest approach**: evaluate the posterior on a grid in parameter space.
- **But:** not very efficient in high-dimensional parameter spaces.

Sampling posterior distributions

- **Alternative**: stochastic approach. Generate a sequence of samples,

$$\{\vec{\theta}_1, \ldots, \vec{\theta}_N | \vec{\theta}_i \sim p(\vec{\theta}|d, M)\}$$

- Integrals over the posterior distribution can then be evaluated using a sum over the samples

$$\int f(\vec{\theta})\, p(\vec{\theta}|d, M)\, \mathrm{d}^n\theta \approx \frac{1}{N} \sum_{i=1}^{N} f(\vec{\theta}_i)$$

Markov Chain Monte Carlo

- Such a sequence of samples can be constructed by generating a reversible Markov chain with stationary distribution equal to the target distribution.
- Such a Markov chain must satisfy *detailed balance*

$$p(\vec{\theta})\, p(\vec{\theta}, \vec{\theta}') = p(\vec{\theta}')\, p(\vec{\theta}', \vec{\theta})$$

- In which

$$p(\vec{\theta}, \vec{\theta}') = p(\vec{\theta}_i = \vec{\theta}' | \vec{\theta}_{i-1} = \vec{\theta})$$

- and $p(\vec{\theta})$ denotes the target distribution, in our case $p(\vec{\theta}|d, M)$.

Metropolis Hastings Algorithm

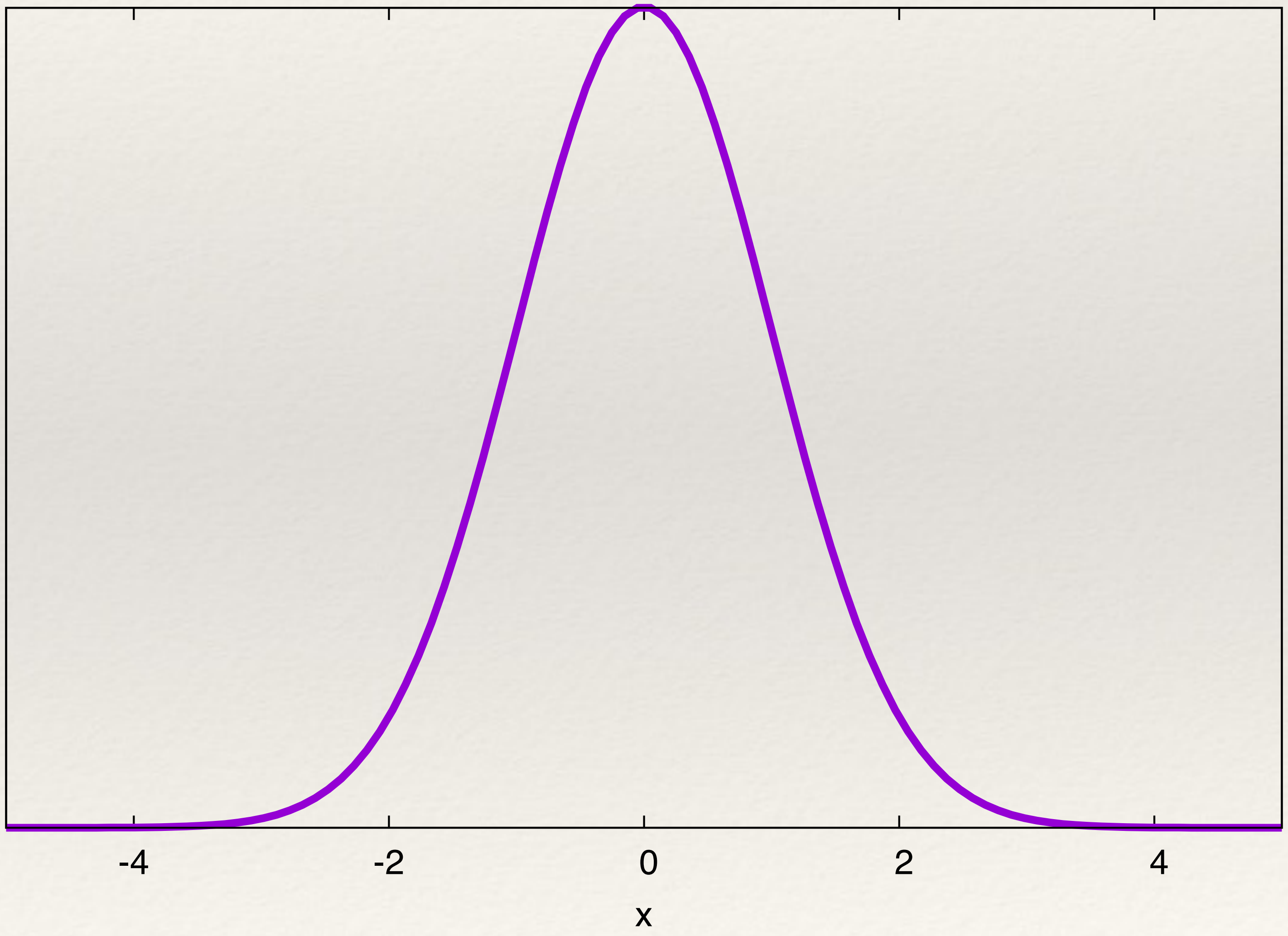
- The Metropolis-Hastings algorithm provides one way to compute a Markov chain with these properties.
- We initialise by choosing a (random) starting point. Then, at step i:
  - propose a new point, $\vec{\theta}'$, by drawing from a *proposal distribution*, $q(\vec{\theta}', \vec{\theta}_i)$.
  - evaluate the target distribution at the new point. Compute the *Metropolis-Hastings ratio*

$$\mathcal{H} = \frac{p(\vec{\theta}')q(\vec{\theta}_i, \vec{\theta}')}{p(\vec{\theta}_i)q(\vec{\theta}', \vec{\theta}_i)}$$

  - and draw a random sample, α, from a U[0,1] distribution. If $\alpha < \mathcal{H}$ set $\vec{\theta}_{i+1} = \vec{\theta}'$, otherwise set $\vec{\theta}_{i+1} = \vec{\theta}_i$. NB if $\mathcal{H} > 1$ the proposed move is definitely accepted.

Proposal Distributions

- Sampling efficiency strongly influenced by choice of proposal distribution.
- Uniform proposal (random sampling) very inefficient - better to use a grid.
- Ideally want a proposal tuned to the distribution you are sampling.
- Gaussian a good choice, but need to tune width.

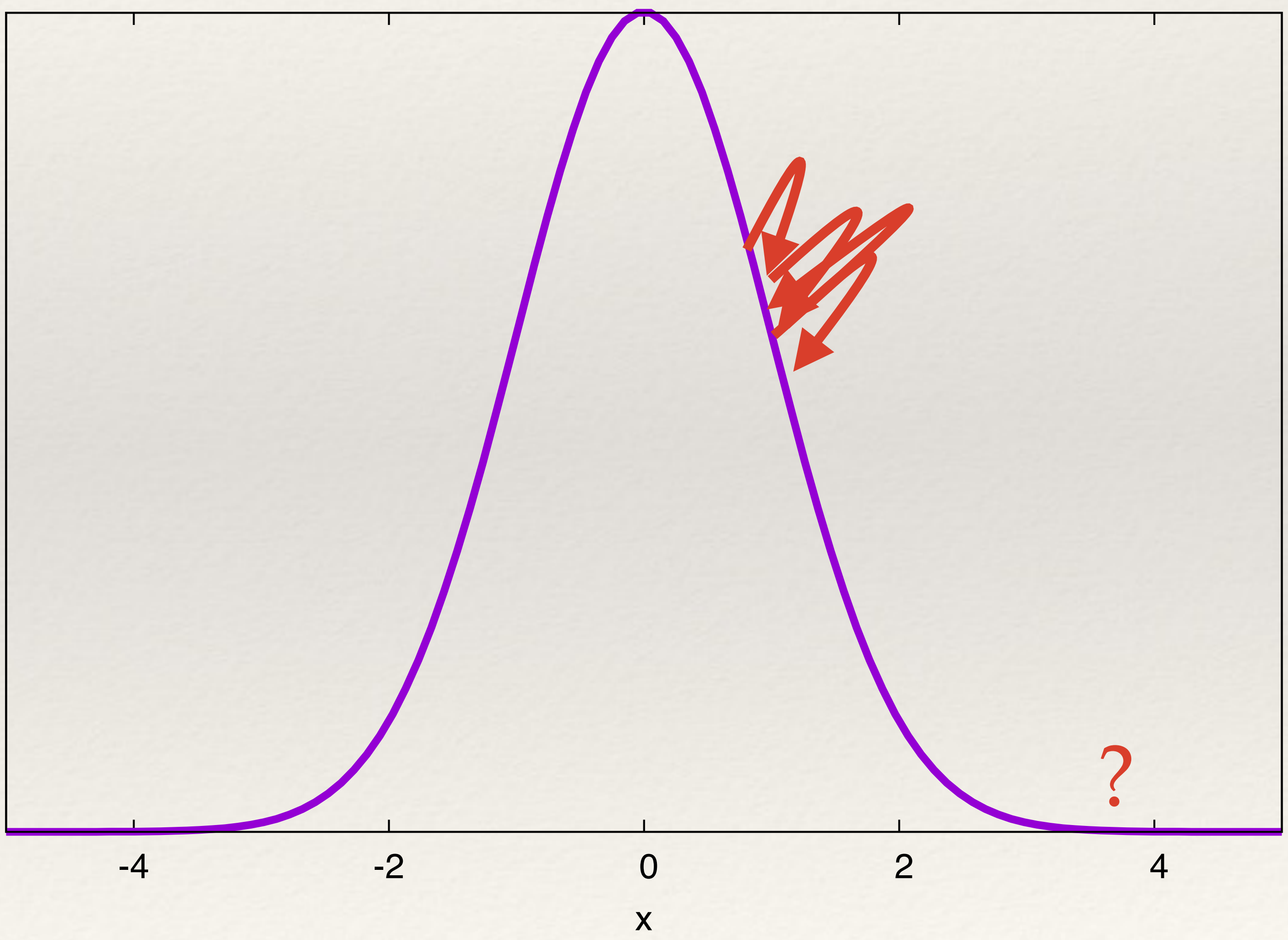


Proposal Distributions

- Sampling efficiency strongly influenced by choice of proposal distribution.
- Uniform proposal (random sampling) very inefficient - better to use a grid.
- Ideally want a proposal tuned to the distribution you are sampling.
- Gaussian a good choice, but need to tune width.
    - **too wide:** low acceptance rate;

Proposal Distributions

- Sampling efficiency strongly influenced by choice of proposal distribution.
- Uniform proposal (random sampling) very inefficient - better to use a grid.
- Ideally want a proposal tuned to the distribution you are sampling.
- Gaussian a good choice, but need to tune width.
  - **too wide:** low acceptance rate;
  - **too narrow:** high acceptance rate; low effective samples.



Annealing

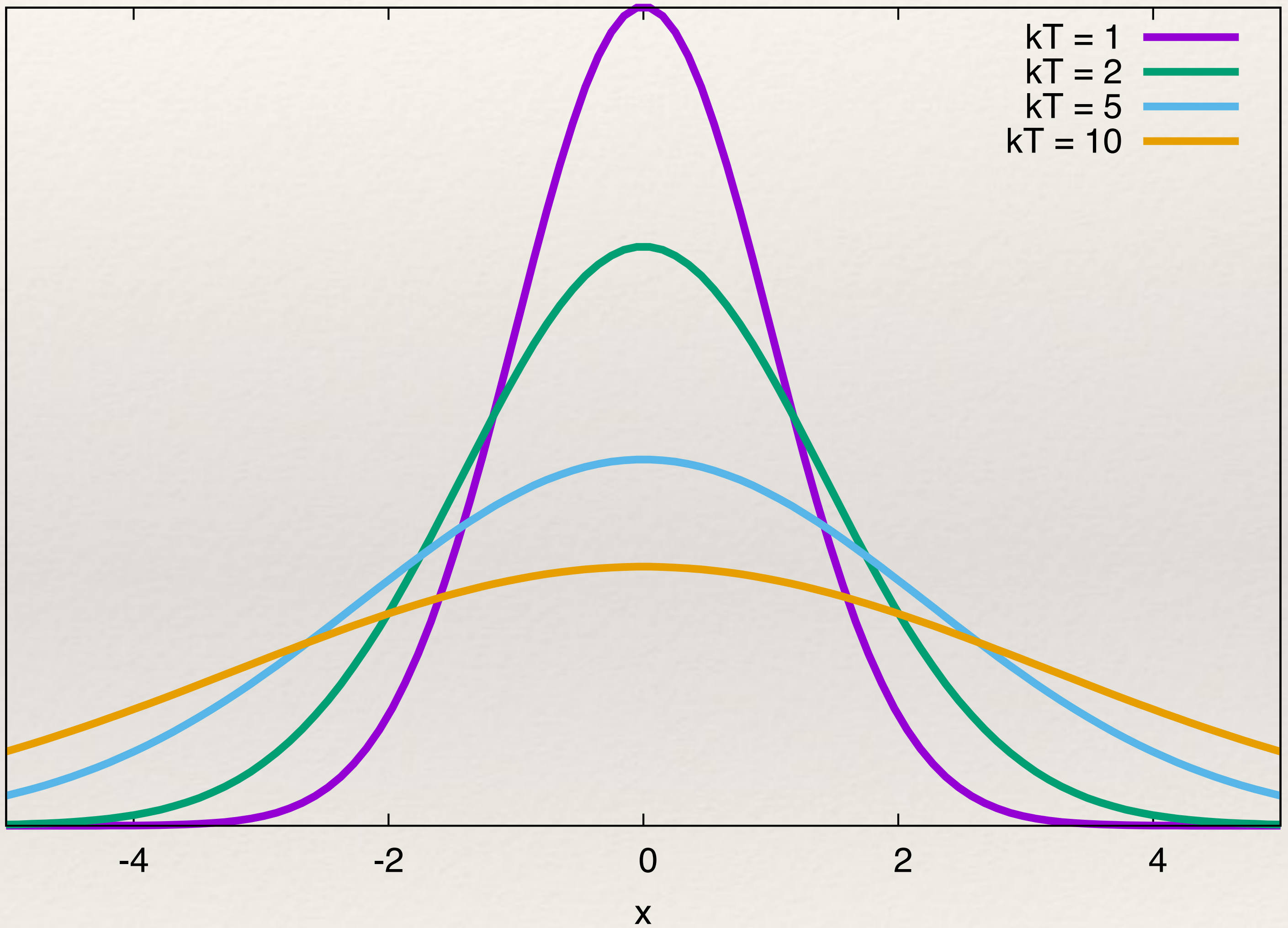
- One way to accelerate convergence is to use *simulated annealing*.
- "Heat up" posterior by making the replacement

$$p(\vec{\theta}|d, M) \to \left[p(\vec{\theta}|d, M)\right]^{\beta}$$

- where

$$\beta = \frac{1}{kT}$$

- Choosing a high temperature smoothes out the posterior which can then be more easily sampled.
- Allows identification of interesting parts of parameter space.



Annealing

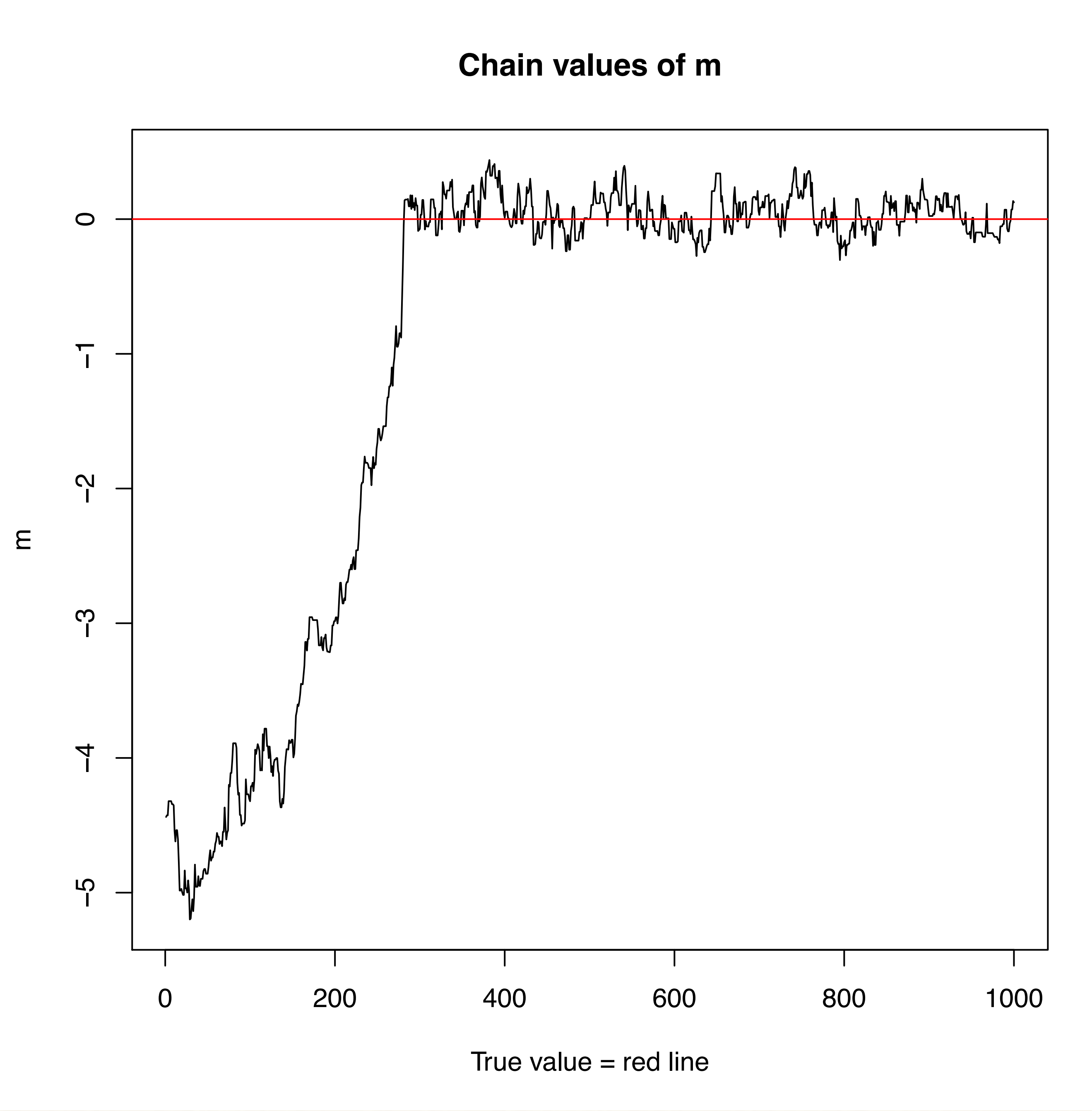
- It is common to use *parallel tempering*. A sequence of M MCMC chains are run simultaneously at different temperatures, $\{T_1, \ldots, T_M\}$.
- The chains can exchange information, which is achieved by proposing a swap of the states of two chains with different temperatures. The swap is accepted with probability

$$\min\left(1, \frac{p_i(\vec{\theta}_j)\, p_j(\vec{\theta}_i)}{p_i(\vec{\theta}_i)\, p_j(\vec{\theta}_j)}\right)$$

- where i, j label the two temperature chains, $\vec{\theta}_k$ denotes the current state of the k'th chain and $p_k(\vec{\theta})$ denotes the target (annealed) distribution for the k'th chain.

Burn-in

- The MCMC chain does not sample from the target distribution immediately.
- There is a residual "memory" of the initial state. Need to discard the first few samples.
- This is called the burn-in.
- Can identify number of samples to discard by looking at *trace plots*.
- Usually a few hundred to a thousand samples is sufficient for burn-in.



Autocorrelation and Effective sample size

- Consecutive samples in the MCMC chain are not independent samples from the target distribution.
- Can use all samples for posterior inference *but* do need to know how many *independent* samples the chain contains in order to assess the precision of inferences.
- Compute the (lag-k) autocorrelation

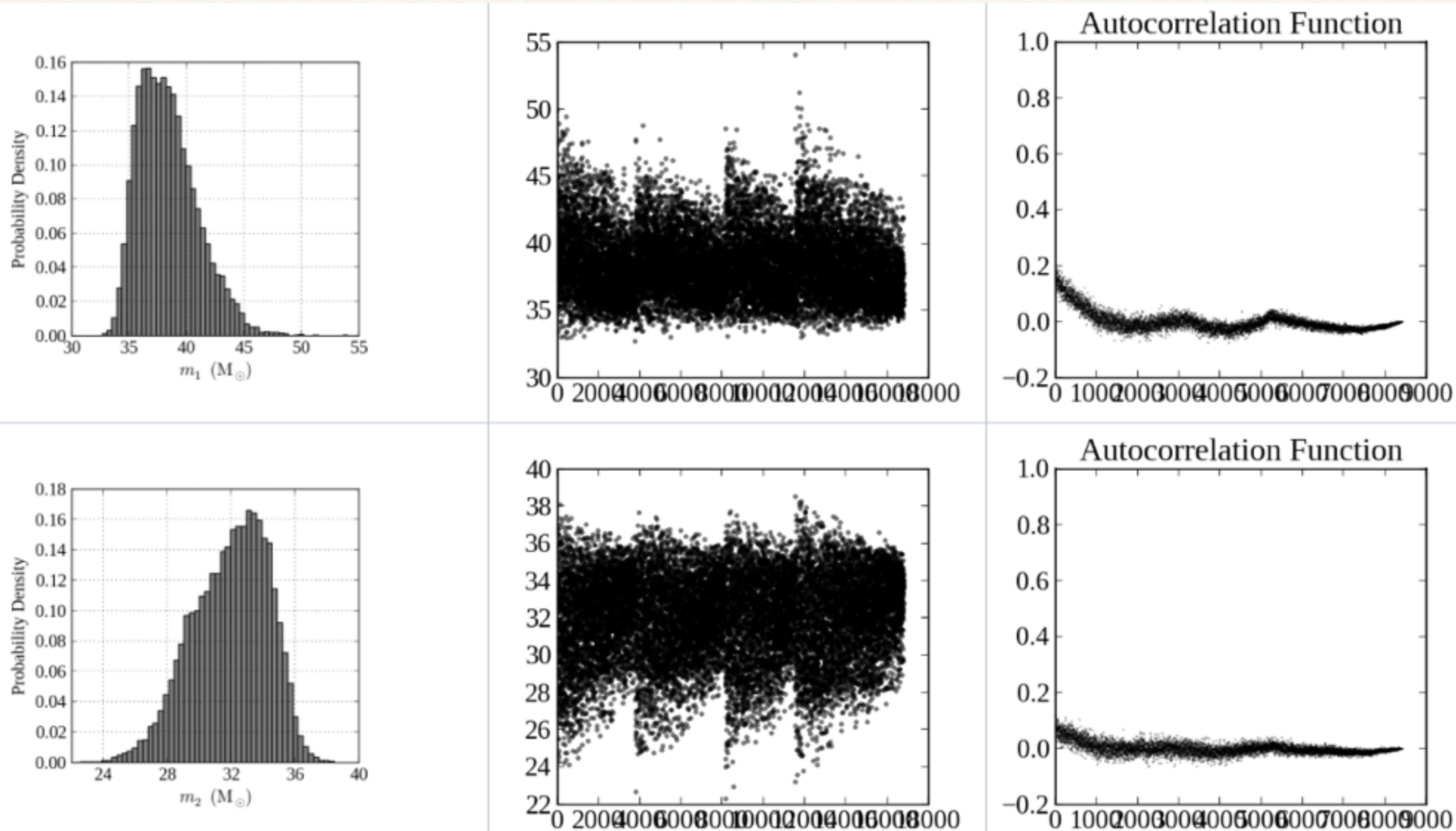
$$\rho_k = \frac{\sum_{i=1}^{N-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

- where x now denotes one of the components of $\vec{\theta}$. Choose k=K large enough that $\rho_k << 1$. Effective sample size is $\sim N/K$. Can "thin" chain by keeping only every K'th sample without affecting accuracy of posterior inference.

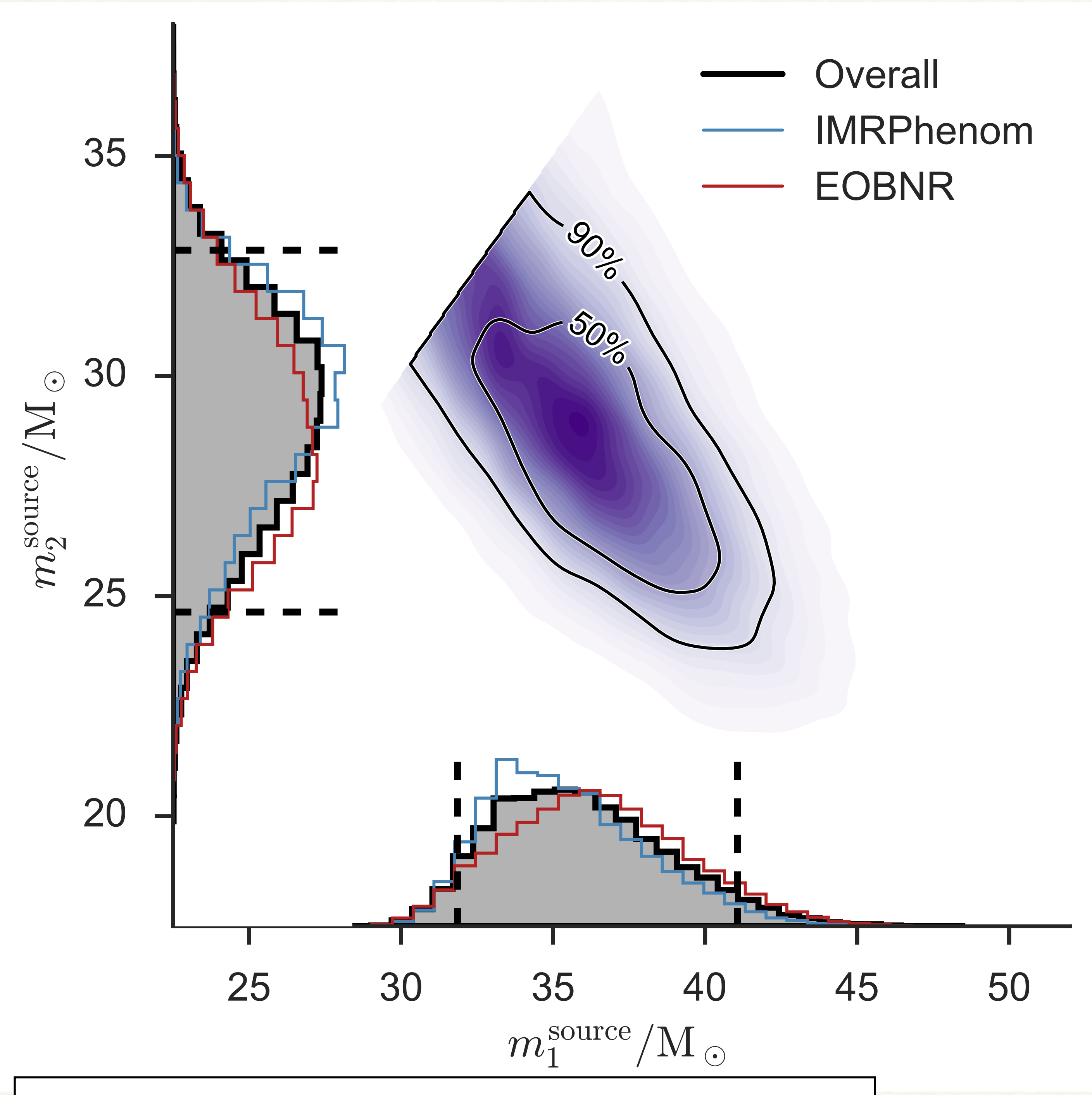
Diagnostics

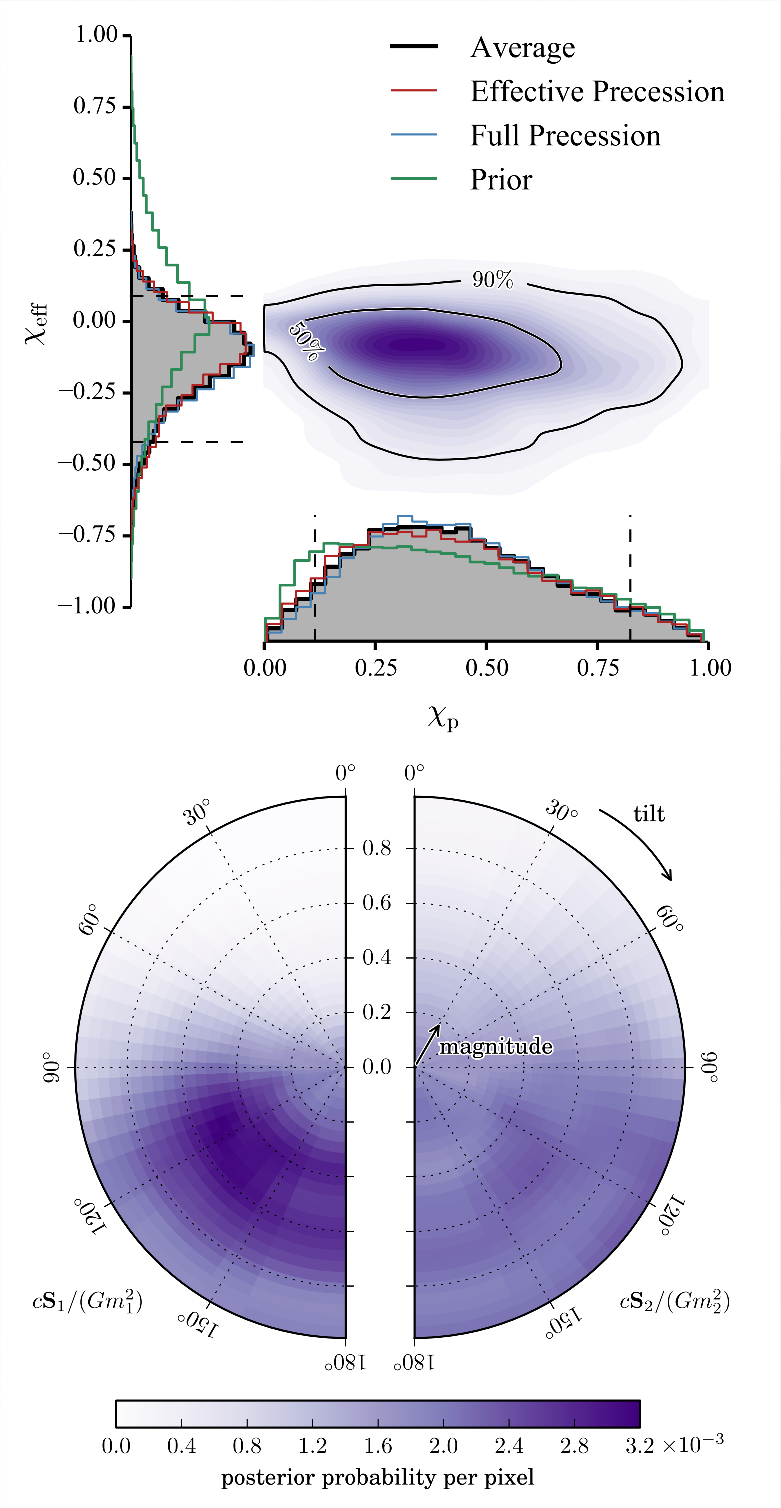
- There are various techniques to diagnose the quality of results from a given MCMC run.
  - compute acceptance rate, i.e., fraction of proposed points that are accepted. Acceptance rate ~25% is optimal.
  - look at one and two dimensional posterior distributions — do they look smooth and well sampled?
  - look at trace plots — is the chain moving back and forth or unidirectionally?
  - run multiple MCMC chains starting at different points. Do they give consistent results?
  - use Gelman-Rubin convergence diagnostic.

Convergence diagnostics: GW150914



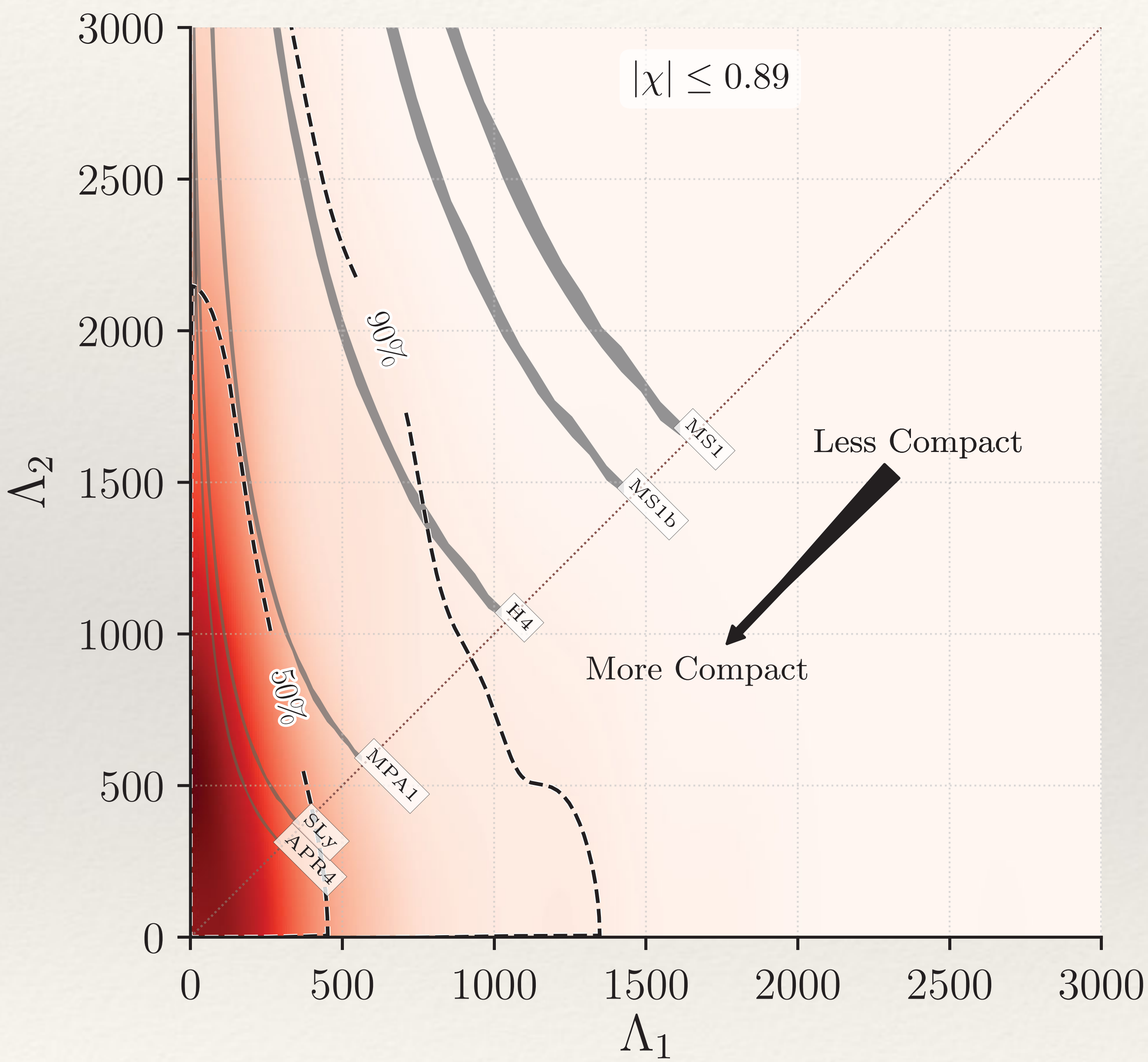
Examples of Parameter Posteriors

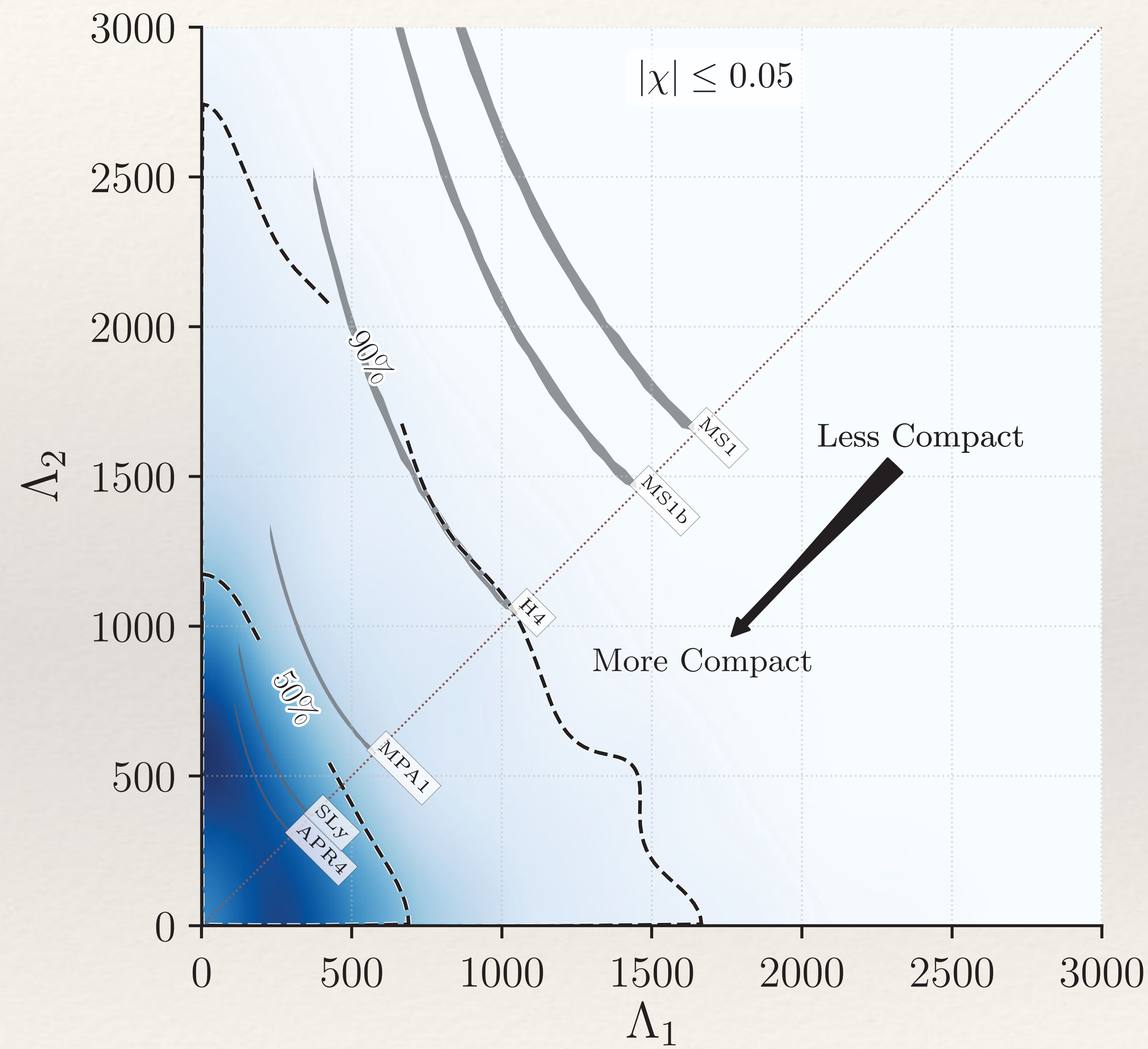




LVC, *Phys. Rev. Lett.* **116**, 061102 (2016)

Examples of Parameter Posteriors

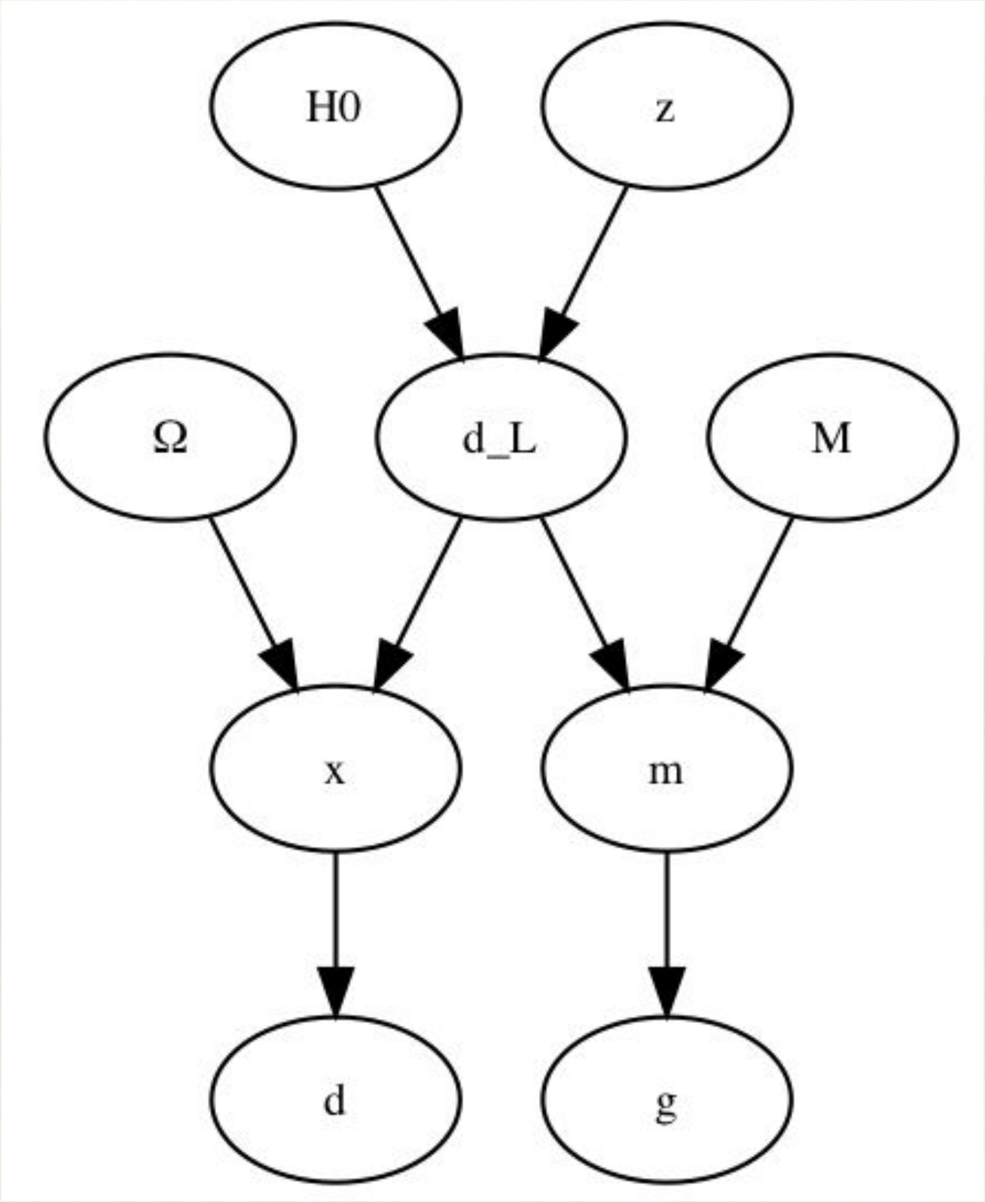




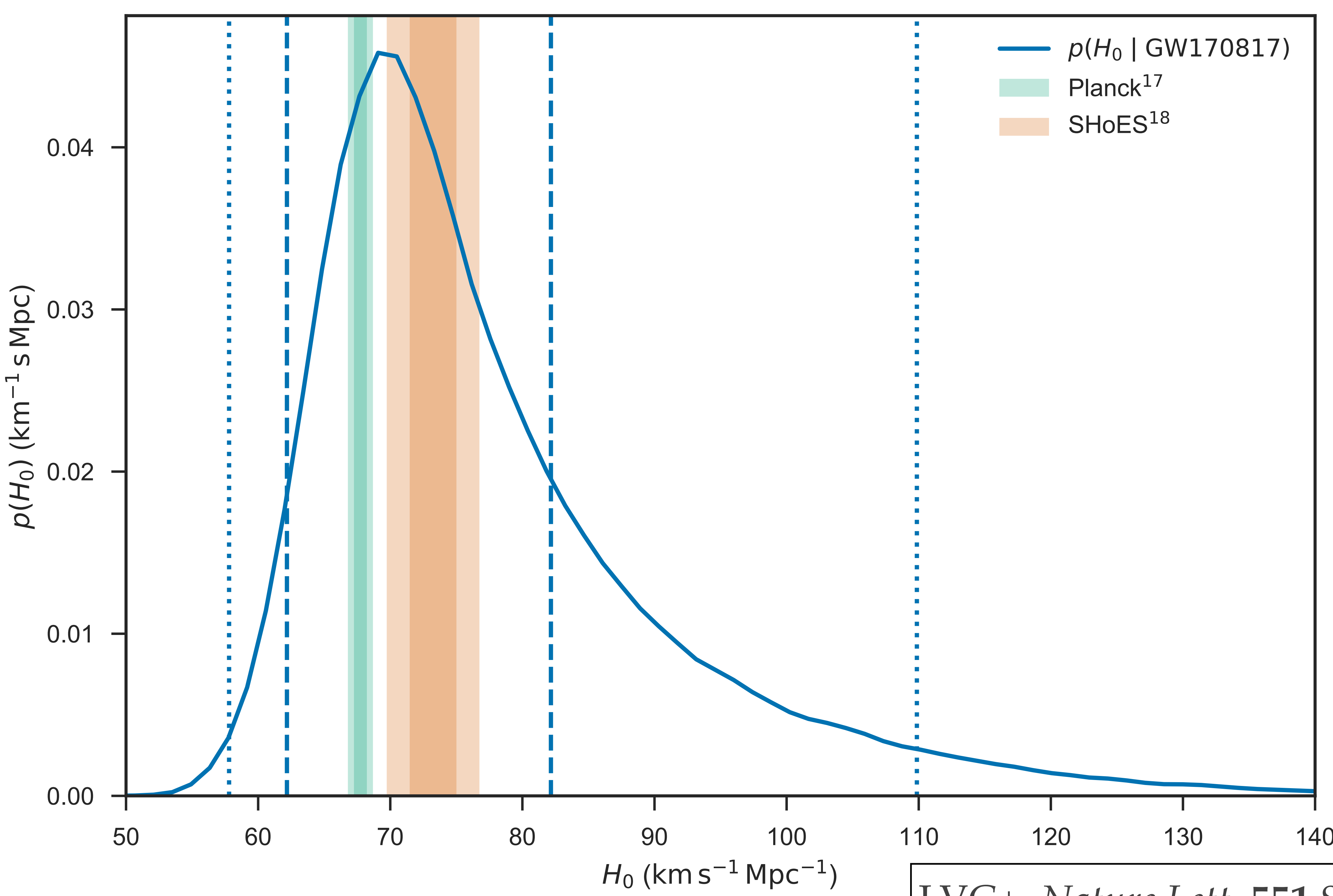
LVC, *Phys. Rev. Lett.* **119** 161101 (2017)

Hierarchical Models

- To infer population properties, we can use *hierarchical models.*
- Parameter prior for individual events depends on further parameters that characterise the population.
- Construct combined posteriors from which inference on either individual events or populations can be derived.
- Models can quickly get complicated! Simplify by imposing conditional independence structure, e.g., $p(x,y,z) = p(x \mid z)\, p(y \mid z)\, p(z)$.

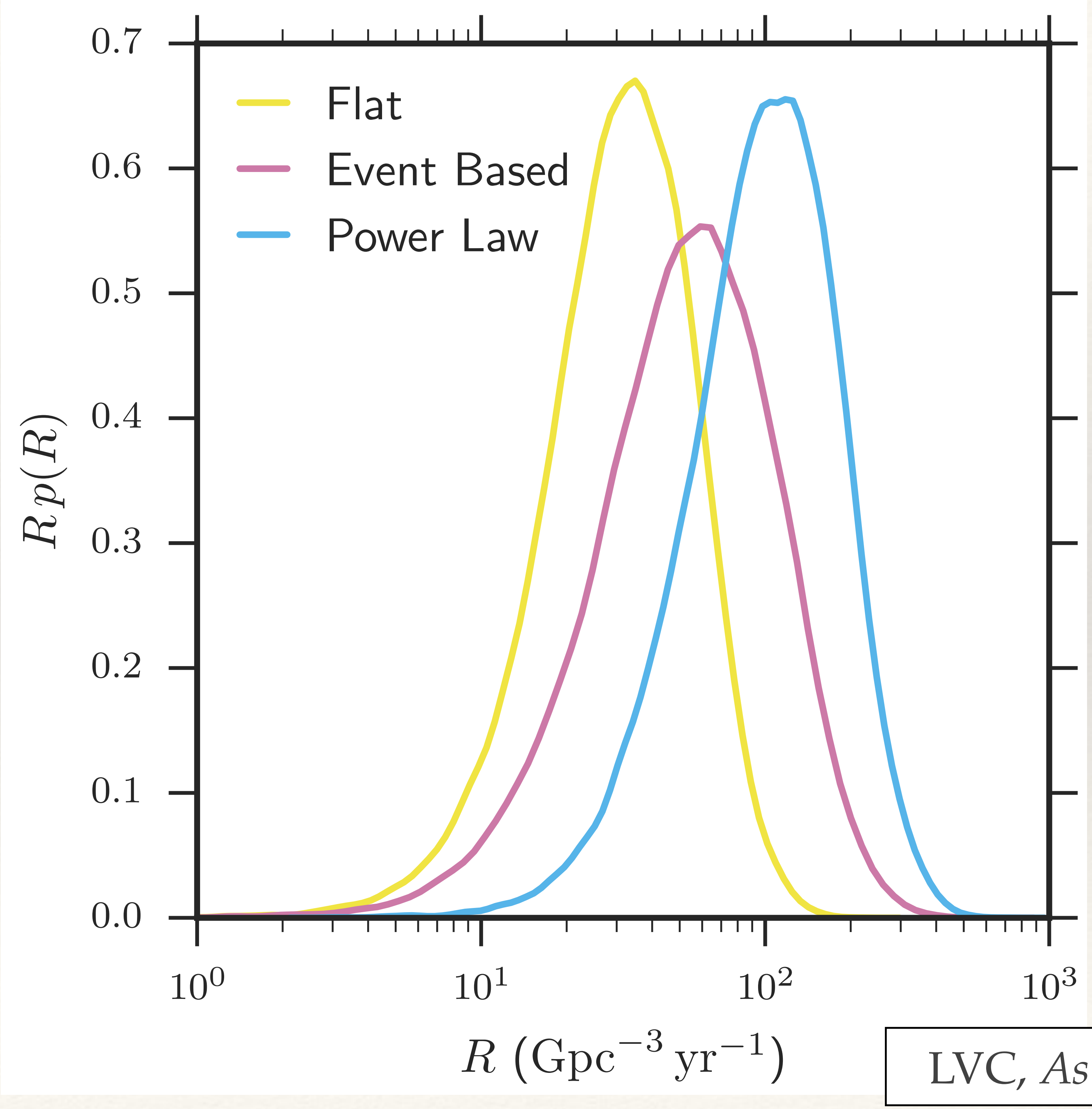


Examples of Parameter Posteriors



LVC+, *Nature Lett.* **551** 85 (2017)

Examples of Parameter Posteriors



LVC, *Astrophys. J. Lett.* **833**, 1 (2016)

Reversible Jump MCMC

- Often the number of sources in the data set is also unknown.
- *Reversible Jump Markov Chain Monte Carlo* is a technique applied in such situations, by periodically proposing jumps between *models*. In GW applications these normally correspond to different numbers of events.
- Represent a proposed move by tuples $(\mathbf{x}, \mathbf{u})$ and $(\mathbf{x}', \mathbf{u}')$. Here $\mathbf{x}$ and $\mathbf{x}'$ denote the parameters of the current and proposed state (which may have different numbers of dimensions) and $\mathbf{u}$, $\mathbf{u}'$ are sets of random numbers that lead to a proposed move from $\mathbf{x}$ to $\mathbf{x}'$ and back.
- Generalisation of acceptance ratio is

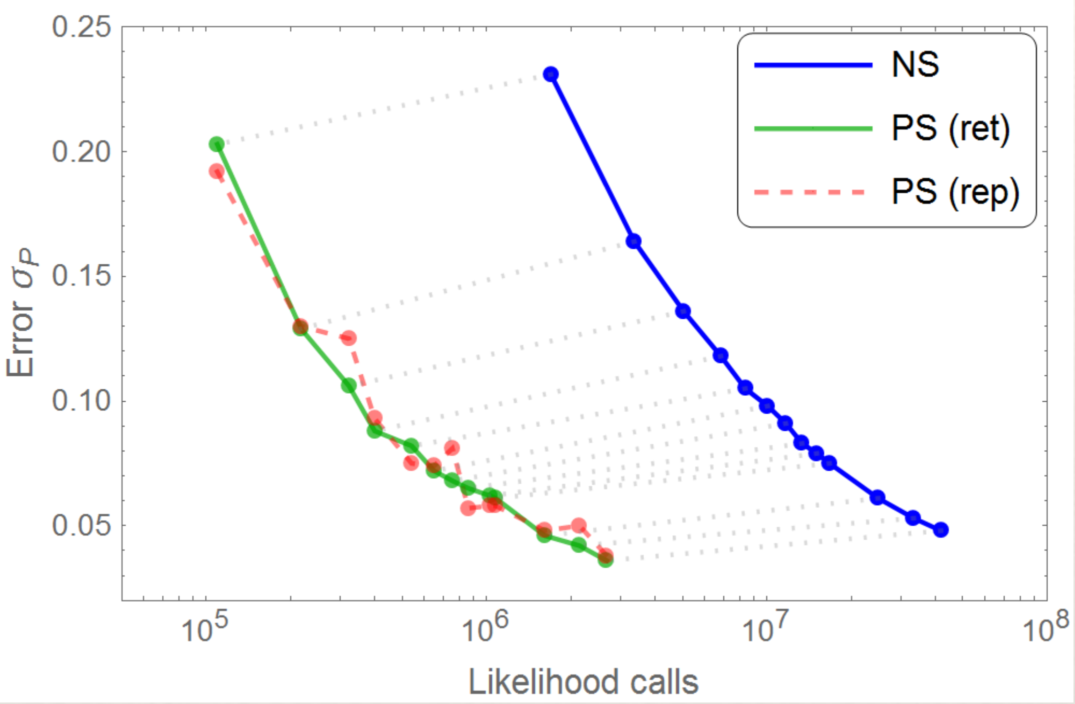
$$\alpha = \min\left(1, \frac{p(\mathbf{x}')q(\mathbf{u}')}{p(\mathbf{x})q(\mathbf{u})}\left|\frac{\partial(\mathbf{x}', \mathbf{u}')}{\partial(\mathbf{x}, \mathbf{u})}\right|\right)$$

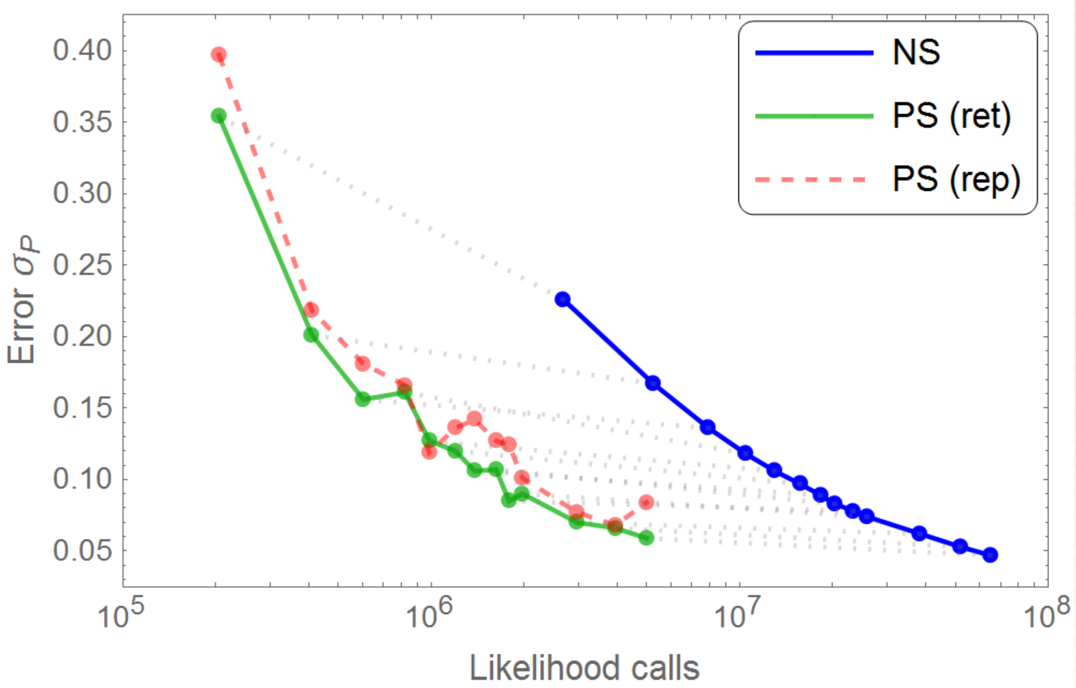
Product Space MCMC

- An alternative to RJMCMC is to use standard MCMC but with an extended parameter space

$$\left\{\vec{\lambda}_1, \vec{\lambda}_2, \ldots, \vec{\lambda}_k \ldots, \vec{\lambda}_M, K\right\}$$

- K is the current parameter space dimension, i.e., number of sources.
- Parameter values with $k > K$ are varied but do not contribute to the likelihood.
- Method can be more efficient than RJMCMC.





Chua et al., *MNRAS* **478**, 28 (2018)

Bayesian Evidence Calculation

- The denominator in Bayes' theorem is the *Bayesian Evidence*.

$$p(\vec{\theta}|d, M) = \frac{p(d|\vec{\theta}, M)p(\vec{\theta}|M)}{p(d|M)}$$

- This is the probability that the observed data *d* would be generated by model *M*. If we have competing models we can use the evidence for *model selection* by computing the *posterior odds*.

$$\frac{p(M_1|d)}{p(M_2|d)} = \frac{p(d|M_1)}{p(d|M_2)}\frac{p(M_1)}{p(M_2)} = \frac{\mathcal{Z}_1}{\mathcal{Z}_2}\frac{p(M_1)}{p(M_2)}$$

Bayesian Evidence Calculation

- The *Bayesian Evidence* is an integral over the model parameter space

$$\mathcal{Z}_i = p(d|M_i) = \int p_i(\vec{\lambda}) p_i(d|\vec{\lambda}) \, \mathrm{d}\vec{\lambda}$$

- This can be rewritten as

$$\frac{1}{\mathcal{Z}_i} = \int \frac{1}{p_i(d|\vec{\lambda})} \frac{p_i(\vec{\lambda}) p_i(d|\vec{\lambda})}{\mathcal{Z}_i} \, \mathrm{d}\vec{\lambda}$$

- which is an integral over the posterior and so can be calculated from MCMC samples via

$$\frac{1}{\mathcal{Z}_i} = \sum_k \frac{1}{p_i(d|\vec{\lambda}_k)}$$

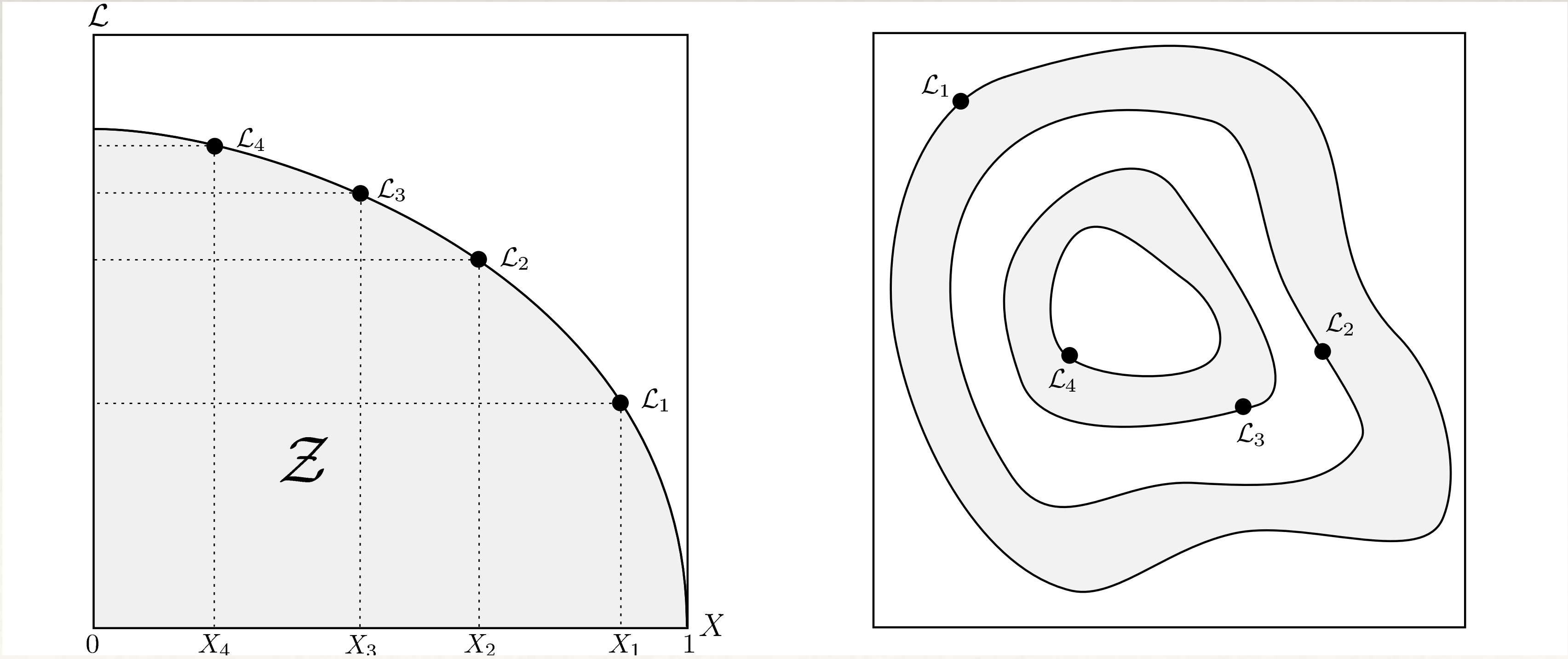
- This is very unstable numerically, due to low number of samples in tails.

Nested Sampling

- Nested Sampling (Skilling 04) provides an efficient way to compute evidences, using a 1D integral over the prior

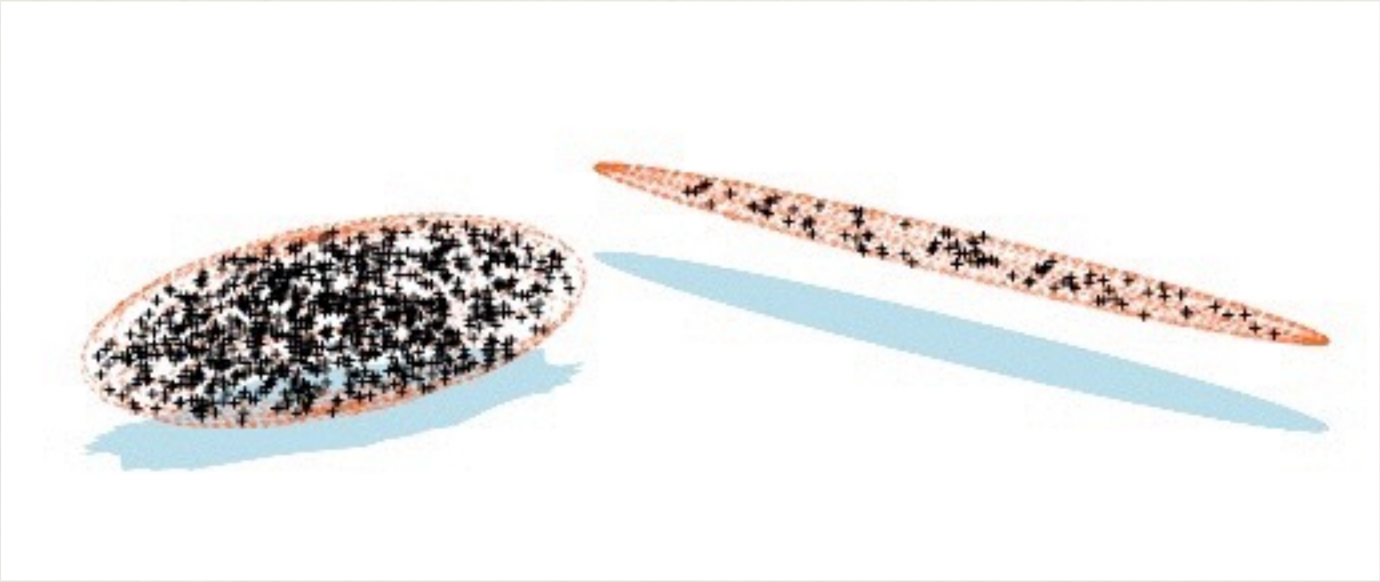
$$\mathcal{Z} = \int \mathcal{L}(\boldsymbol{\Theta})\pi(\boldsymbol{\Theta})\mathrm{d}^N\boldsymbol{\Theta} = \int_0^1 \mathcal{L}(X)\mathrm{d}X, \text{ where } X(\lambda) = \int_{\mathcal{L}(\boldsymbol{\Theta})>\lambda} \pi(\boldsymbol{\Theta})\mathrm{d}^N\boldsymbol{\Theta}$$

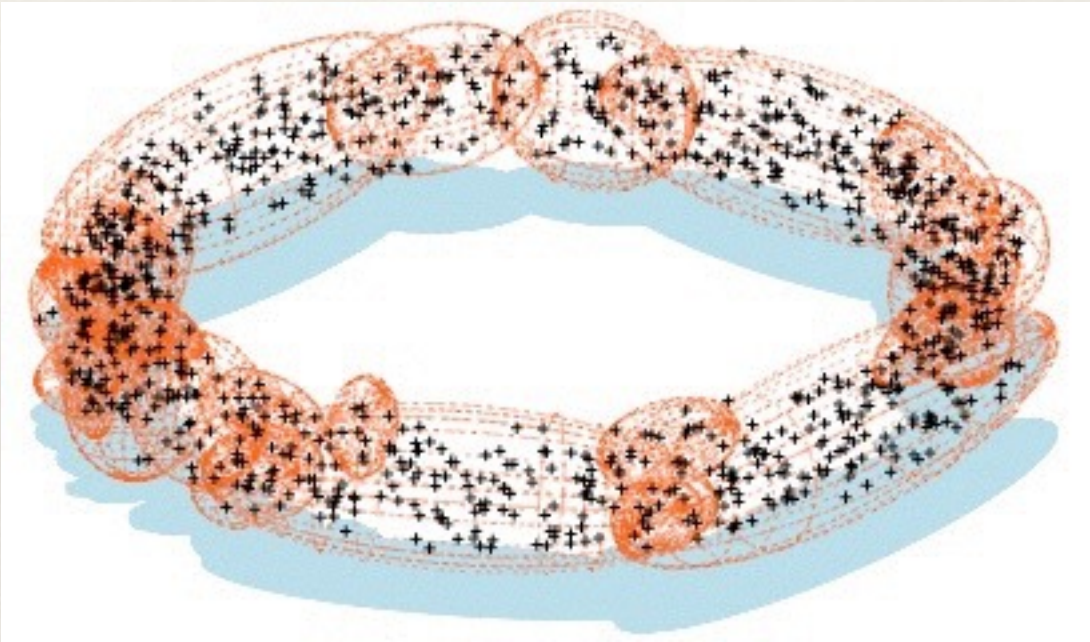
- Use N 'live points', initially chosen at random from the prior. At step i, the point of lowest likelihood, $\mathcal{L}_i$, is replaced by a new point with likelihood $\mathcal{L} > \mathcal{L}_i$. The prior volume is reduced by a factor t, drawn from $p(t) = Nt^{N-1}$, at each step. We climb through nested contours of increasing likelihood as the algorithm proceeds.



MultiNest

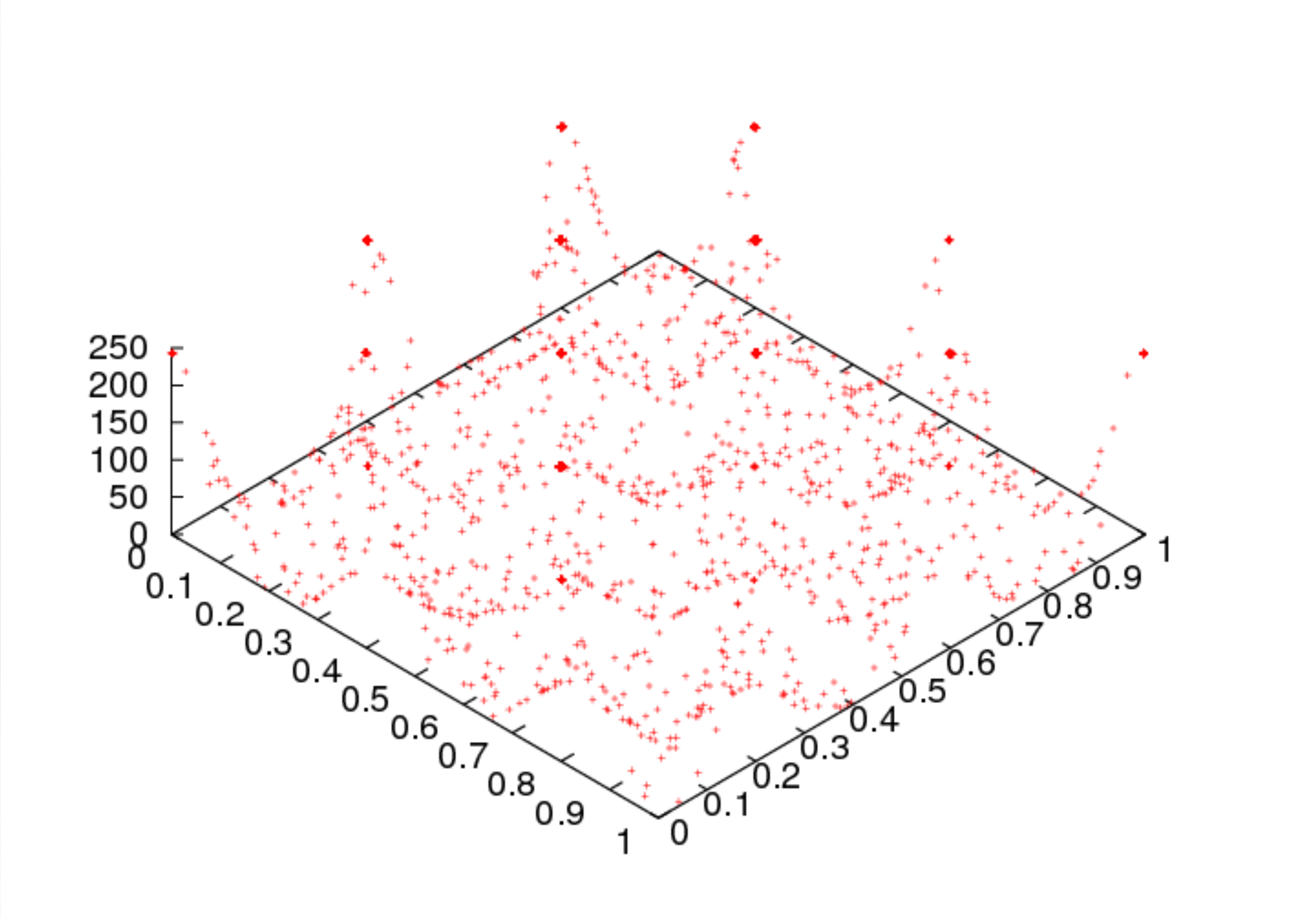
- The trick is to sample efficiently from the prior within the hard constraint that $\mathcal{L} > \mathcal{L}_i$. MultiNest achieves this using an ellipsoidal rejection sampling scheme. The live point set is partitioned into a number of (possibly overlapping) ellipsoids.





- The algorithm is well suited to exploring likelihoods with multiple modes.
- Although designed to compute the evidence, MultiNest also returns the posterior probability distribution.

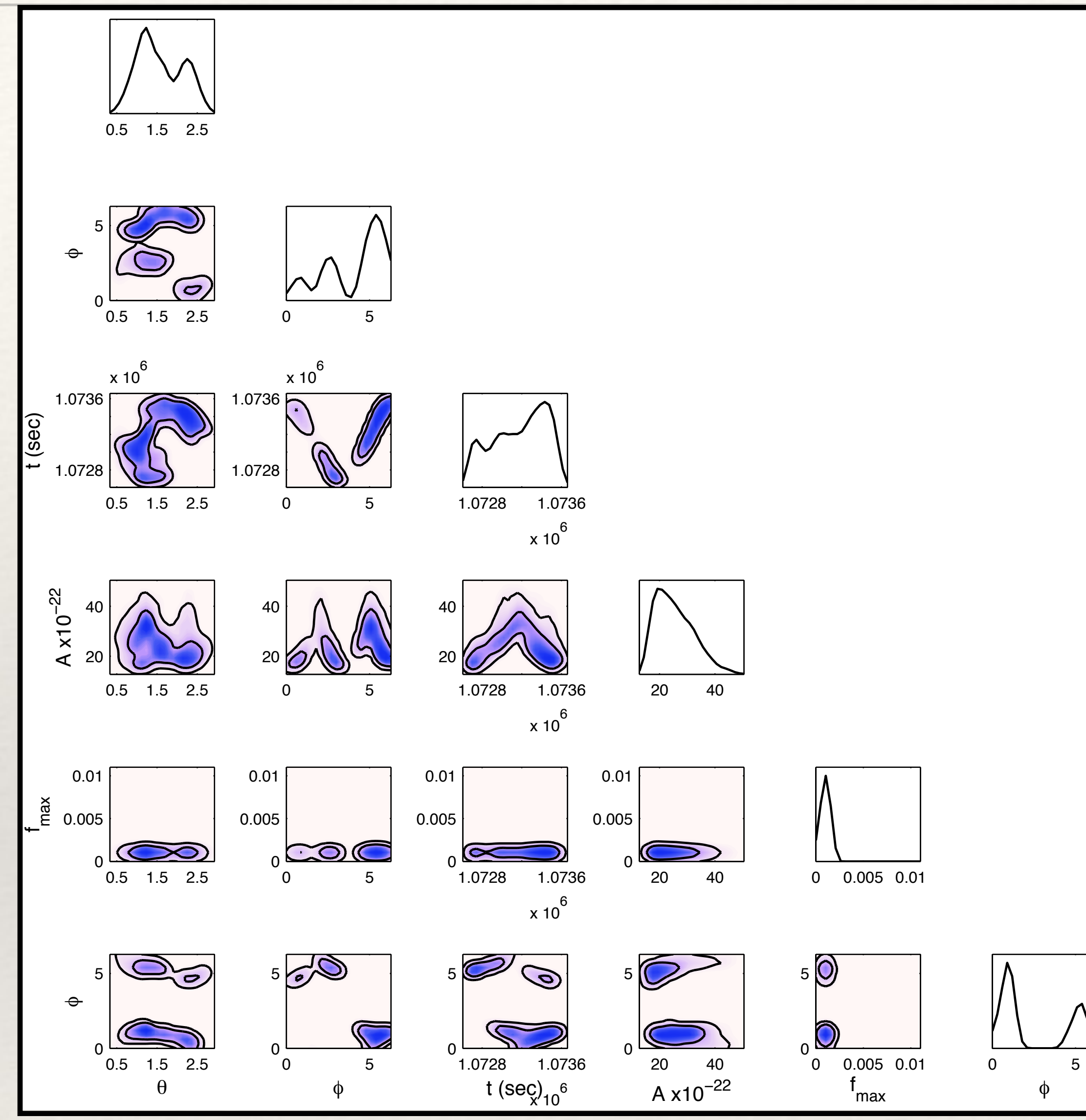
MultiNest



250
200
150
100
50
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

MultiNest for GWs

- MultiNest widely used in astrophysics and other fields.
- There have been a number of applications to gravitational wave detection. For example, **cosmic string** detection in the Mock LISA Data Challenges.
- Identified correct number of signals (3), and recovered waveforms with better than 99% overlap in all cases.



Source	t_c(s)	Channel	True SNR	Recovered SNR		
				Mode 1	Mode 2	Mode 3
1	1.6×10^6	A E	41.0 14.5	41.2 14.5	41.0 14.6	N/A
2	1.1×10^6	A E	30.7 13.9	30.7 13.9	N/A	N/A
3	6×10^5	A E	18.8 36.9	18.9 36.7	18.5 37.1	18.4 36.8

MultiNest for GWs

- MultiNest widely used in astrophysics and other fields.
- There have been a number of applications to gravitational wave detection. For example, **cosmic string** detection in the Mock LISA Data Challenges.
- Identified correct number of signals (3), and recovered waveforms with better than 99% overlap in all cases.
- Evidence ratio identifies burst origin as cosmic string versus generic sine-Gaussian alternative.