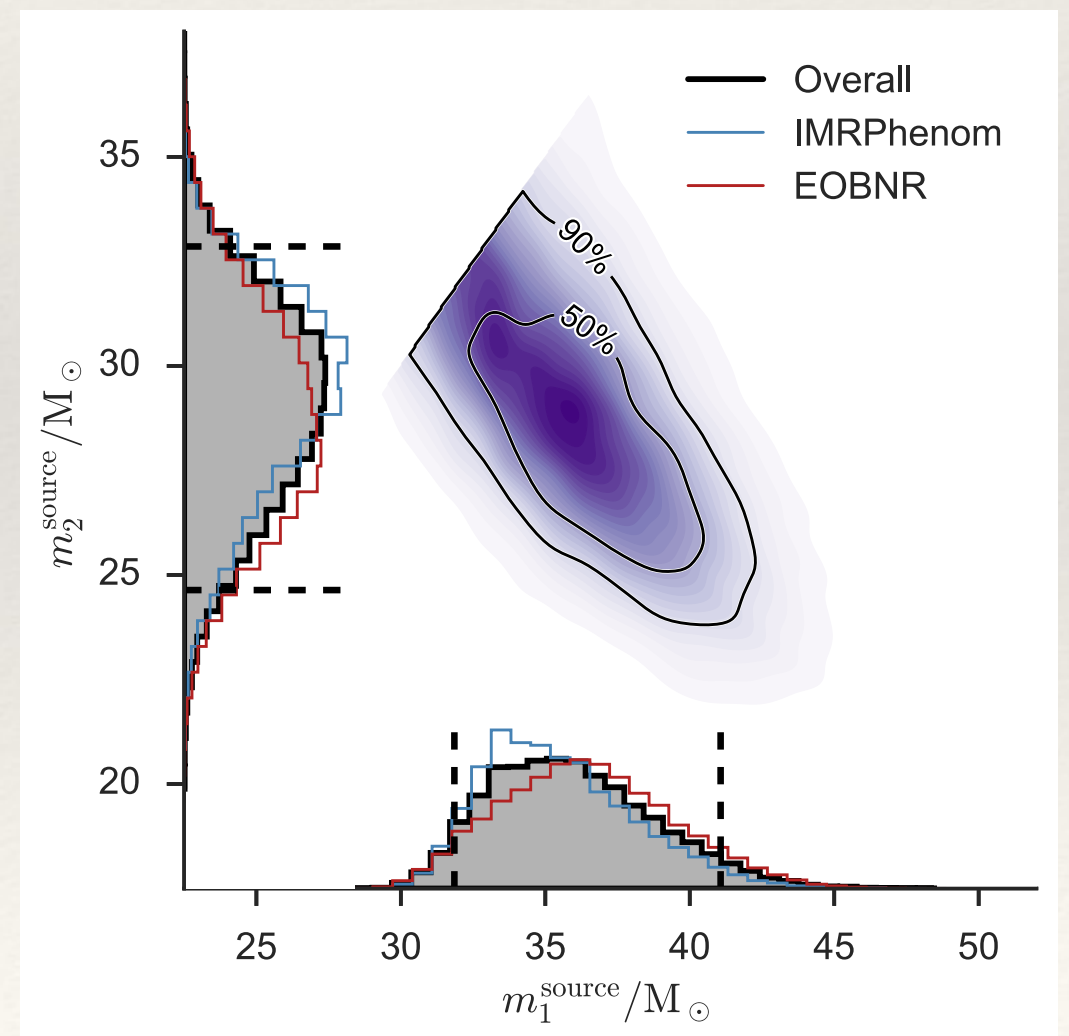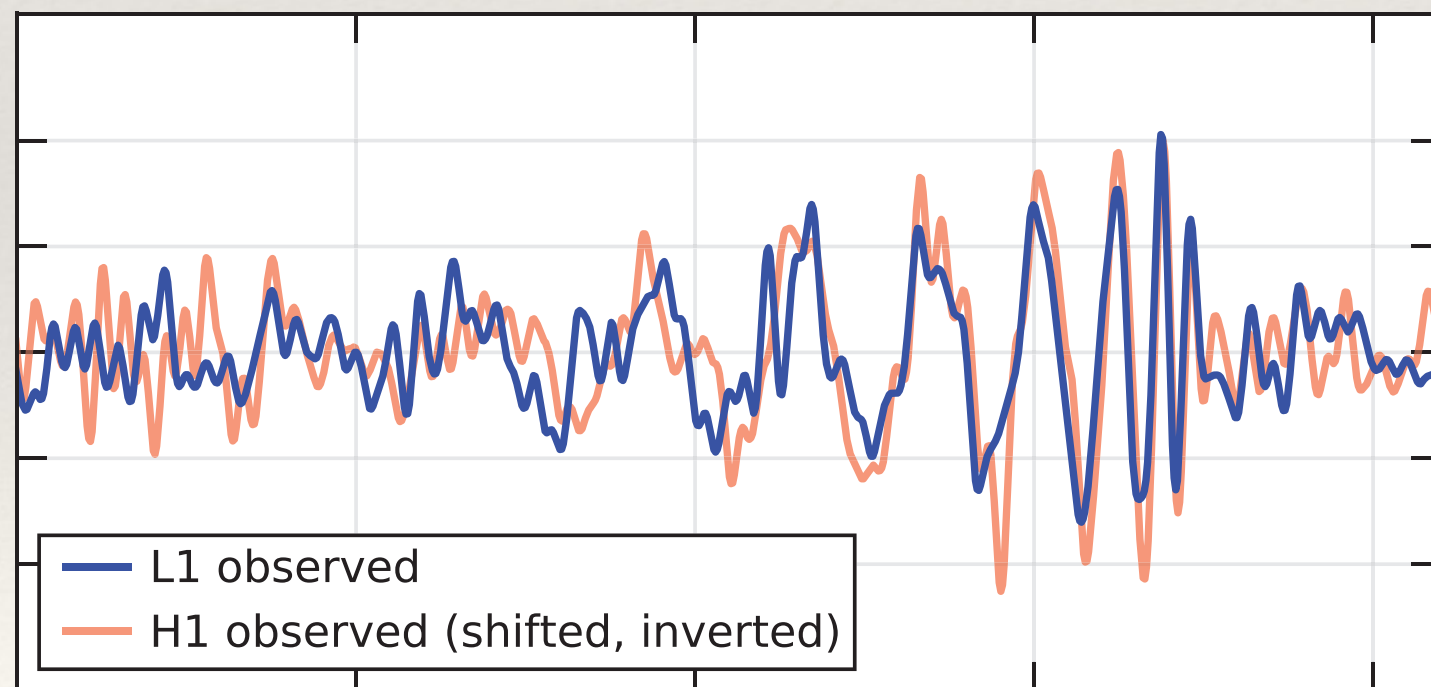# Astrophysics and Data Analysis
Lecture 3: Data Analysis Techniques
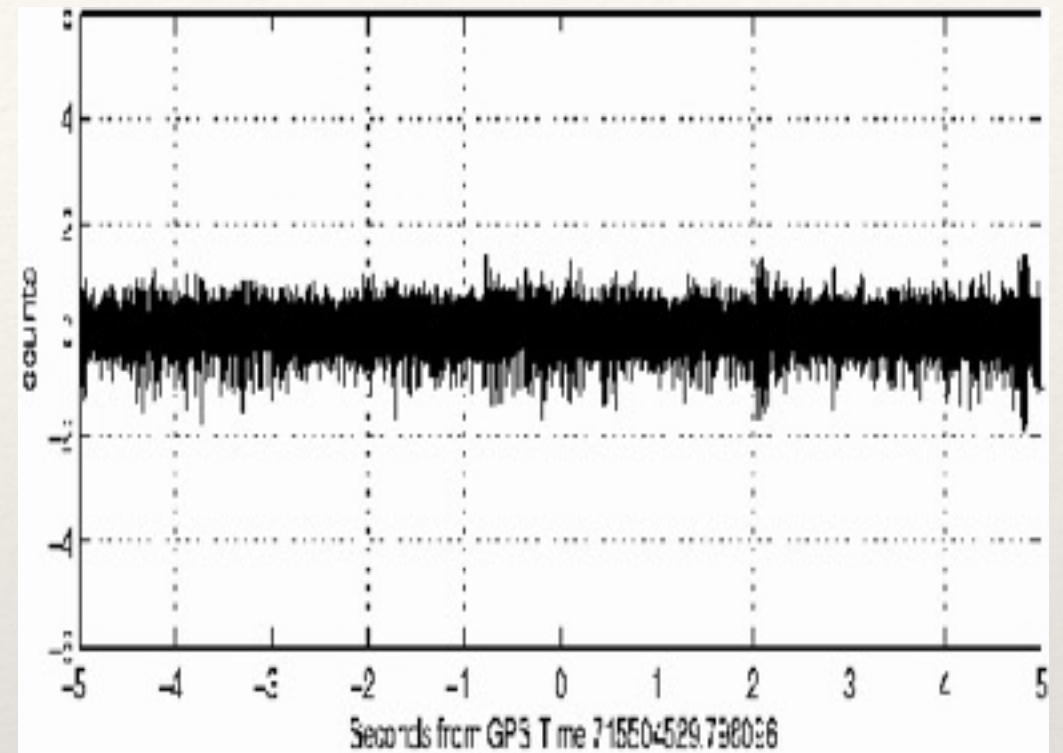
Jonathan Gair, University of Edinburgh

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
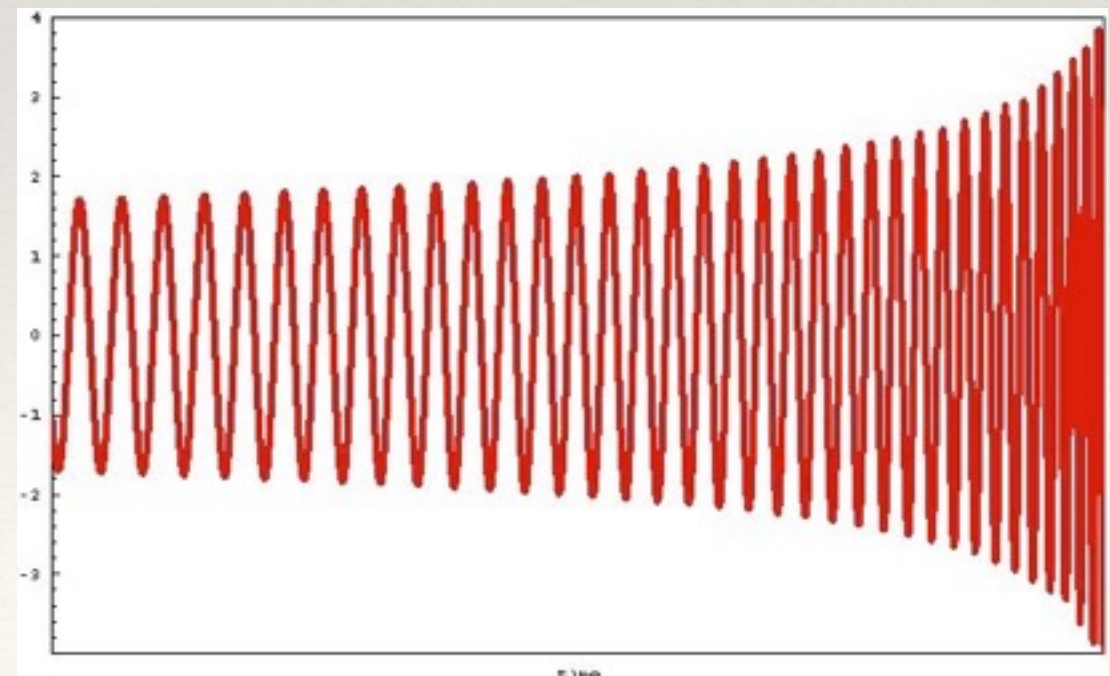
Matched Filtering

- 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.

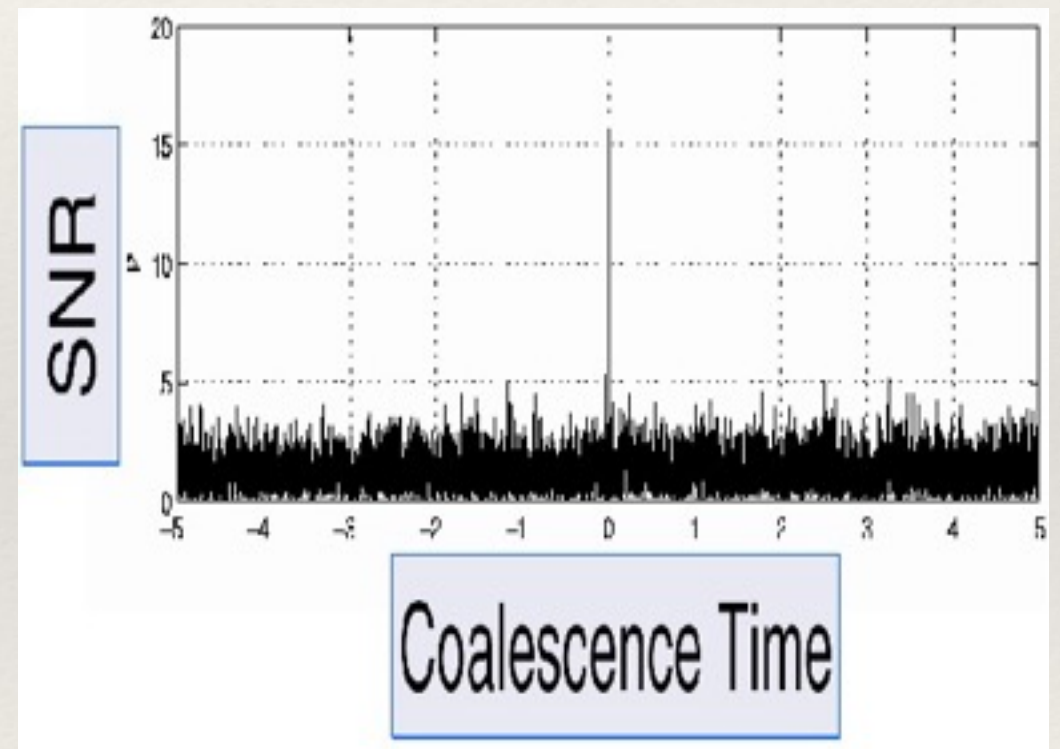- IF a template in the bank matches a signal in the data, we can pull it out of the noise

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.

Linear Signal Approximation

❖ If we write

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

❖ and expand

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

❖ 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

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

❖ 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.
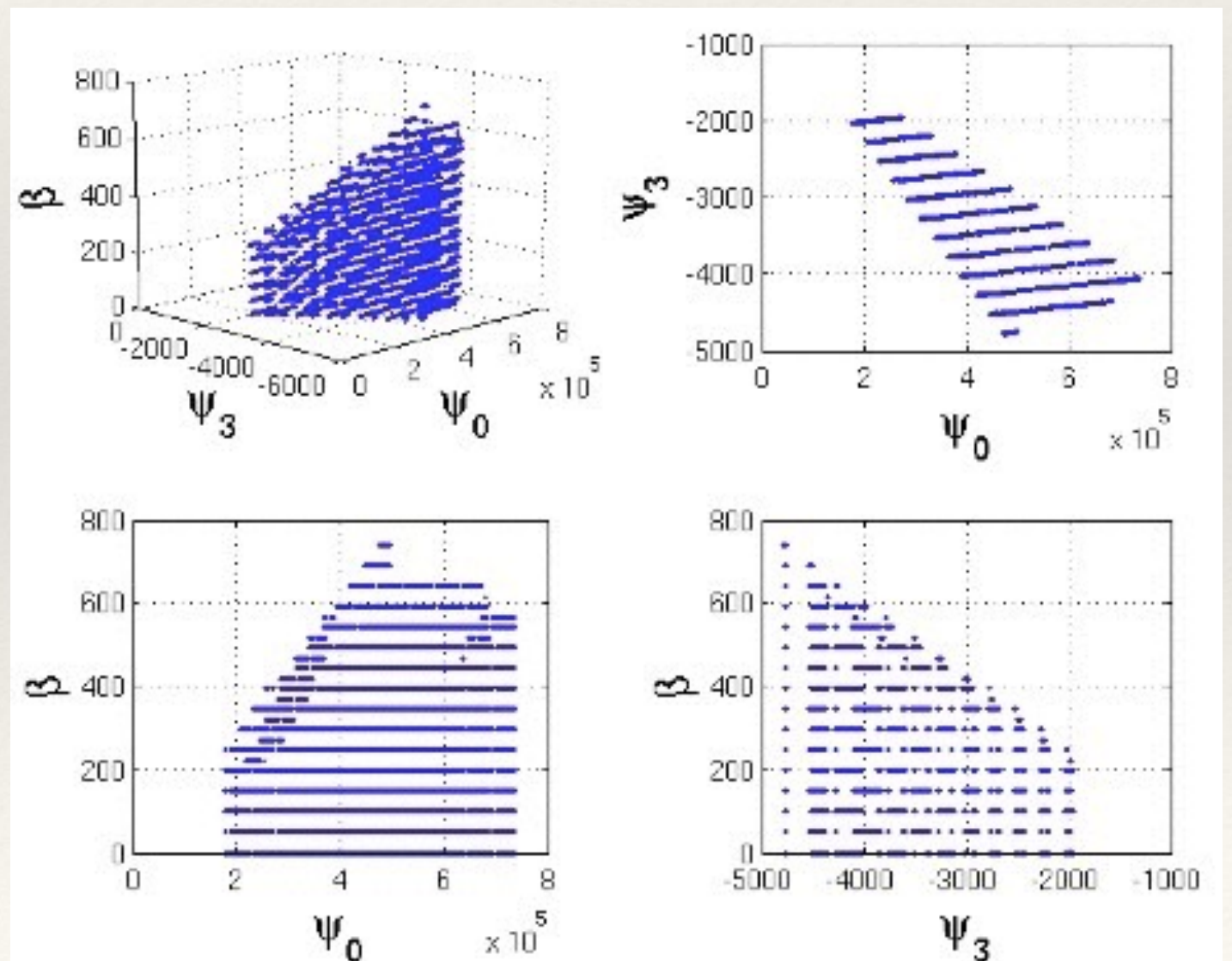
Waveform Consistency

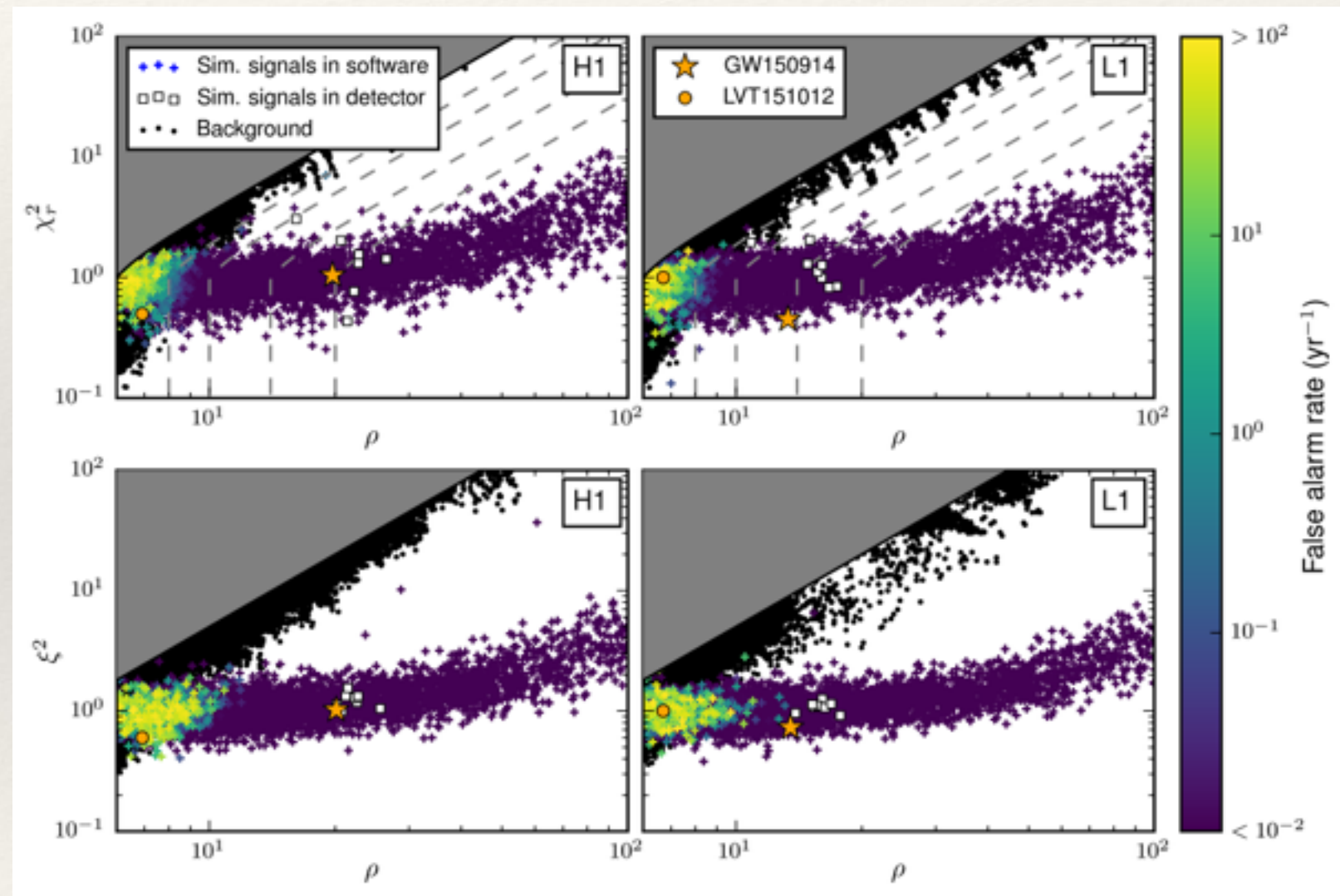❖ If we subtract the correct template the residual at each frequency should be Normally distributed.

❖ 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

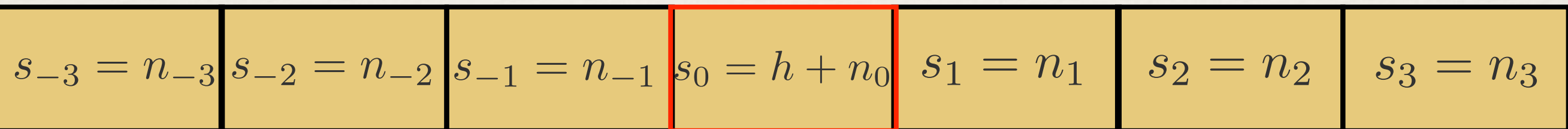$$\hat{\rho} = \frac{\rho}{(1 + (\chi^2/N)^3)^{\frac{1}{6}}}$$

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
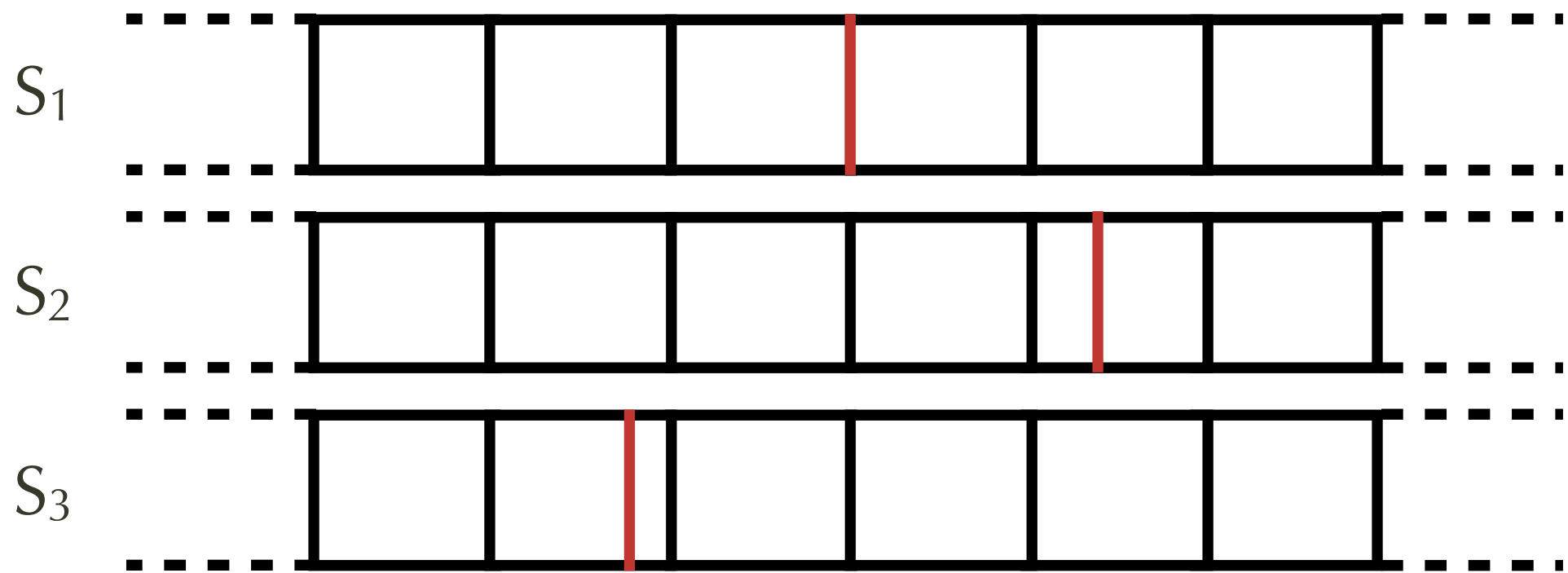
- noise + signal coincidences are not background

- 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.
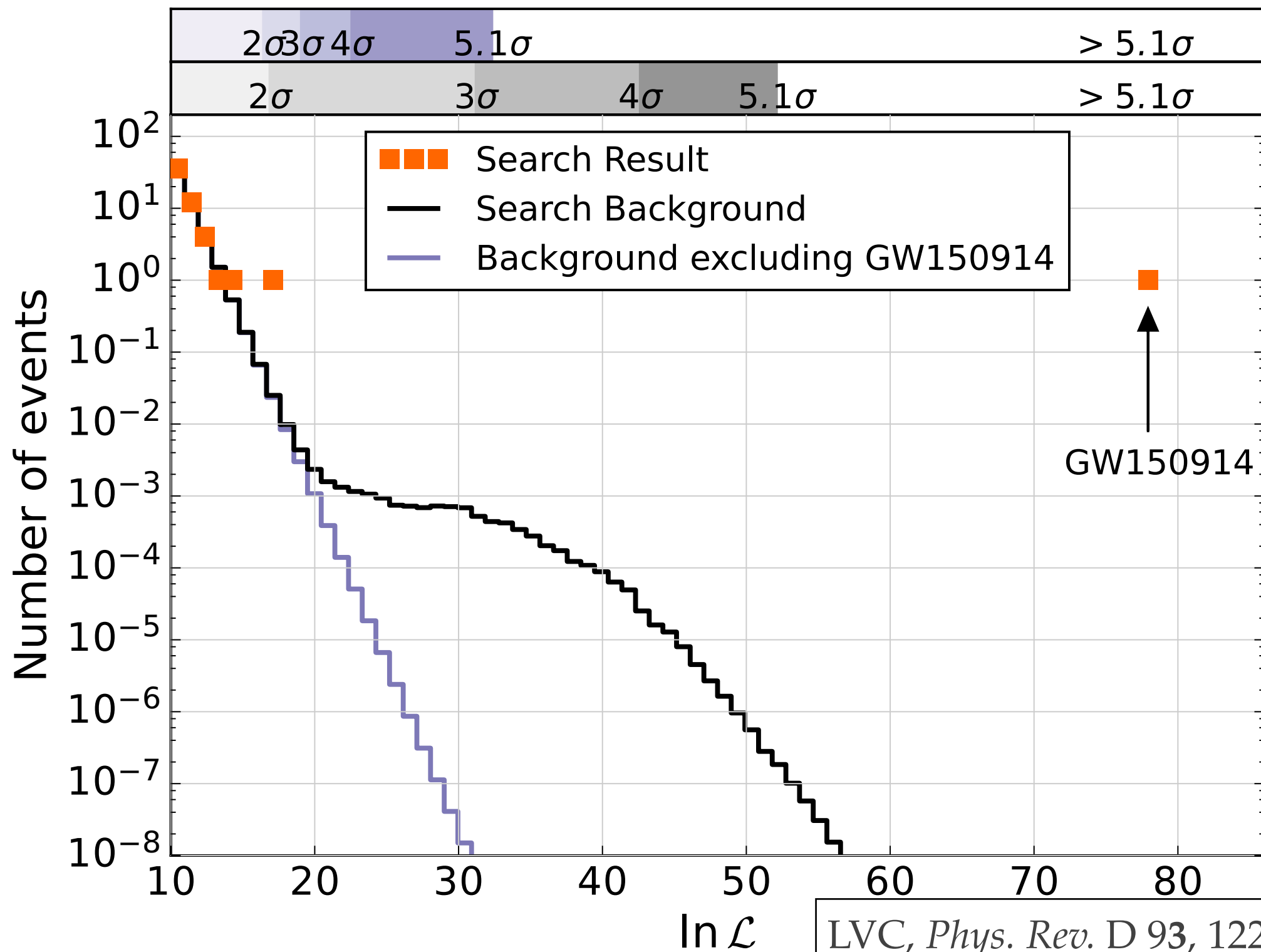
LIGO Pipelines

❖ 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.
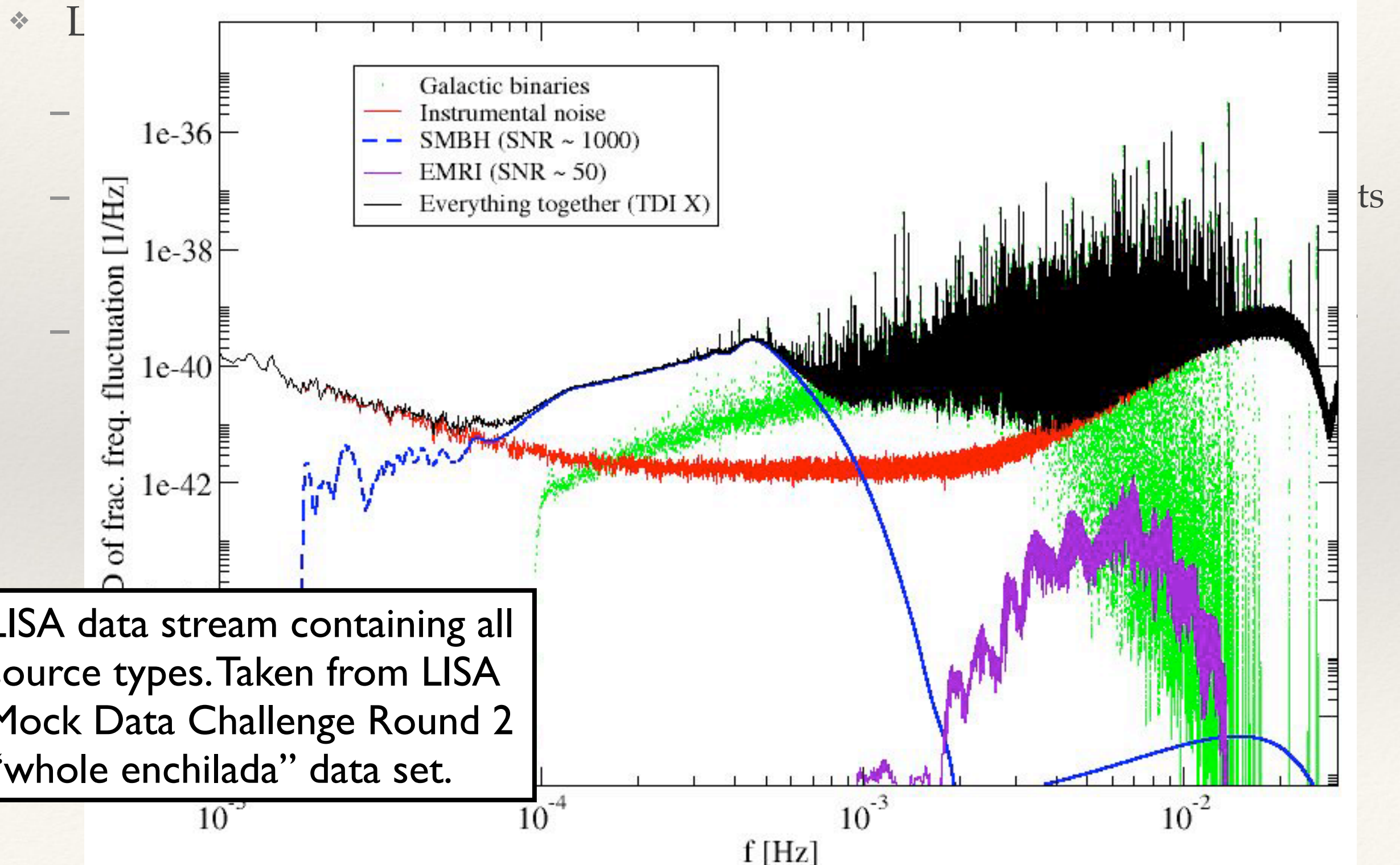
LIGO Pipelines

LIGO Pipelines
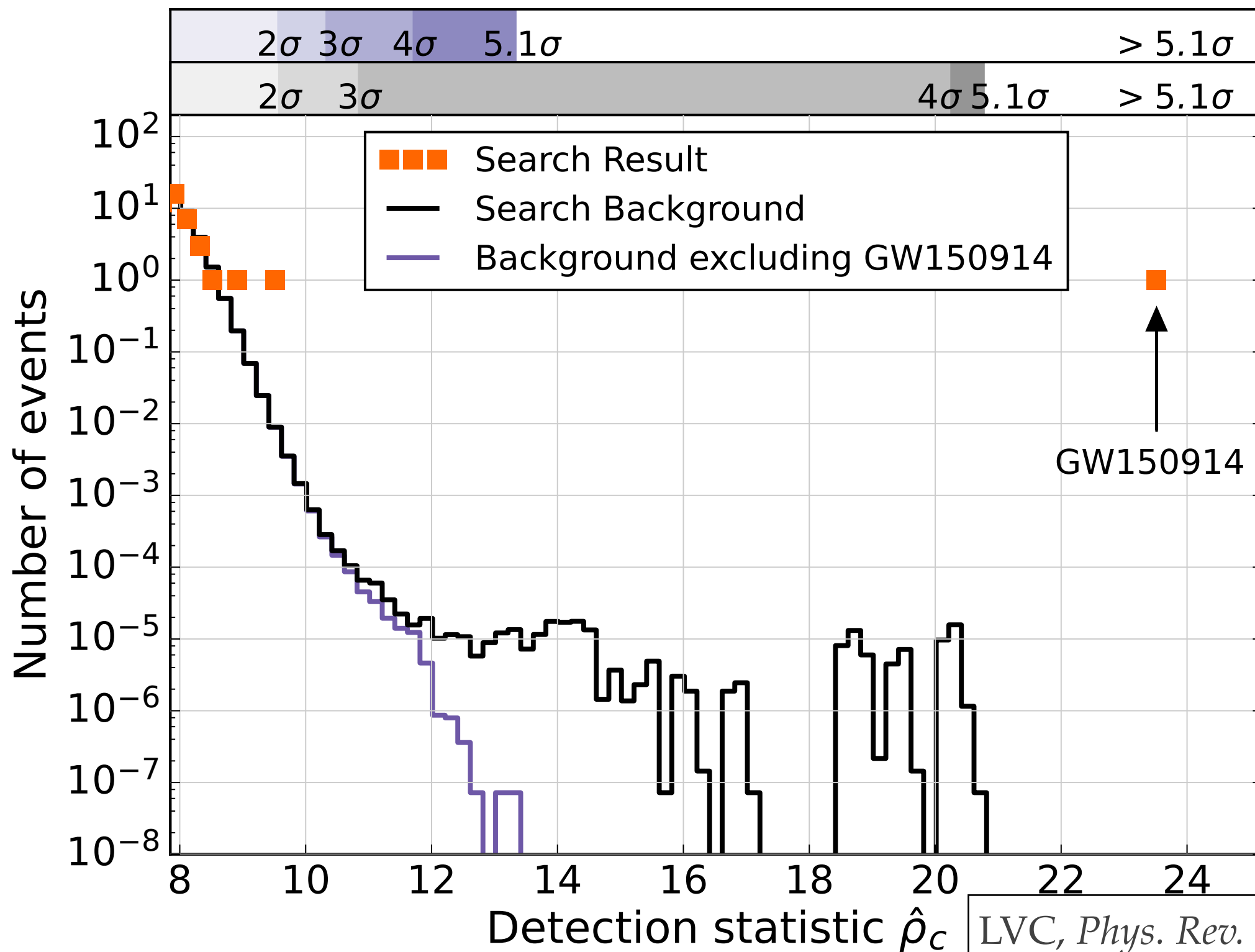
Matched Filtering for LISA

❖ LISA sources are not isolated in time or frequency

  – Every compact binary in the galaxy radiates in the LISA band continuously

  – Typically, there will be a few SMBH merger signals per year, each of which lasts several months and has SNR~1000

  – EMRI events last for the full mission lifetime, and there will be several 1000 of them (confusion noise + resolvable signals)

Matched Filtering for LISA

LISA data stream containing all source types. Taken from LISA Mock Data Challenge Round 2 "whole enchilada" data set.

Matched Filtering for LISA

❖ LISA sources are not isolated in time or frequency

   – Every compact binary in the galaxy radiates in the LISA band continuously

   – Typically, there will be a few SMBH merger signals per year, each of which lasts several months and has SNR~1000

   – EMRI events last for the full mission lifetime, and there will be several 1000 of them (confusion noise + resolvable signals)

❖ Should fit simultaneously for all sources

   – Use techniques such as Markov Chain Monte Carlo that can simultaneously fit an unknown number of sources of multiple types

❖ Cannot run multi-parameter fits on whole parameter space – need techniques to approximately identify the parameters of sources present to constrain MCMC.

Matched Filtering for LISA

For LISA, various matched filtering methods were used in the context of the Mock LISA Data Challenges.

For example, galactic binaries. Algorithms included *gCLEAN*, *Slice and Dice* (used F-statistic), *genetic algorithms*, variants of MCMC including *RJMCMC* and *BAM*.
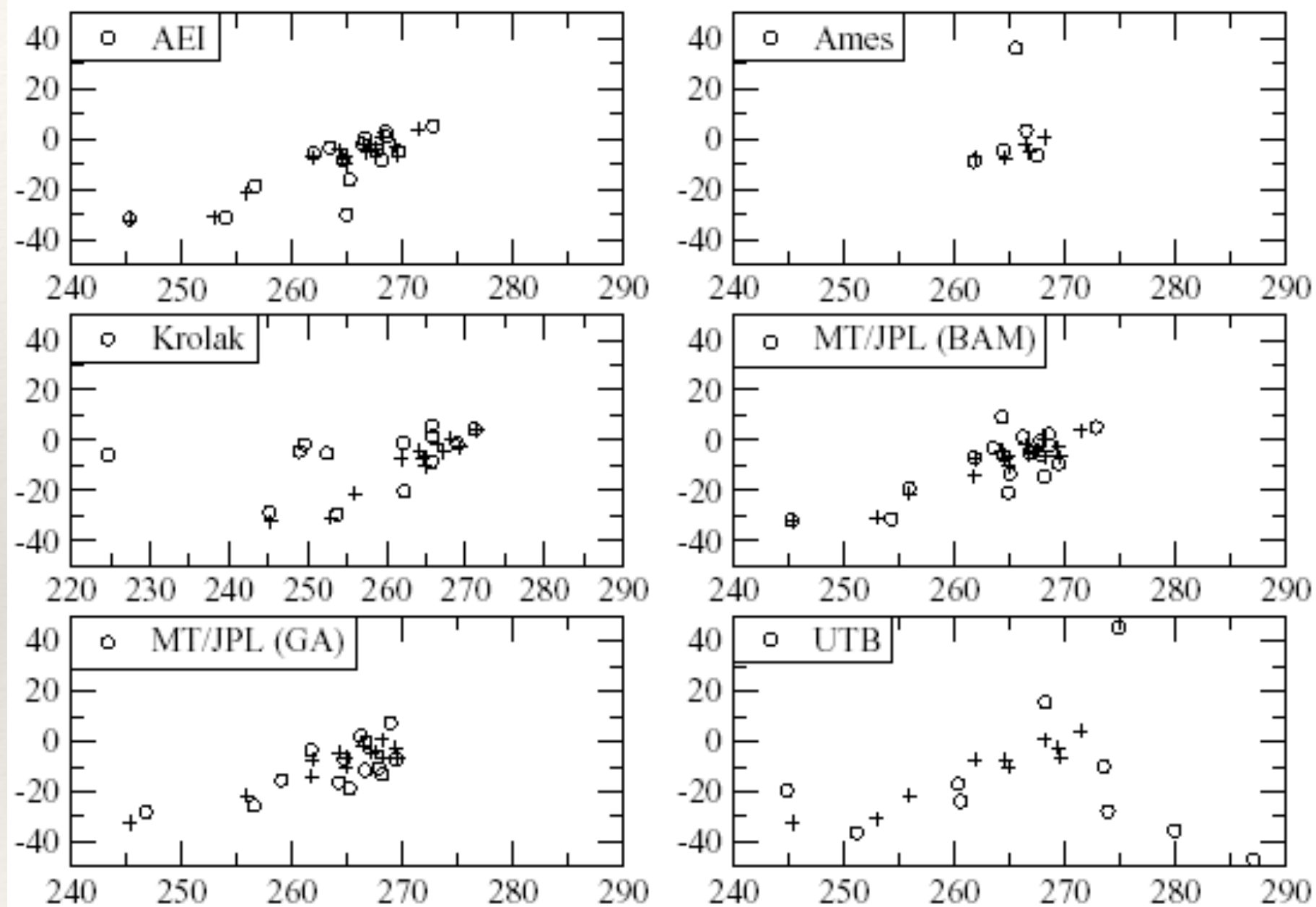
Figure 1. Recovered sky positions from each entry for Challenge 1.1.3. The source positions are indicated by + and the recovered positions are indicated by ∘. Each plot includes only those sources which are within a frequency bin of a recovered source. The plots are of ecliptic latitude vs. longitude in degrees.

Matched Filtering for LISA

❖ Similar approaches successful for supermassive black hole binaries.

source (SNR$_{\text{true}}$)	group	$\Delta M_c/M_c$ $\times 10^{-5}$	$\Delta\eta/\eta$ $\times 10^{-4}$	Δt_c (sec)	Δsky (deg)	Δa_1 $\times 10^{-3}$	Δa_2 $\times 10^{-3}$	$\Delta D/D$ $\times 10^{-2}$	SNR	FF$_A$	FF$_E$
MBH-1	AEI	2.4	6.1	62.9	11.6	7.6	47.4	8.0	1657.71	0.9936	0.9914
(1670.58)	CambAEI	3.4	40.7	24.8	2.0	8.5	79.6	0.7	1657.19	0.9925	0.9917
	MTAPC	24.8	41.2	619.2	171.0	13.3	28.7	4.0	1669.97	0.9996	0.9997
	JPL	40.5	186.6	23.0	26.9	39.4	66.1	6.9	1664.87	0.9972	0.9981
	GSFC	1904.0	593.2	183.9	82.5	5.7	124.3	94.9	267.04	0.1827	0.1426
MBH-3	AEI	9.0	5.2	100.8	175.9	6.2	18.6	2.7	846.96	0.9995	0.9989
(847.61)	CambAEI	13.5	57.4	138.9	179.0	21.3	7.2	1.5	847.04	0.9993	0.9993
	MTAPC	333.0	234.1	615.7	80.2	71.6	177.2	16.1	842.96	0.9943	0.9945
	JPL	153.0	51.4	356.8	11.2	187.7	414.9	2.7	835.73	0.9826	0.9898
	GSFC	8168.4	2489.9	3276.9	77.9	316.3	69.9	95.6	218.05	0.2815	0.2314
MBH-4	AEI	4.5	75.2	31.4	0.1	47.1	173.6	9.1	160.05	0.9989	0.9994
(160.05)	CambAEI	3.2	171.9	30.7	0.2	52.9	346.1	21.6	160.02	0.9991	0.9992
	MTAPC	48.6	2861.0	5.8	7.3	33.1	321.1	33.0	149.98	0.8766	0.9352
	JPL	302.6	262.0	289.3	4.0	47.6	184.5	28.3	158.34	0.8895	0.9925
	GSFC	831.3	1589.2	1597.6	94.4	59.8	566.7	95.4	-45.53	-0.1725	-0.2937
MBH-2	AEI	1114.1	952.2	38160.8	171.1	331.7	409.0	15.3	20.54	0.9399	0.9469
(18.95)	CambAEI	88.7	386.6	6139.7	172.4	210.8	130.7	24.4	20.36	0.9592	0.9697
	MTAPC	128.6	45.8	16612.0	8.9	321.4	242.4	13.1	20.27	0.9228	0.9260
	JPL	287.0	597.7	11015.7	11.8	375.3	146.3	9.9	18.69	0.9661	0.9709
MBH-6	AEI	1042.3	1235.6	82343.2	2.1	258.2	191.6	26.0	13.69	0.9288	0.9293
(12.82)	CambAEI	5253.2	1598.8	953108.0	158.3	350.8	215.4	29.4	10.17	0.4018	0.4399
	MTAPC	56608.7	296.7	180458.8	119.7	369.2	297.6	25.1	11.34	-0.0004	0.0016

Matched Filtering for LISA

❖ Multi stage matched filtering approach used for EMRIs

type[1]	ν (mHz)	$\mu/M_\odot$	$M/M_\odot$	e_0	θ_S	φ_S	λ	a/M^2	SNR
True	0.1920421	10.296	9517952	0.21438	1.018	4.910	0.4394	0.69816	120.5
Found	0.1920437	10.288	9520796	0.21411	1.027	4.932	0.4384	0.69823	118.1
True	0.34227777	9.771	5215577	0.20791	1.211	4.6826	1.4358	0.63796	132.9
Found	0.34227742	9.769	5214091	0.20818	1.172	4.6822	1.4364	0.63804	132.8
True	0.3425731	9.697	5219668	0.19927	0.589	0.710	0.9282	0.53326	79.5
Found	0.3425712	9.694	5216925	0.19979	0.573	0.713	0.9298	0.53337	79.7
True	0.8514396	10.105	955795	0.45058	2.551	0.979	1.6707	0.62514	101.6
Found	0.8514390	10.106	955544	0.45053	2.565	1.012	1.6719	0.62534	96.0
True	0.8321840	9.790	1033413	0.42691	2.680	1.088	2.3196	0.65829	55.3
Found	0.8321846	9.787	1034208	0.42701	2.687	1.053	2.3153	0.65770	55.6
Blind									
True	0.1674472	10.131	10397935	0.25240	2.985	4.894	1.2056	0.65101	52.0
Found	0.1674462	10.111	10375301	0.25419	3.023	4.857	1.2097	0.65148	51.7
True	0.9997627	9.7478	975650	0.360970	1.453	4.95326	0.5110	0.65005	122.9
Found	0.9997626	9.7479	975610	0.360966	1.422	4.95339	0.5113	0.65007	116.0
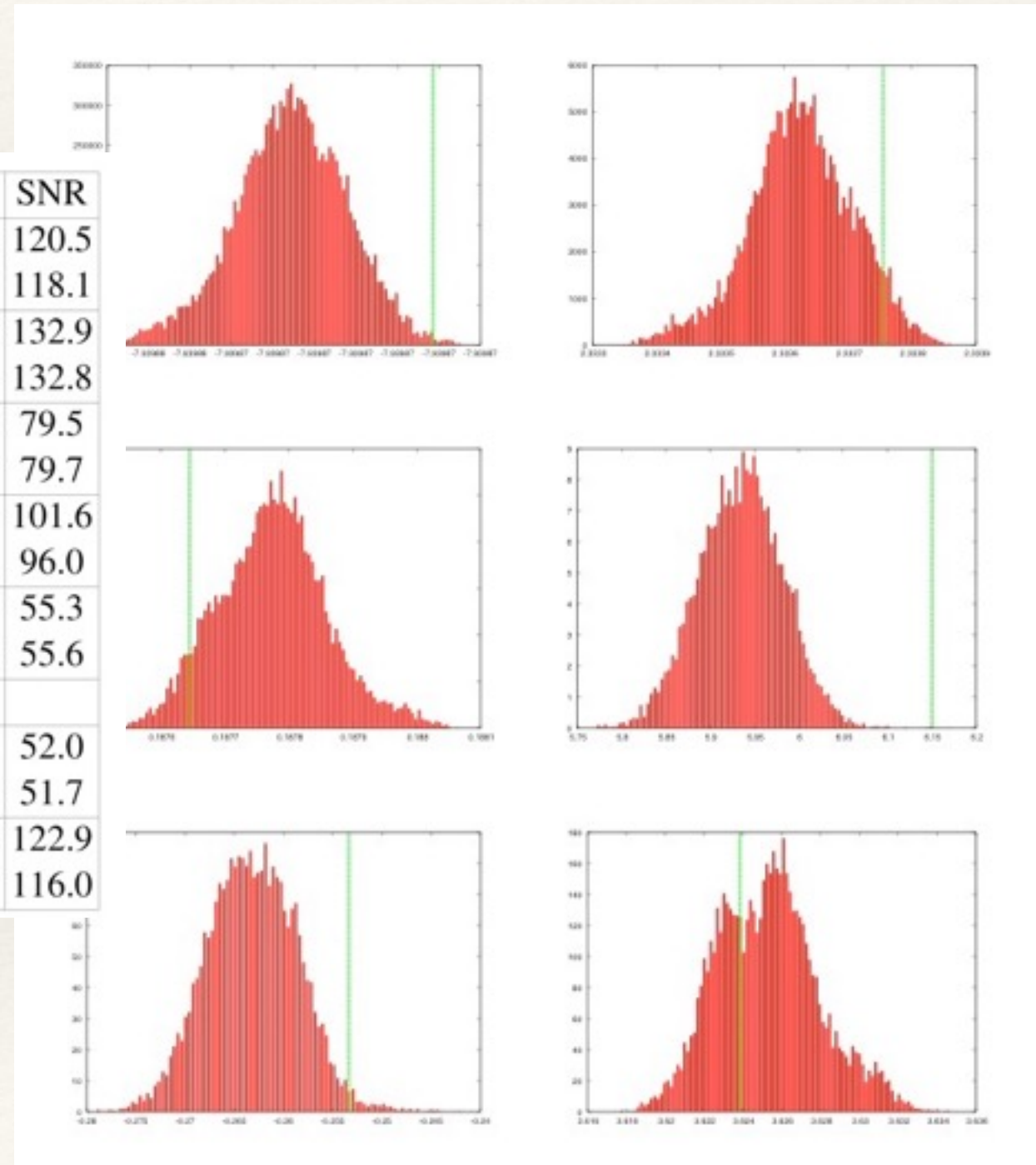
Babak, JG & Porter (2009)

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.
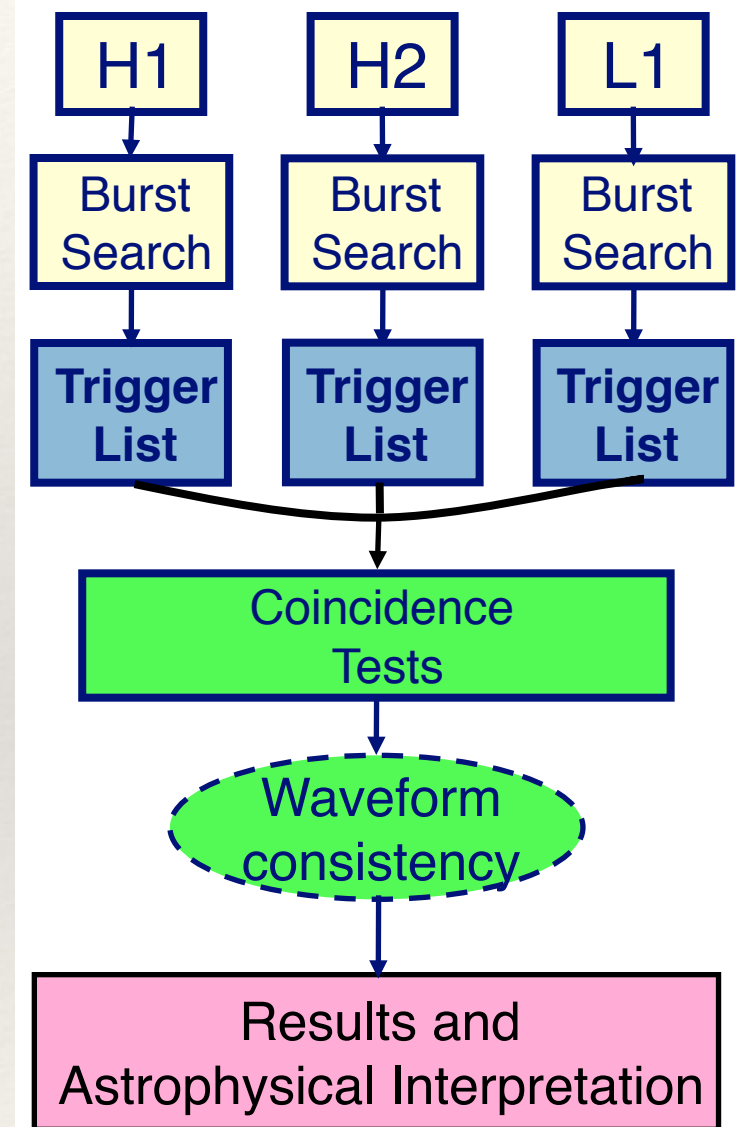
Unmodelled Searches: Coherent Wave Burst

❖ 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.

Network pixel x[i] = {a_k(i, τ_k)}

$$E_c = \sum_{m \neq n} L_{mn}$$

$$N = |X - \xi_\sigma|^2$$

$$cc = \frac{E_c}{N + E_c}$$

Background

Simulation

cluster halo

core
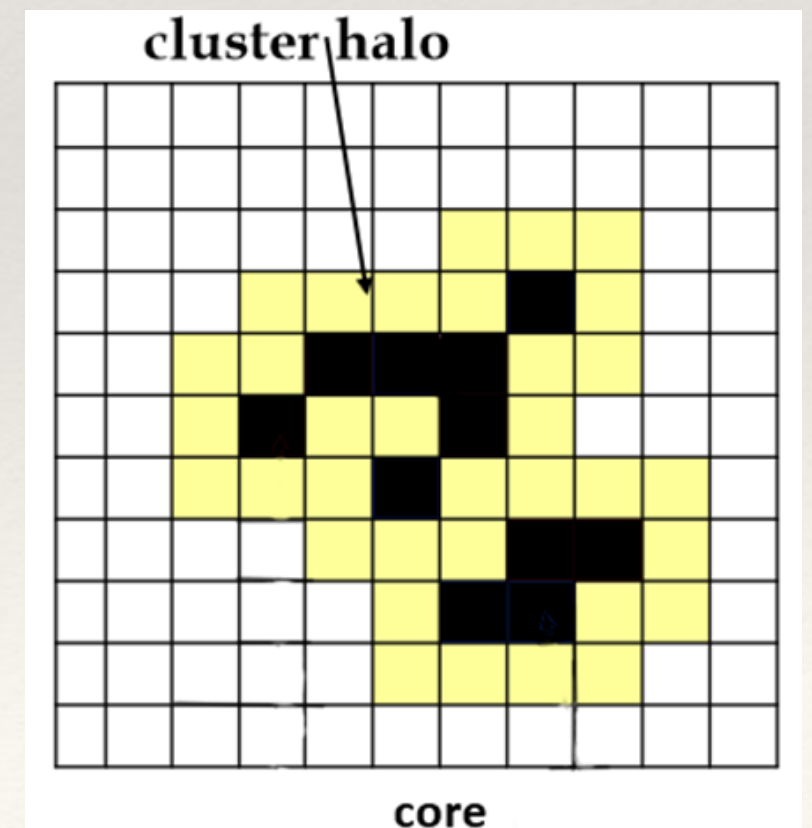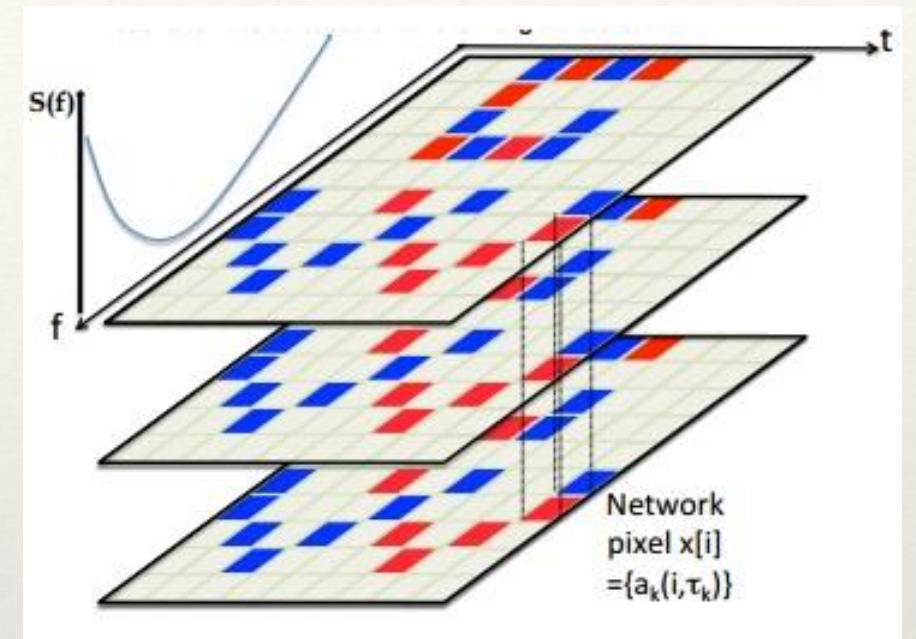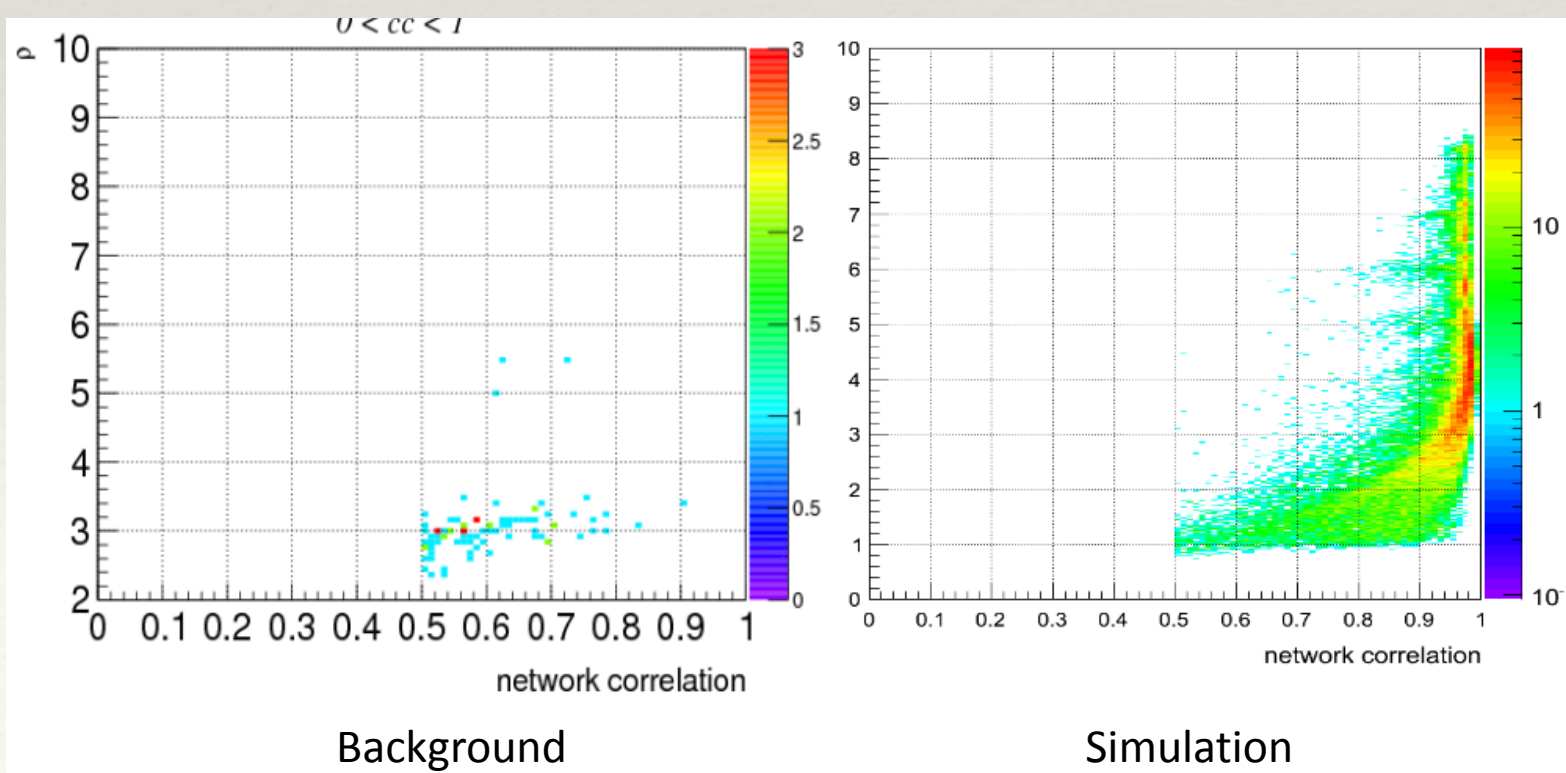
Unmodelled Searches: Coherent Wave Burst
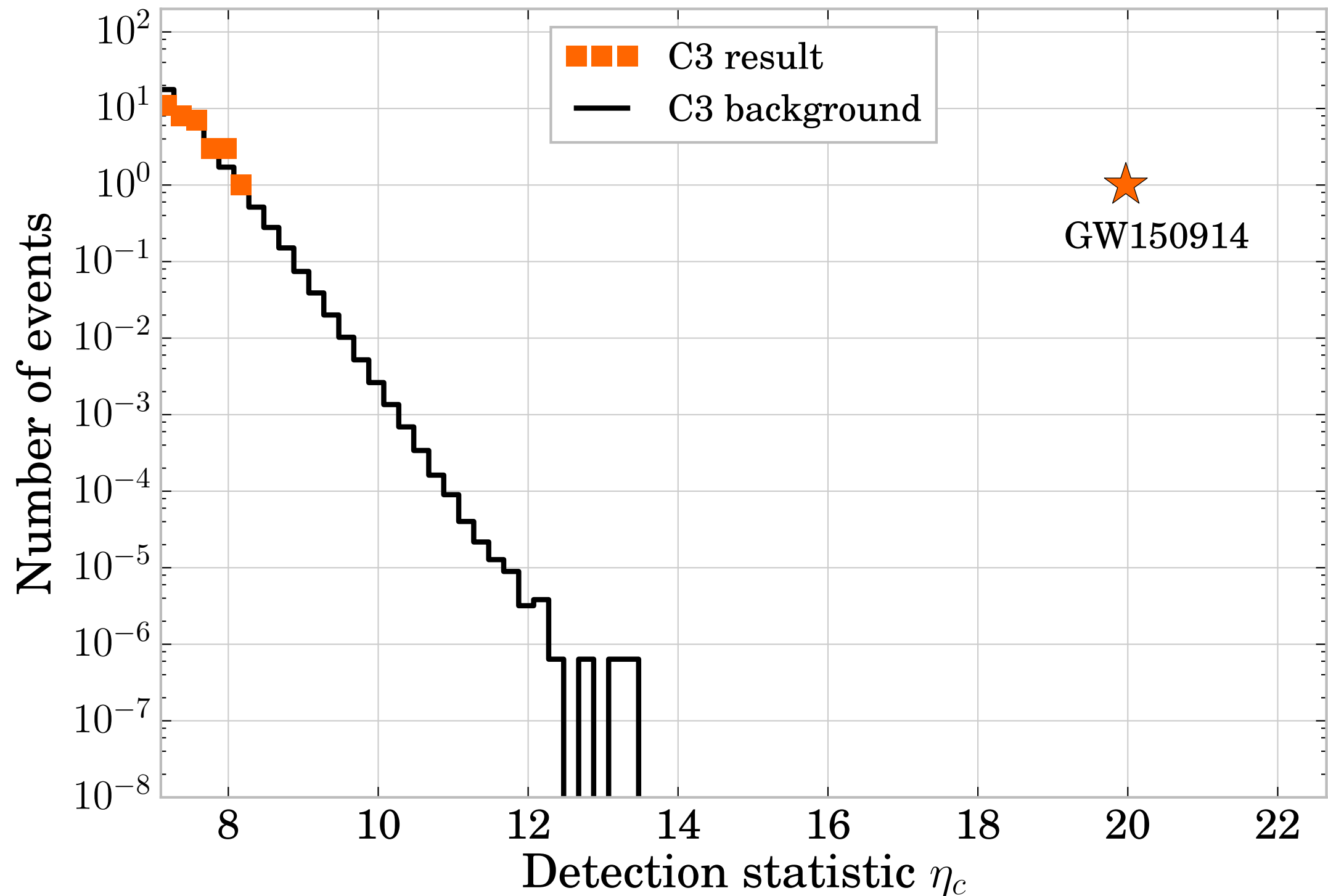
Unmodelled Searches: X-pipeline

- *X-pipeline* uses similar methods to CWB, but different implementation.

- Data is whitened and FFT'd at multiple resolutions. Data from different detectors is then summed to construct various energy measures.

- High energy (black) pixels are identified and clustered.

- Each event characterised by certain parameters - start time, peak time, duration, start frequency, peak frequency, bandwidth, number of pixels, energy measures, sky position, FFT length etc.

- Loop over sky positions and length of FFTs.
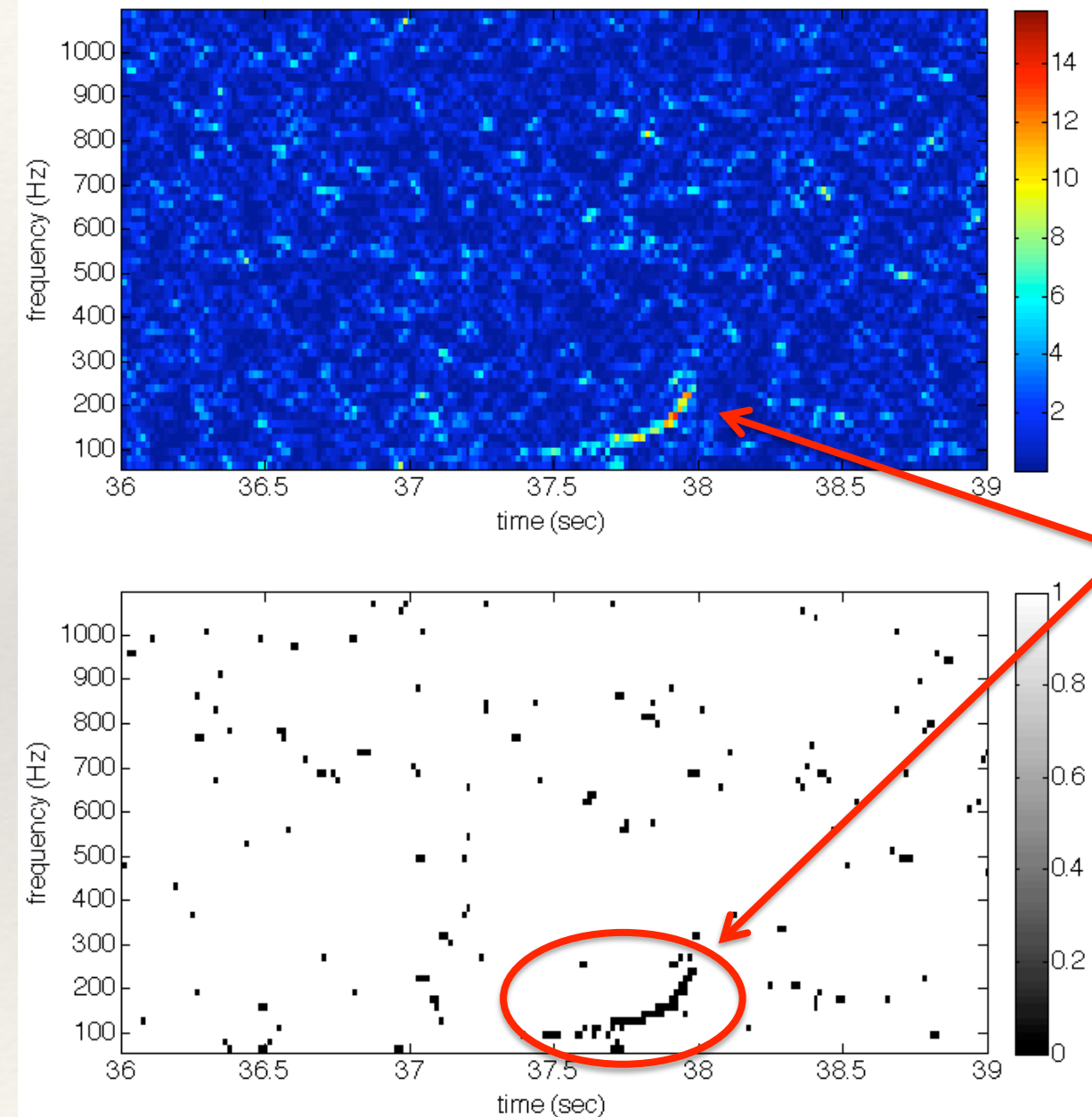
Unmodelled Searches: X-pipeline

- 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.
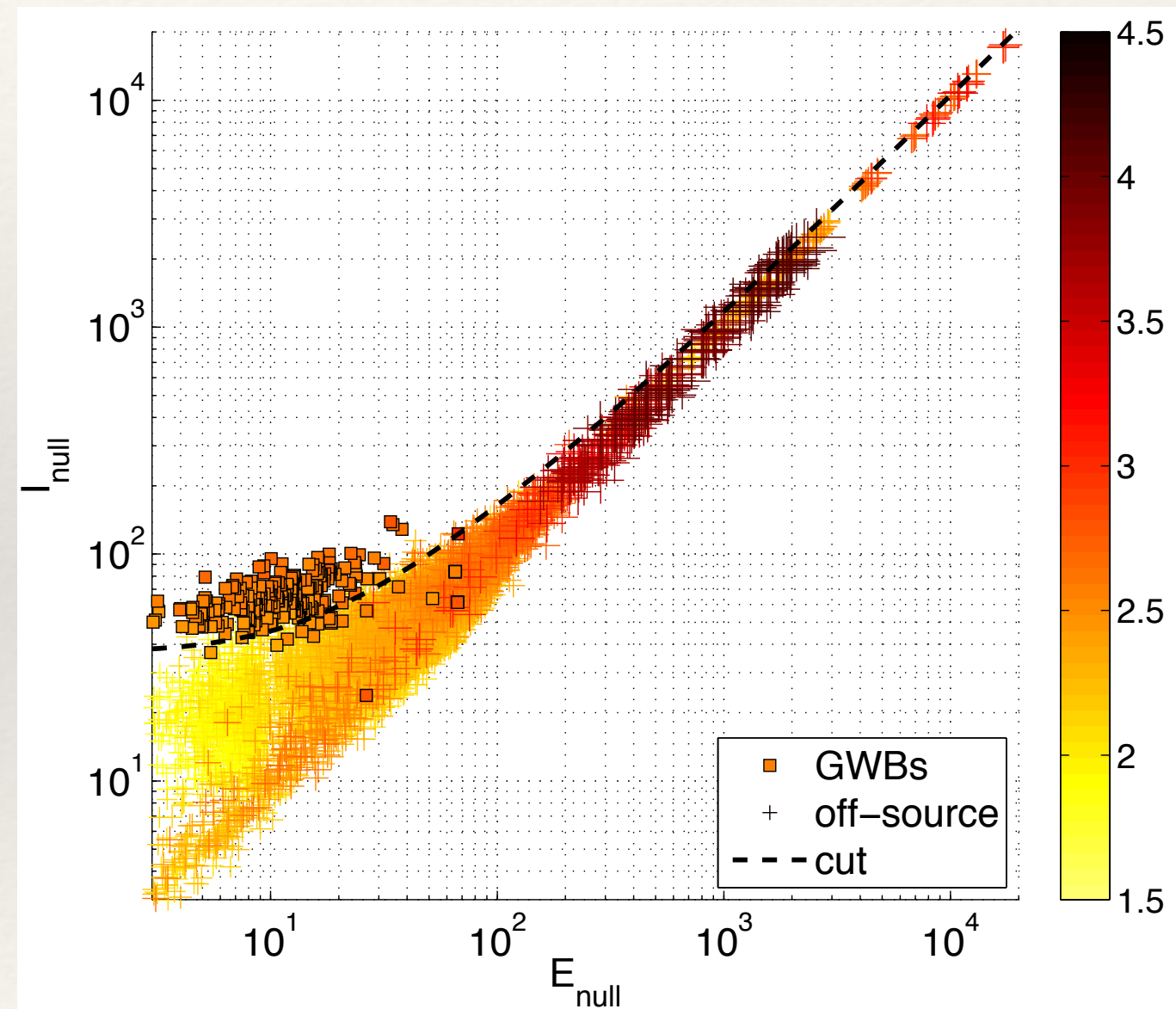
Unmodelled Searches: BayesWave

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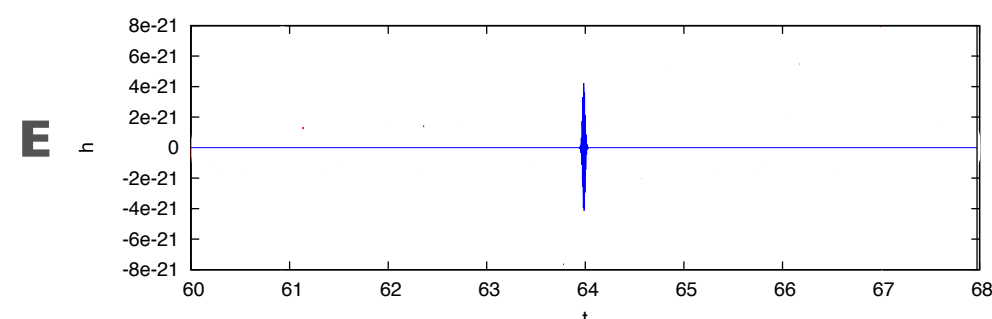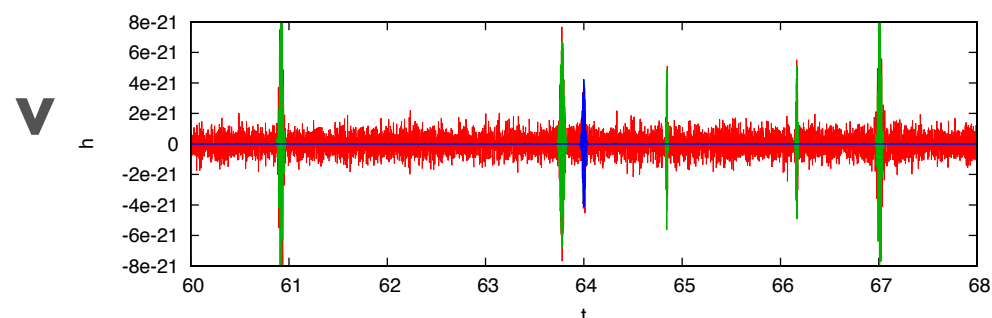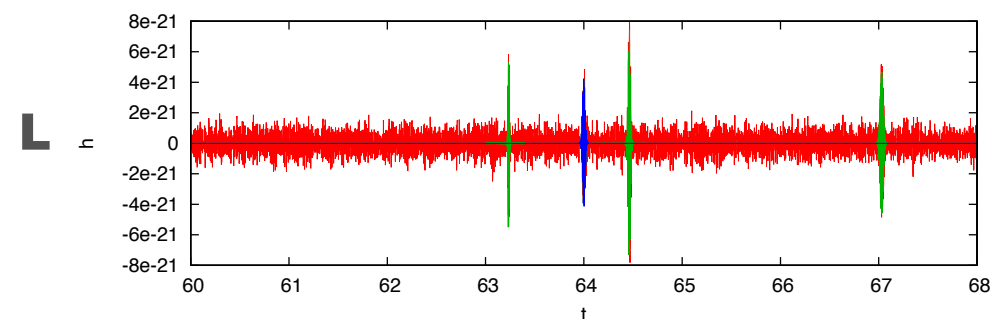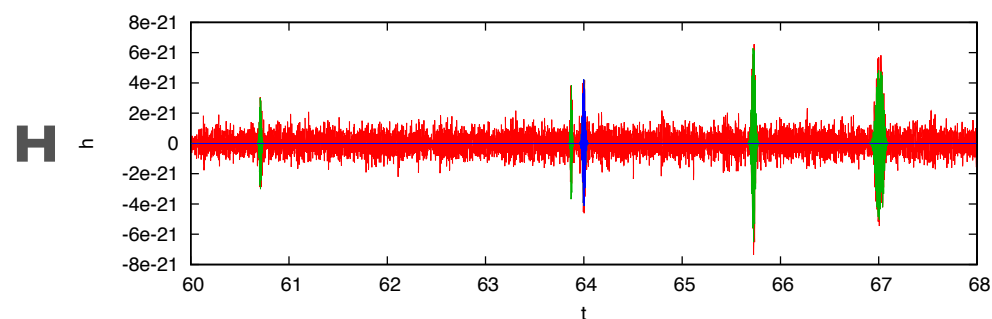
$$d^H = n^H + \sum_i^{N_G^H} \psi(\vec{\gamma}_i^H) + h^H(N_{\text{GW}}^\oplus, \vec{\gamma}^\oplus, \vec{\lambda}\,)$$

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

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

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

Extrinsic   Intrinsic

$$\{\vec{\Omega}, \delta\phi_\oplus, \delta t_\oplus, \delta\mathcal{A}, \psi, e\} \qquad \{t_0, f_0, \mathcal{A}, \text{etc.}\}$$

sky location, orientation, etc.   morphological params.

Unmodelled Searches: BayesWave
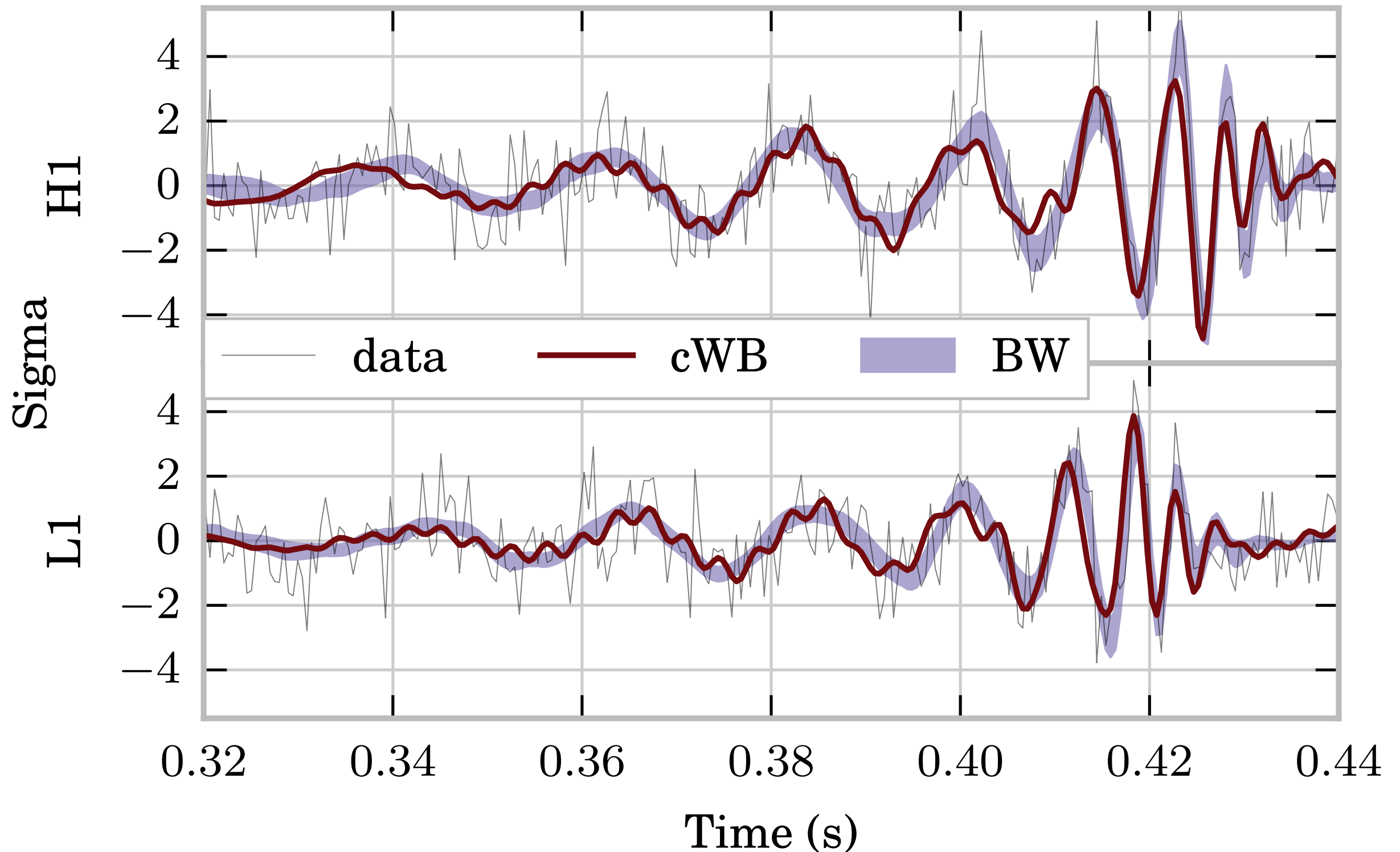
Unmodelled Searches: LISA

❖ Time-frequency methods were also applied for EMRI detection for LISA.

– Search for tracks in time-frequency spectrogram of data.

– Three algorithms tried - Excess Power, HACR, CATS. Estimate detection threshold at ~2Gpc. Good parameter recovery in MLDC, but likely to fail when presented with weak or confused sources.
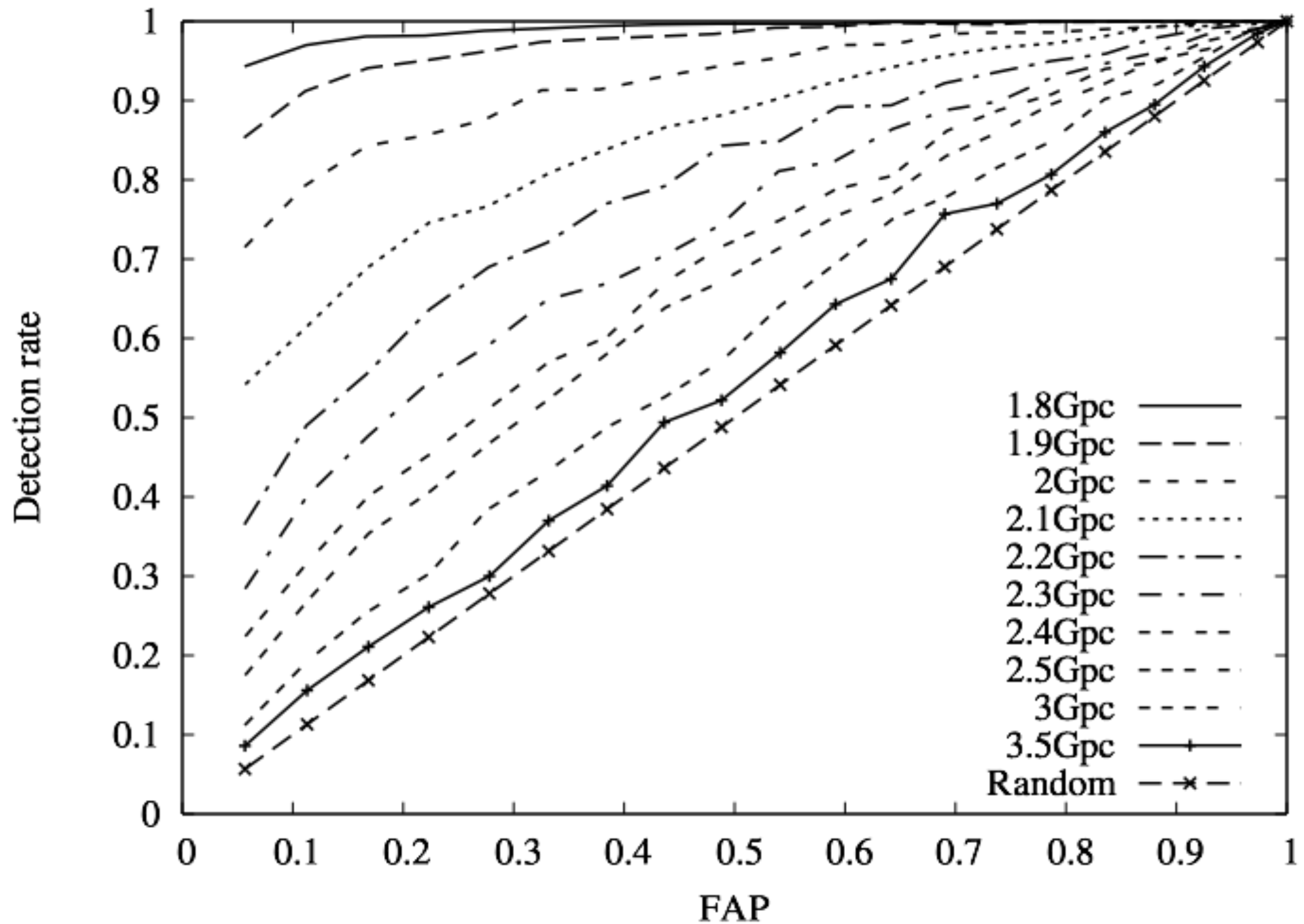
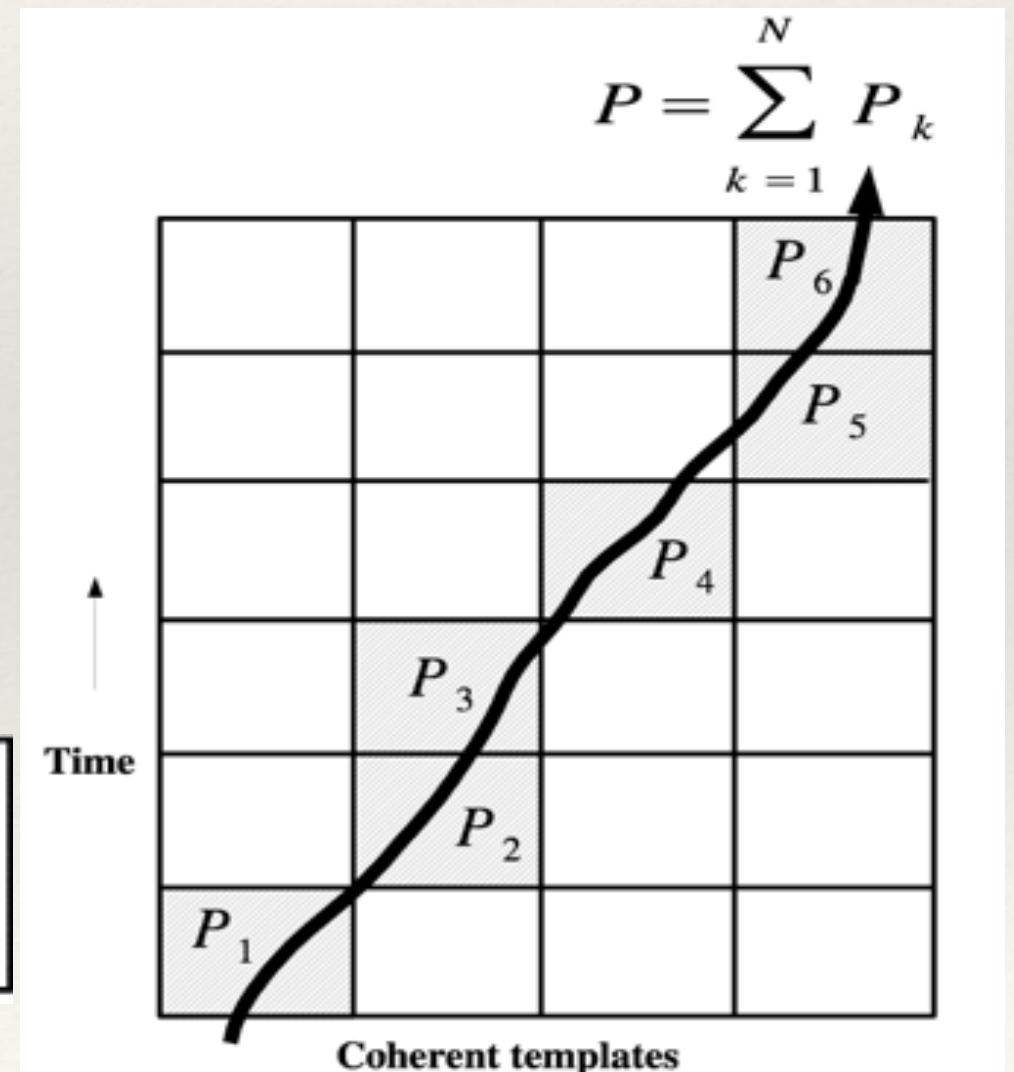# Unmodelled Searches: LISA
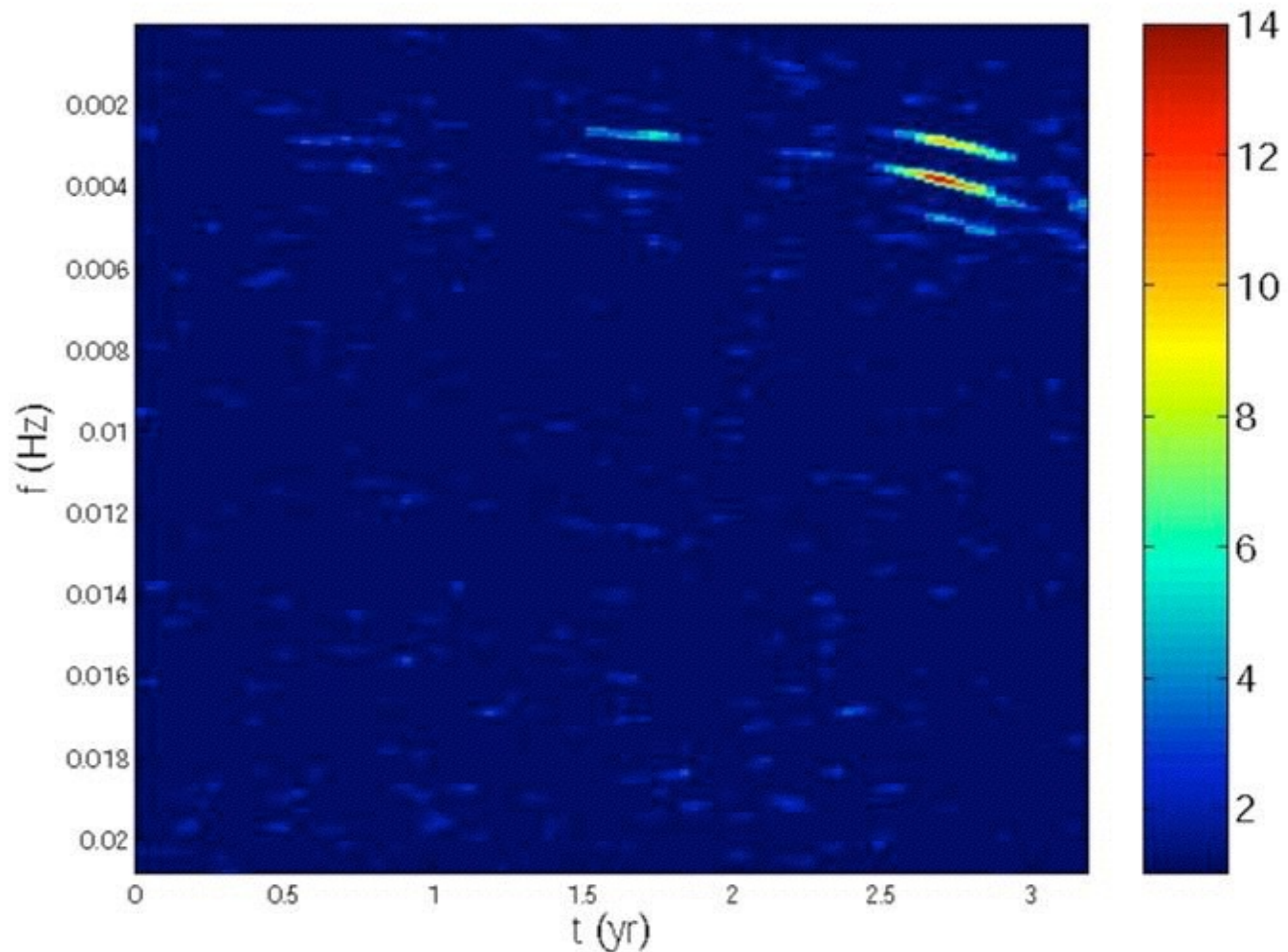
Unmodelled Searches: LISA

Searching for signals: Semi-coherent methods

Semi-coherent searches: EMRIs

❖ First stage is coherent matched filtering of shorter (~2 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 $\langle a, b \rangle = 4\,\Re\left[\int_0^\infty \frac{\tilde{a}^*(f)\,\tilde{b}(f)}{S_b(f)}\,\mathrm{d}f\right]$

$$P = \sum_{k=1}^{N} P_k$$

Time

Coherent templates

Semi-coherent searches: EMRIs

❖ First stage is coherent matched filtering of shorter (~2 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.
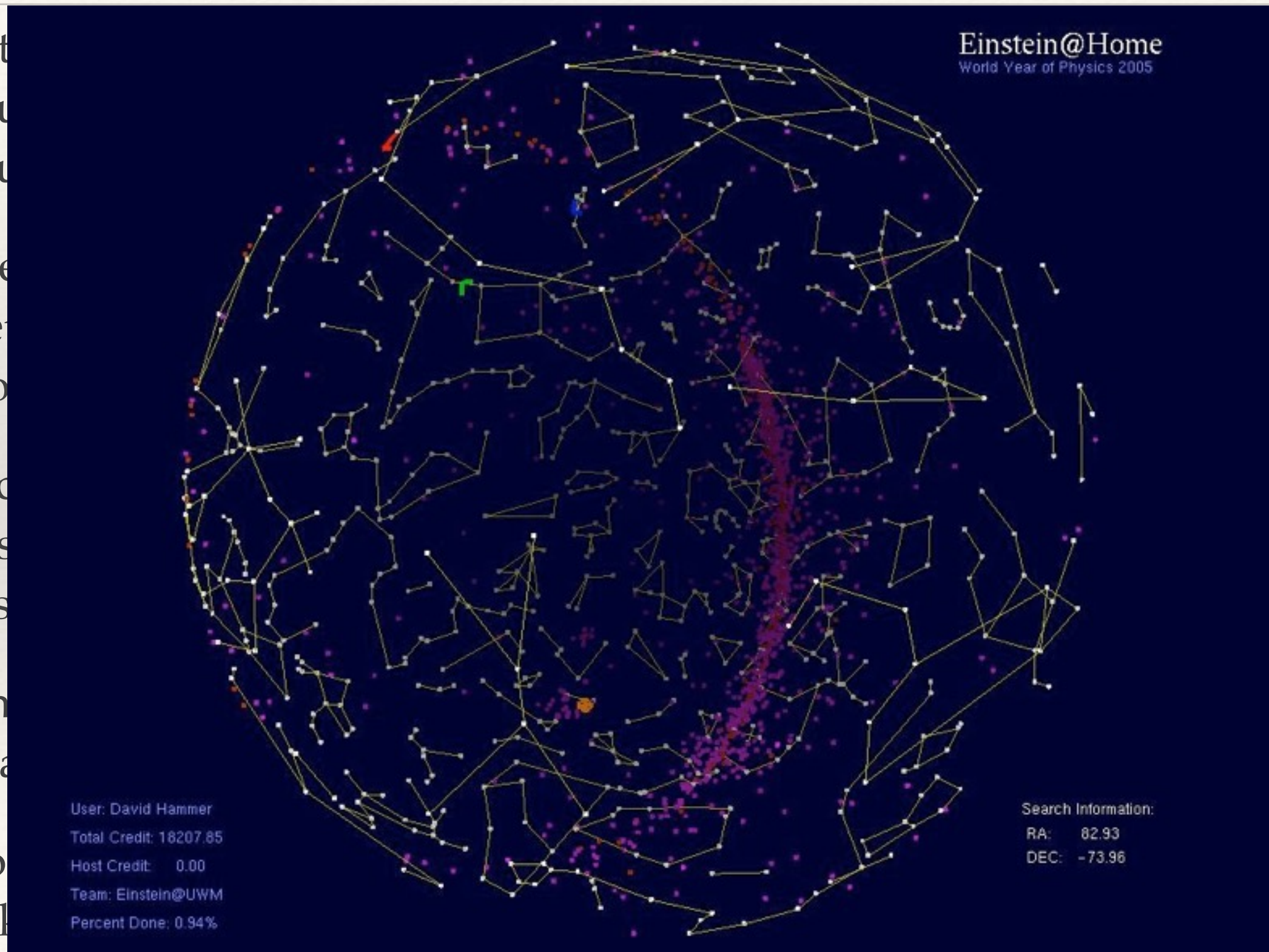
Semi-coherent searches: pulsars

❖ LIGO unknown pulsar search also uses semi-coherent techniques.

❖ *Stack-Slide* algorithm is very similar to EMRI algorithm described above.

❖ *Hough Transform* 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.

Einstein@Home
World Year of Physics 2005

User: David Hammer
Total Credit: 18207.85
Host Credit: 0.00
Team: Einstein@UWM
Percent Done: 0.94%

Search Information:
RA: 82.93
DEC: −73.96

❖ In t   pu
   hou   arch.
   You

❖ The
   Ne   e
   Wo

❖ Eac   r sky
   pos
   ens

❖ Ein   nd a
   tota

❖ No   d
   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.

$$Y_Q = \int_0^T \mathrm{d}t_1 \int_0^T \mathrm{d}t_2 \, h_1(t_1) Q(t_1 - t_2) h_2(t_2)$$

$$= \int_{-\infty}^{\infty} \mathrm{d}f \int_{-\infty}^{\infty} \mathrm{d}f' \delta_T(f - f') \tilde{h}_1^*(f) Q(f') \tilde{h}_2(f')$$

Stochastic Gravitational Wave Fore/Backgrounds

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

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

❖ 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

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

❖ 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

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

Stochastic Gravitational Wave Fore/Backgrounds

- ❖ In the ...ation to the Di...

- ❖ and $G_...$ ...letectors is und... ross-correl...

- ❖ The fu... ...res the loss o... of the two d...

Overlap Reduction Function
(LIGO-LA and other detectors)

Legend:
- LIGO-WA
- ALLEGRO (optimal alignment)
- GEO-600
- TAMA-300
- VIRGO

Frequency (Hz)
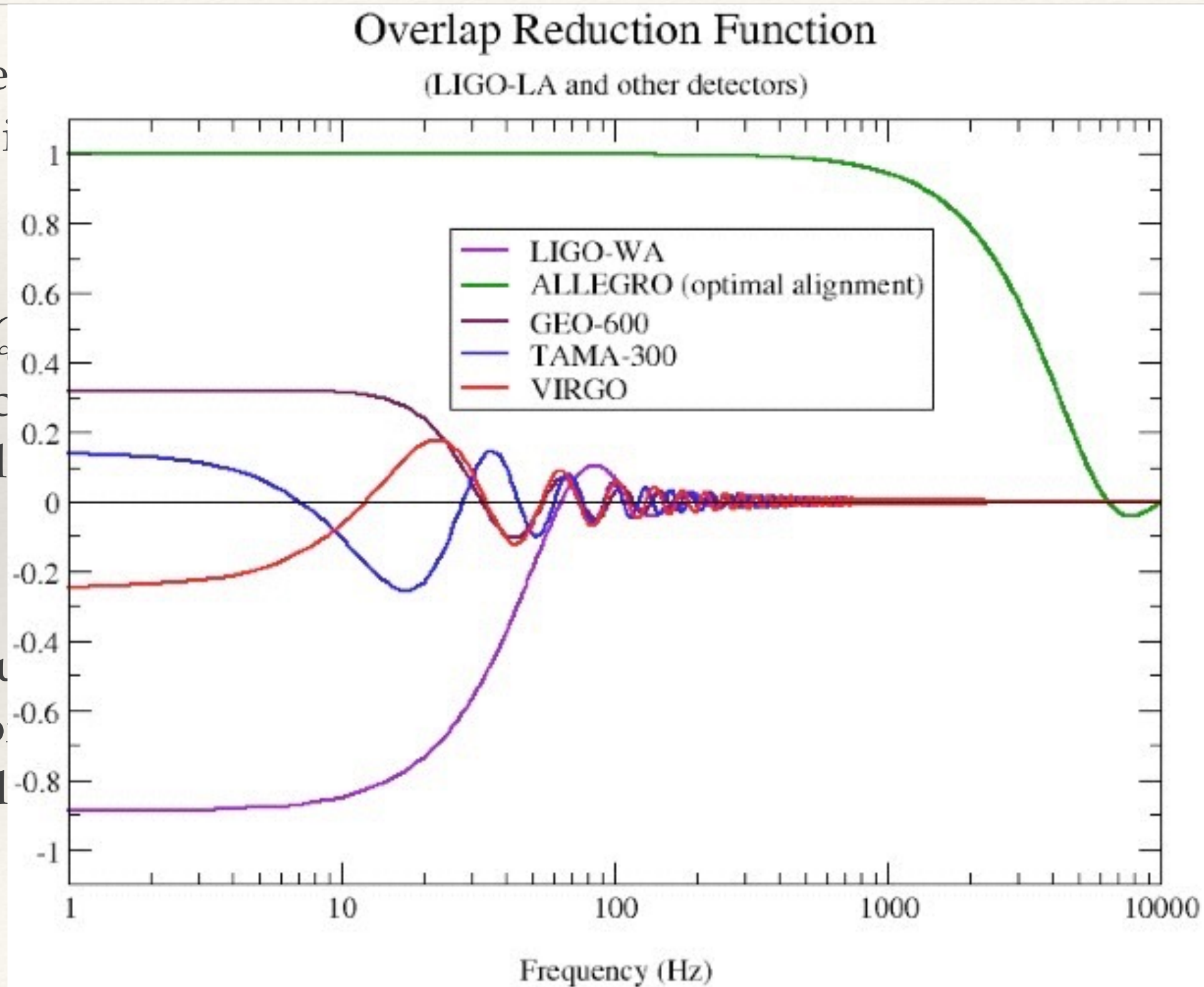
Stochastic Gravitational Wave Fore/Backgrounds

❖ For pulsar timing, the overlap reduction function for an isotropic background is the Hellings and Downs curve.
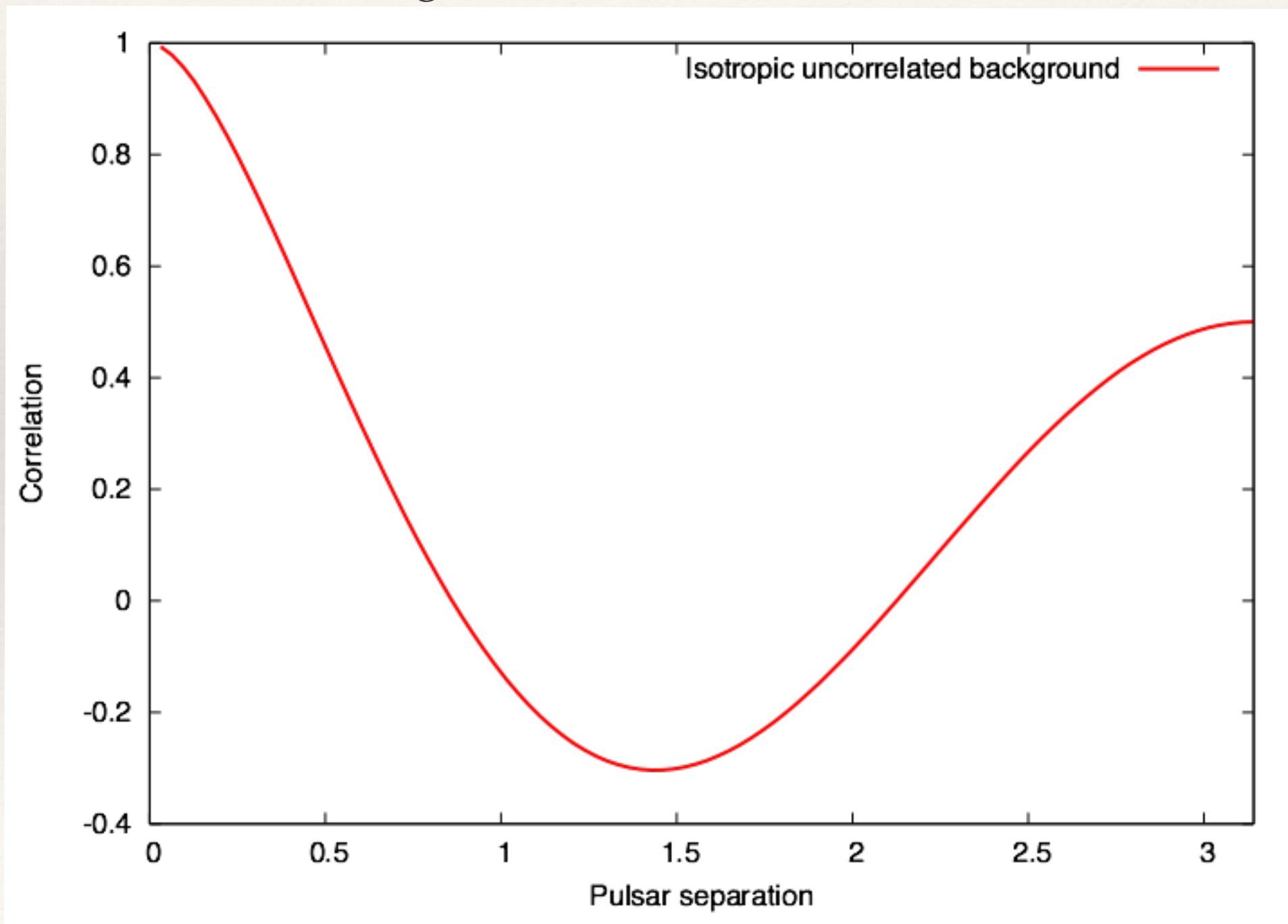
Stochastic Gravitational Wave Fore/Backgrounds

- ❖ Uncorrelated anisotropic and correlated backgrounds have different correlation functions.
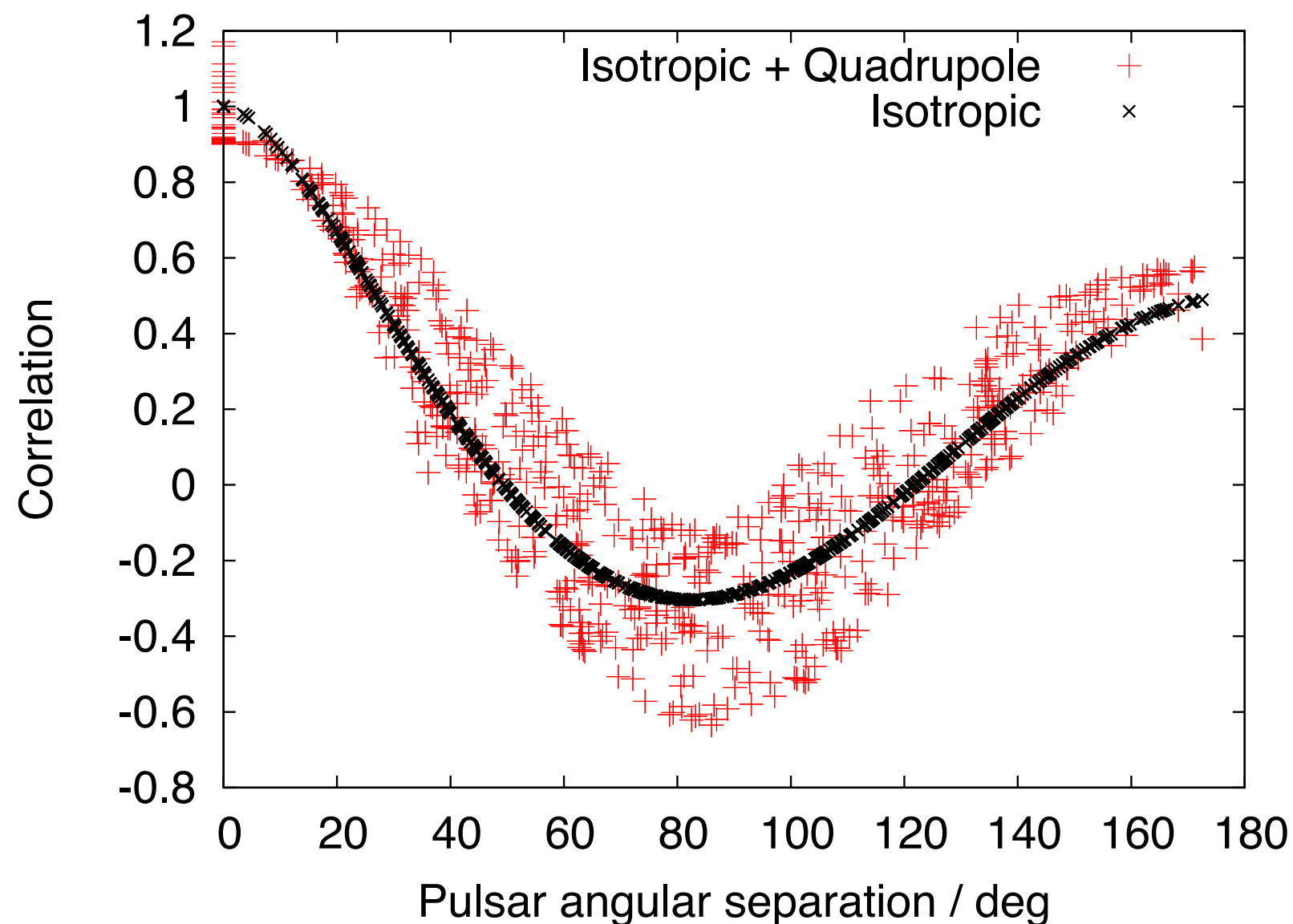
Correlation function for uncorrelated quadrupole background

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
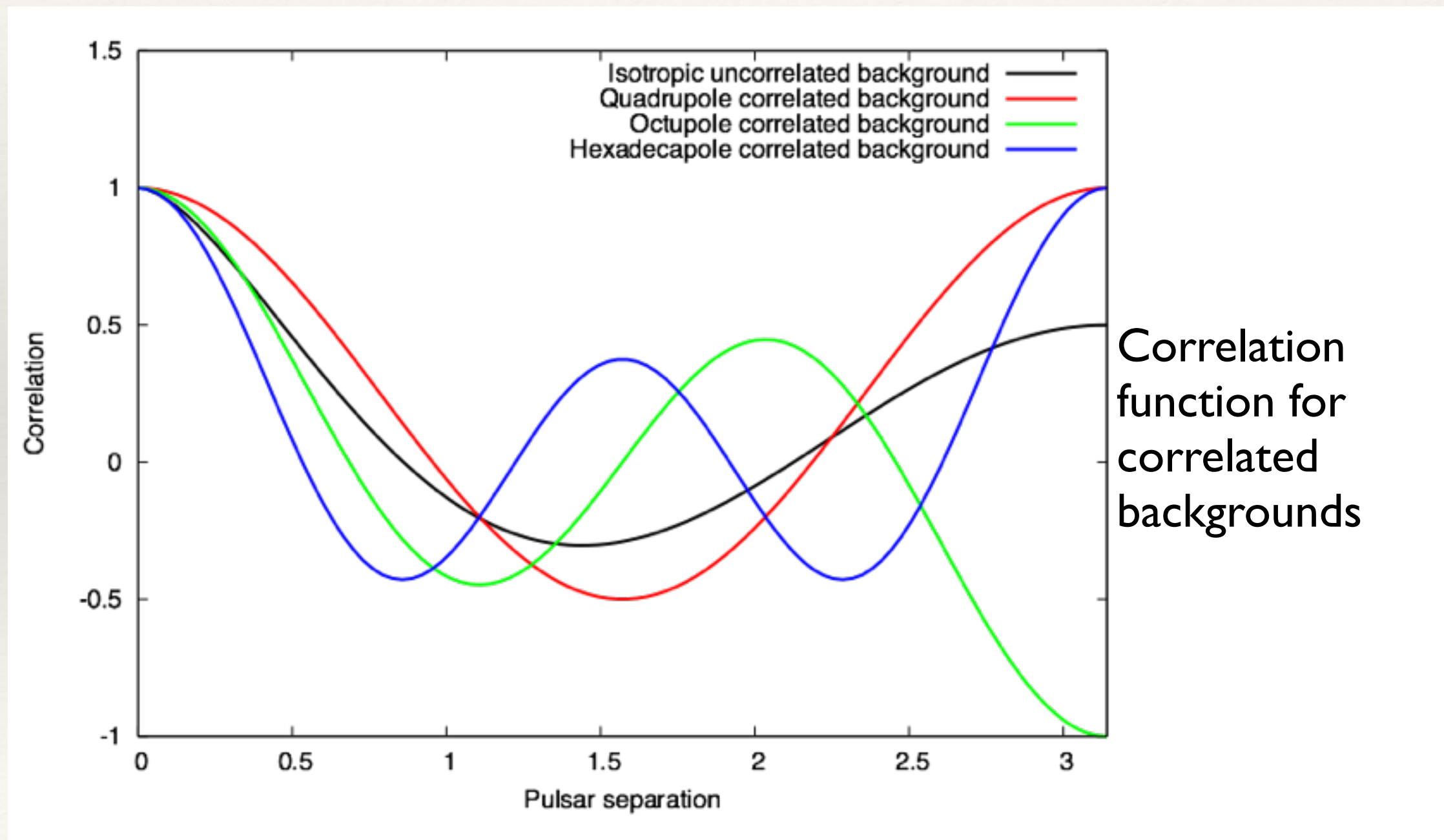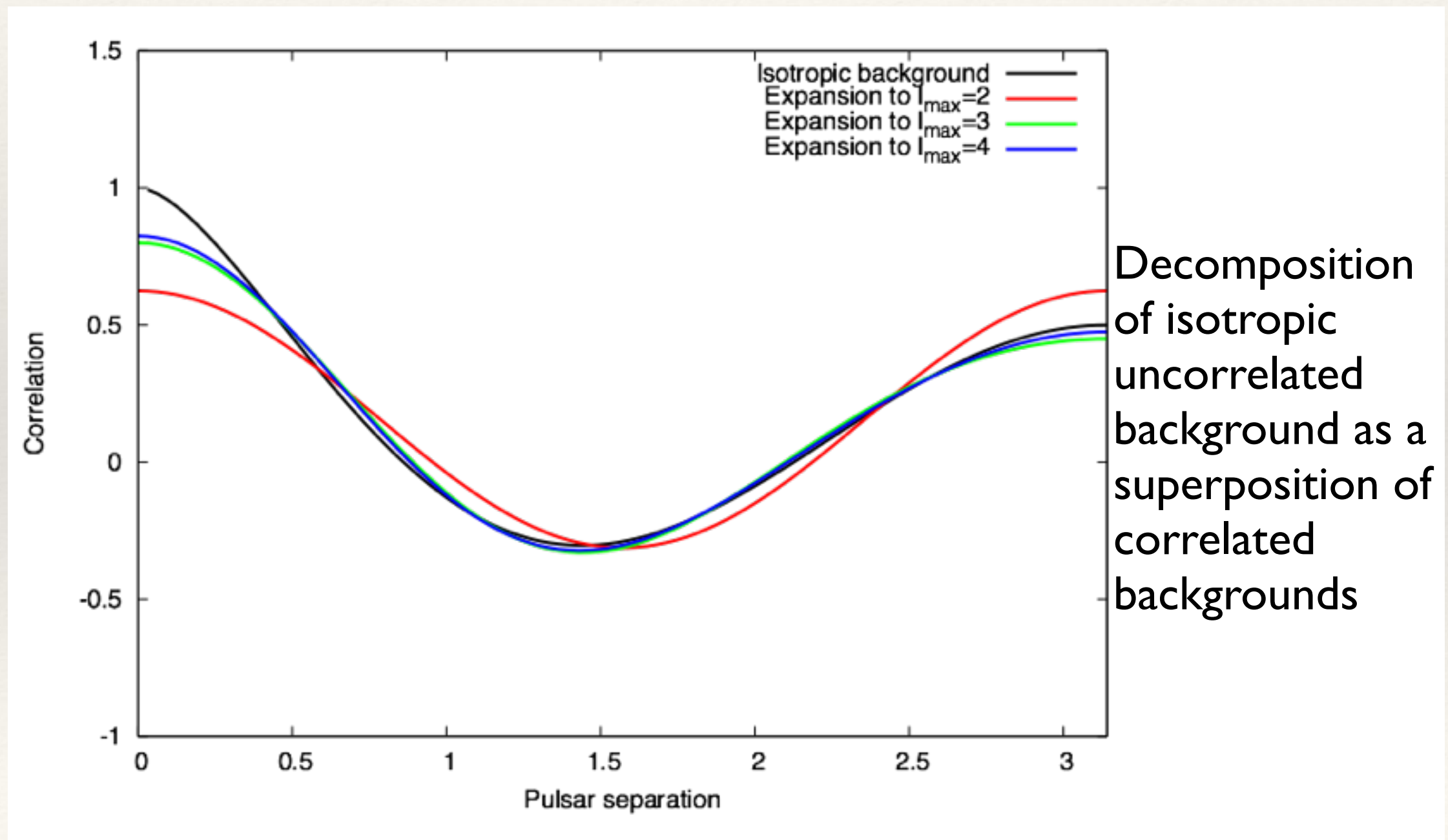
Stochastic Gravitational Wave Fore/Backgrounds

- Best published result from a LIGO/ALLEGRO cross-correlation search in the frequency range 850-950Hz was a 90% confidence limit of

$$h_{100}^2 \Omega_{\mathrm{w}}(f) \leq 0.53$$

- This corresponds to a GW strain at 915Hz of $1.5 \times 10^{-23} \mathrm{Hz}^{-1/2}$.

- Can also search for anisotropic and non-Gaussian backgrounds

  - *Popcorn noise.* If the duration of a background source is short compared to the average time between events, the Central Limit Theorem can no longer guarantee the noise is Gaussian.

  - *Periodic noise.* If the sources are anisotropic on the sky, e.g., a dipole anisotropy, a galactic disk population or a galactic halo population, we expect to see the background amplitude to vary harmonically with the detector's motion.

- In each case, can still use cross-correlation, but optimal filter now depends on type of noise source.

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.
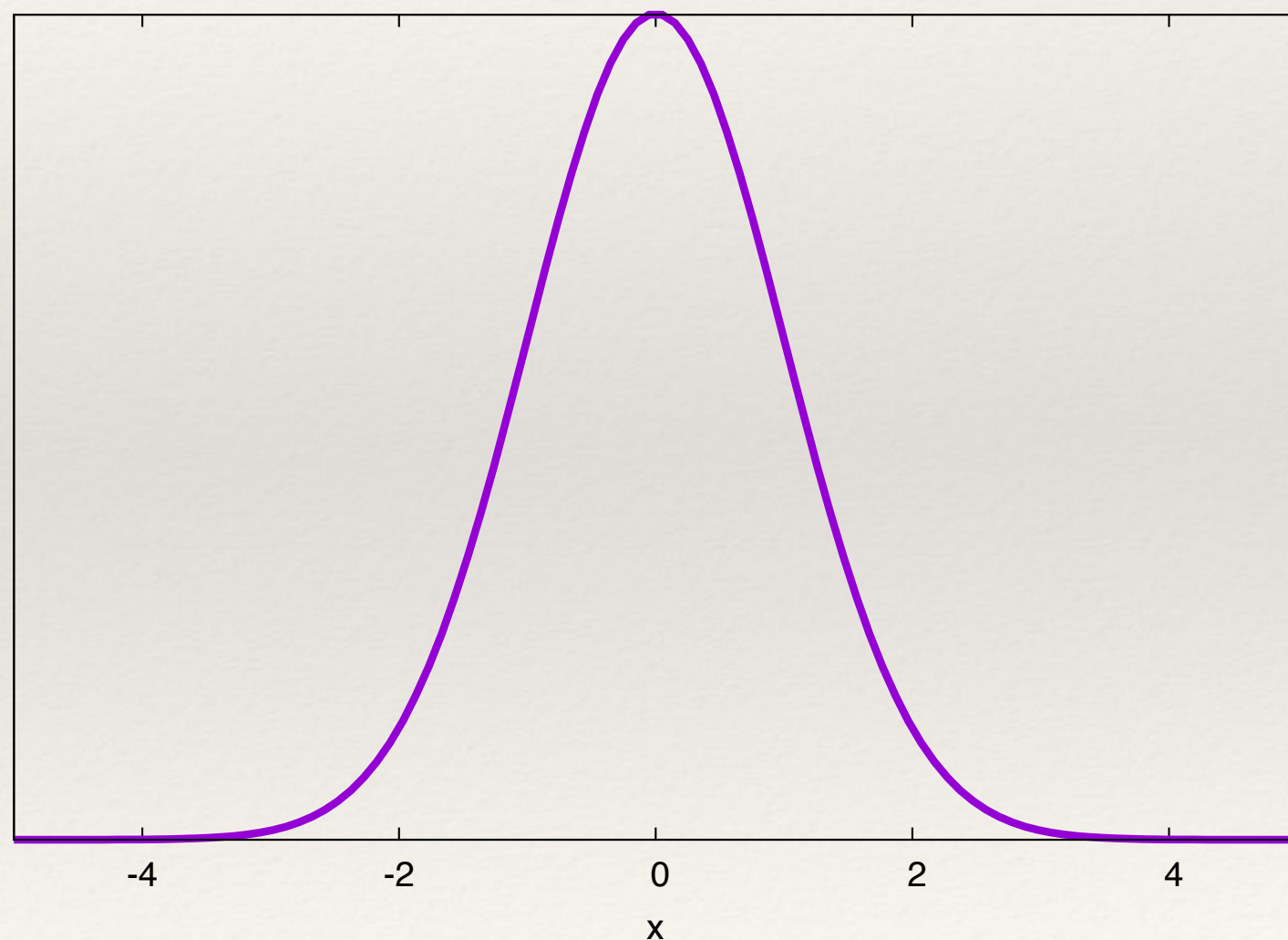
    ❖ **too wide:** low acceptance rate;

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.
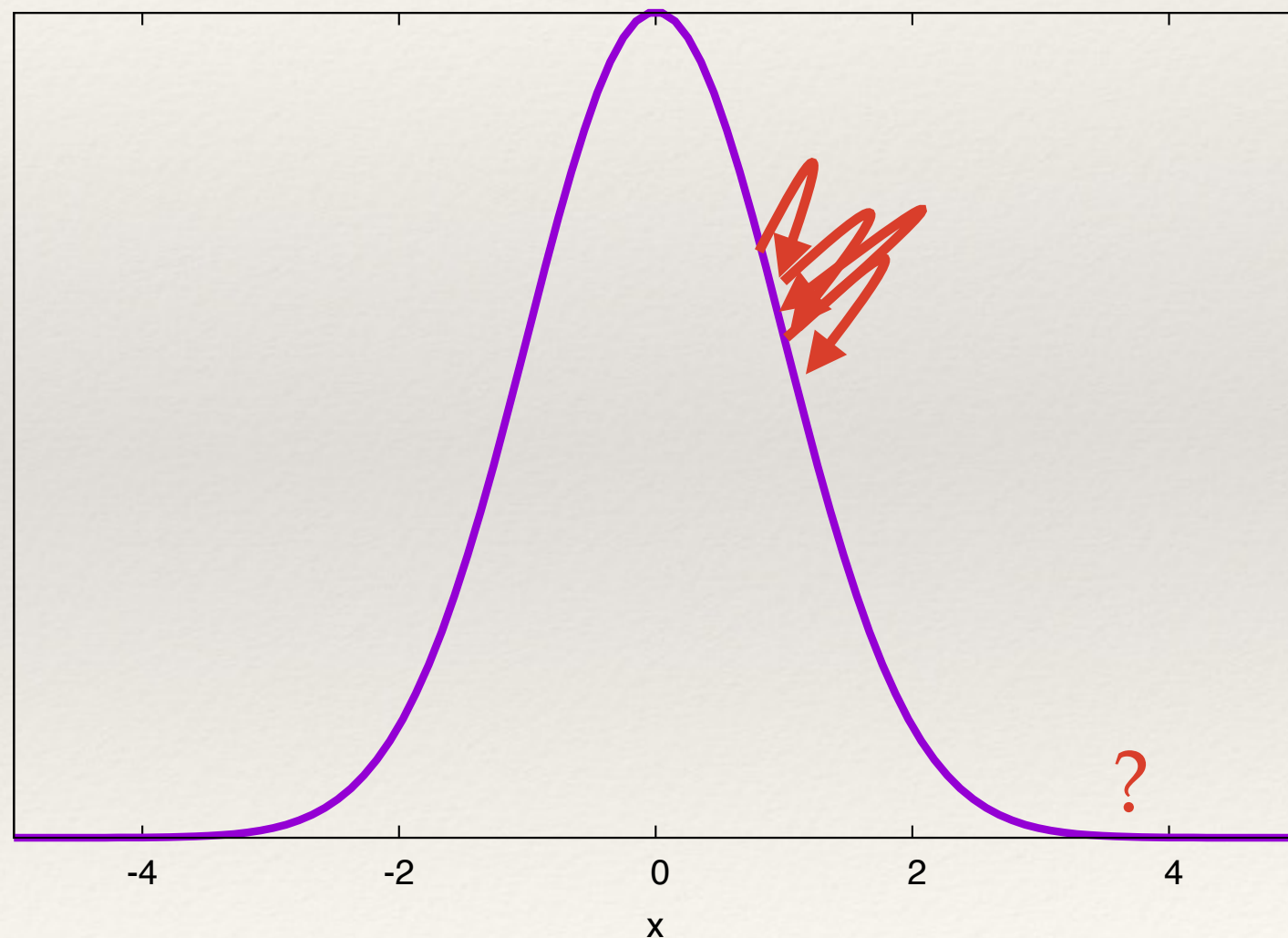
Annealing

❖ 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.
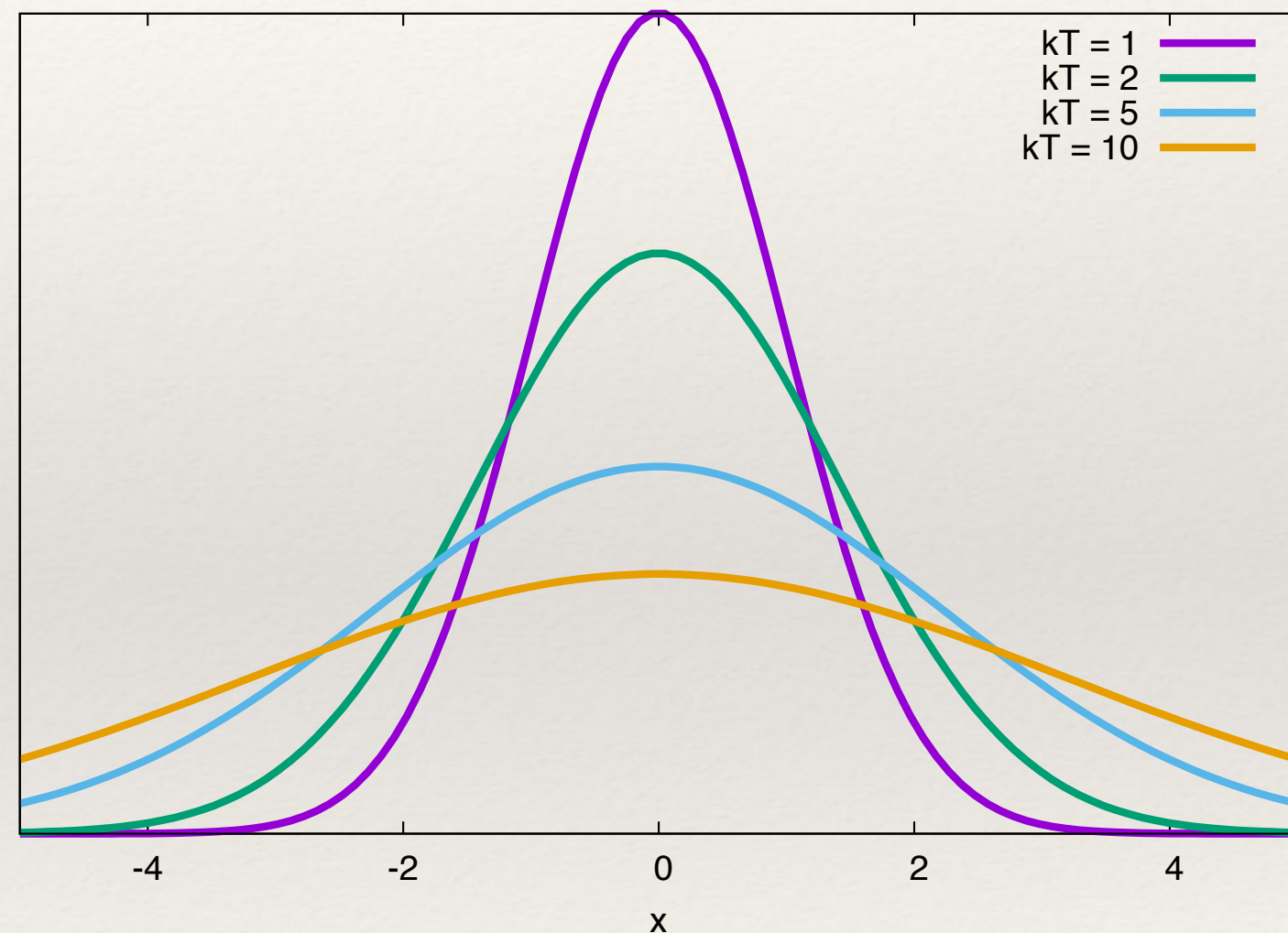
Annealing

❖ 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.

Chain values of m

True value = red line
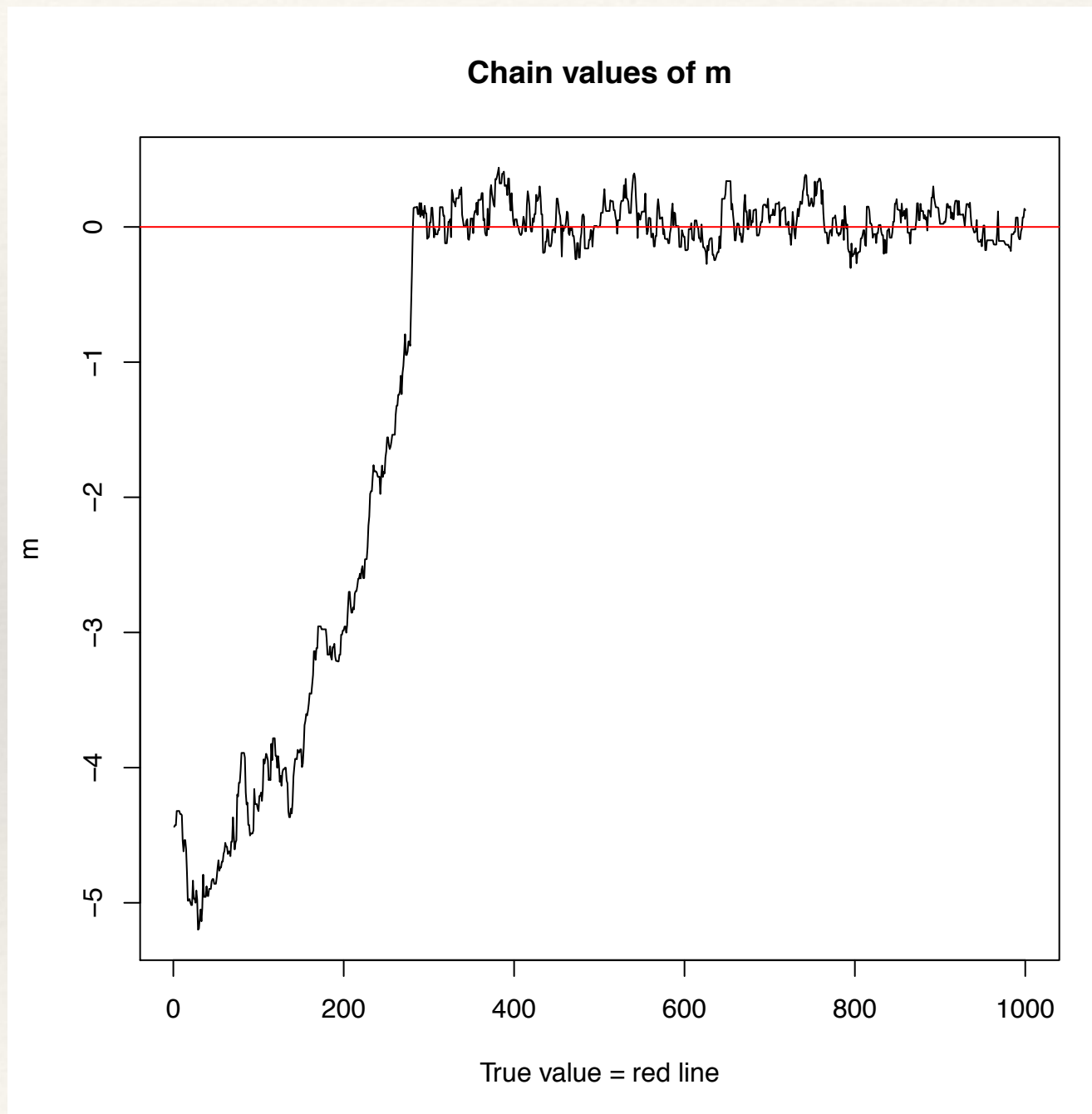
Autocorrelation and Effective sample size

❖ 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.

Gelman-Rubin convergence diagnostic

- Run m (at least 2) chains and discard first half of samples from each.

- Calculate the within-chain variance

$$W = \frac{1}{m} \sum_{j=1}^{m} \frac{1}{N-1} \sum_{i=1}^{N} (x_{ij} - \bar{x}_j)^2$$

- Calculate the between-chain variance

$$B = \frac{N}{m-1} \sum_{j=1}^{m} (\bar{x}_j - \bar{\bar{x}})^2, \qquad \bar{\bar{x}} = \frac{1}{m} \sum_{j=1}^{m} \bar{x}_j$$

- Calculate the estimated variance of a given parameter

$$\mathrm{Var}(x) = \left(1 - \frac{1}{N}\right) W + \frac{1}{N} B$$

- Calculate the potential scale-reduction factor

$$\hat{R} = \sqrt{\frac{\hat{\mathrm{Var}}(x)}{W}}$$

- If R is greater than ~1.1 or 1.2, need to run chains for longer.

Convergence diagnostics: GW150914

Examples of Parameter Posteriors
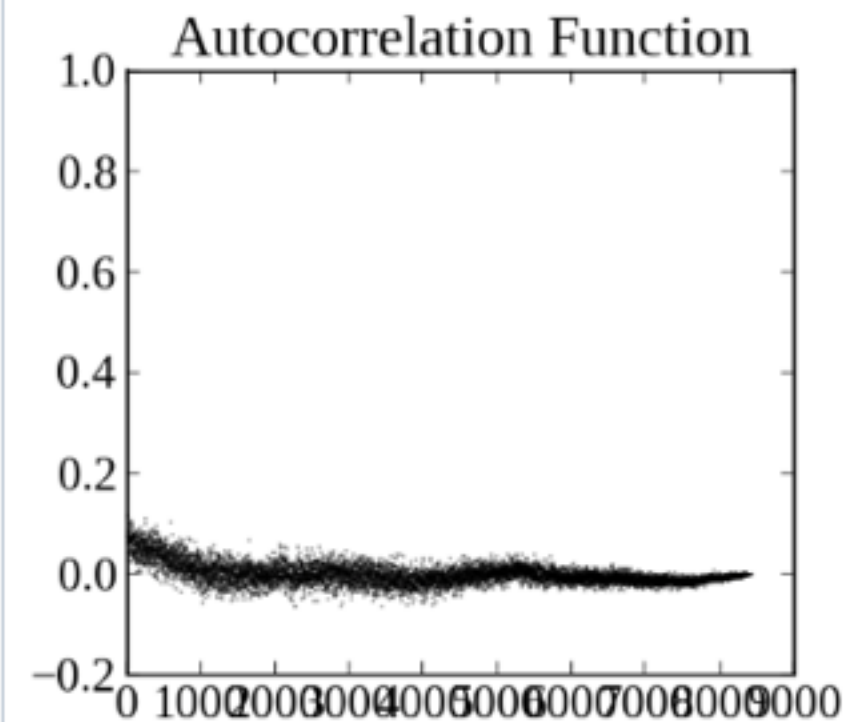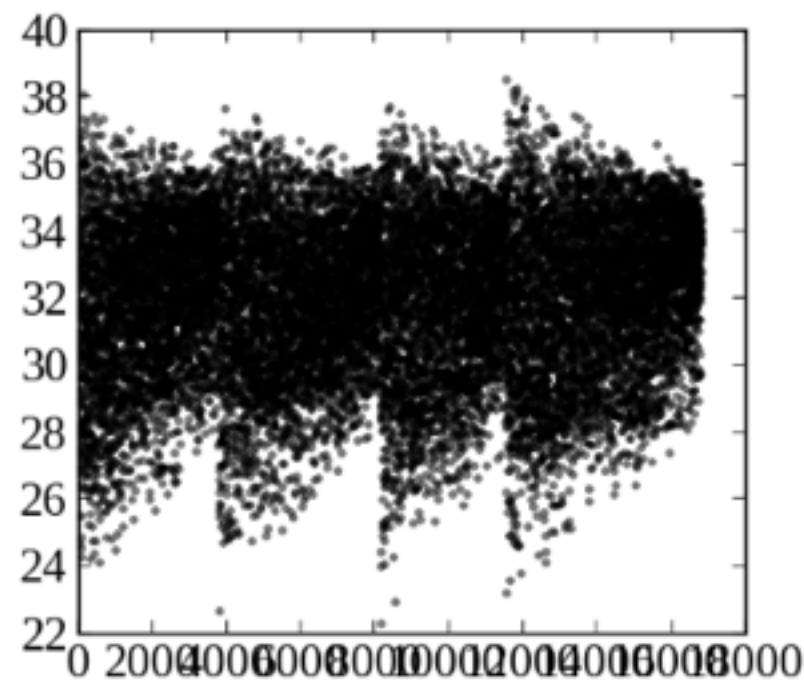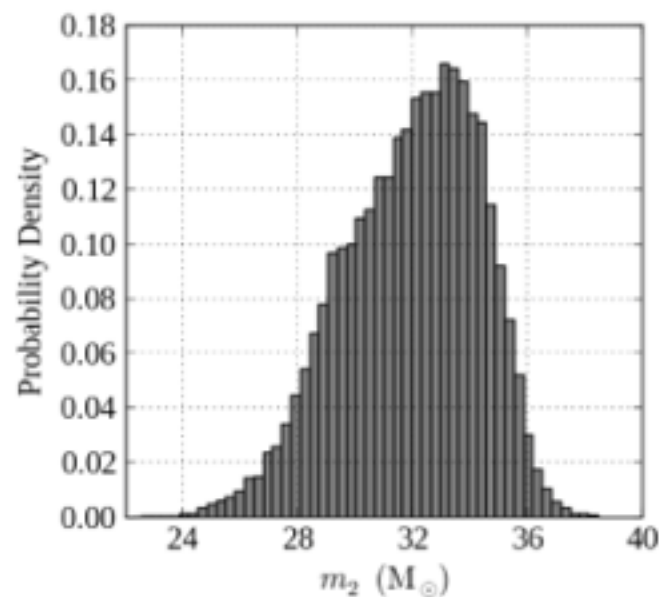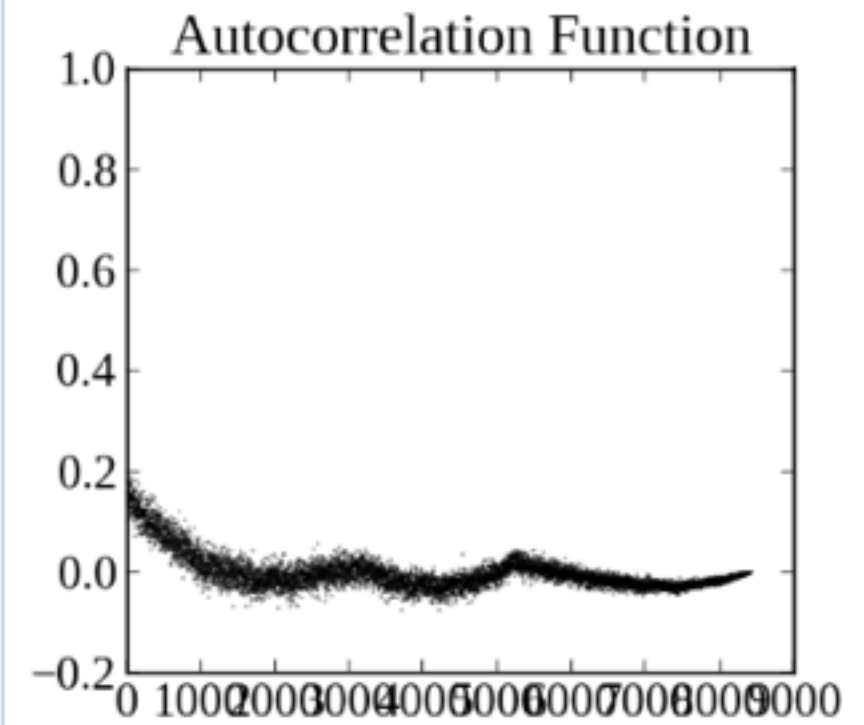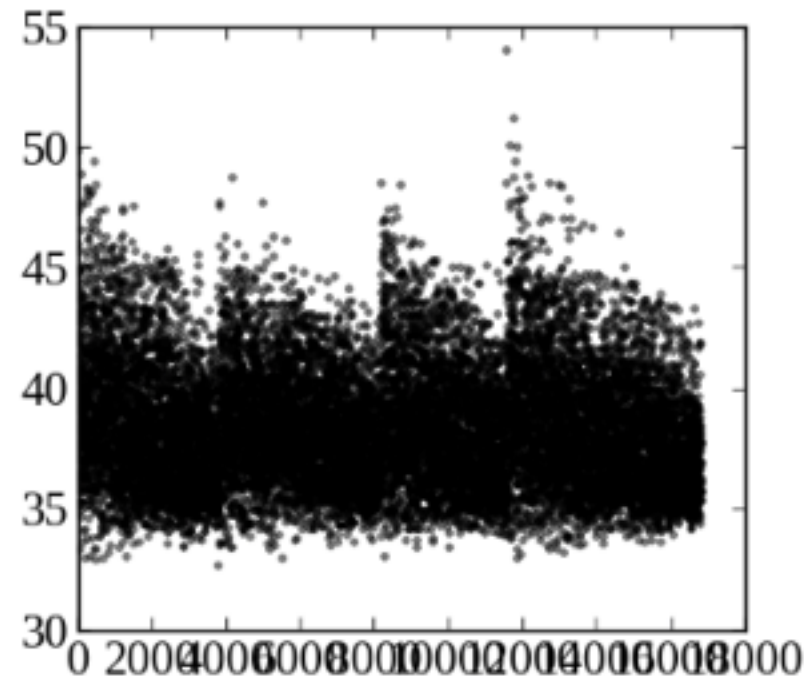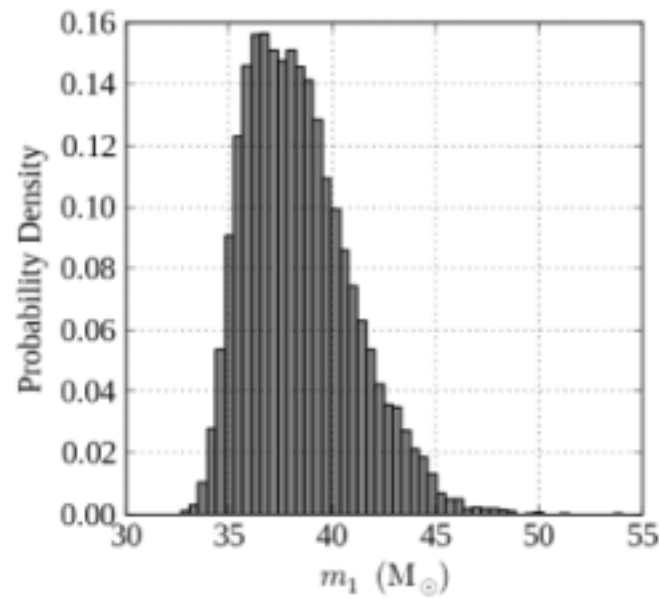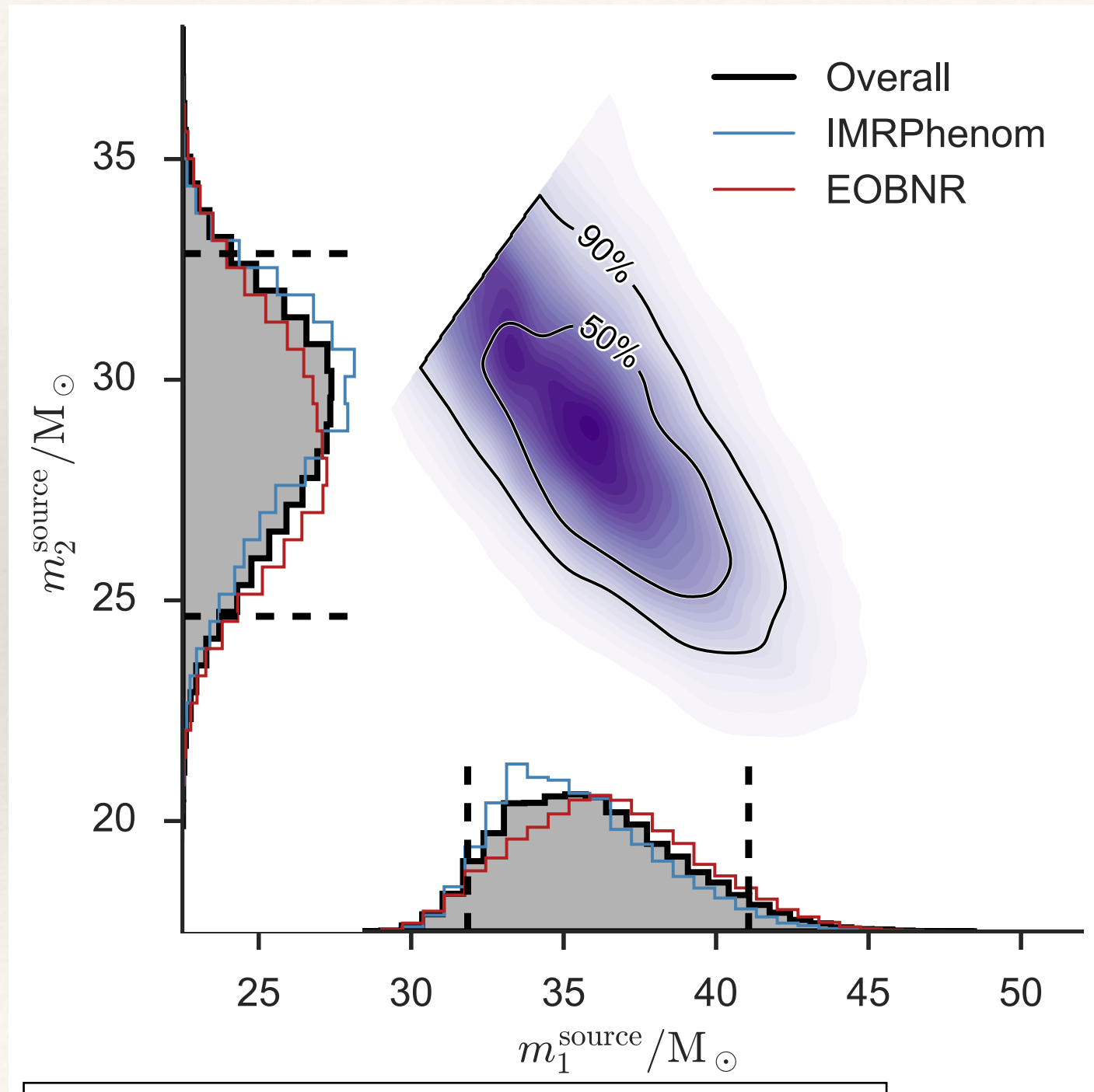
Examples of Parameter Posteriors
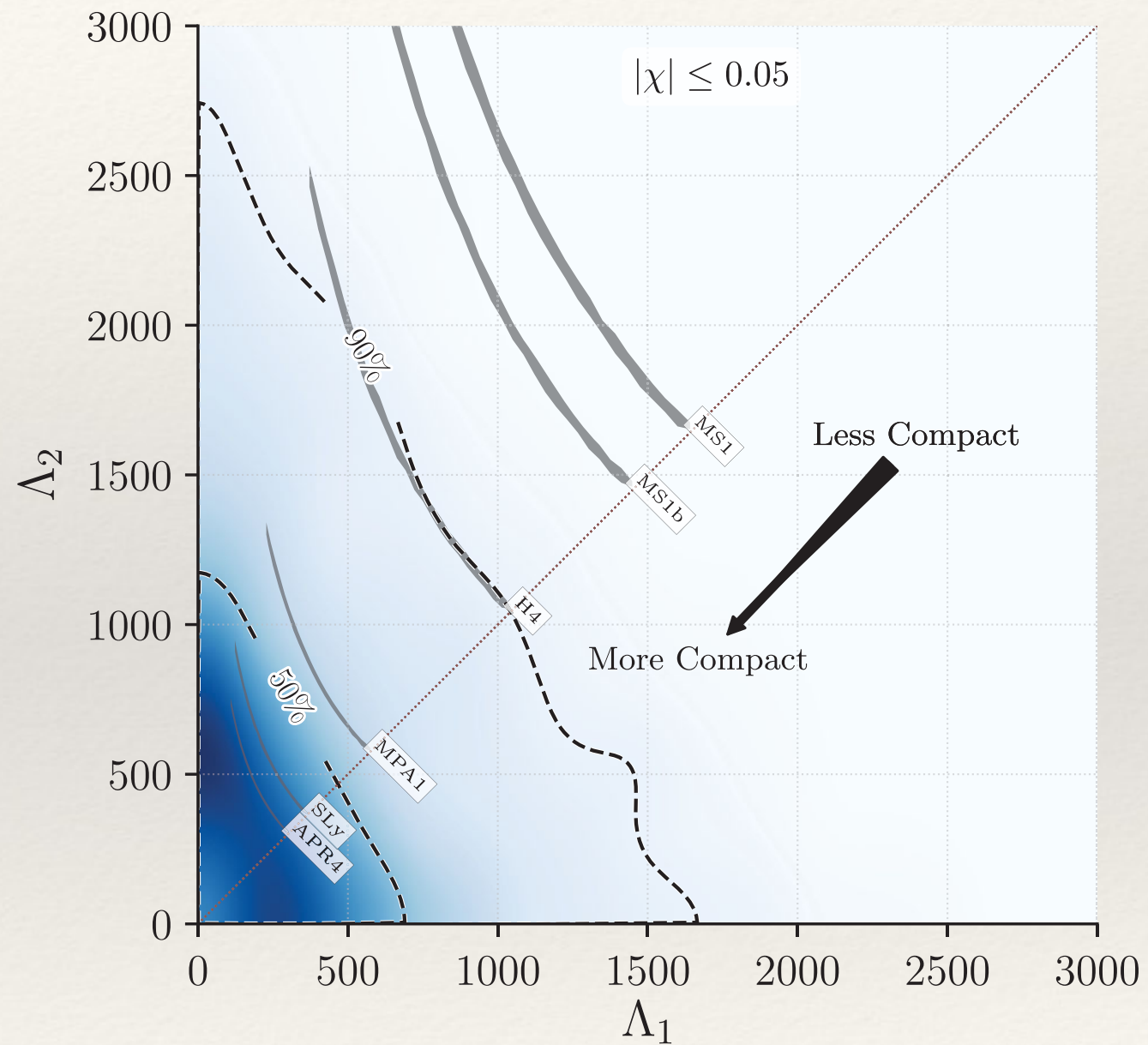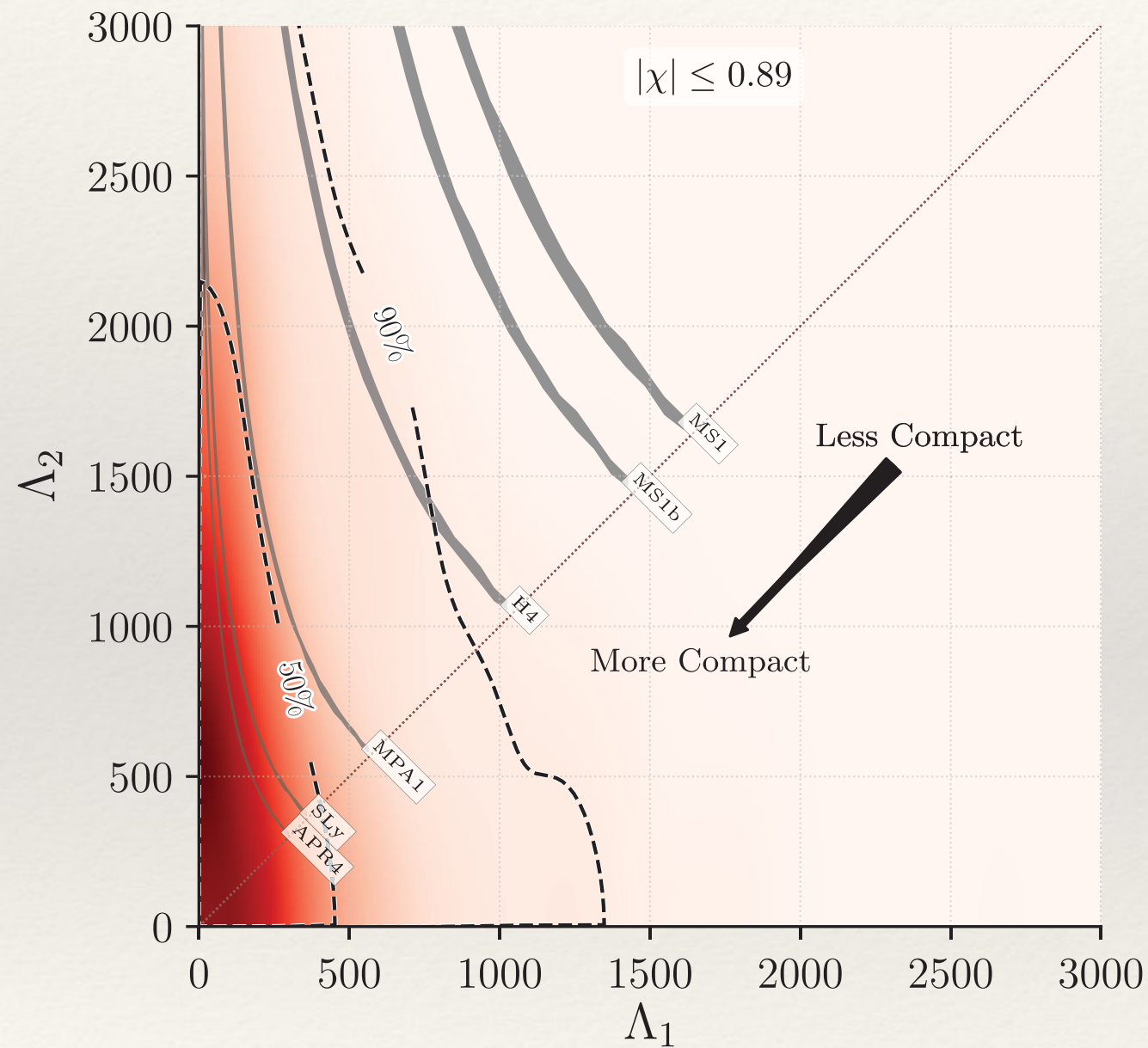
LVC, *Phys. Rev. Lett.* **119** 161101 (2017)

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., $p(x,y,z) = p(x|z) \, p(y|z) \, p(z)$.
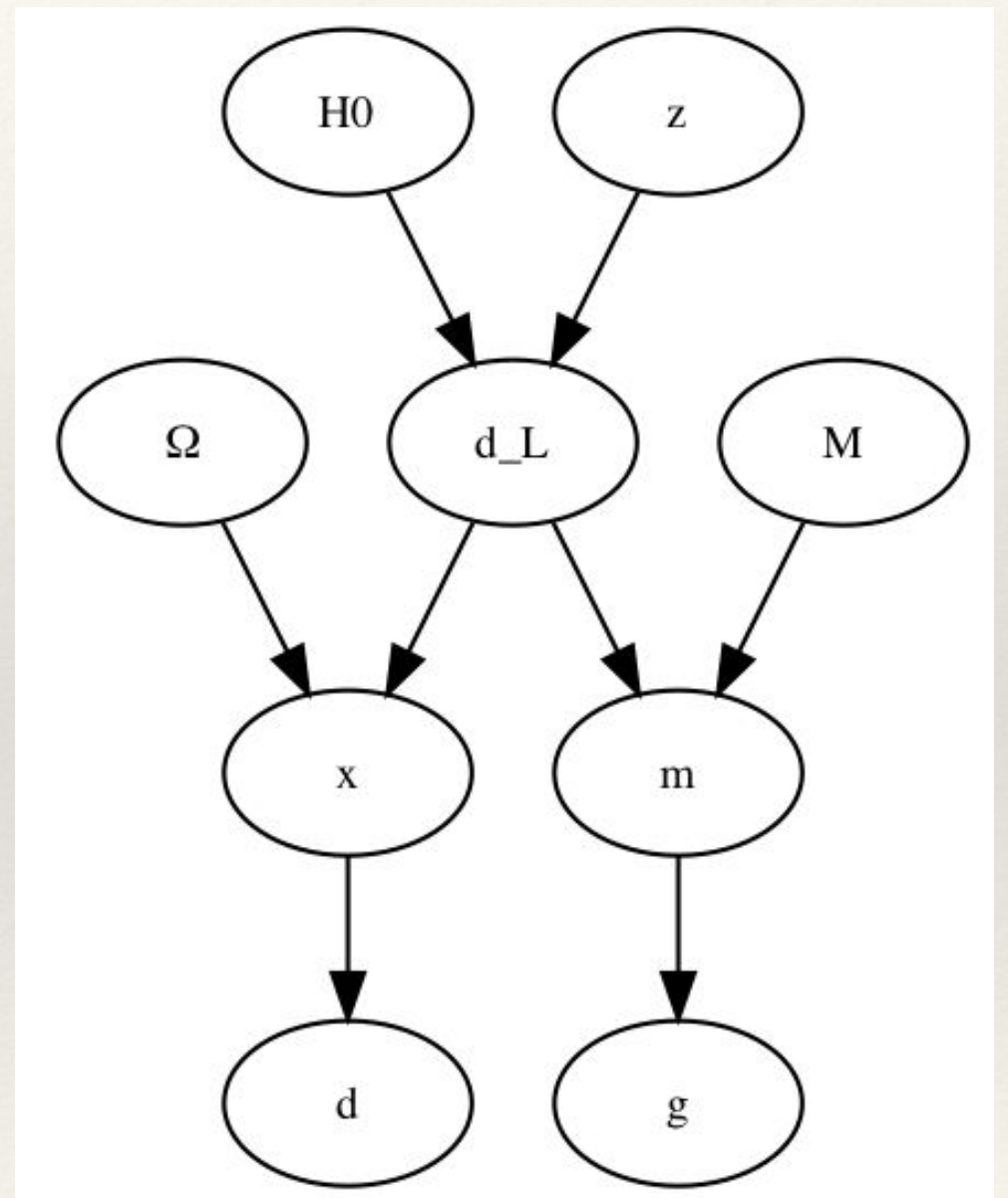
Examples of Parameter Posteriors
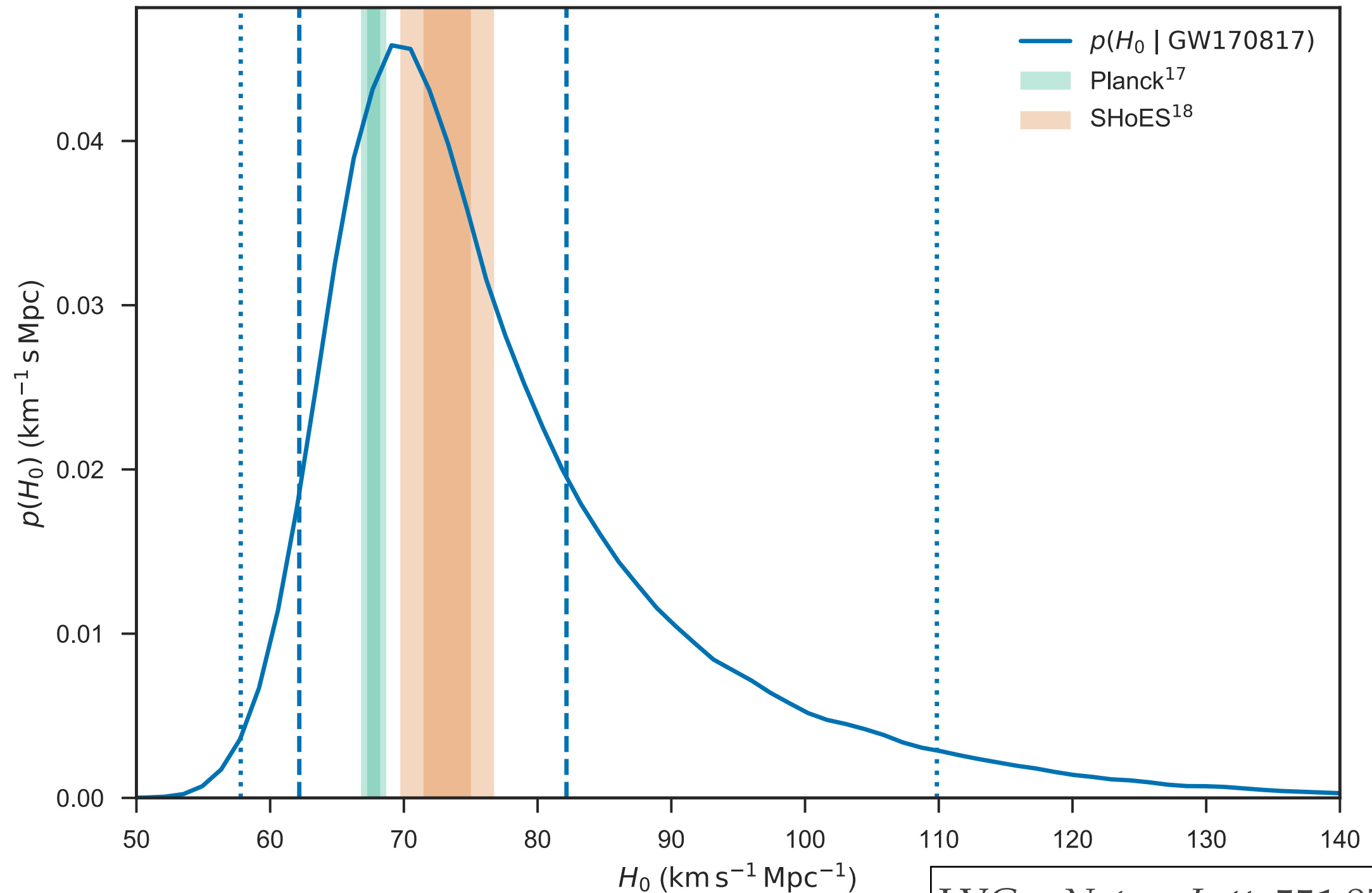
Examples of Parameter Posteriors
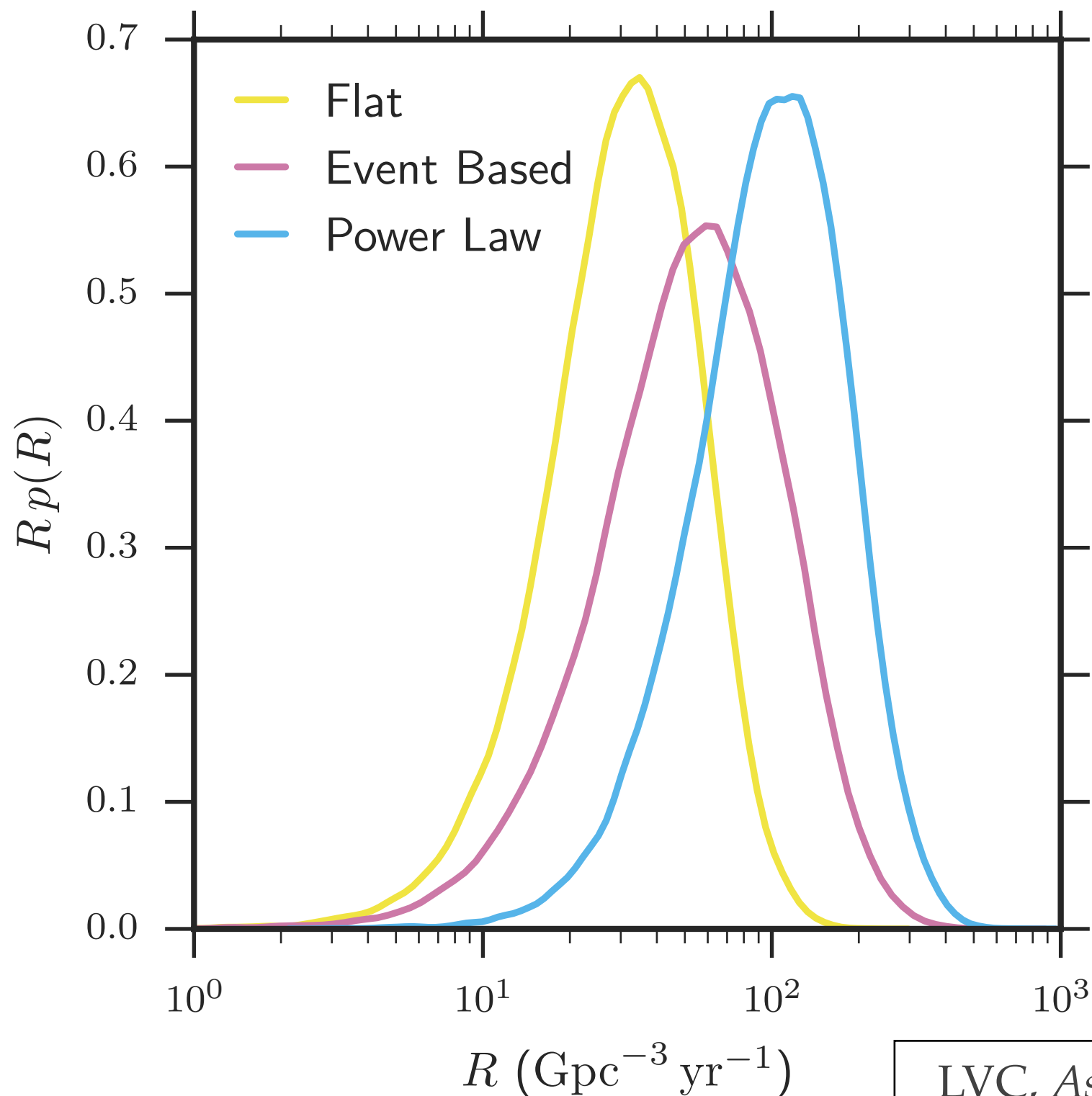
Reversible Jump MCMC

* 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 *(x, u)* and *(x', u')*. Here **x** and **x'** denote the parameters of the current and proposed state (which may have different numbers of dimensions) and **u**, **u'** are sets of random numbers that lead to a proposed move from **x** to **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)
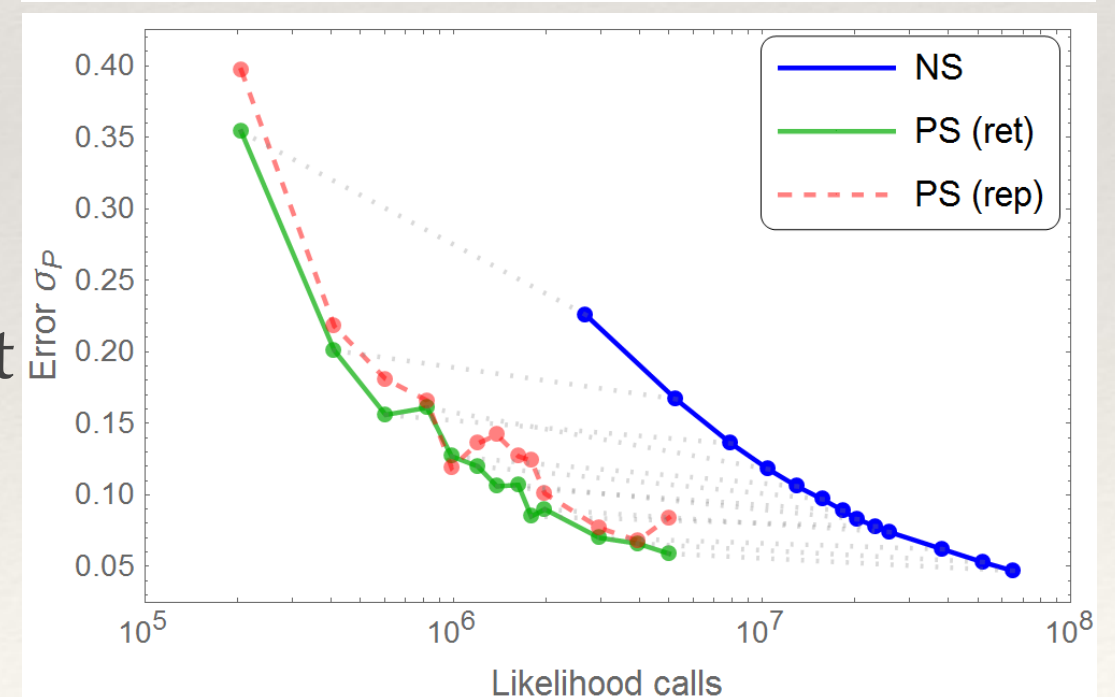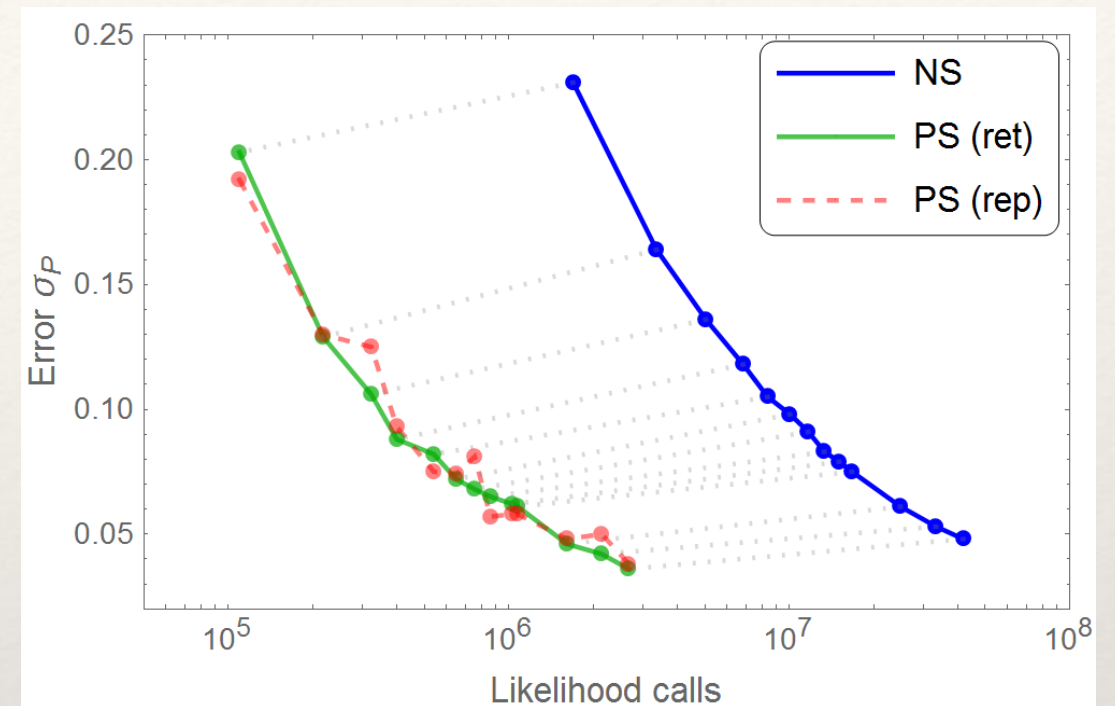
Bayesian Evidence Calculation

❖ 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.
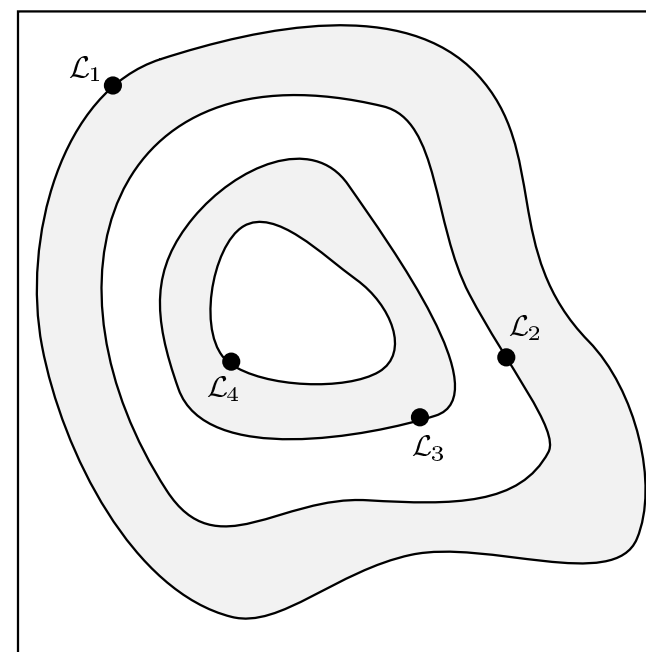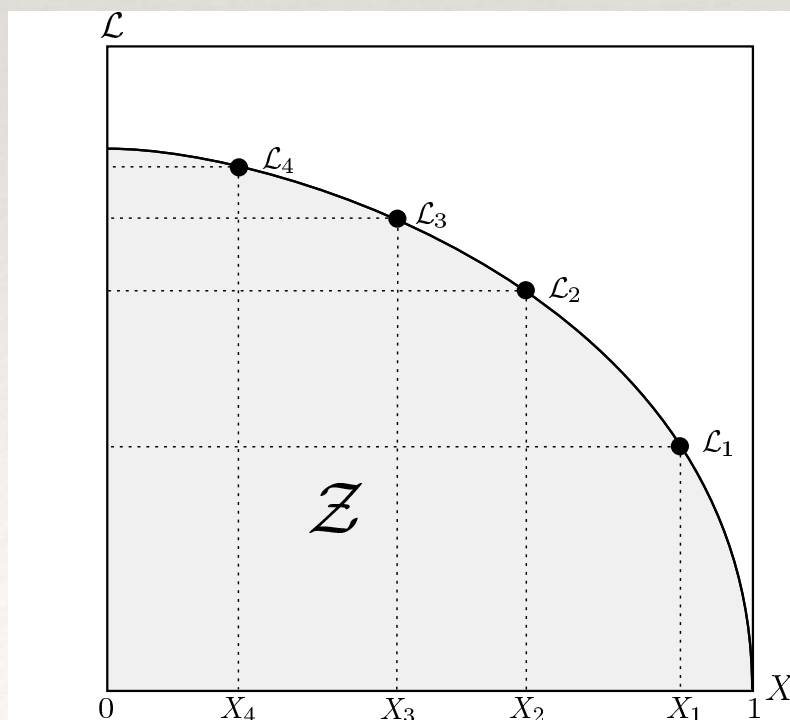
MultiNest

- 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.
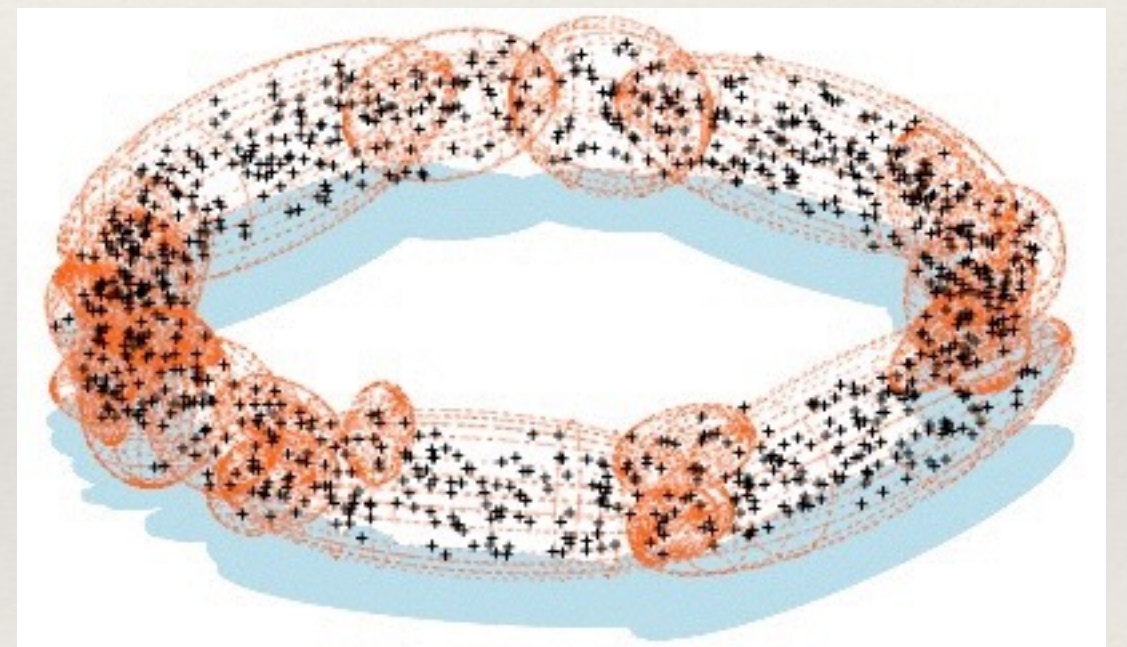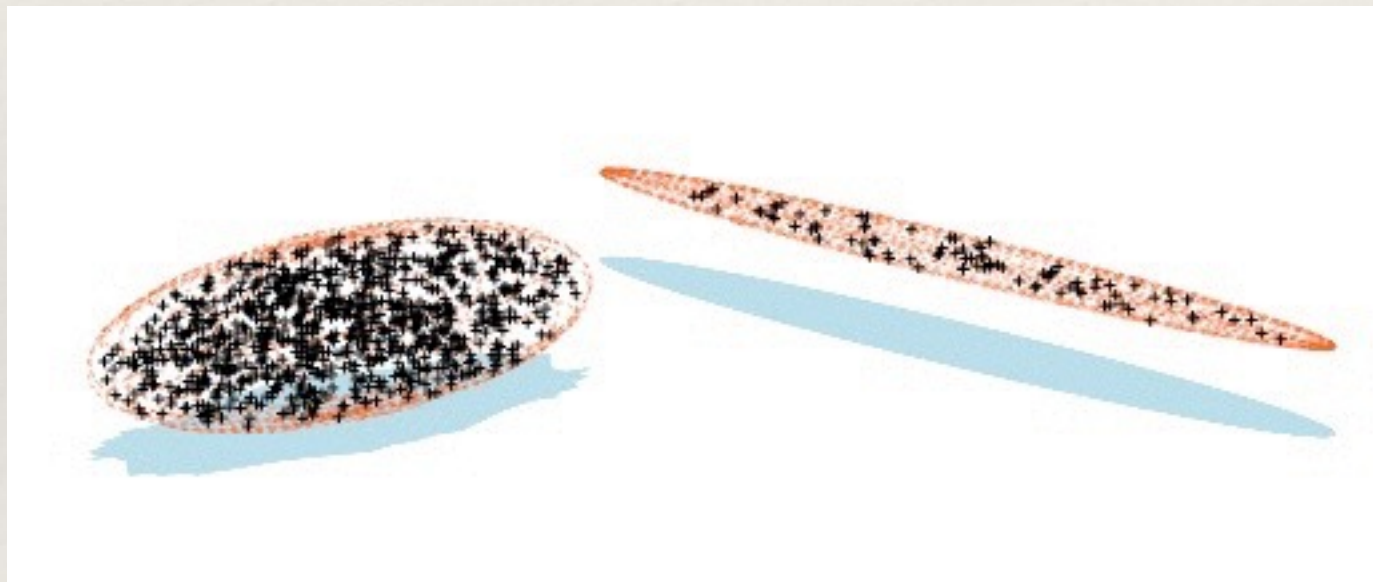
- The algorithm is well suited to exploring likelihoods with multiple modes.

- Although designed to compute the evidence, MultiNest also returns the posterior probability distribution.

MultiNest
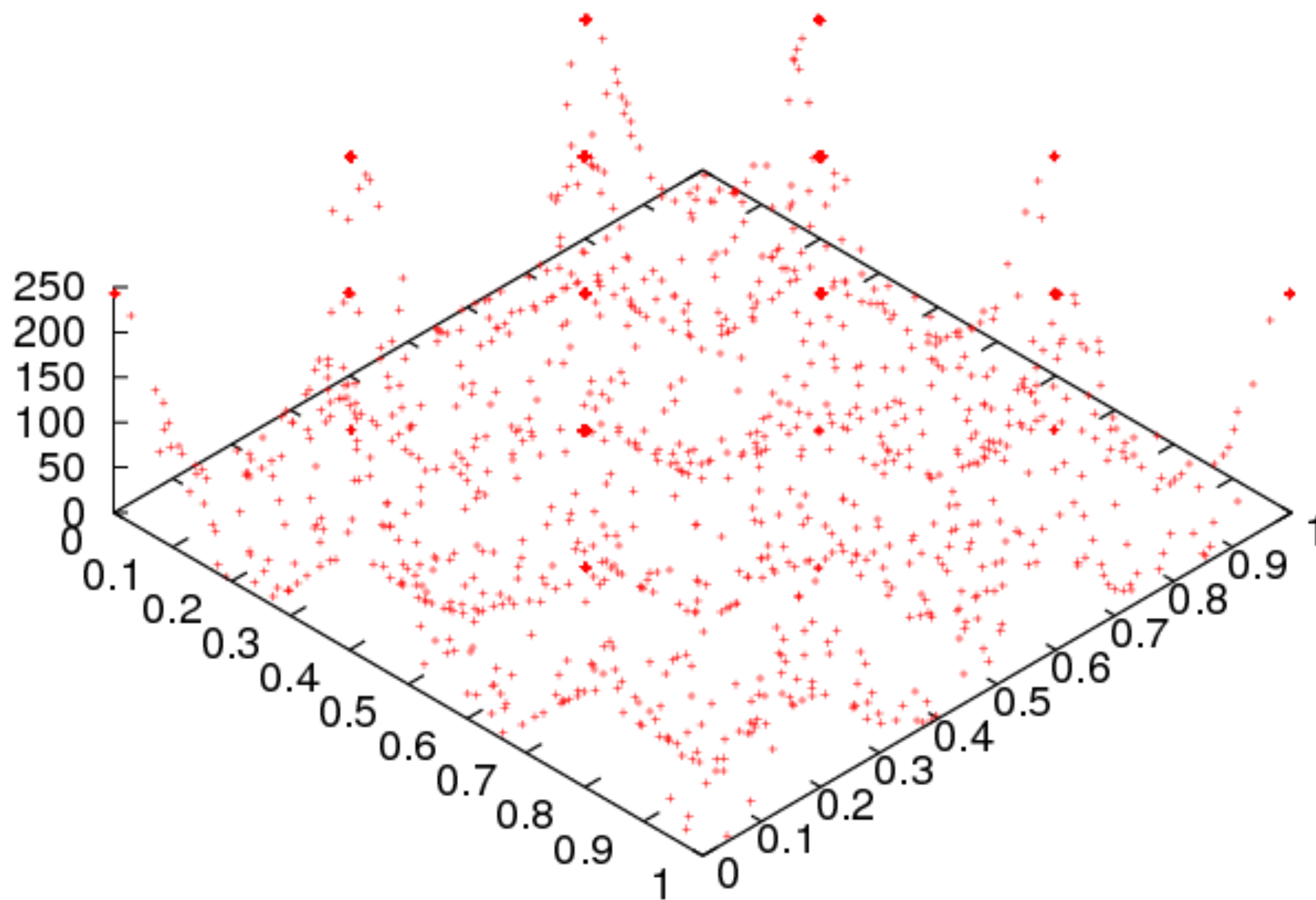
MultiNest for GWs

❖ 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.

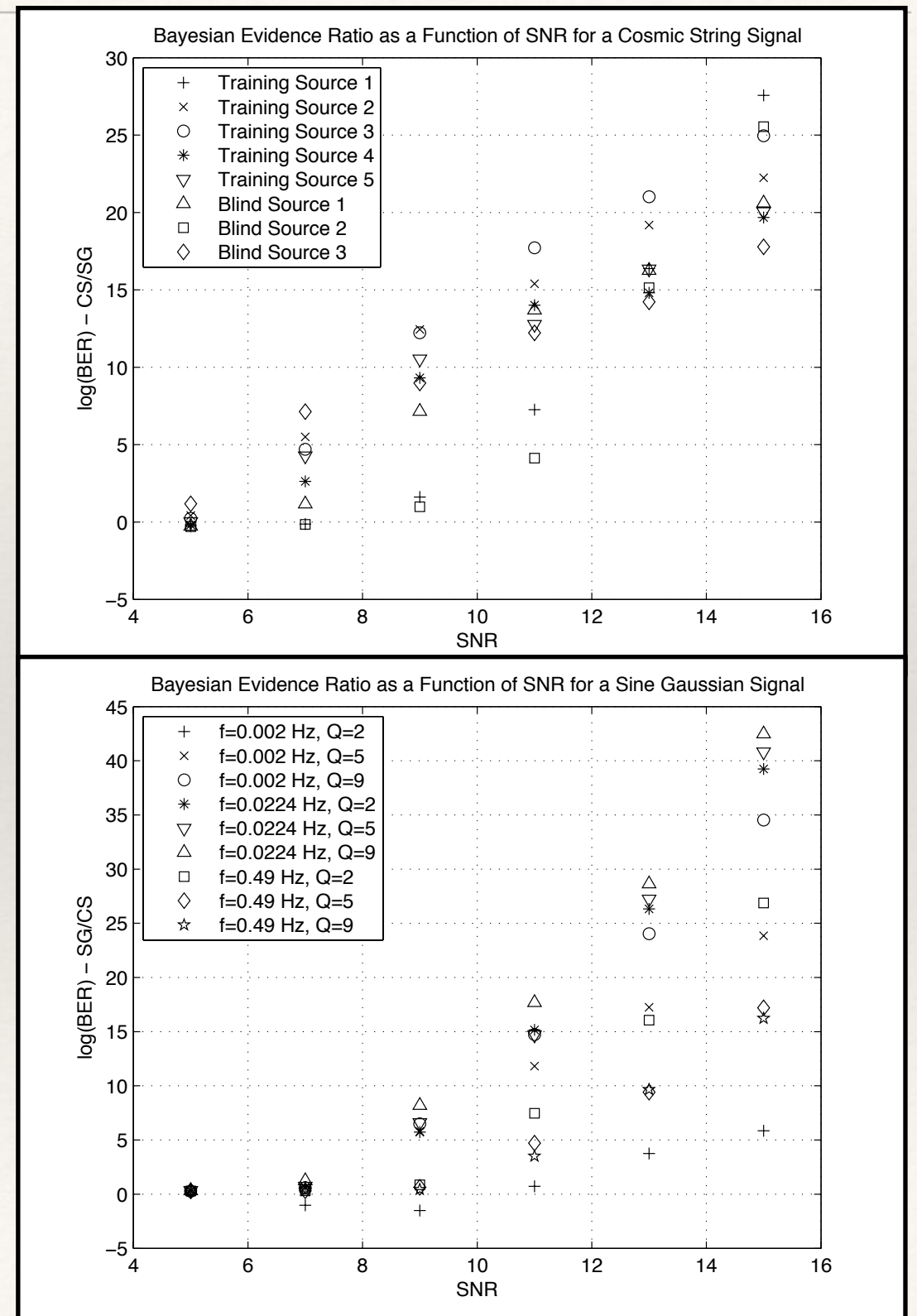❖ Identified correct number of signals (3), and recovered waveforms with better than 99% overlap in all cases.
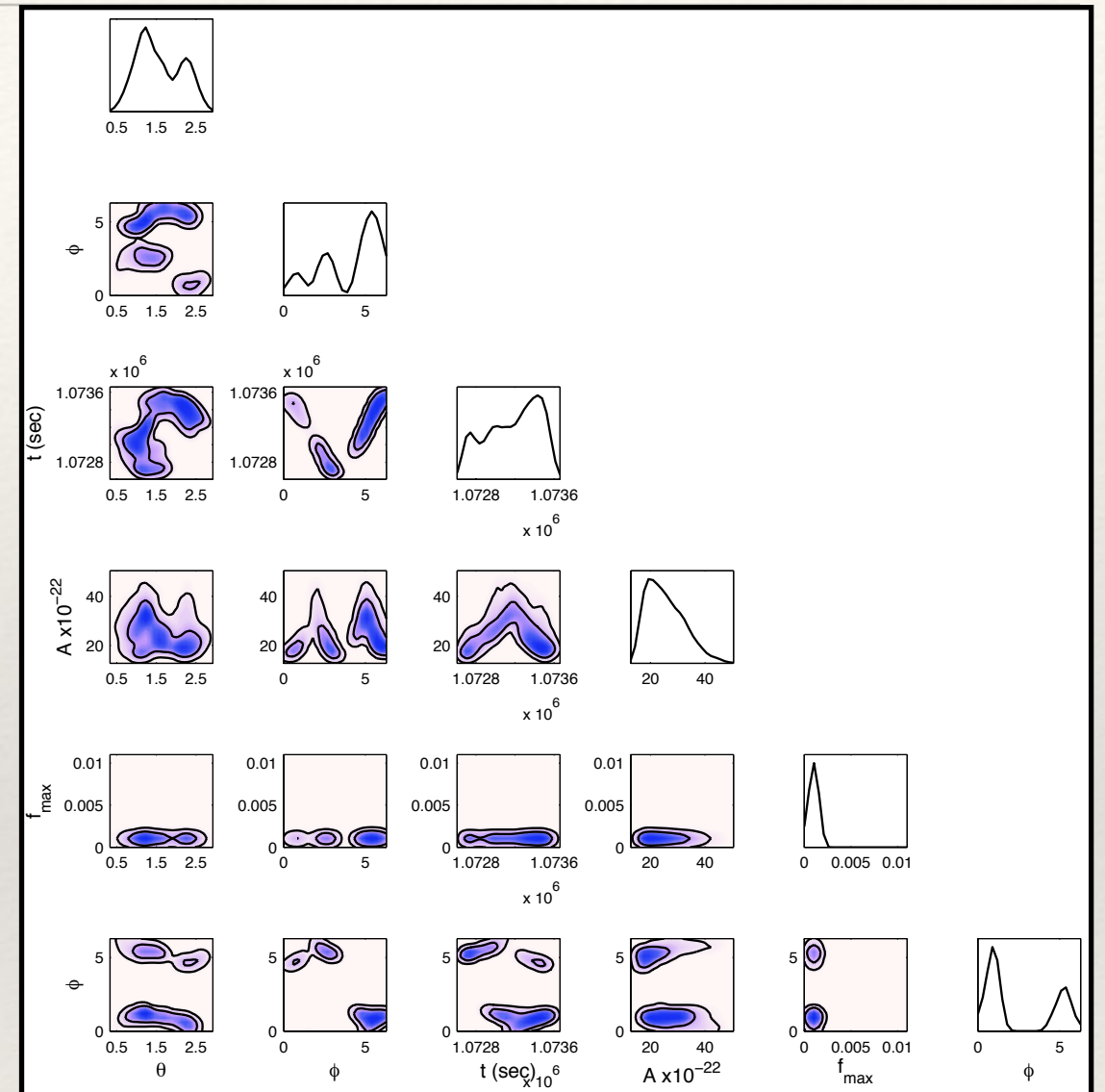
Source	$t_c(s)$	Channel	True SNR	Recovered SNR		
				Mode 1	Mode 2	Mode 3
1	1.6×10^6	A	41.0	41.2	41.0	N/A
		E	14.5	14.5	14.6	
2	1.1×10^6	A	30.7	30.7	N/A	N/A
		E	13.9	13.9		
3	6×10^5	A	18.8	18.9	18.5	18.4
		E	36.9	36.7	37.1	36.8

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.

Bayesian Evidence Ratio as a Function of SNR for a Cosmic String Signal

Bayesian Evidence Ratio as a Function of SNR for a Sine Gaussian Signal

PolyChord

❖ An alternative nested sampling algorithm is PolyChord (Handley et al. 2015). It uses slice sampling to sample within the likelihood constraint, and affine transformations to make contours more spherical.
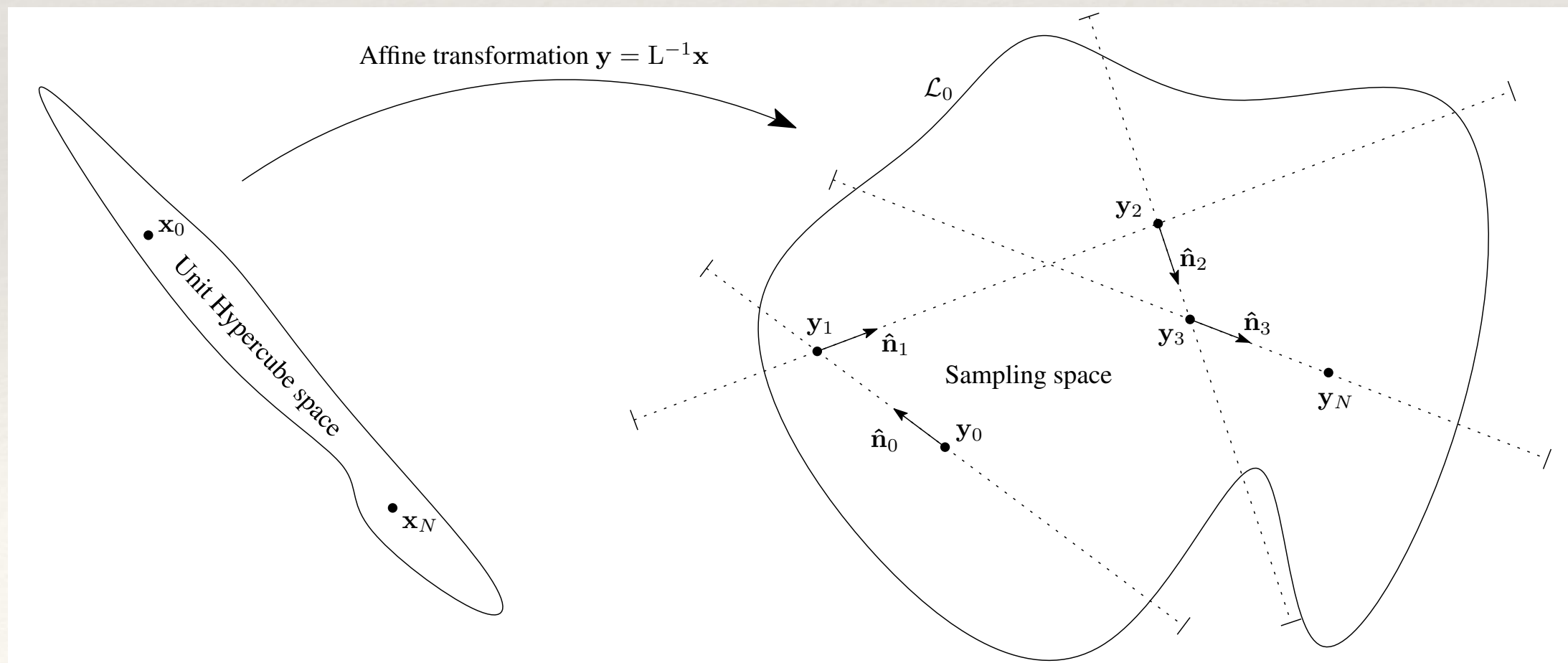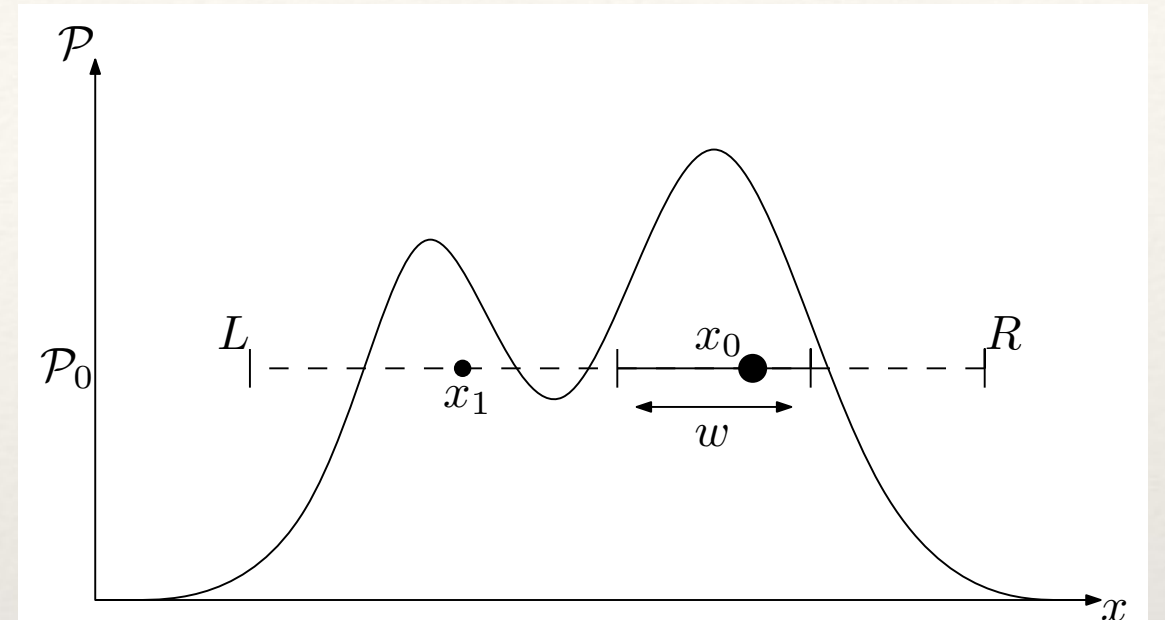
PolyChord

❖ PolyChord outperforms MultiNest in large numbers of dimensions.
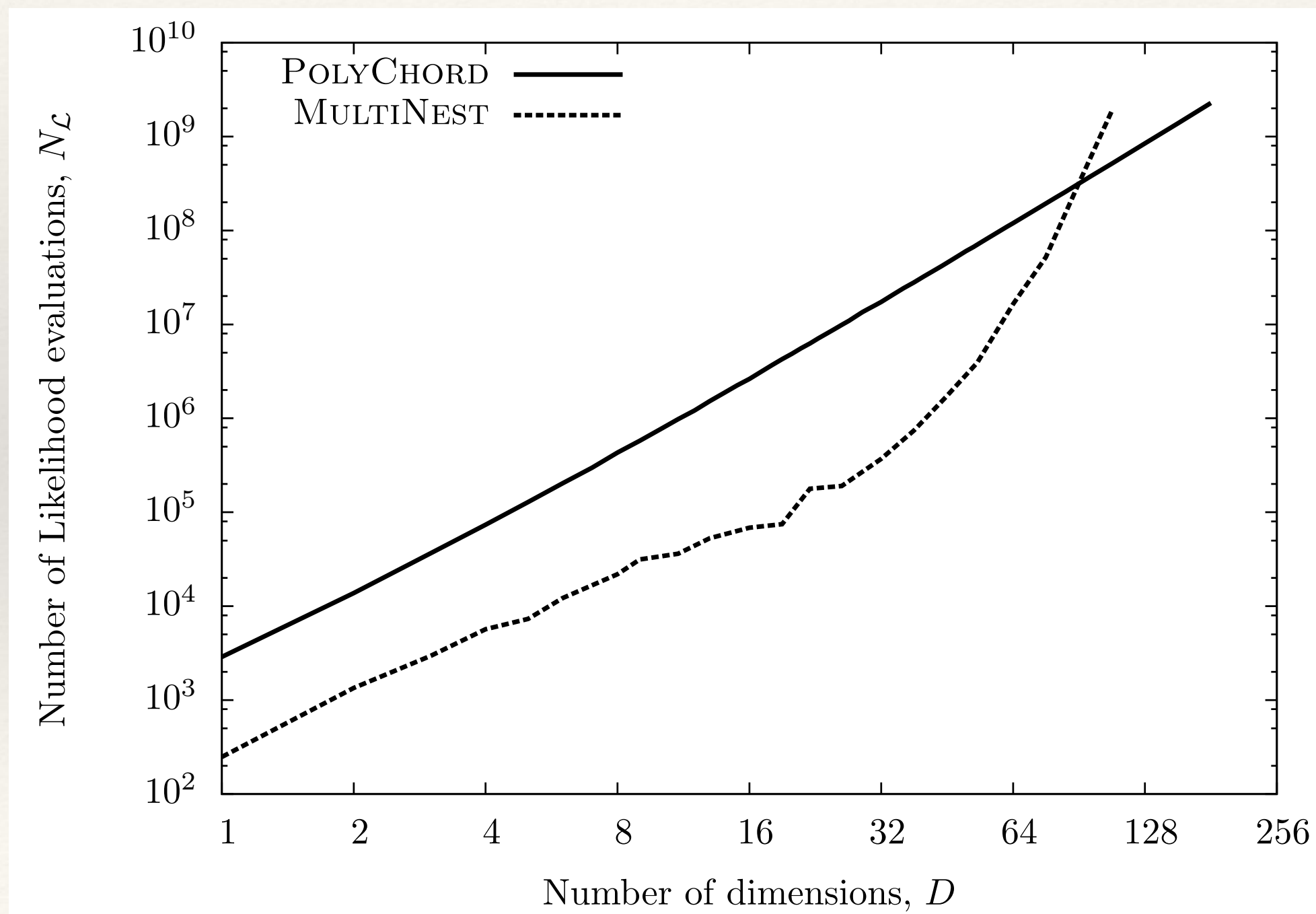
Reduced Order Modelling

❖ Most waveform models are too expensive for parameter estimation. Use reduced order modelling to develop computationally more efficient waveforms.

❖ Either interpolation of waveforms (Puerrer et al.).

❖ Or build a basis for the waveform space using a greedy algorithm (Field, Tiglio, Galley, …)

❖ Typically get order of magnitude computational saving.

Canizares, JG et al. (2013)
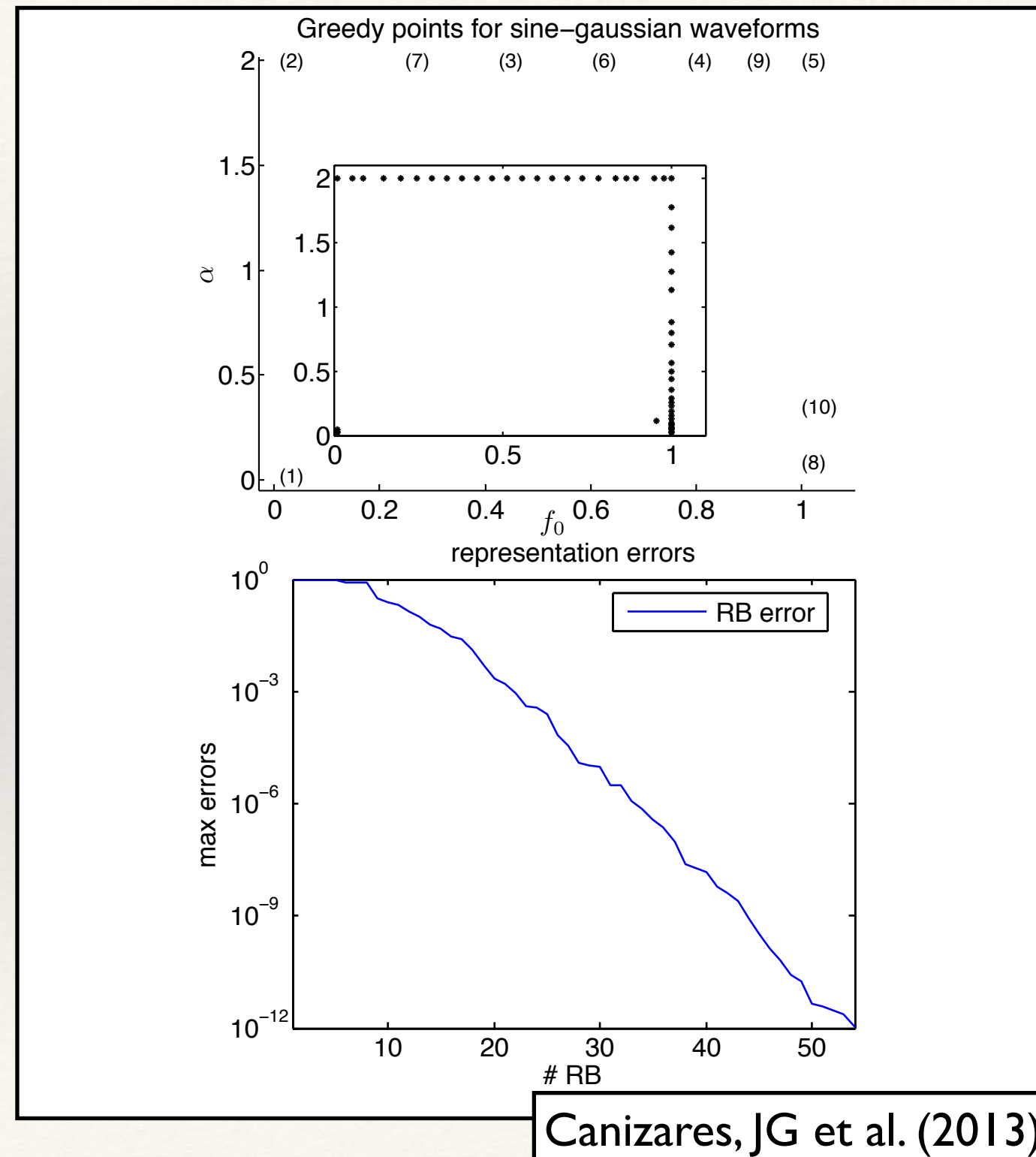
Reduced Order Modelling

❖ Final step - construct a quadrature rule to approximate overlap

$$(h(\vec{\lambda})|d) = 4\Re \int_0^\infty \frac{\tilde{h}(\vec{\lambda})\tilde{d}^*(f)}{S_h(f)} \mathrm{d}f$$

$$\approx 4\Re \left[\sum_{k=0}^{N/2} d^*(f_k)\vec{e}^T(f_k)\Delta f A^{-1} \right] \vec{h}(\vec{\lambda})$$

$$= 4\Re \sum_{k=1}^{m} \omega_k h(F_k; \vec{\lambda})$$

❖ Reduced order quadrature (ROQ) is state of the art for LIGO PE. Still need SEOBNRv4-ROQ and IMRPhenomP-ROQ.

❖ Models not yet good enough for LISA!

Full Likelihood

ROQ Likelihood
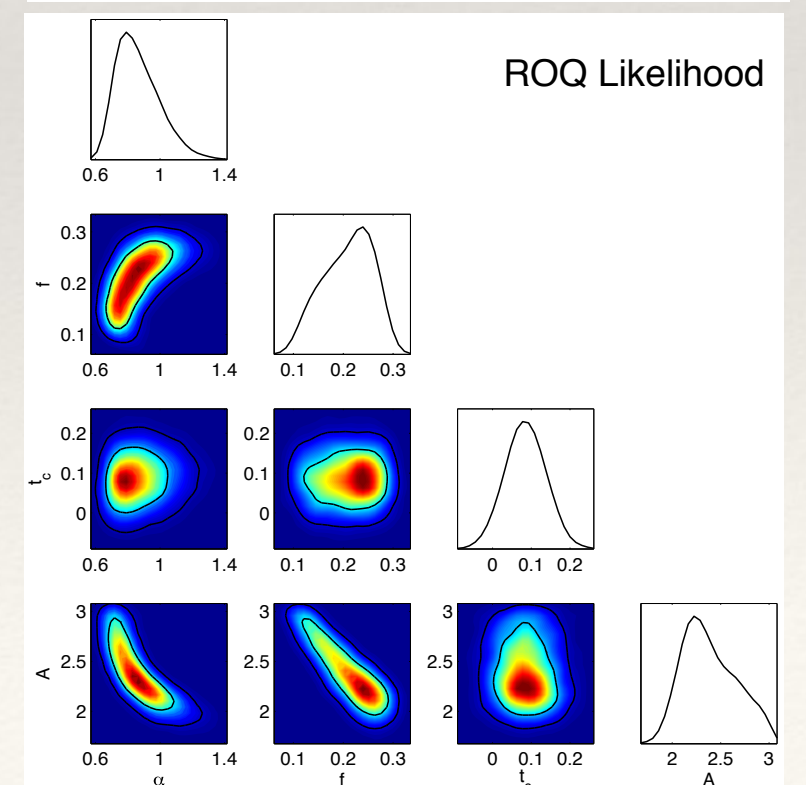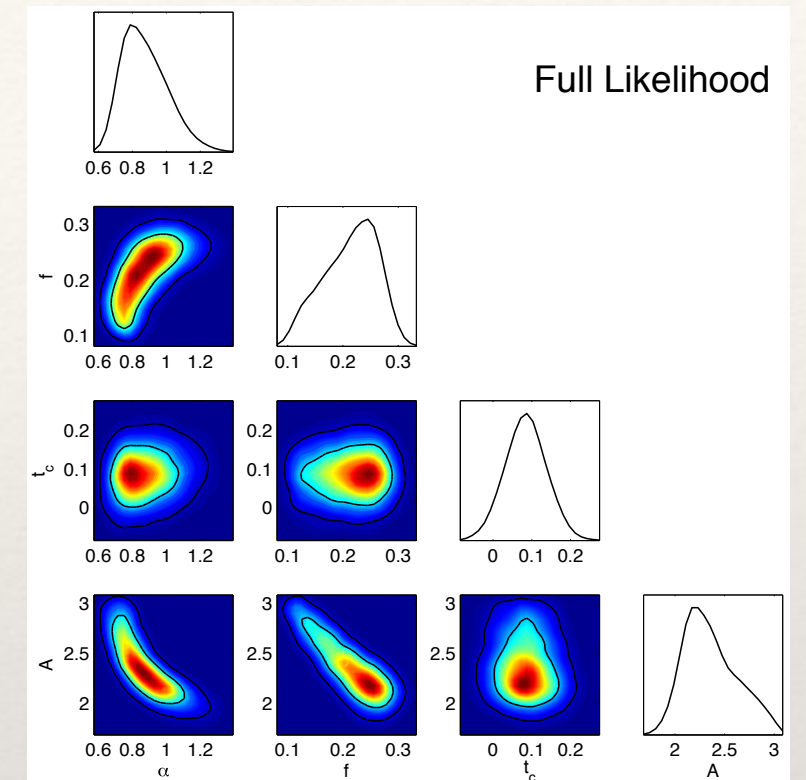
Background Mapping

Representing the GW sky

- ❖ Standard approach to background detection assumes a suitable form for the statistical properties of the background. Alternative: try to produce a phase-coherent map of the sky (**JG** et al. 2014, 2015).

- ❖ Represent arbitrary "skies" using spin-weighted spherical harmonics. A spin-weighted function $f(\hat{k}, \hat{l}, \hat{m})$ maps a point $\hat{k}$ and an orthonormal basis $(\hat{l}, \hat{m})$ on the sphere onto $\mathbb{C}$ and has the property

$$f(\hat{k}, \cos\psi\hat{l} - \sin\psi\hat{m}, \sin\psi\hat{l} + \cos\psi\hat{m}) = \mathrm{e}^{is\psi} f(\hat{k}, \hat{l}, \hat{m})$$

- ❖ where s is the spin weight.

- ❖ Under such a rotation

$$h_+ \to h_+ \cos 2\psi + h_\times \sin 2\psi \qquad h_\times \to -h_+ \sin 2\psi + h_\times \cos 2\psi$$

- ❖ so the quantities $m_\pm^a m_\pm^b h_{ab}(\hat{k})$ for $\hat{m}_\pm^a = \hat{l}^a \pm i\hat{m}^a$ have spin-weight ± 2.
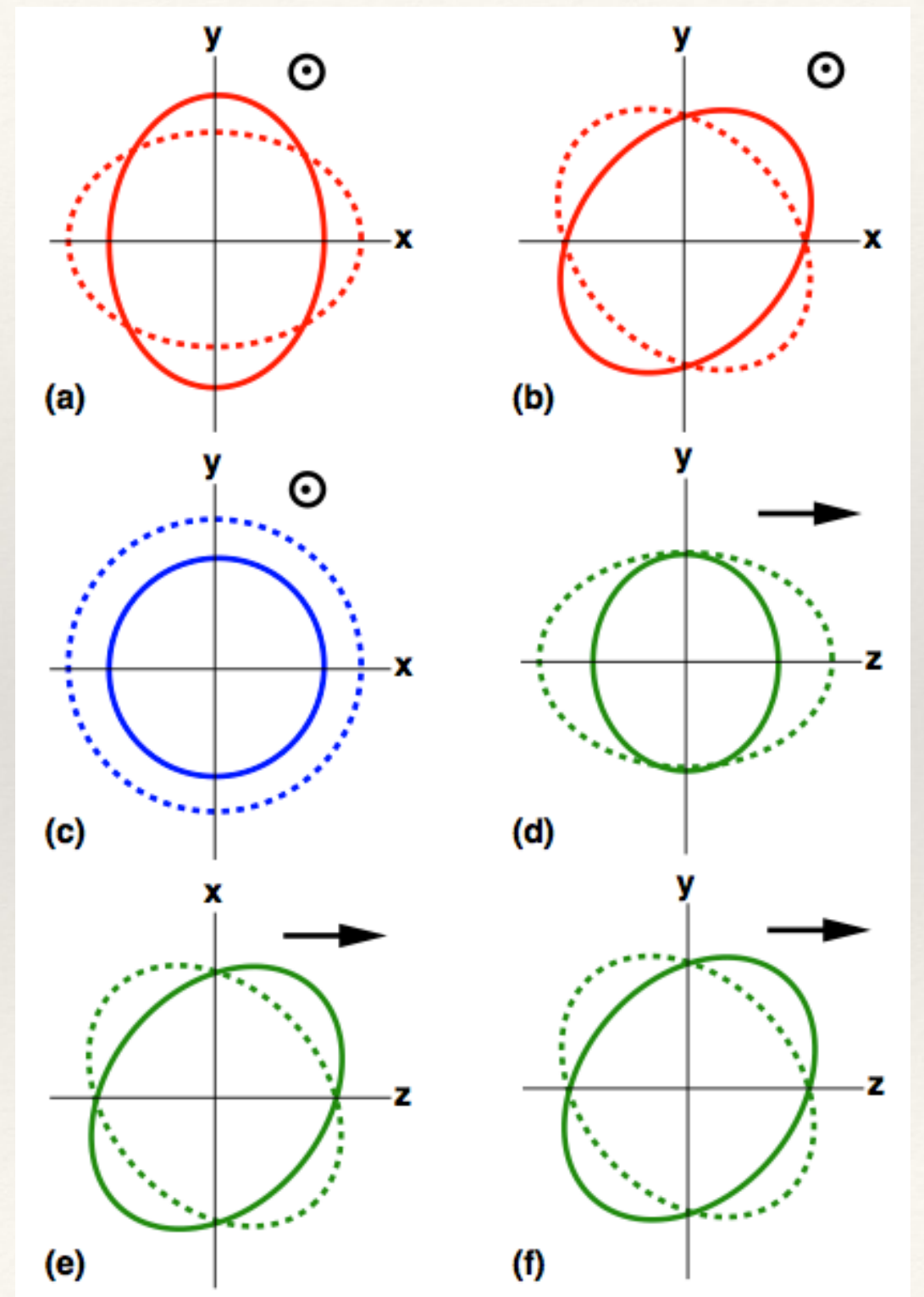
GW polarisation states

- GR admits two transverse and traceless (TT) polarisations

$$e_{ij}^{+} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad e_{ij}^{\times} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- In other metric theories of gravity, can have up to four additional states

$$e_{ij}^{B} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad e_{ij}^{L} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$e_{ij}^{X} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad e_{ij}^{Y} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

GW polarisation states

- A spin-weight s function can be expanded in terms of

$$_sY_{lm}(\theta,\phi) = \sqrt{\frac{(l-s)!}{(l+s)!}}\check{\eth}^s Y_{lm}(\theta,\phi)$$

$$\check{\eth}^s\eta = -(\sin\theta)^s\left[\frac{\partial}{\partial\theta} + i\csc\theta\frac{\partial}{\partial\phi}\right](\sin\theta)^{-s}\eta$$

- We expand tensor (+,x) modes in spin-weight 2 spherical harmonics.

- For scalar modes (B,L), the quantities $\hat{m}^a_{\pm}\hat{m}^b_{\pm}h^B_{ab}(\hat{k})$ and $\hat{m}^a_{\pm}\hat{m}^b_{\pm}h^L_{ab}(\hat{k})$ have spin-weight 0 - expand in standard spherical harmonics.

- For the vector modes (X,Y), the quantities $\hat{m}^a_{\pm}\hat{m}^b_{\pm}h^X_{ab}(\hat{k}), \hat{m}^a_{\pm}\hat{m}^b_{\pm}h^Y_{ab}(\hat{k})$ transform like spin-weight +/-1 objects.

- For tensor and vector modes there is an alternative grad/curl (G/C) representation which we will use here.

Example: Pulsar timing

- A plane gravitational wave induces a redshift in a pulsar signal

$$z(t, \hat{k}) \equiv \frac{\Delta v(t)}{\nu_0} = \frac{1}{2} \frac{\hat{u}^a \hat{u}^b}{1 + \hat{k} \cdot \hat{u}} \Delta h_{ab}(t, \hat{k})$$

- The redshift induced by a GW background can be written as

$$z(t) = \int_{-\infty}^{\infty} df \int_{S^2} d^2\Omega_{\hat{k}} \frac{1}{2} \frac{\hat{u}^a \hat{u}^b}{1 + \hat{k} \cdot \hat{u}} h_{ab}(f, \hat{k}) \left[1 - e^{-i2\pi f L(1 + \hat{k} \cdot \hat{u})/c}\right] e^{i2\pi f(t - \hat{k} \cdot \vec{x}/c)}$$

$$= \int_{-\infty}^{\infty} df \sum_{(lm)} \sum_{P} R^P_{(lm)}(f) a^P_{(lm)}(f) e^{i2\pi ft}$$

- where the response functions for individual modes are given by

$$R^P_{(lm)}(f) = \int_{S^2} d^2\Omega_{\hat{k}} \frac{1}{2} \frac{\hat{u}^a \hat{u}^b}{1 + \hat{k} \cdot \hat{u}} Y^P_{(lm)ab}(\hat{k}) e^{-i2\pi f \hat{k} \cdot \vec{x}/c} \left[1 - e^{-i2\pi f L(1 + \hat{k} \cdot \hat{u})/c}\right]$$

- Will simplify notation elsewhere by writing $y \equiv 2\pi f L/c$

Computing response functions

- ❖ Compute response in computational frame, in which pulsar is in the z-direction. Expansion coefficients transform under a rotation in a similar way to spherical harmonic coefficients.

$$Y^P_{(lm)ab}(\theta, \phi) = \sum_{m'=-l}^{l} \left[D^l{}_{mm'}(\chi_I, \zeta_I, 0) \right]^* Y^P_{(lm')\bar{a}\bar{b}}(\bar{\theta}, \bar{\phi}) \mathbf{R}(\chi_I, \zeta_I, 0)^{\bar{a}}{}_a \mathbf{R}(\chi_I, \zeta_I, 0)^{\bar{b}}{}_b$$

- ❖ Deduce that the response functions in the cosmic frame for a pulsar in direction $\hat{u}^a_I = (\sin \zeta_I \cos \chi_I, \sin \zeta_I \sin \chi_I, \cos \zeta_I)$ takes the form

$$R^P_{I(lm)}(f) = Y_{lm}(\hat{u}_I) \mathcal{R}^P_l(y_I)$$

- ❖ for all polarisation states. Similarly, for the astrometric response

$$_1 R^P_{I(lm)} = {}_1 Y_{lm}(\hat{u}_I) {}_1 \mathcal{R}^P_l(y_I) \qquad _{-1} R^P_{I(lm)} = {}_{-1} Y_{lm}(\hat{u}_I) {}_{-1} \mathcal{R}^P_l(y_I)$$

Results: PTAs

❖ The full set of response functions for PTAs are

$$\mathcal{R}_l^G(y) = \pi\,^{(2)}N_l(-i)^l e^{-iy}\left[(2-2iy+y^2)j_l(y) - i(6+4iy+y^2)\frac{\mathrm{d}j_l}{\mathrm{d}y} - (6iy-y^2)\frac{\mathrm{d}^2 j_l}{\mathrm{d}y^2} - iy^2\frac{\mathrm{d}^3 j_l}{\mathrm{d}y^3}\right]$$

$$\mathcal{R}_l^C(y) = 0$$

$$\mathcal{R}_l^B(y) = 2\pi\frac{1}{\sqrt{2}}\left\{\delta_{l0} - \frac{1}{3}\delta_{l1} - (-i)^l e^{-iy}\left[\left(1-i\frac{l}{y}\right)j_l(y) + ij_{l+1}(y)\right]\right\}$$

$$\mathcal{R}_l^L(y) = 2\pi\left\{-\delta_{l0} + \frac{1}{3}\delta_{l1} + (-i)^l e^{-iy}\left[\left(1-i\frac{l}{y}\right)j_l(y) + ij_{l+1}(y)\right] + \frac{1}{2}H_l(y)\right\}$$

$$H_l(y) = \int_{-1}^{1}\mathrm{d}x\,\frac{1}{(1+x)}P_l(x)\left(1 - e^{-iy(1+x)}\right)$$

$$R_l^{V_G}(y) = \pi\,^{(1)}N_l\left\{\frac{4}{3}\delta_{l1} + 2(-i)^l e^{-iy}\left[\left(1-\frac{il}{y}\right)(l+1)j_l(y) - (y-i(2l+3))j_{l+1}(y) - iyj_{l+2}(y)\right]\right\}$$

$$R_l^{V_C}(y) = 0$$

PTA response to tensor modes
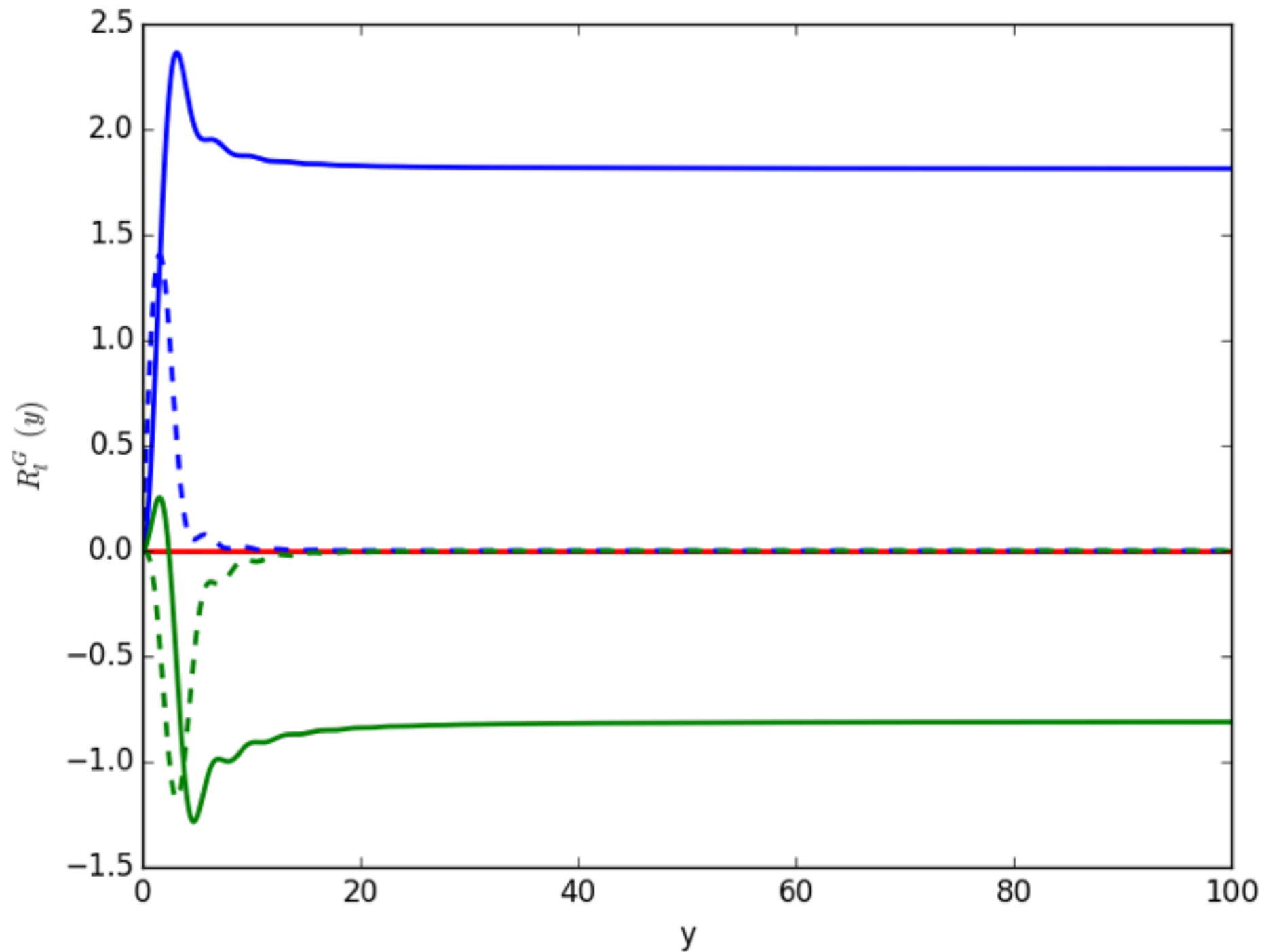
PTA response to breathing modes
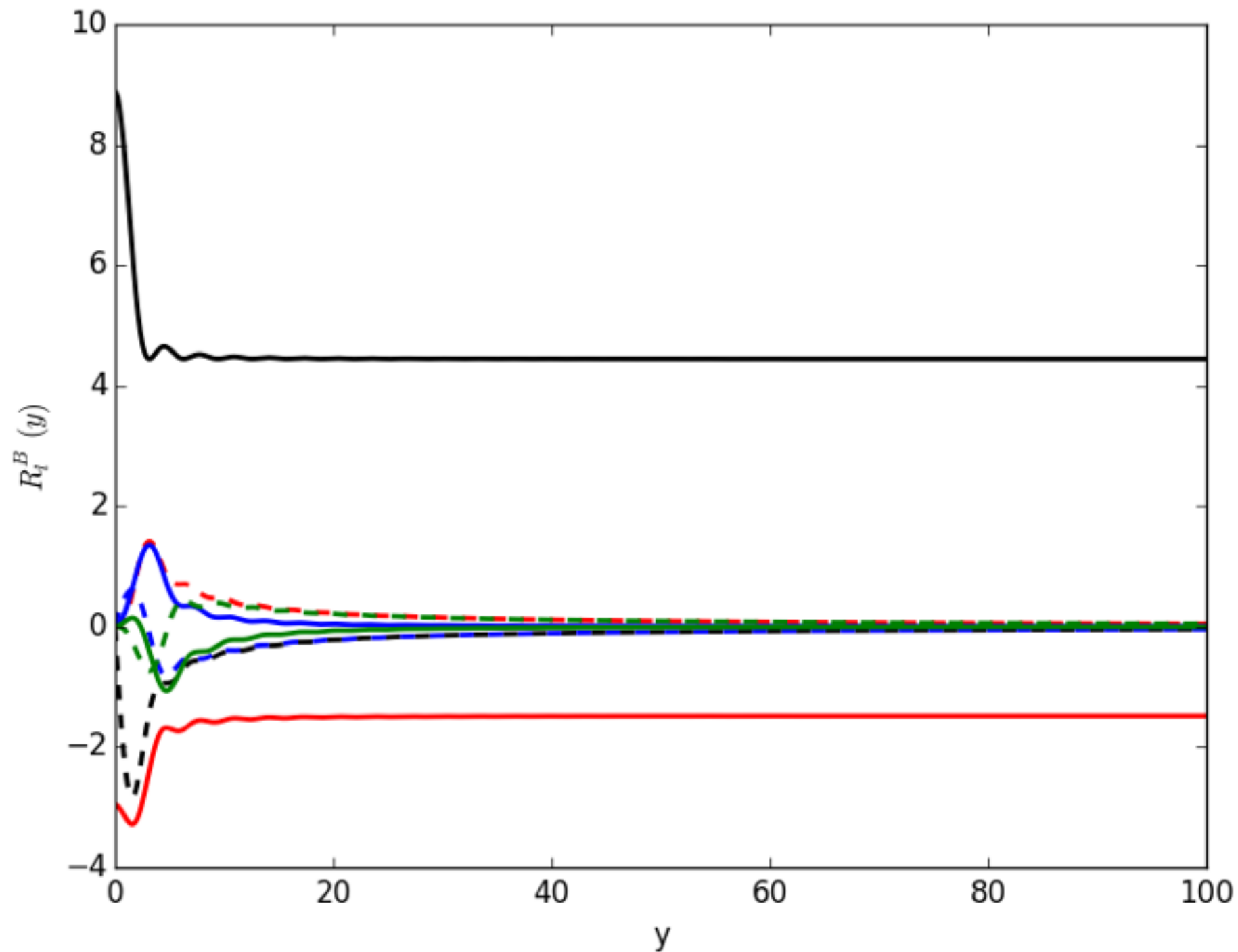
PTA response to vector modes
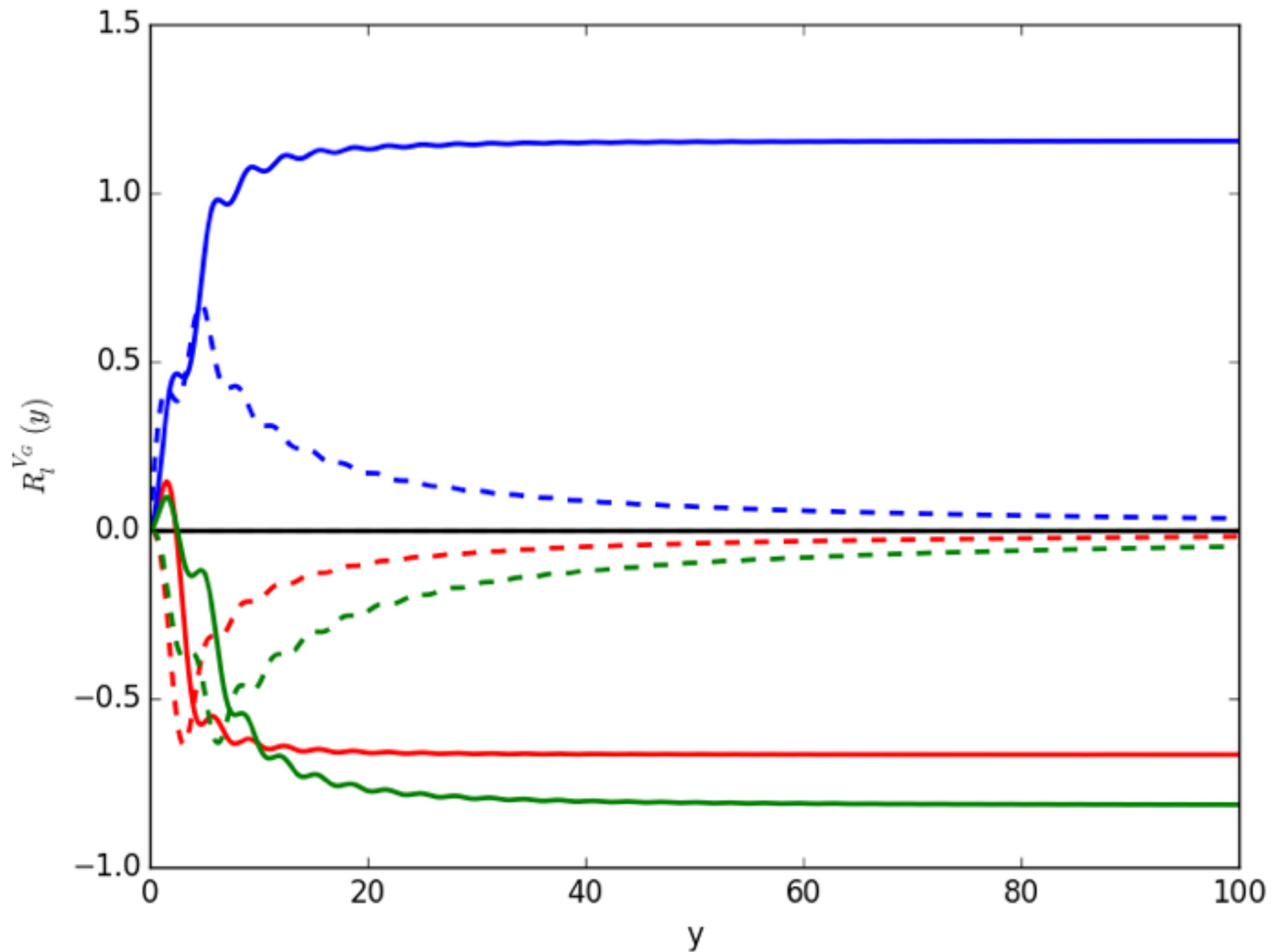
PTA response to scalar-longitudinal modes
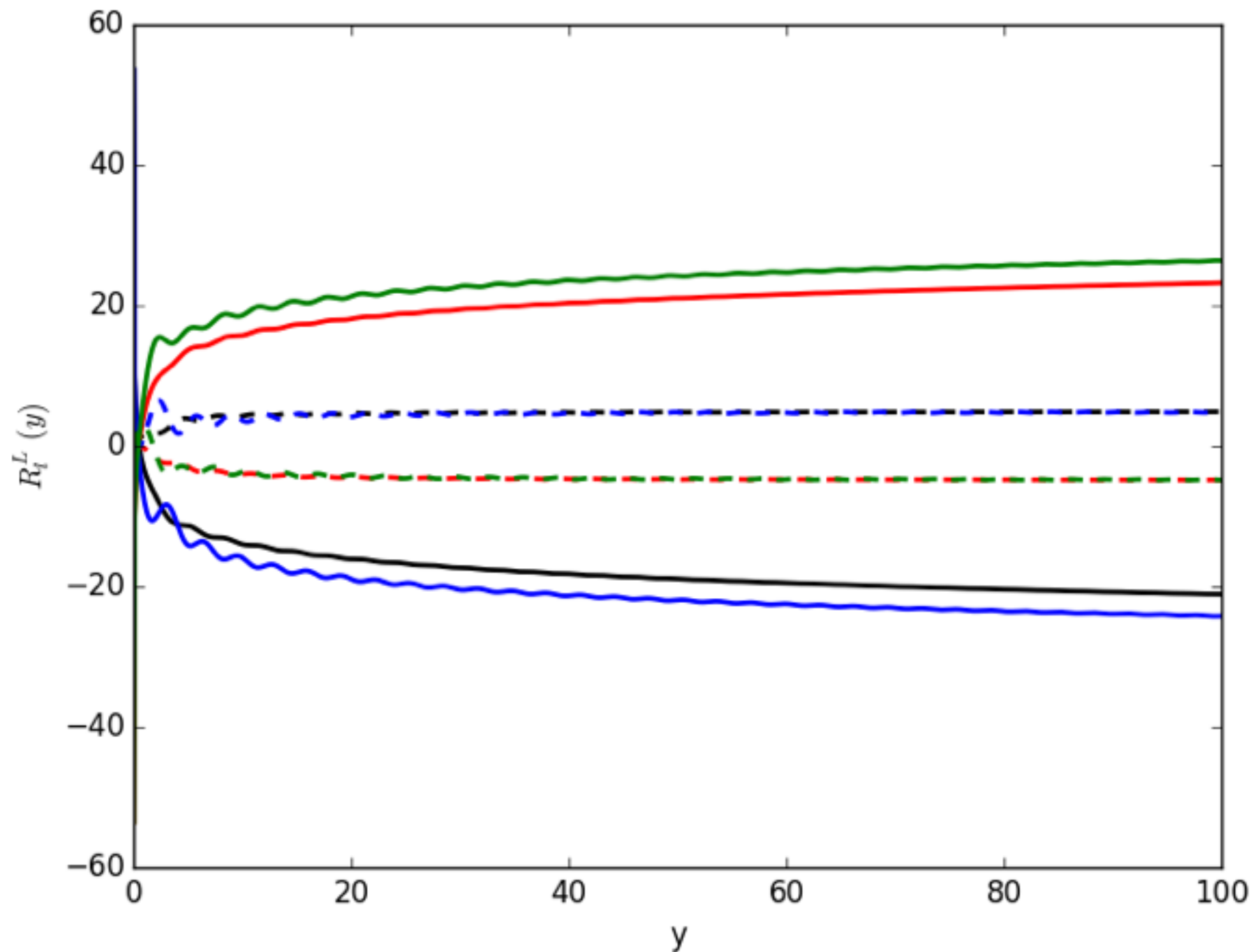
Background mapping: theory

- ❖ The total response of a pulsar in direction $\hat{u}_I$ is

$$R_I(f) = \sum_{lm} \left(a^B_{(lm)}(f) \mathcal{R}^B_l(y_I) + a^L_{(lm)}(f) \mathcal{R}^L_l(y_I) \right.$$

$$\left. + a^{V_G}_{(lm)}(f) \mathcal{R}^{V_G}_l(y_I) + a^G_{(lm)}(f) \mathcal{R}^G_l(y_I) \right) Y_{lm}(\hat{u}_I)$$

- ❖ If we have pulsars all over the sky, can decompose "pulsar response" map into spherical harmonic basis. Coefficients are linear combinations of different polarisations.

- ❖ No confusion between B and G modes due to range of l. Confusion with V^G and L possible unless have pulsars at several distances, i.e., several y's.

- ❖ Addition of astrometric measurements provides higher accuracy and an additional mechanism for breaking degeneracies.

Background mapping: in practice

- We can use observed timing residuals, s, to infer the coefficients, a, of the background. The likelihood takes the form

$$p(s|F, \vec{a}) \propto \exp\left[-\frac{1}{2}(\vec{s} - H\vec{a})^\dagger F^{-1}(\vec{s} - H\vec{a})\right]$$

- At a given frequency we make only $2N_p$ measurements - an amplitude and phase for each of the N_p pulsars. Can only hope to recover N_p combinations of the (complex) $a^G_{(lm)}$'s.

- This shows up in a singular-value decomposition of H, $H = U\Sigma V^\dagger$

- The rectangular matrix Σ has at most N_p non-zero elements on the diagonal.

- We can write $U = [H_{\text{range}} H_{\text{null}}]$ where the N_p columns of H_{range} span the range of H.

Background mapping: in practice

❖ In a search we can replace $H\vec{a}$ by $H_{\text{range}}\vec{b}$ in the likelihood. The value of $\vec{a}$ corresponding to a given value of $\vec{b}$ is given in terms of the pseudo-inverse of Σ, Σ^+, by $\vec{a} = V\Sigma^+\vec{b}$

❖ Which components do we expect to be able to measure? Since

$$R^G_{I(lm)} \sim \frac{1}{l^{\frac{3}{2}}} \qquad \text{as } l \to \infty$$

❖ we expect to measure the low-l modes more precisely. To reach an angular resolution of l_{max} we therefore need an array of

$$N_p \approx (l_{\text{max}} + 1)^2 - 4$$

❖ Need $N_p \approx 21$ pulsars to reach l_{max}=4 required for an isotropic background; $N_p \approx 100$ to reach single source resolution at l_{max}=10.

Background mapping: in practice

❖ Gradient piece of background behaves as expected. Adding more pulsars increases resolution of map and reduces residual.
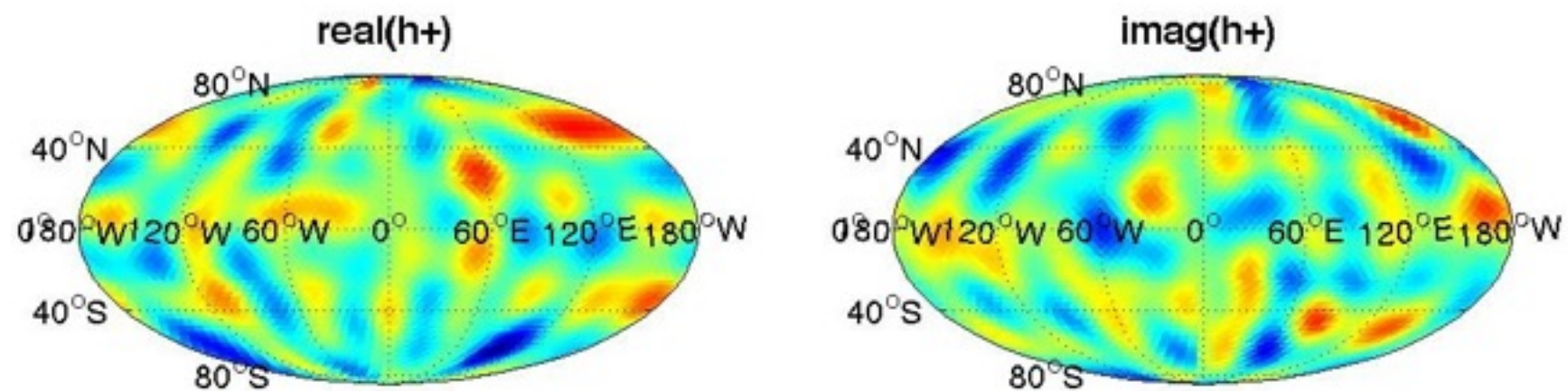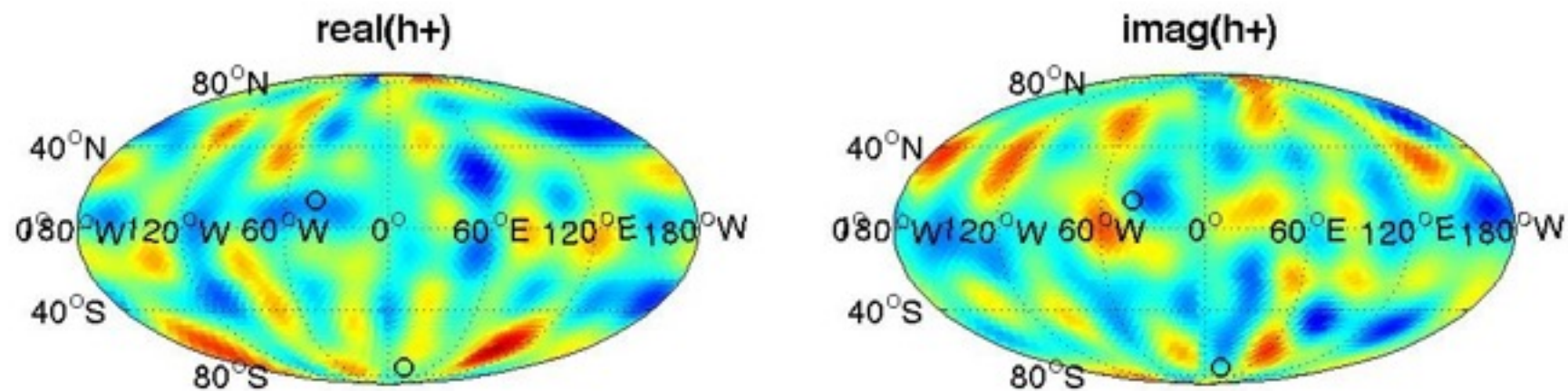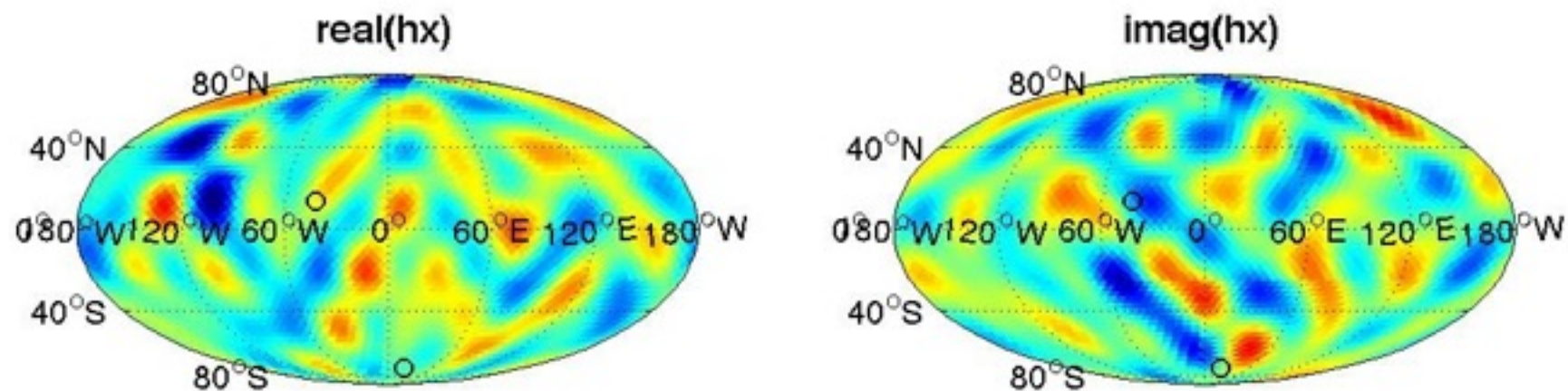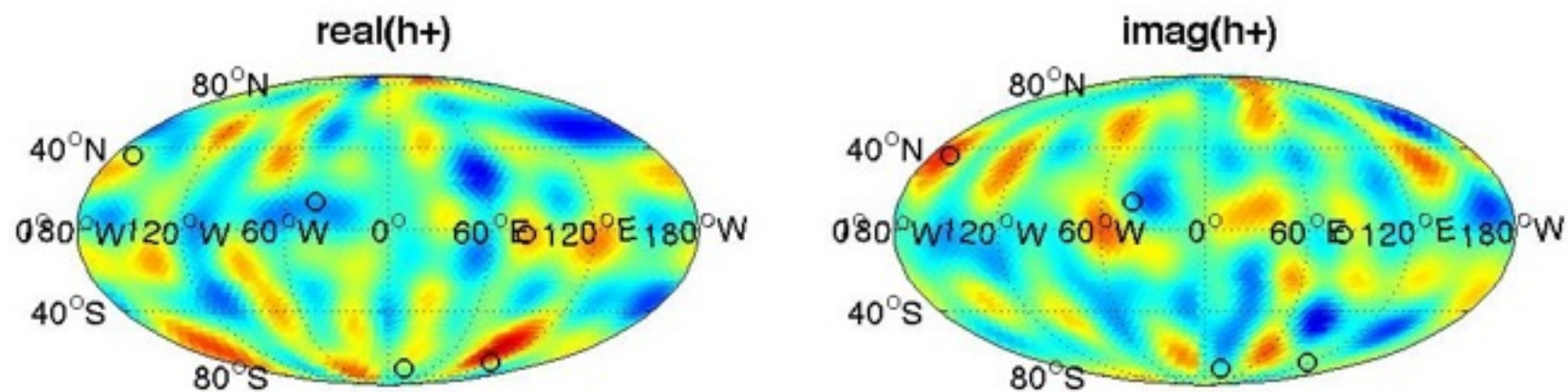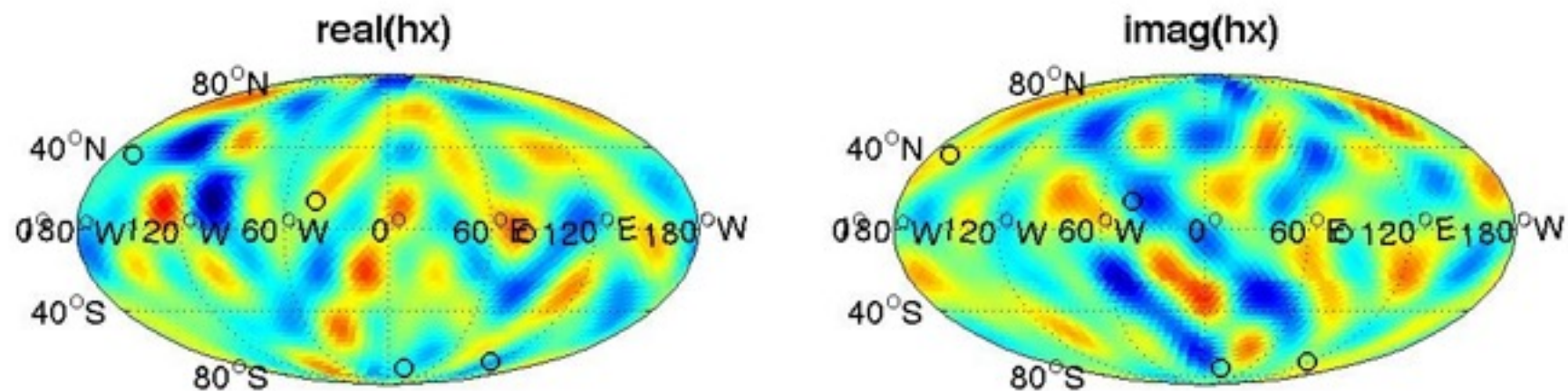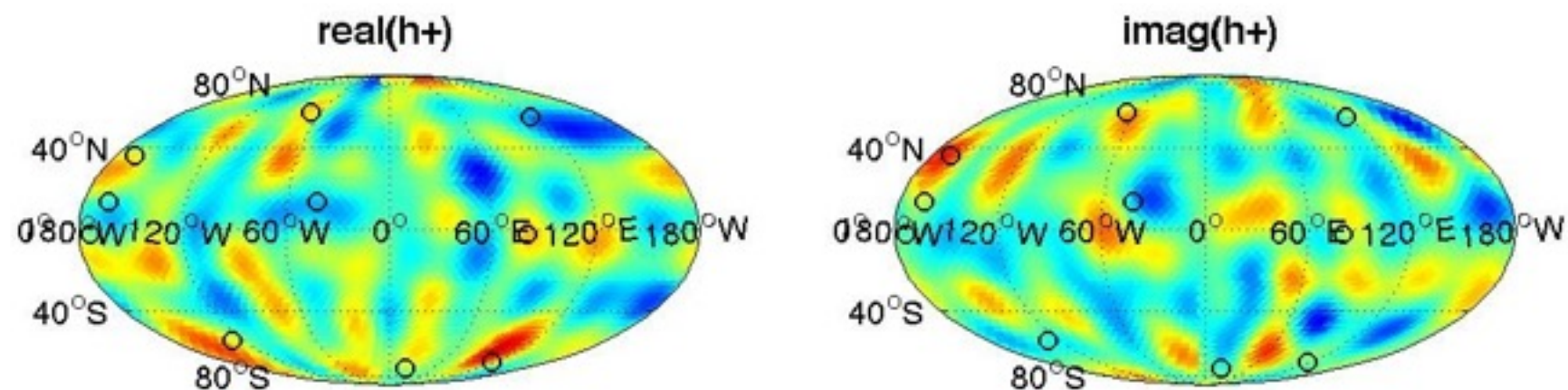

Injected background
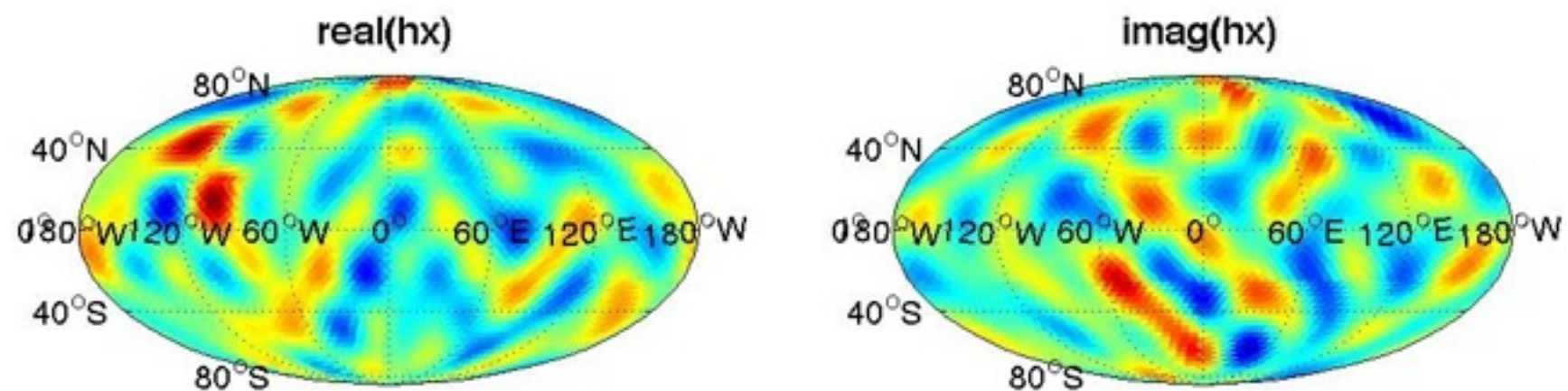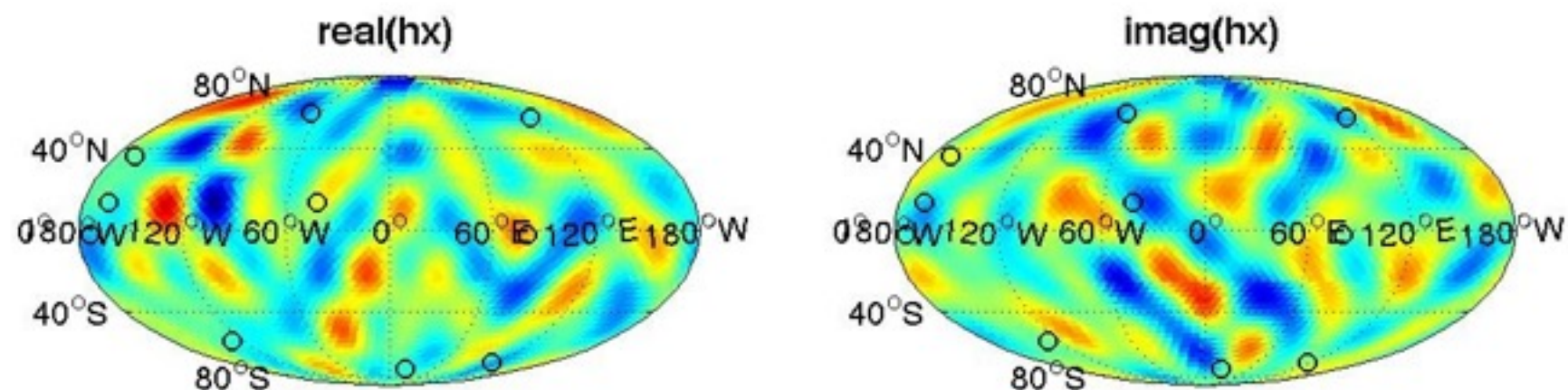
Background mapping: in practice

❖ Gradient piece of background behaves as expected. Adding more pulsars increases resolution of map and reduces residual.
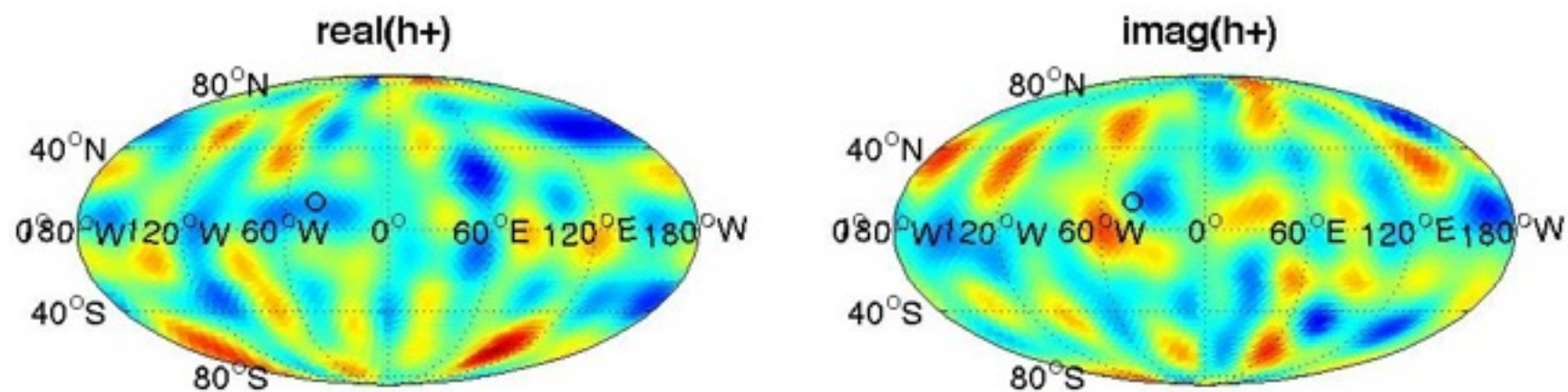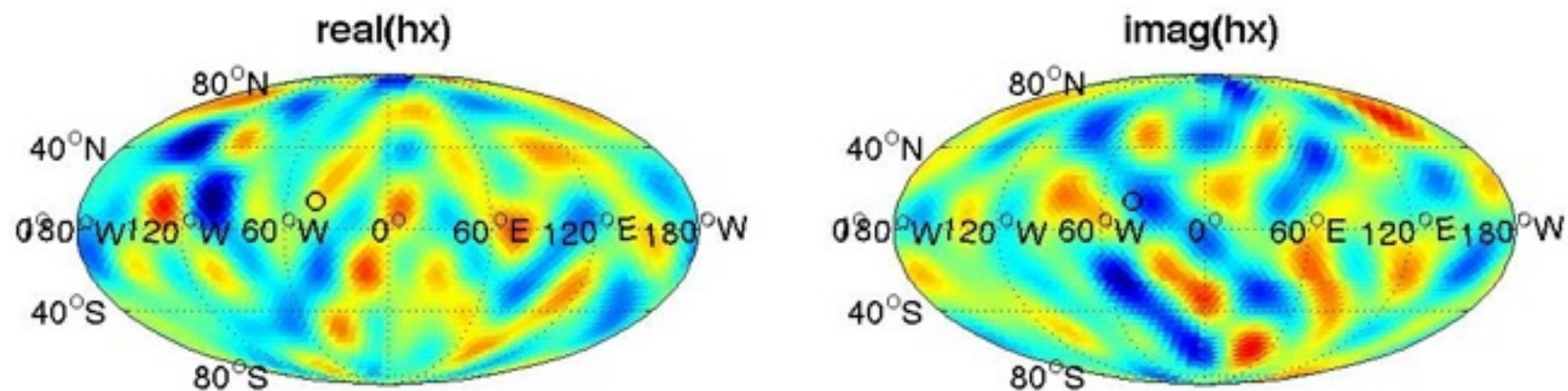
Residual $N_p=1$

Background mapping: in practice

❖ Gradient piece of background behaves as expected. Adding more pulsars increases resolution of map and reduces residual.

Residual N_p=2

Background mapping: in practice

❖ Gradient piece of background behaves as expected. Adding more pulsars increases resolution of map and reduces residual.
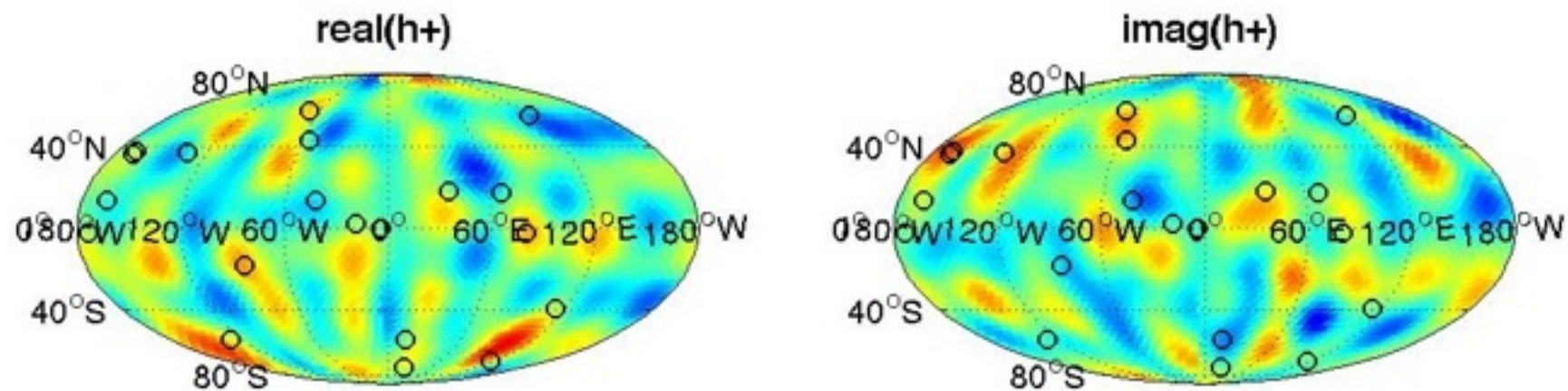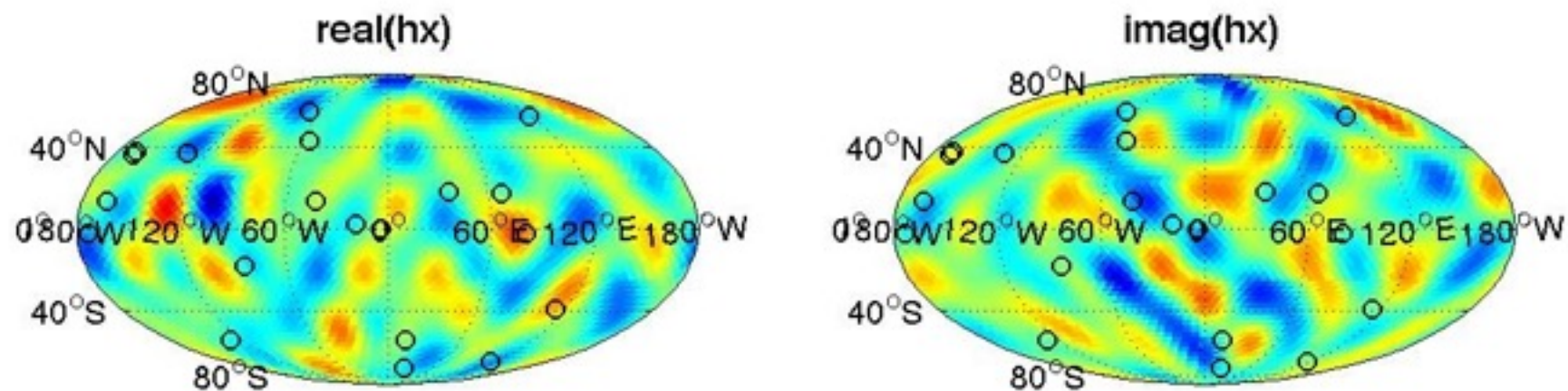
Residual $N_p=5$

Background mapping: in practice

❖ Gradient piece of background behaves as expected. Adding more pulsars increases resolution of map and reduces residual.

Residual N_p=10

Background mapping: in practice

❖ Gradient piece of background behaves as expected. Adding more pulsars increases resolution of map and reduces residual.
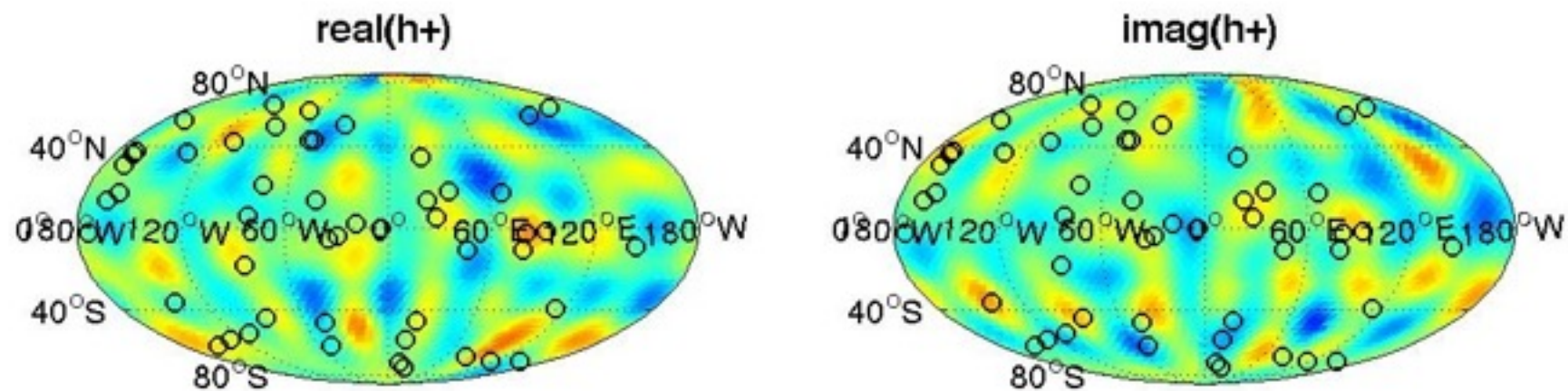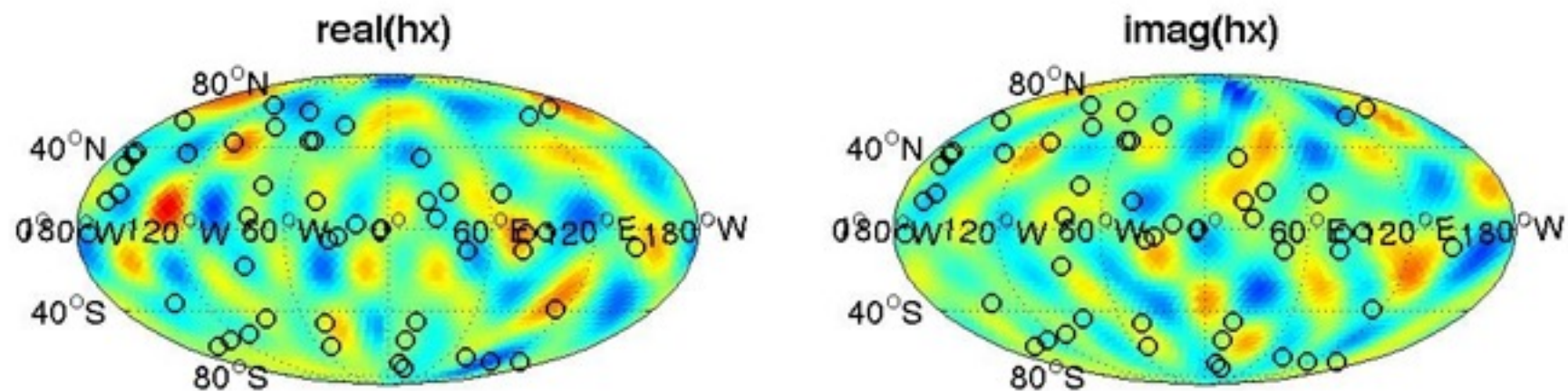
Residual N_p=20

Background mapping: in practice

❖ Gradient piece of background behaves as expected. Adding more pulsars increases resolution of map and reduces residual.

Residual N_p=50

Background mapping: in practice

❖ Gradient piece of background behaves as expected. Adding more pulsars increases resolution of map and reduces residual.

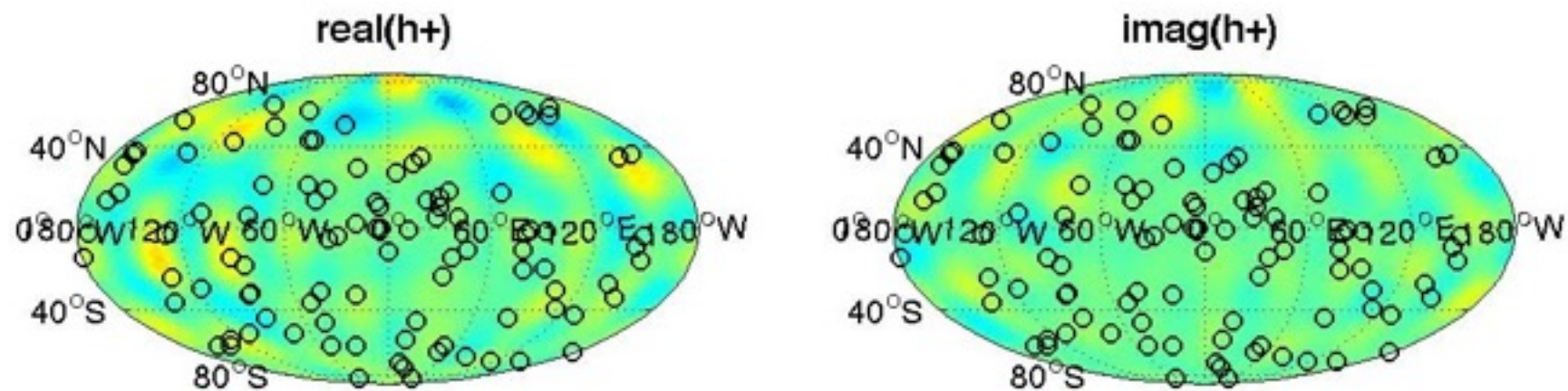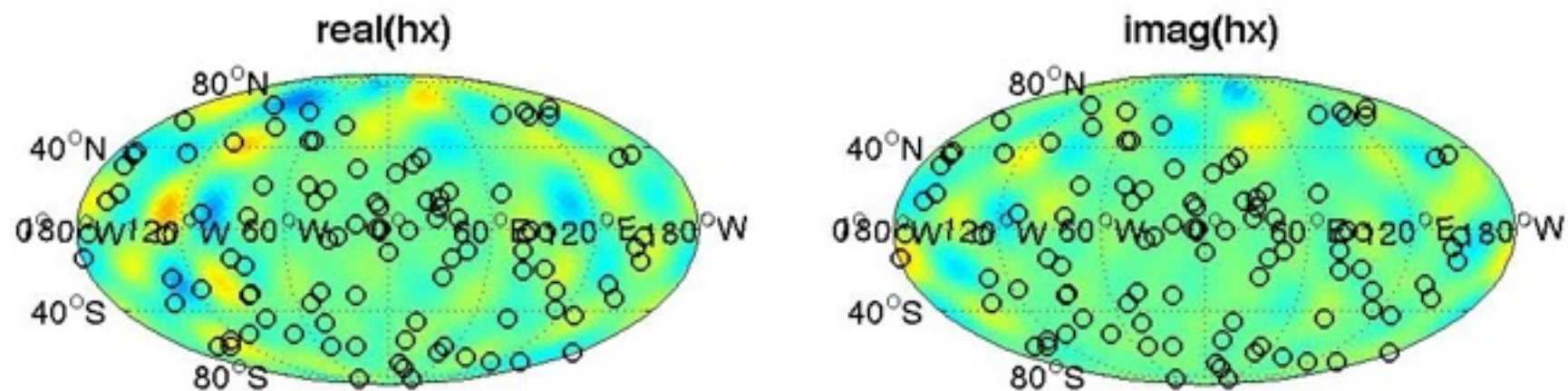

Residual N_p=100

Isotropic uncorrelated backgrounds

❖ An isotropic, uncorrelated and unpolarised background is described by the two-point functions

$$\langle h_+(f, \hat{k}) h_+^*(f', \hat{k}')\rangle = \langle h_\times(f, \hat{k}) h_\times^*(f', \hat{k}')\rangle = \frac{1}{2} H(f) \delta^2(\hat{k}, \hat{k}') \delta(f - f')$$

$$\langle h_+(f, \hat{k}) h_\times^*(f', \hat{k}')\rangle = \langle h_\times(f, \hat{k}) h_+^*(f', \hat{k}')\rangle = 0$$

❖ or in terms of the grad and curl expansion coefficients

$$\langle a_{(lm)}^G(f) a_{(l'm')}^{G*}(f')\rangle = \langle a_{(lm)}^C(f) a_{(l'm')}^{C*}(f')\rangle = \delta_{ll'} \delta_{mm'} H(f) \delta(f - f')$$

$$\langle a_{(lm)}^G(f) a_{(l'm')}^{C*}(f')\rangle = \langle a_{(lm)}^C(f) a_{(l'm')}^{G*}(f')\rangle = 0$$

❖ The expected correlation between the response of two pulsars for such a background is

$$\langle h_1(t) h_2(t')\rangle = \int_{-\infty}^{\infty} \mathrm{d}f \, e^{i2\pi f(t-t')} H(f) \Gamma_{12}(f)$$

$$\Gamma_{12}(f) = \sum_{l=2}^{\infty} \sum_{m=-l}^{l} \sum_P R_{1(lm)}^P(f) R_{2(lm)}^{P*}(f) = \sum_{l=2}^{\infty} (N_l)^2 (2l+1) \pi P_l(\hat{u}_1 \cdot \hat{u}_2)$$

Overlap reduction function: PTAs