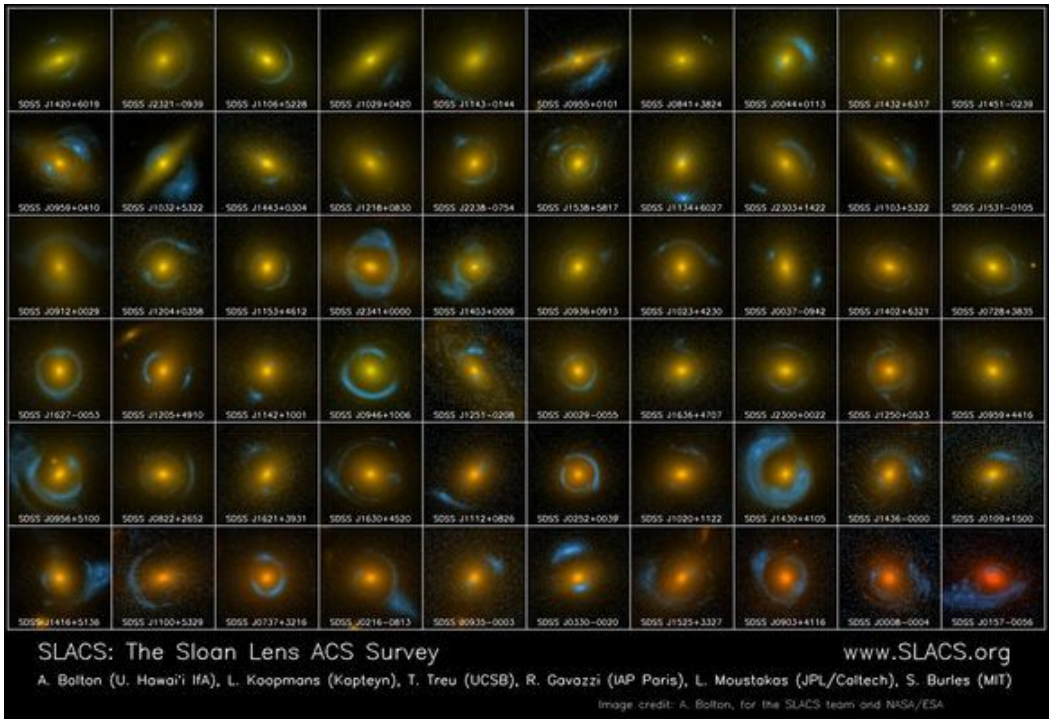
Fully Bayesian Analysis of Extremely Large Astronomy / Cosmology Datasets

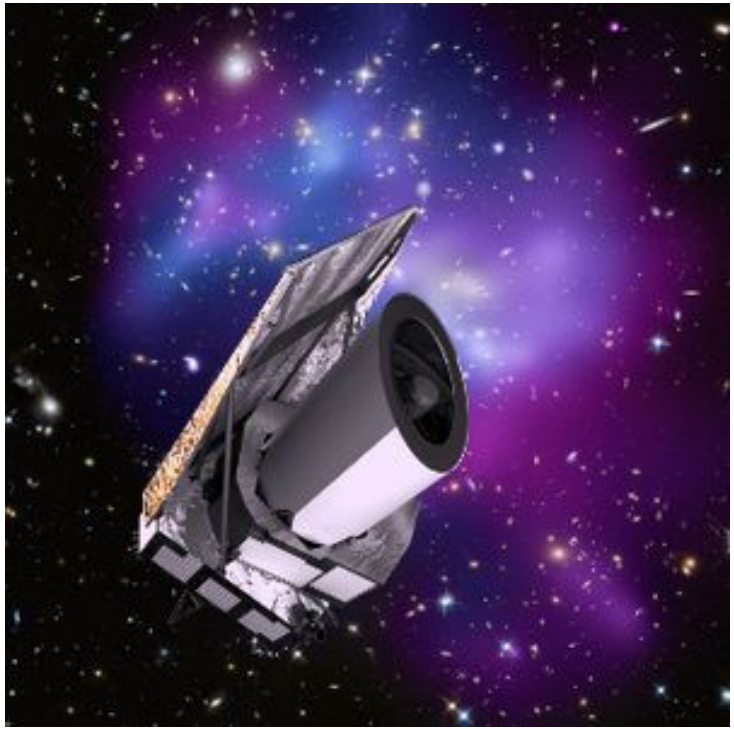
James Nightingale

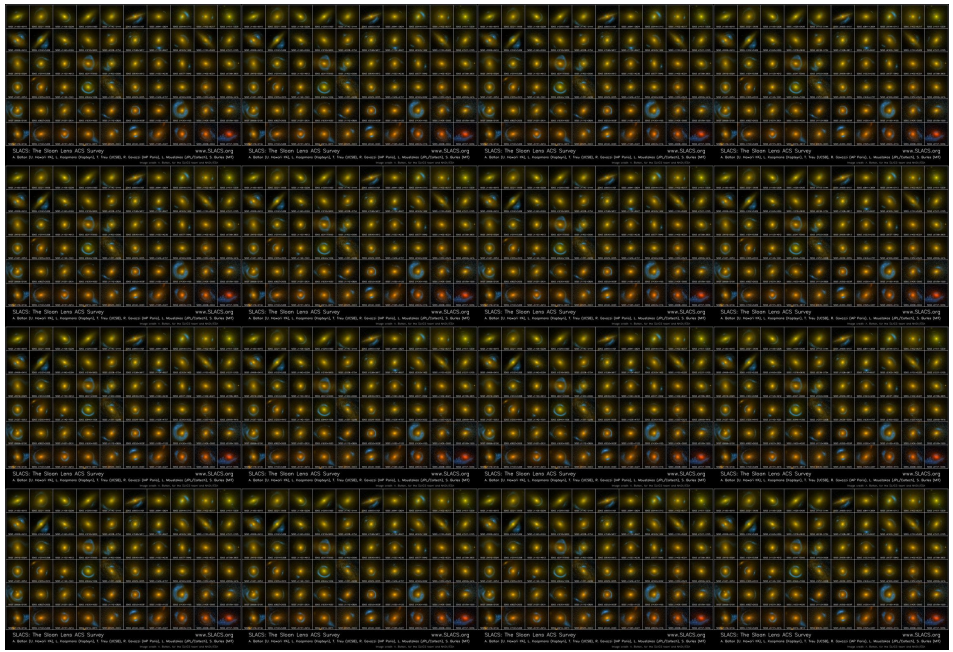
Richard Hayes, Matthew Griffiths, Richard Massey

Big Data – 50 years of strong lensing



Big Data – 1 Week of Euclid



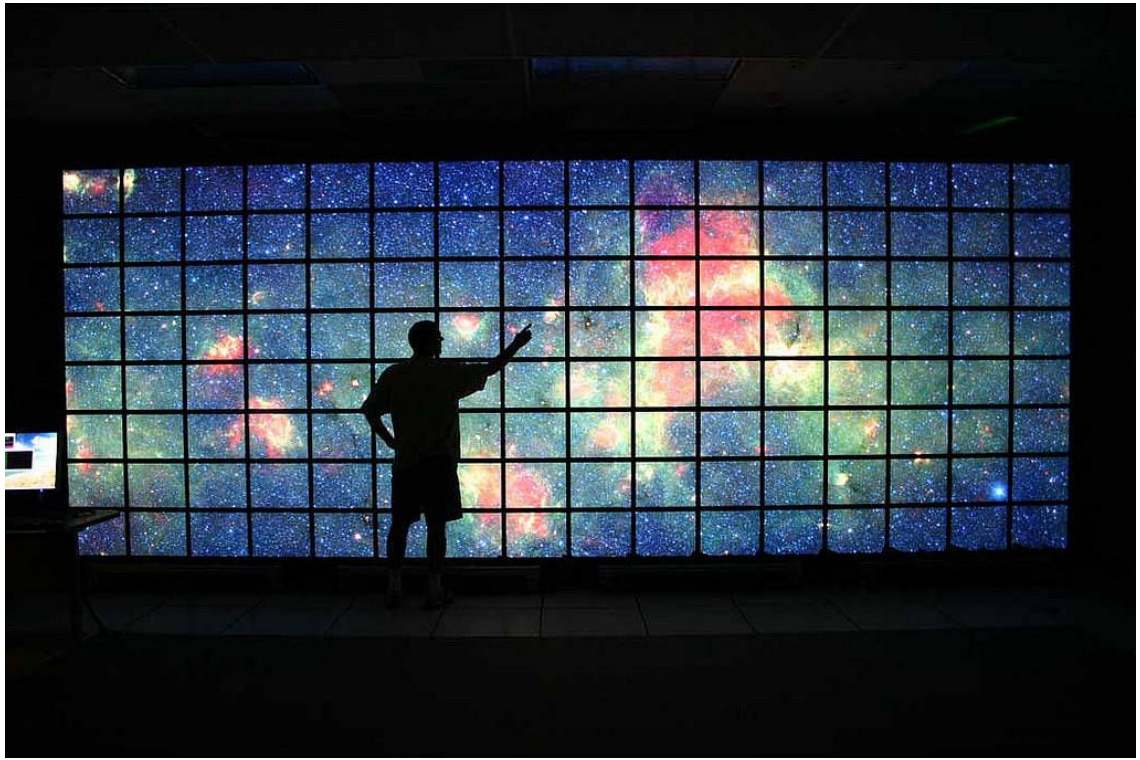


Big Data Analysis

We need statistical techniques that can:

- **Scale:** To datasets of this size.
- **Learn:** extract the wealth of meaningful information these datasets contain.

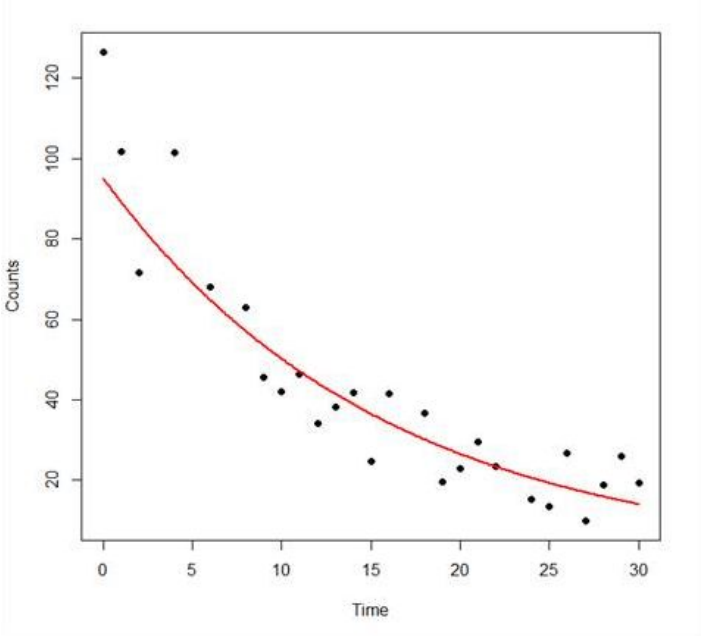
It would be nice if it was Bayesian too (i.e. no machine learning)!



Model Fitting

Model Fitting

Given some data and a model, finding the set of model parameters that provide the best fit to the data.



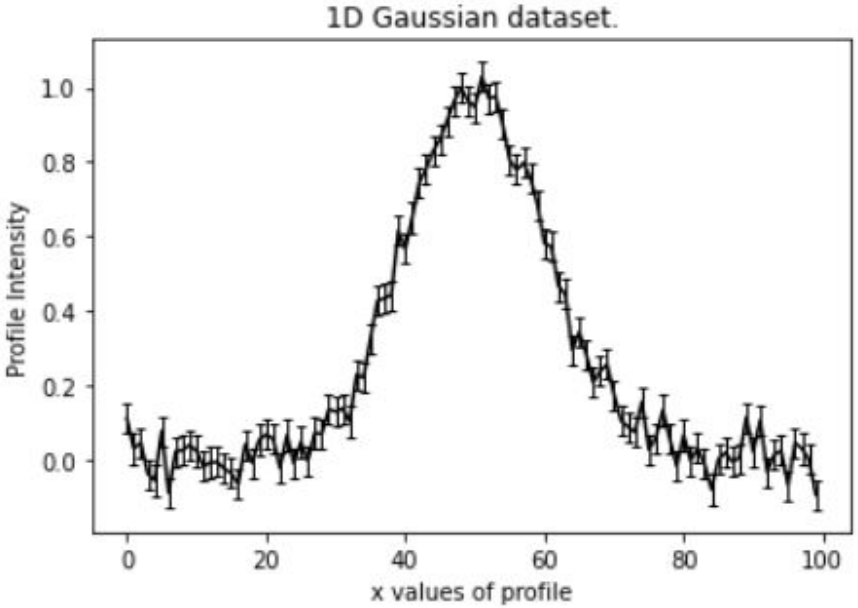
$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Model Fitting

Model -> Gaussian:

- Centre
- Normalization
- Sigma

1) Draw a set of parameters.
2) Create Model Gaussian.
3) Fit to Dataset.
4) Compute Likelihood.
5) Repeat using non-linear search.

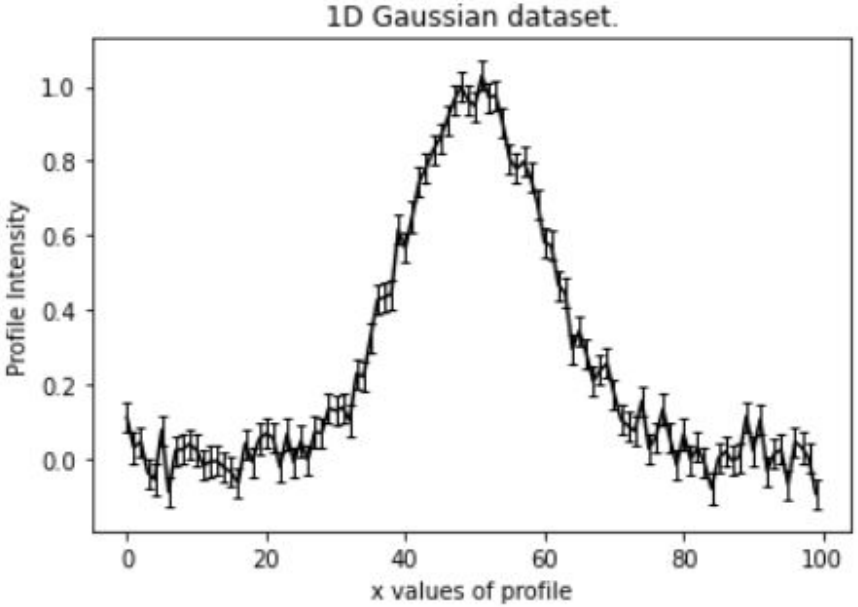


Model Fitting

Model -> Gaussian:

- Centre **= 60.0**
- Normalization **= 20.0**
- Sigma **= 15.0**

1) **Draw a set of parameters.**
2) Create Model Gaussian.
3) Fit to Dataset.
4) Compute Likelihood.
5) Repeat using non-linear search.

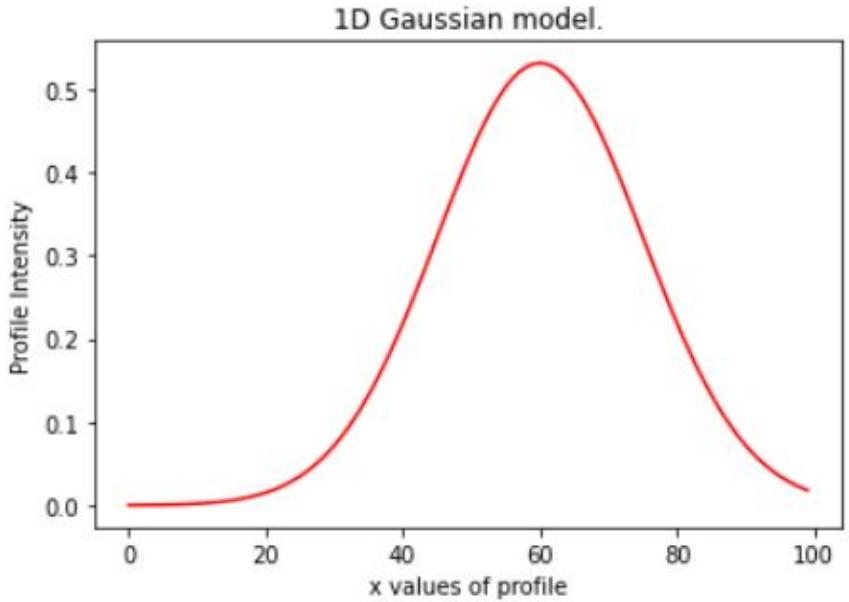


Model Fitting

Model -> Gaussian:

- Centre **= 60.0**
- Normalization **= 20.0**
- Sigma **= 15.0**

1) Draw a set of parameters.
2) **Create Model Gaussian.**
3) Fit to Dataset.
4) Compute Likelihood.
5) Repeat using non-linear search.

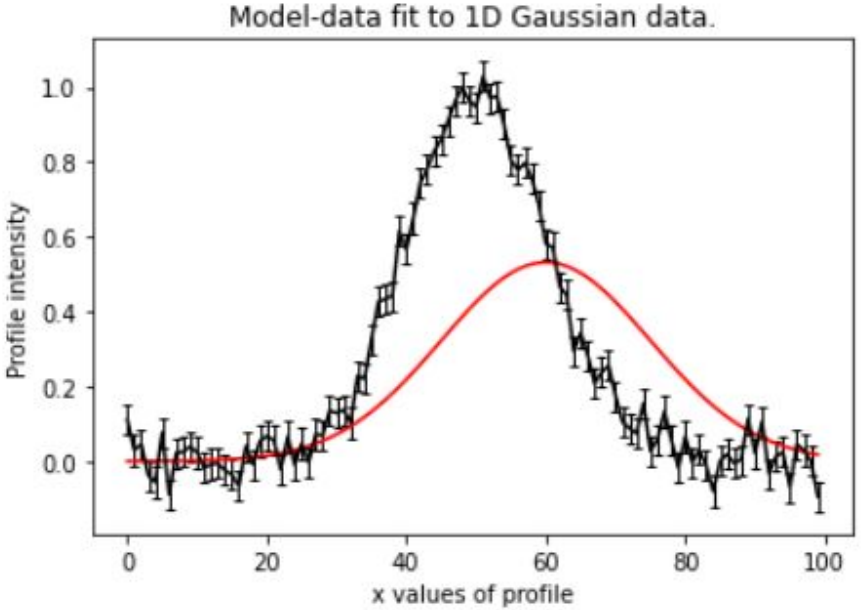


Model Fitting

Model -> Gaussian:

- Centre **= 60.0**
- Normalization **= 20.0**
- Sigma **= 15.0**

1) Draw a set of parameters.
2) Create Model Gaussian.
3) **Fit to Dataset.**
4) Compute Likelihood.
5) Repeat using non-linear search.

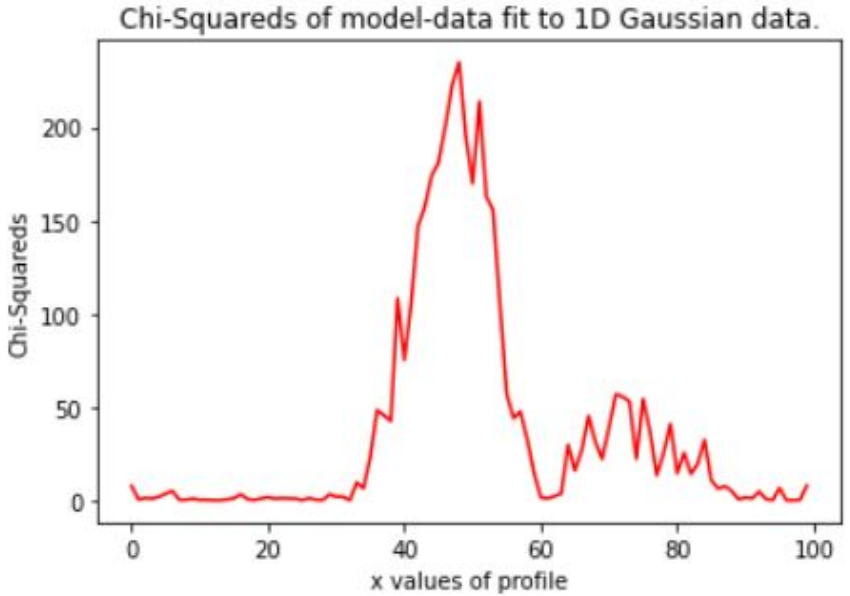


Model Fitting

Model -> Gaussian:

- Centre **= 60.0**
- Normalization **= 20.0**
- Sigma **= 15.0**

1) Draw a set of parameters.
2) Create Model Gaussian.
3) Fit to Dataset.
4) **Compute Likelihood.**
5) Repeat using non-linear search.

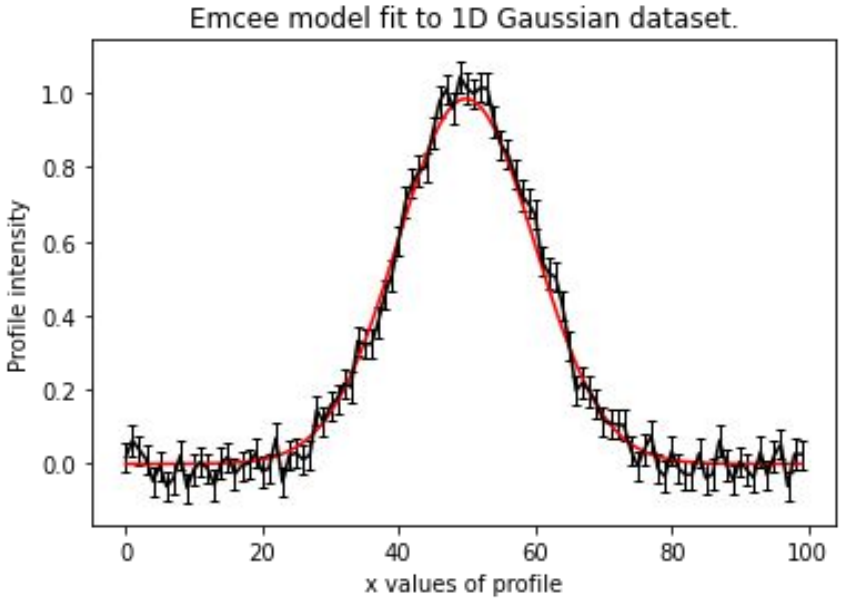


Model Fitting

Model -> Gaussian:

- Centre **= 50.0**
- Intensity **= 10.0**
- Sigma **= 5.0**

1) Draw a set of parameters.
2) Create Model Gaussian.
3) Fit to Dataset.
4) Compute Likelihood.
5) **Repeat using non-linear search.**



Non-linear Search

Many different methods for fitting the model to data (e.g. integrating the likelihood function, sampling parameter space, model-fitting).

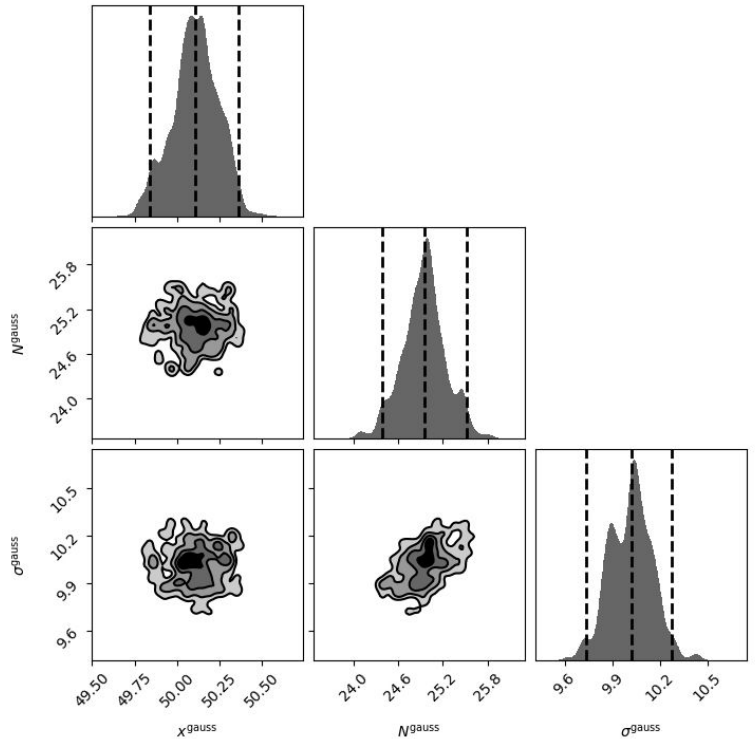
- Markov Chain Monte Carlo (MCMC).
- Maximum Likelihood Estimators (MLE).
- Nested sampling.

I will use the nested sampler dynesty (https://arxiv.org/abs/1904.02180) throughout this talk.

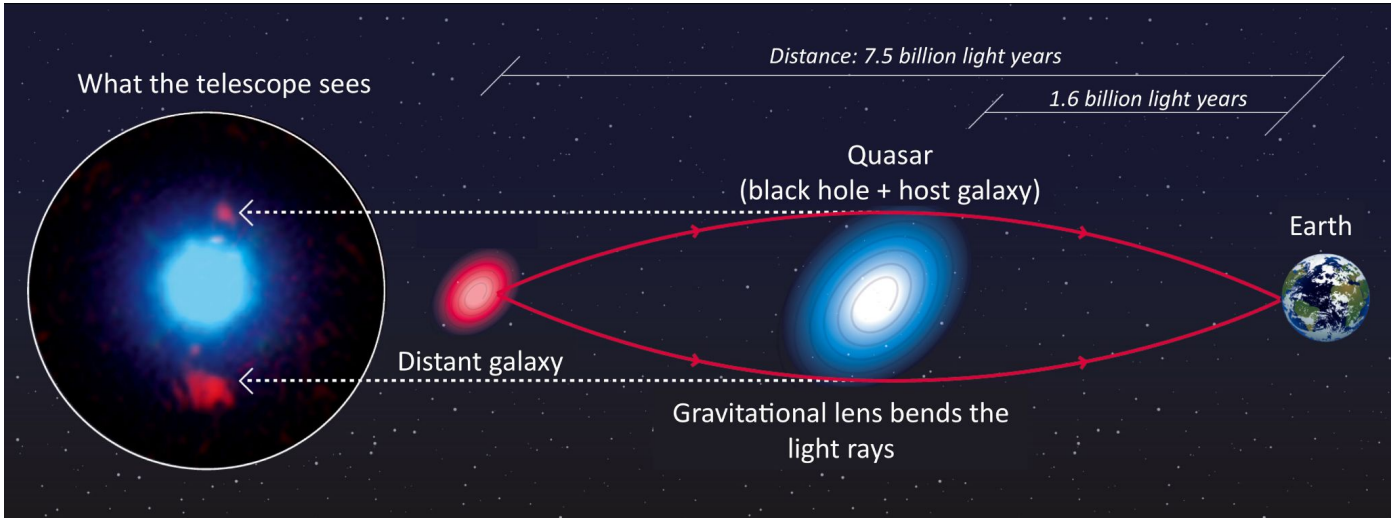
Nothing I show depends on the choice of non-linear search.

Probability Distribution Function

Results of model-fit give probability distribution function (PDF) of parameters in 1D, 2D and more dimensions.



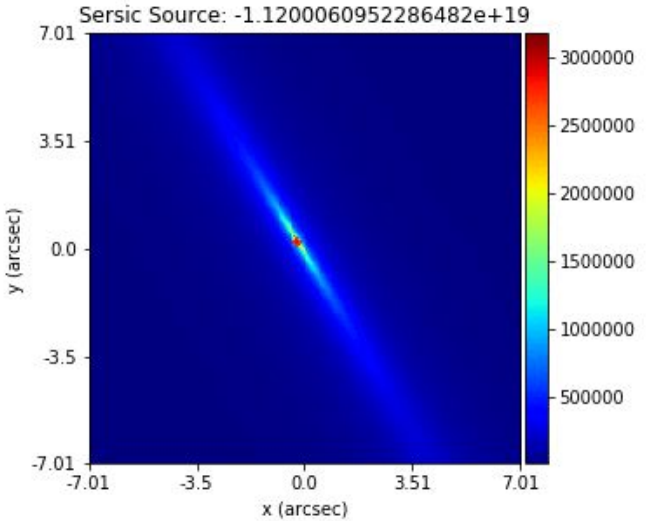
Strong Gravitational Lensing

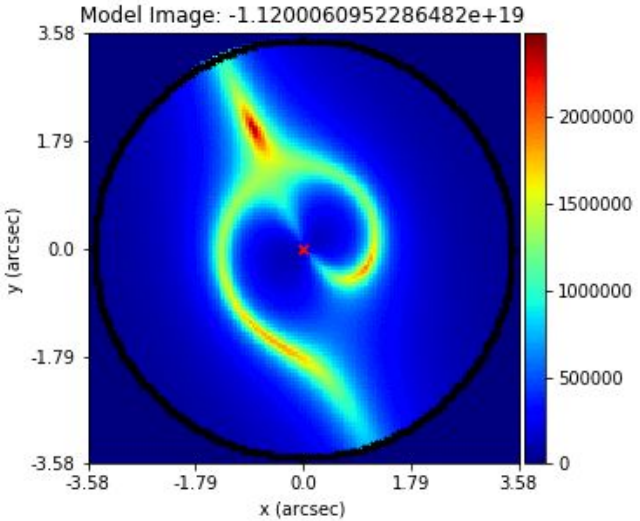


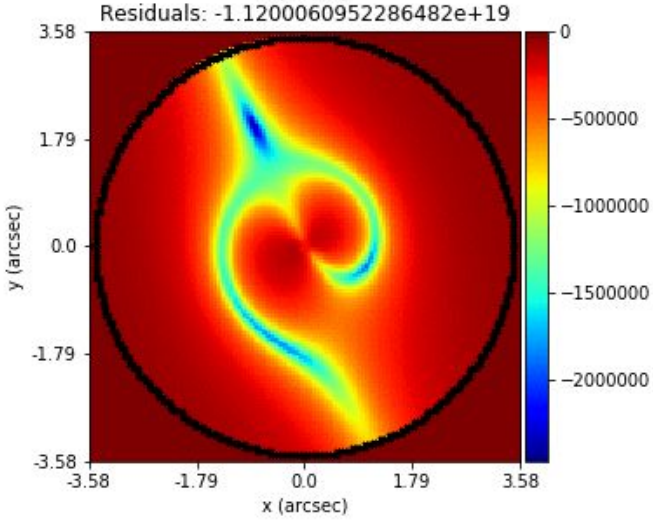
Bayesian Methods

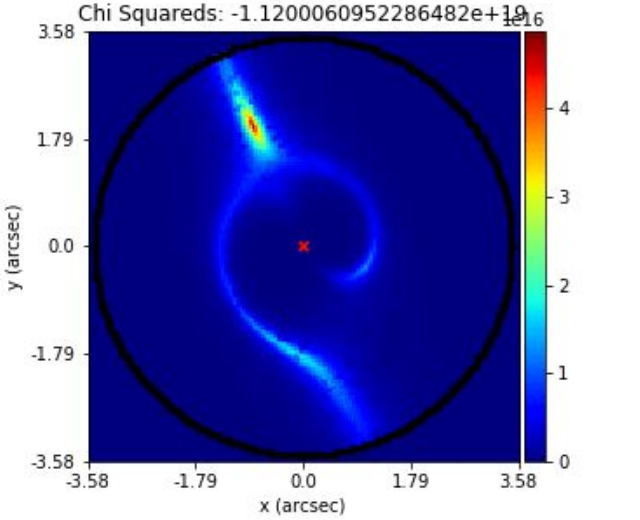
For example, use **nested sampling** to analyse Astronomy data.

Loads of tools for statistical inference required (model composition, advanced inference methods, managing large results, etc.)







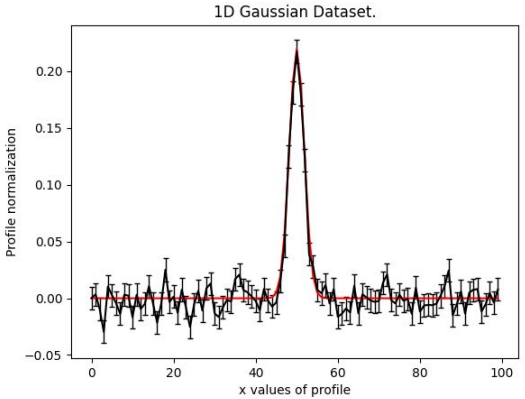


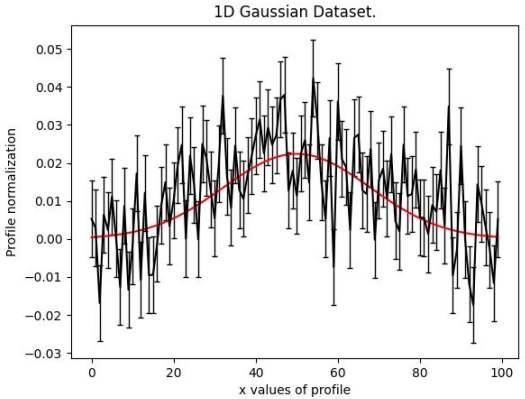
Fitting Multiple Datasets

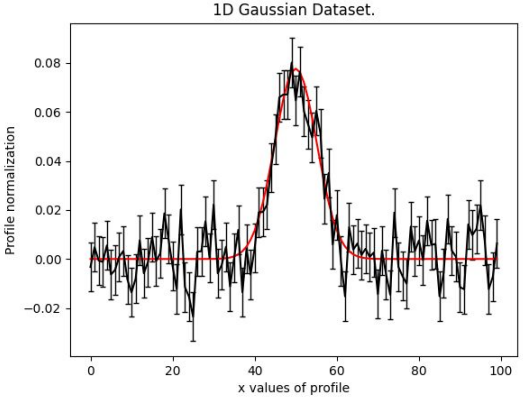
Toy Model: 1D Gaussian Pulses

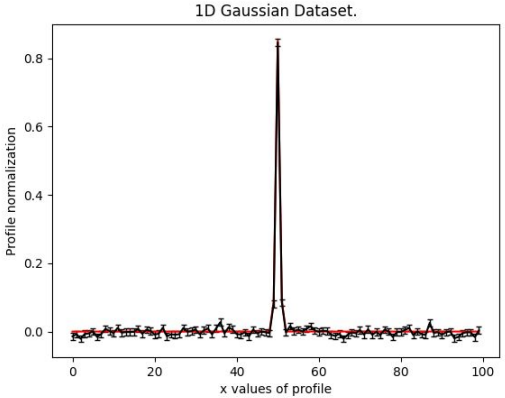
We observe many 1D Gaussians which all have the same centre (50 pixels).

Goal: Estimate this centre.









Weighted Average / Joint PDF

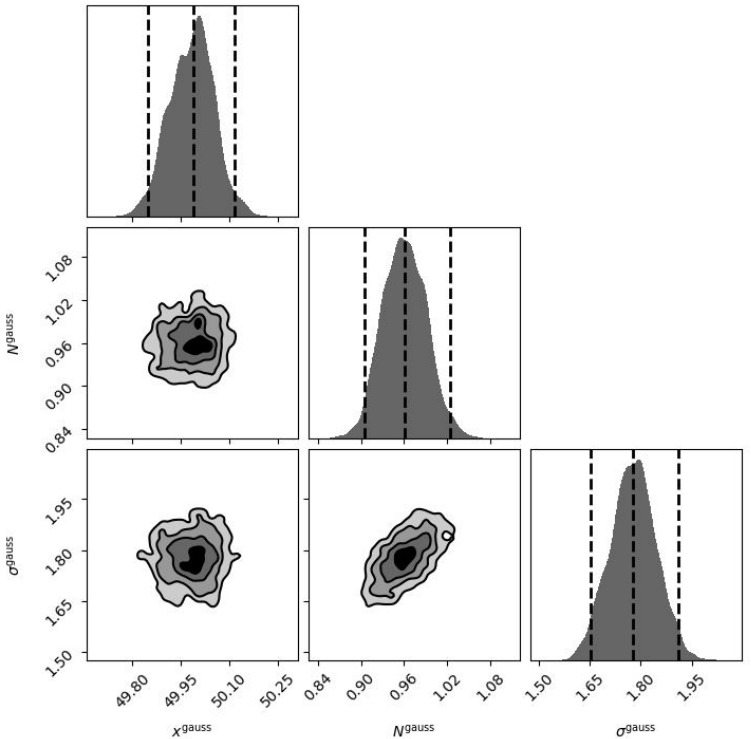
The “traditional” approach.

Fit each dataset, one-by-one, and combine inferred centres at the end.

- Could take the **weighted average** of the inferred centre of every fit.

- Could multiply the inferred likelihoods of each fit and then multiply by prior (requires use of **Kernel Density Estimators).**

Number of free parameters in each fit N = 3.



Weighted Average / Joint PDF

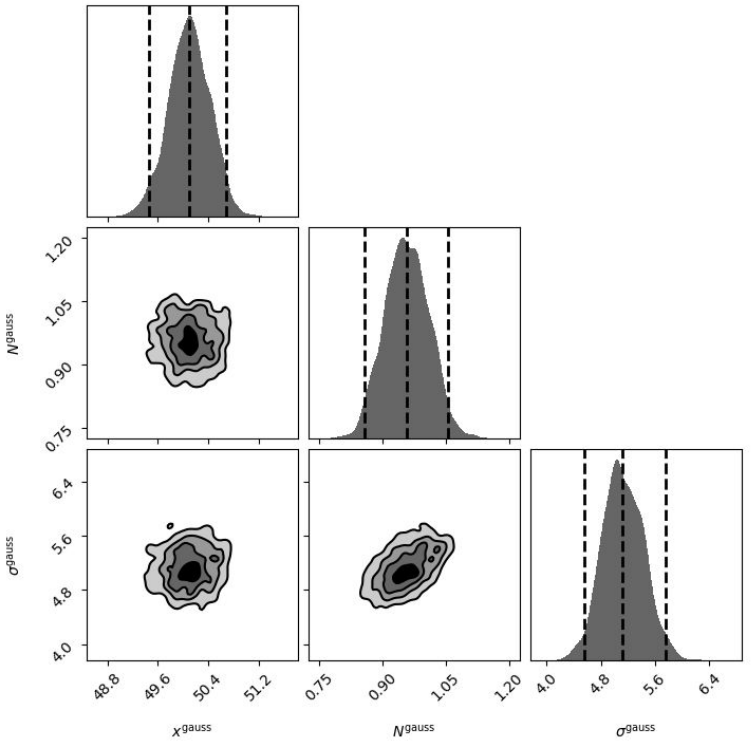
The “traditional” approach.

Fit each dataset, one-by-one, and combine inferred centres at the end.

- Could take the **weighted average** of the inferred centre of every fit.

- Could multiply the inferred likelihoods of each fit and then multiply by prior (requires use of **Kernel Density Estimators).**

Number of free parameters in each fit N = 3.



Weighted Average / Joint PDF

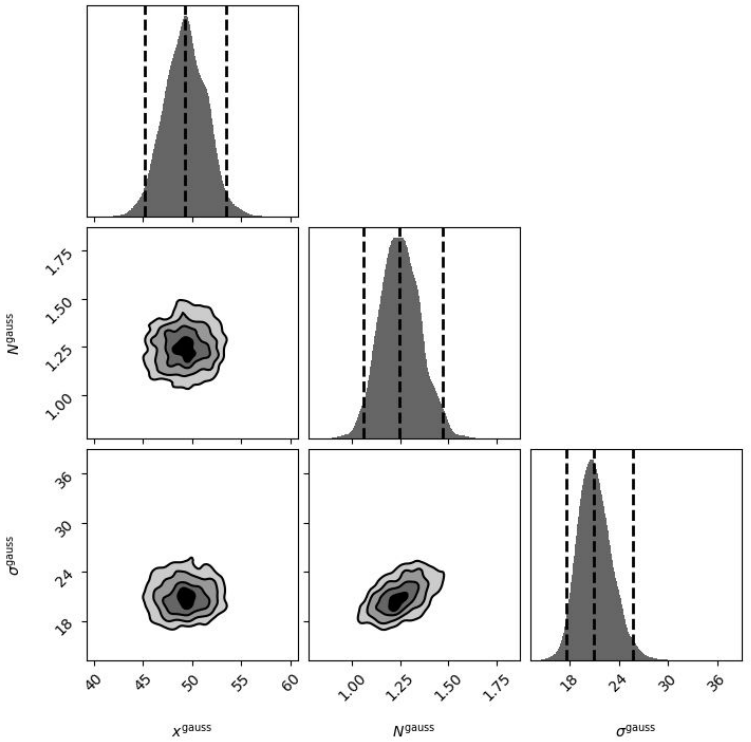
The “traditional” approach.

Fit each dataset, one-by-one, and combine inferred centres at the end.

- Could take the **weighted average** of the inferred centre of every fit.

- Could multiply the inferred likelihoods of each fit and then multiply by prior (requires use of **Kernel Density Estimators).**

Number of free parameters in each fit N = 3.



Weighted Average / Joint PDF

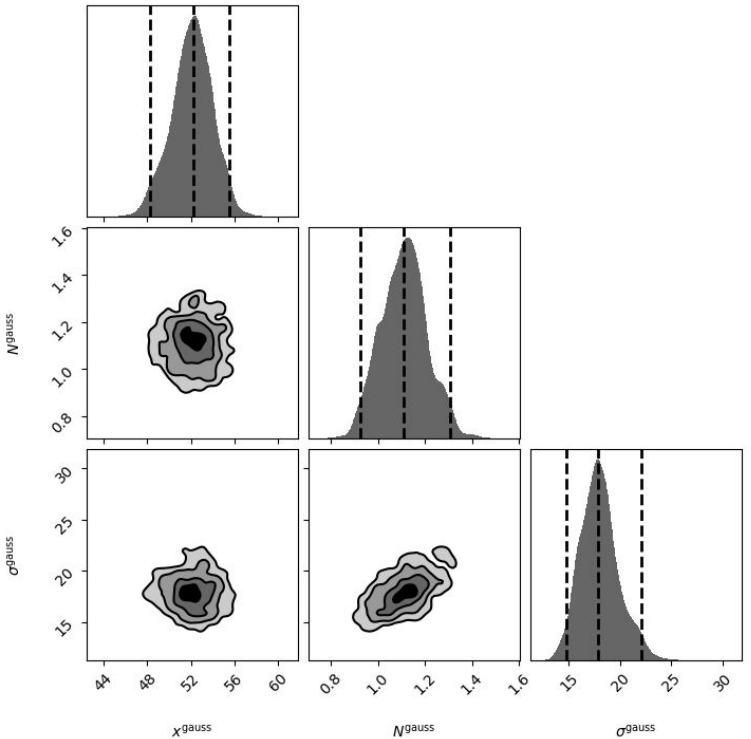
The “traditional” approach.

Fit each dataset, one-by-one, and combine inferred centres at the end.

- Could take the **weighted average** of the inferred centre of every fit.

- Could multiply the inferred likelihoods of each fit and then multiply by prior (requires use of **Kernel Density Estimators).**

Number of free parameters in each fit N = 3.

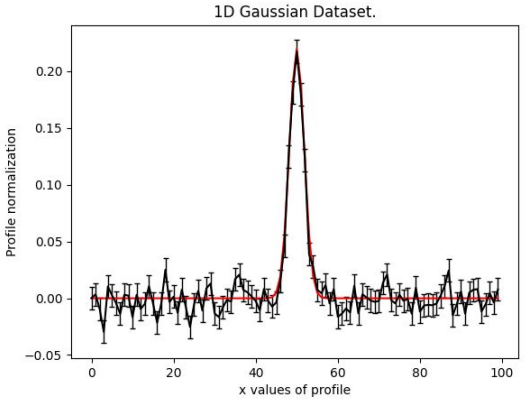


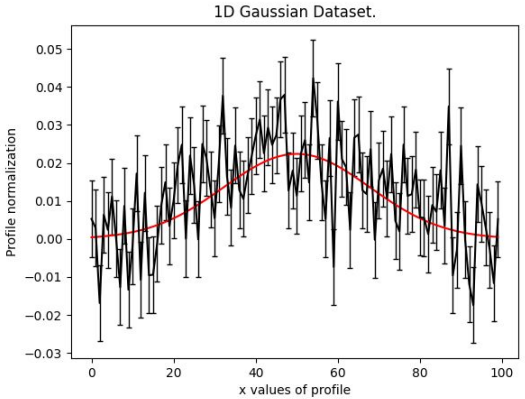
Weighted Average / Joint PDF

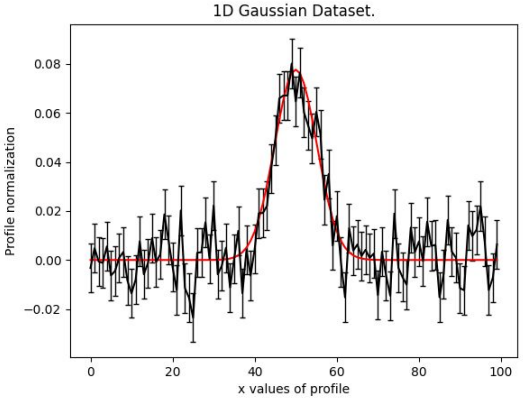
Weighted average gives:

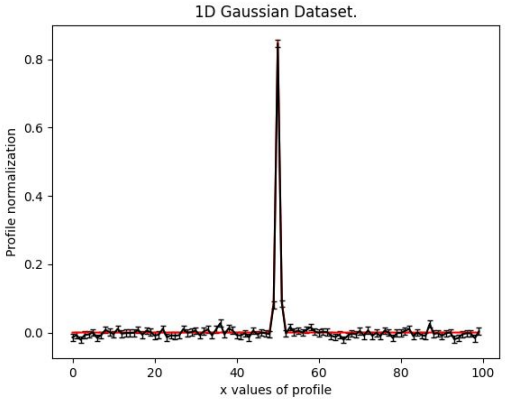
Centre = 50.004 +- 0.035

It works!









Graphical Models

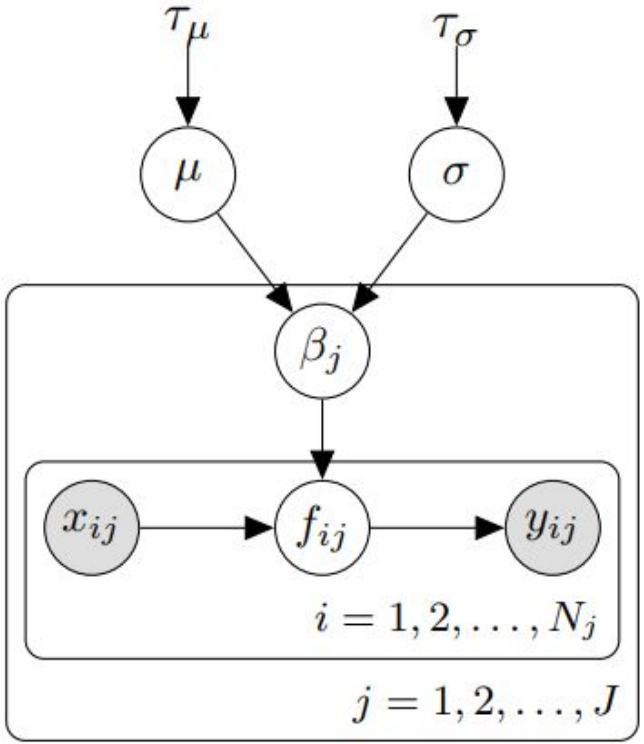
Graphical Model

Fit all 4 datasets simultaneously, with a single shared centre parameter across all datasets.

Traditional Fit: N = 4 x 3 = 12 free parameters (3 parameters per fit).

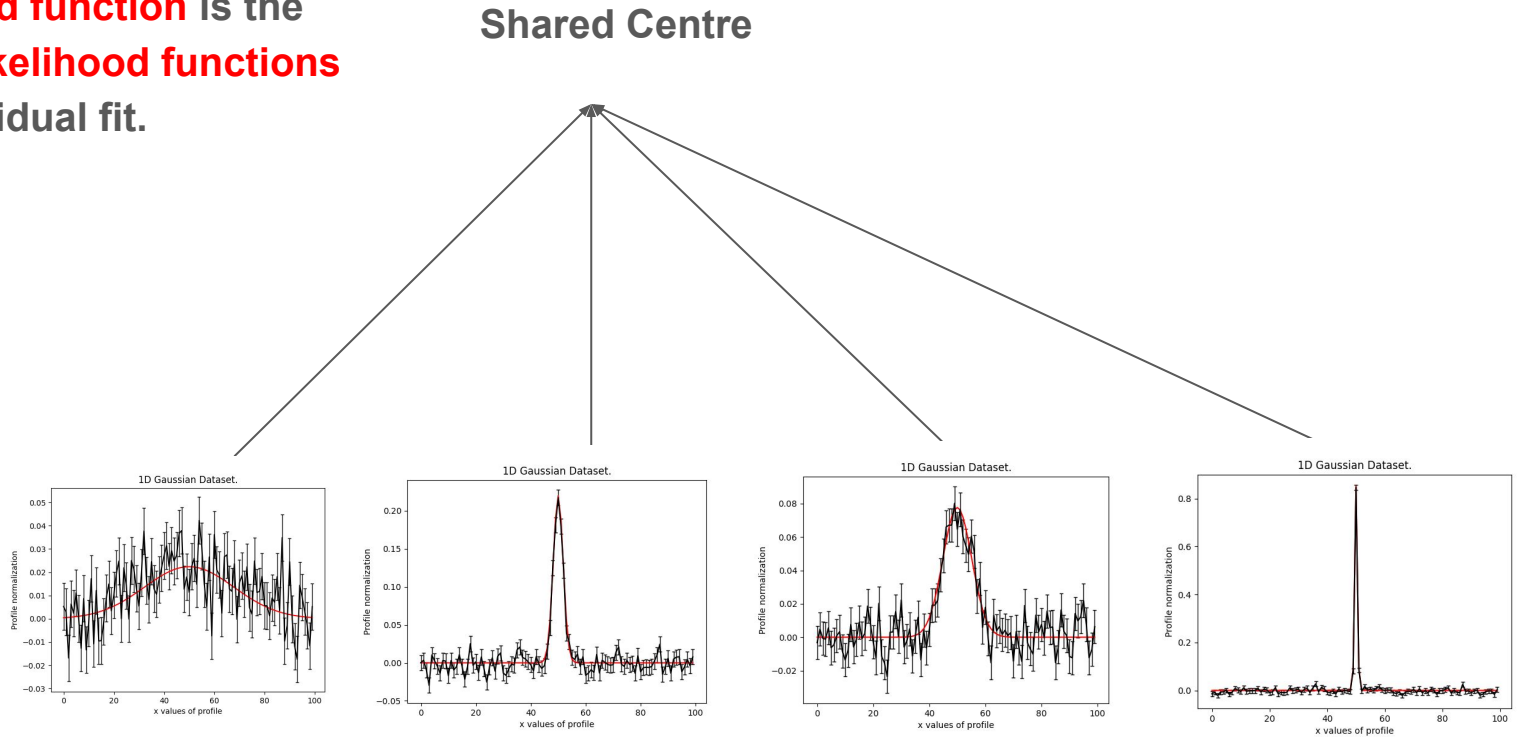
Graphical: Number of free parameters in fit N = 9 (x4 σ, x4 normalizations, x1 centre).

The likelihood function is the sum of the likelihood functions of each individual fit.



Graph

The likelihood function is the sum of the likelihood functions of each individual fit.



■ ■ ■

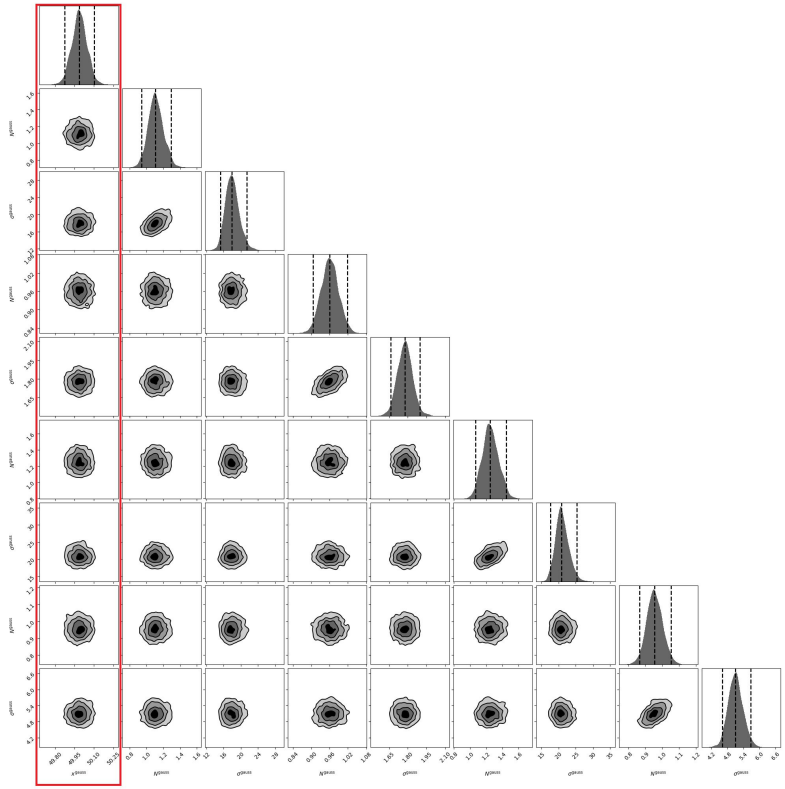
Weighted Average / Joint PDF

The graphical model gives:

Centre = 50.12 +- 0.30 [1**σ**]

In agreement with the simpler traditional approach:

Centre = 50.14 +- 0.25 [1**σ**]

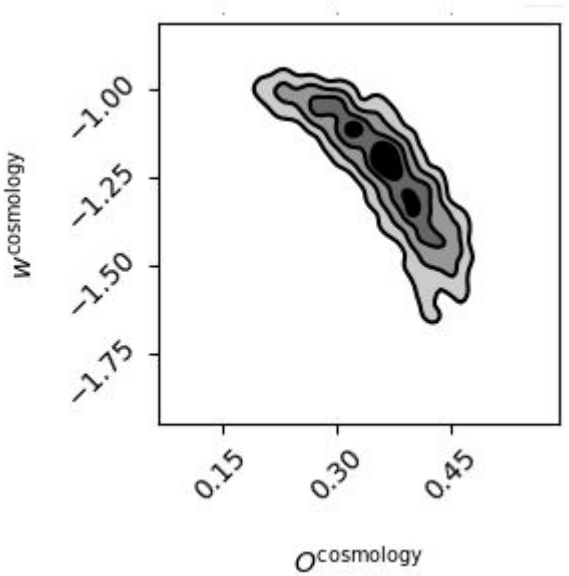


Benefits of Graphical Model:

Well defined priors:

- **Simple Approach:** the prior on the centre of each dataset is defined multiple times.
- **Graphical Model:** Only one prior, consistent with dataset properties.

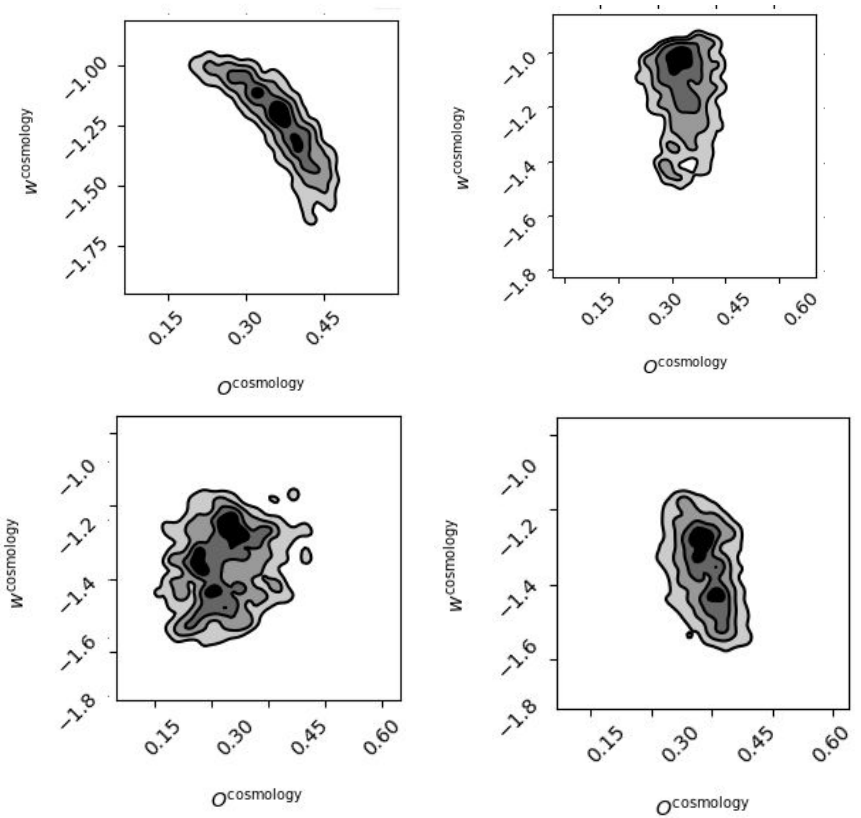
Cannot make hierarchical.



Benefits of Graphical Model:

Multiplying posteriors together (e.g. via Kernel Density Estimators) is unreliable for high dimensionality (e.g. smoothing produces errors).

No smoothing for a graphical model, final posterior is sampled.



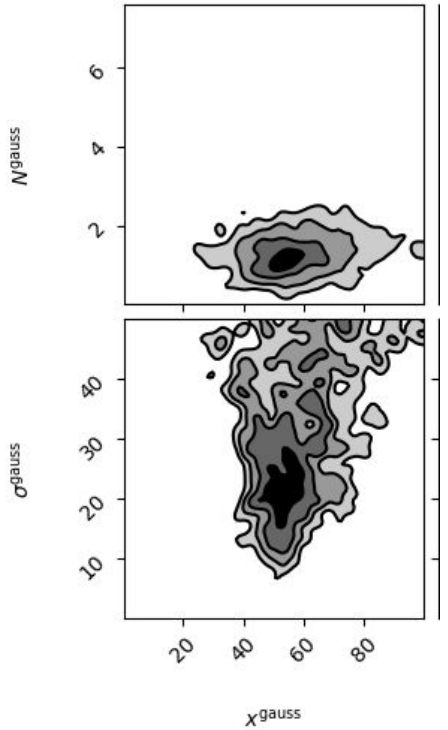
Nuisance Parameter Inference

Graphical model improves individual parameter constraints (e.g. σ / N).

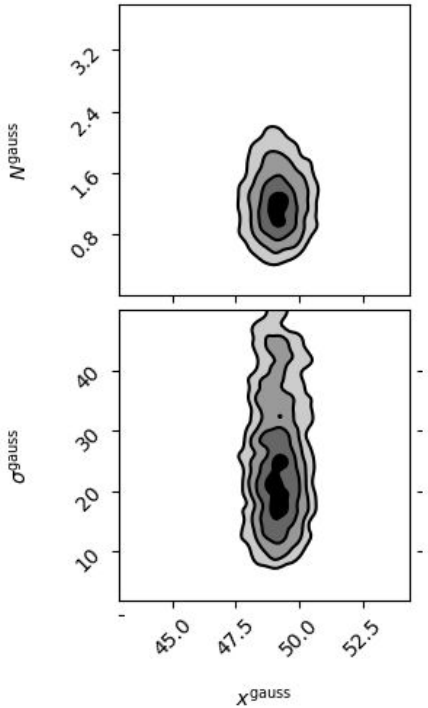
The fit to each dataset now "knows" about the constraints from the other datasets.

For individual model-fits this is not the case!

Individual Models:



Graphical Model:

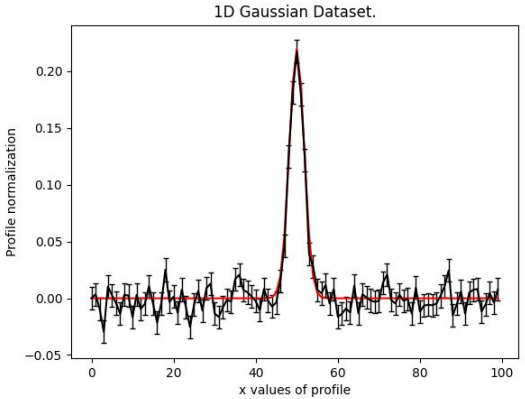


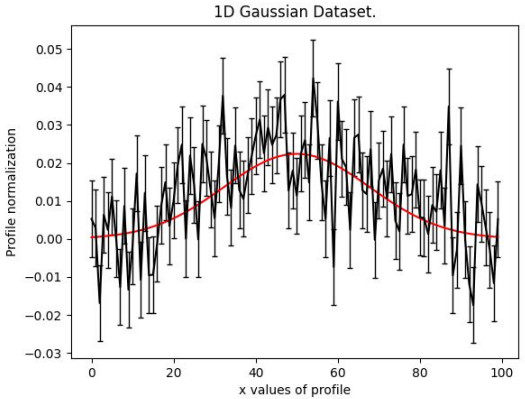
Weighted Average / Joint PDF

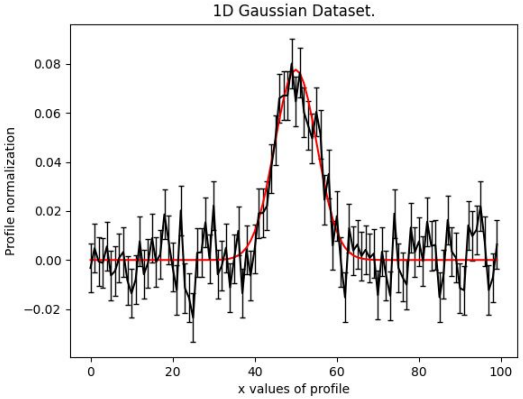
Graphical model improves individual parameter constraints (e.g. σ / N).

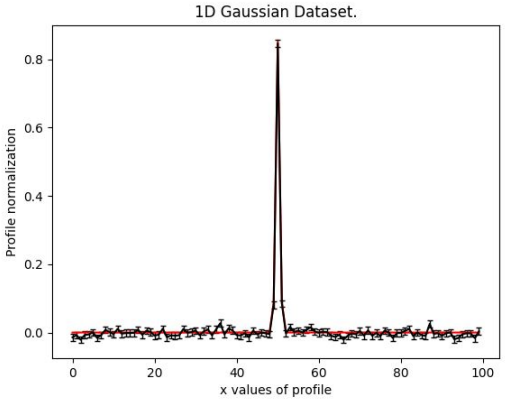
The fit to each dataset now "knows" about the constraints from the other datasets.

For individual model-fits this is not the case!









Weighted Average / Joint PDF

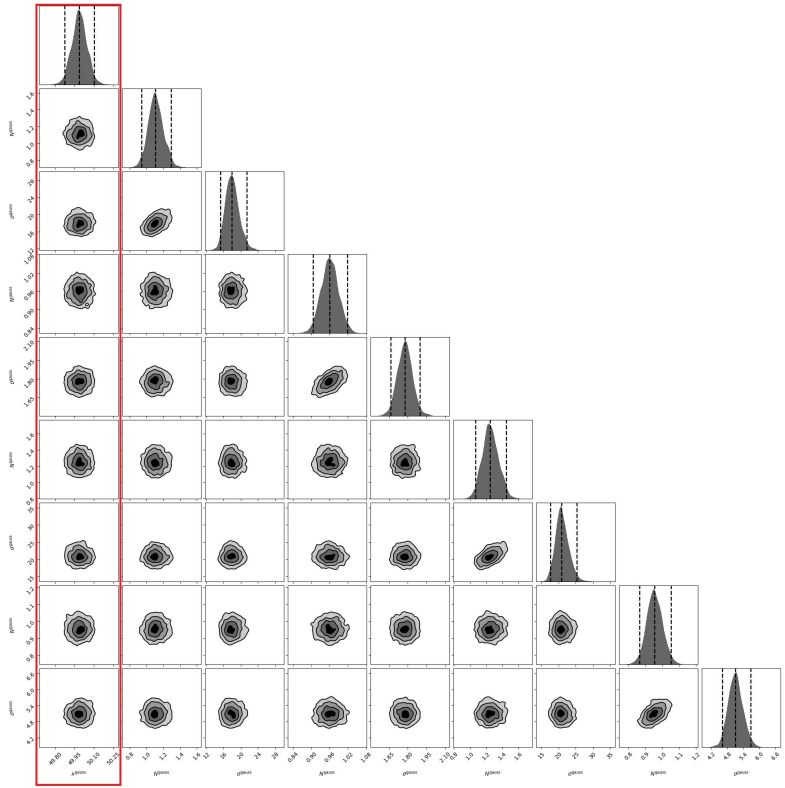
The graphical model gives:

Centre = 50.12 +- 0.30 [1σ]

In agreement with the simpler traditional approach:

Centre = 50.14 +- 0.25 [1σ]

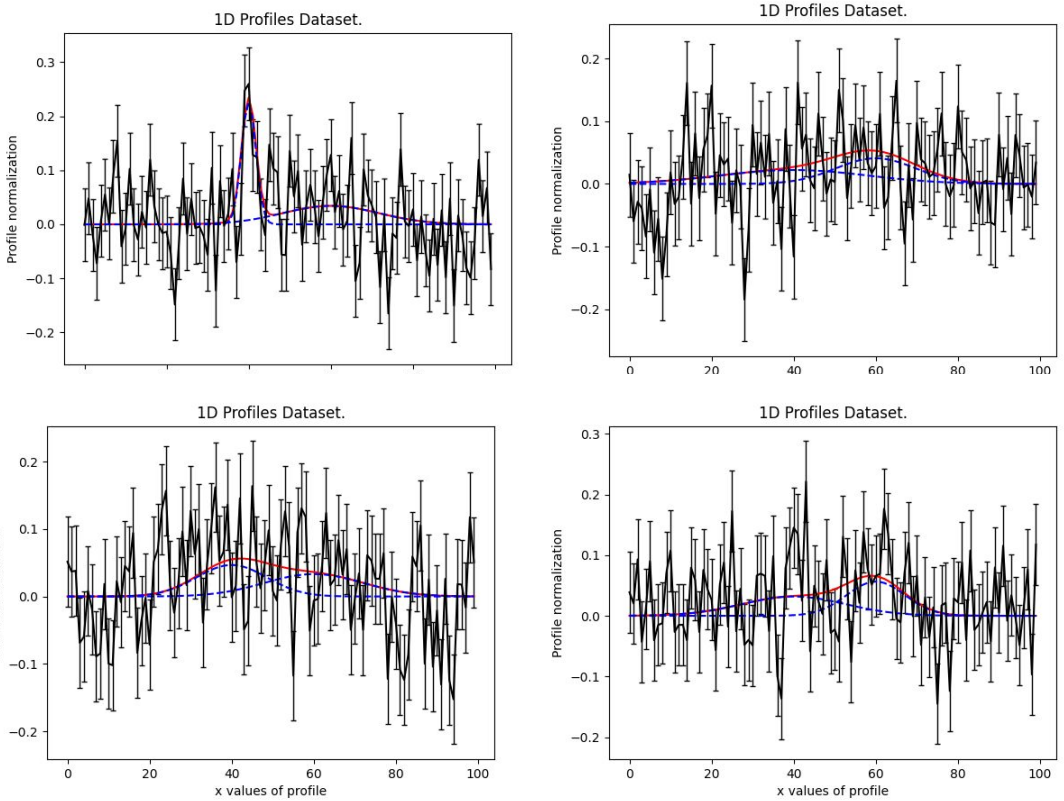
Formally for posterior multiplication they should give the same result.



Toy Model: x2 1D Gaussian Pulses Low S/N

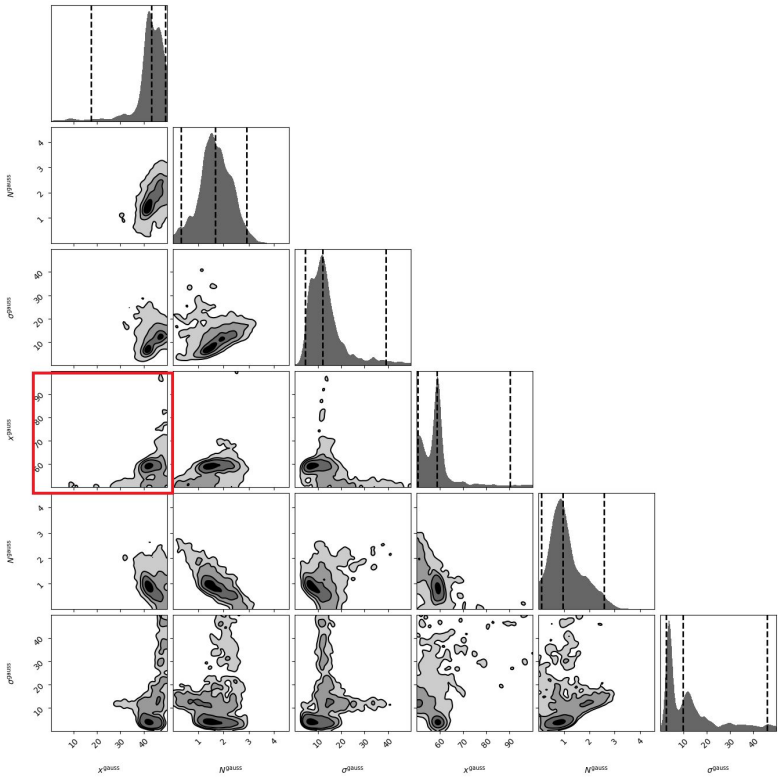
We observe many pairs of 1D Gaussians which all have the same centre (40 & 60 pixels).

Goal: Estimate both centres.



Individual Fits

Give a poor estimate of the centres due to degeneracy of centres in each model-fit.



Individual Fits

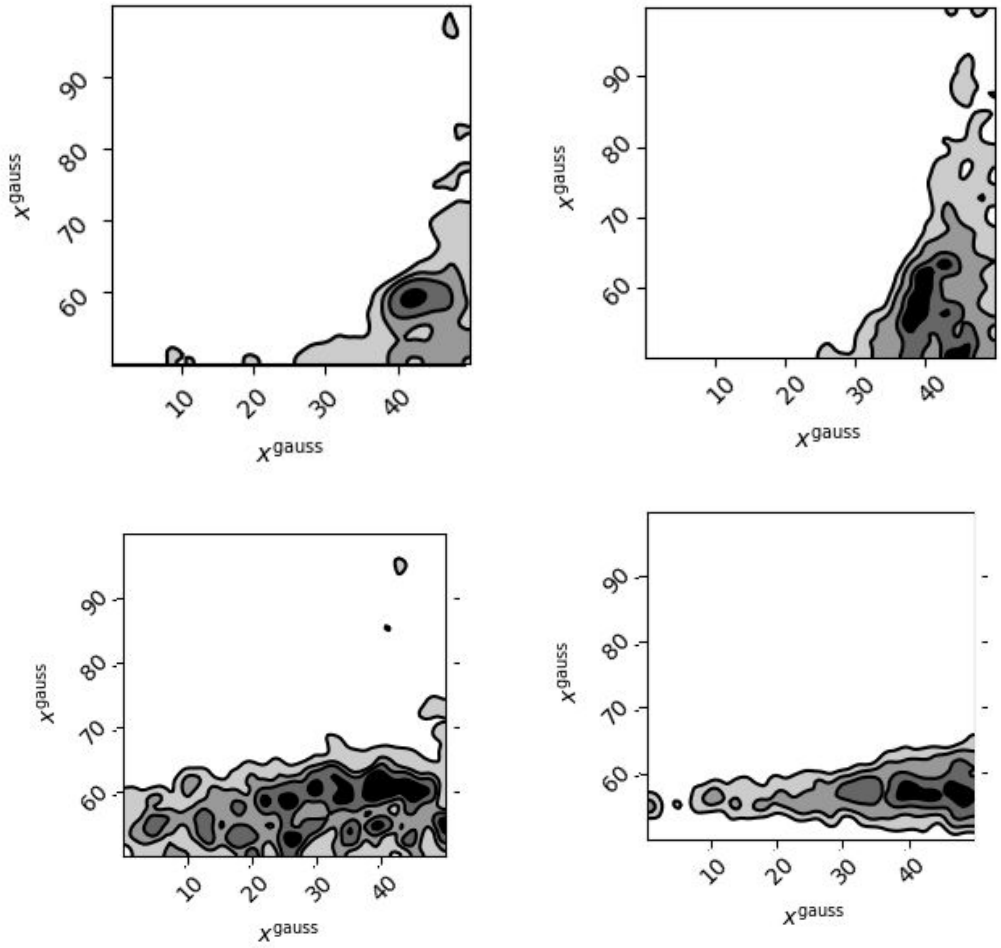
Give a poor estimate of the centres due to degeneracy of centres in each model-fit.

True Centres:

40.0 and 60.0

Inferred values:

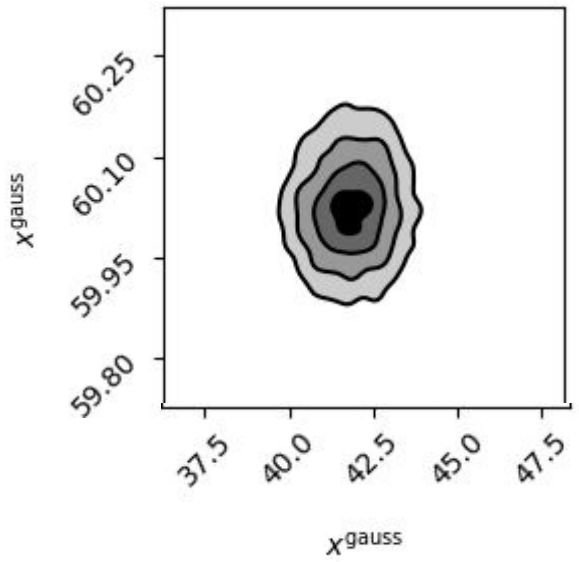
29.51 +- 7.10 [1**σ**]
75.03 +- 8.58 [1**σ**]



Graphical Model

Properly accounts for degeneracies when sampling parameters:

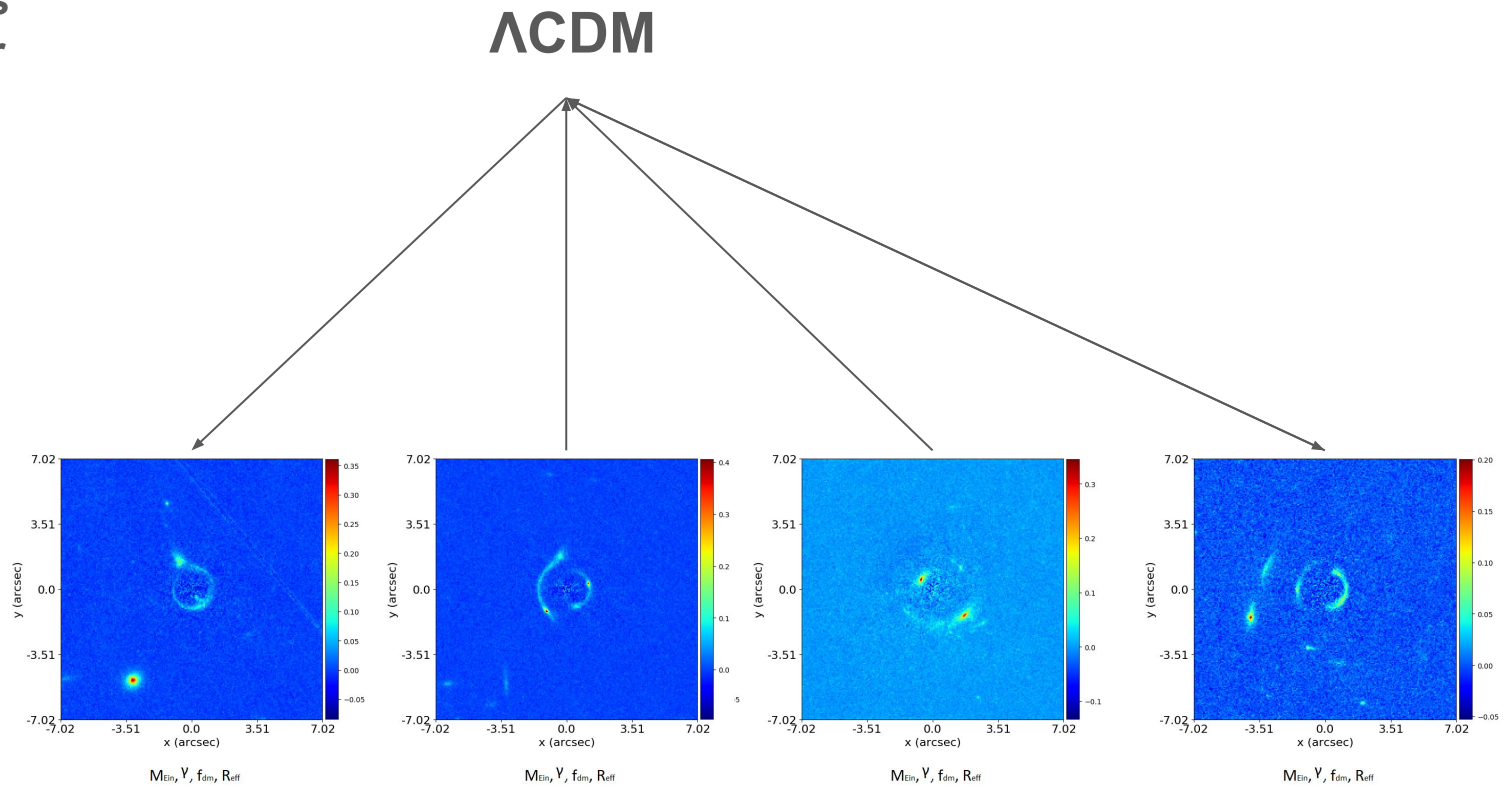
41.52 +- 3.78 [1**σ**]
60.06 +- 2.03 [1**σ**]



Where am I going with this?

The Gaussians are galaxies or strong lenses.

The shared centre(s) are cosmological Parameters.



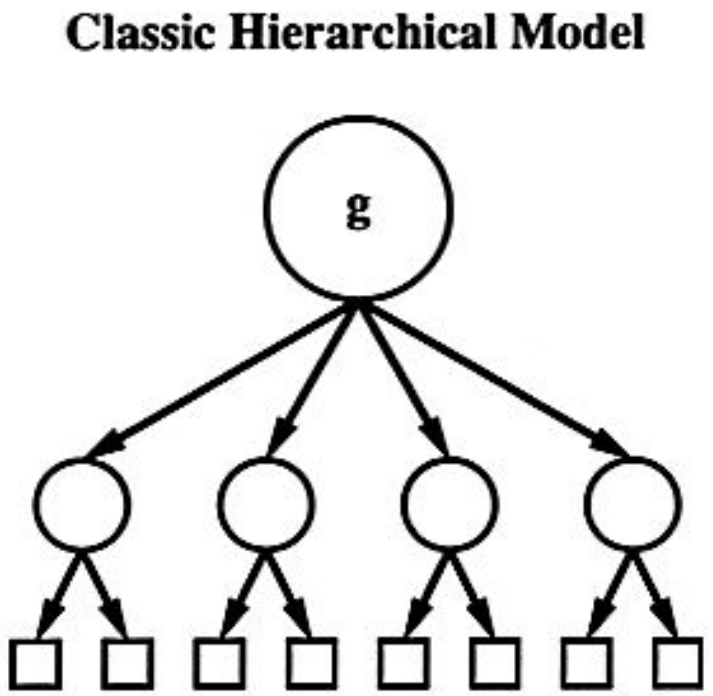
Hierarchical Models

Hierarchical Models (“Bayesian Hierarchical Analysis”)

Posit that certain parameters are drawn from a parent distribution.

For example, a Gaussian with (mean, scatter) = (μ, σ)

- Break degeneracies in fits to individual datasets.

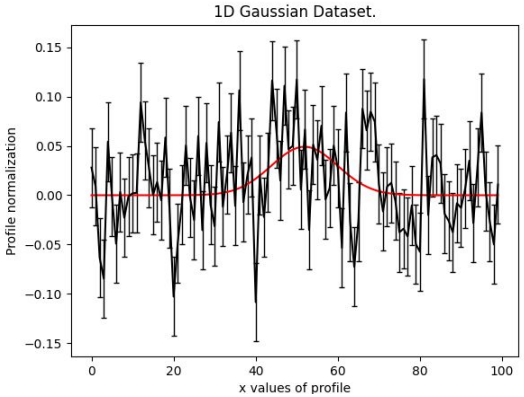


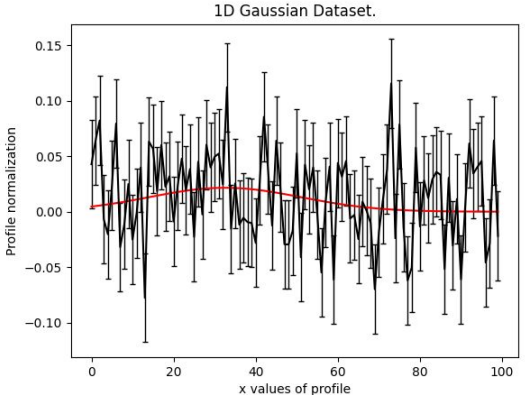
Toy Model: 1D Gaussian Pulses

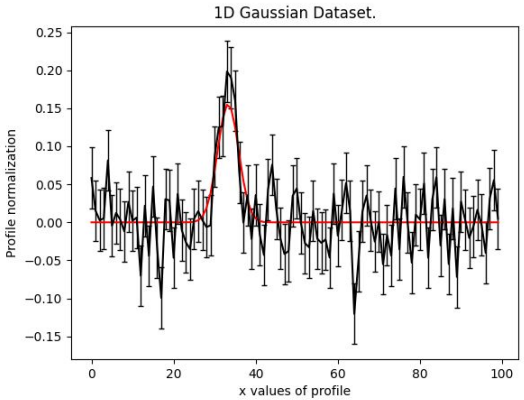
Observe many 1D Gaussians which now all have different centres.

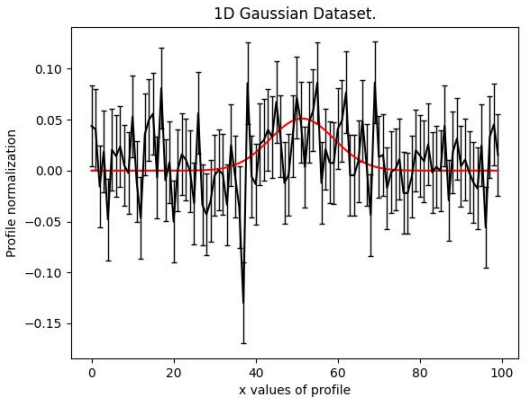
Centres are drawn from a parent Gaussian distribution with parameters (μ, σ)

Goal: Estimate the mean of the distribution μ and its scatter σ.









Traditional Method

1) Fit every dataset one-by-one.
2) Fit Gaussian distribution to the inferred centres (using their errors).

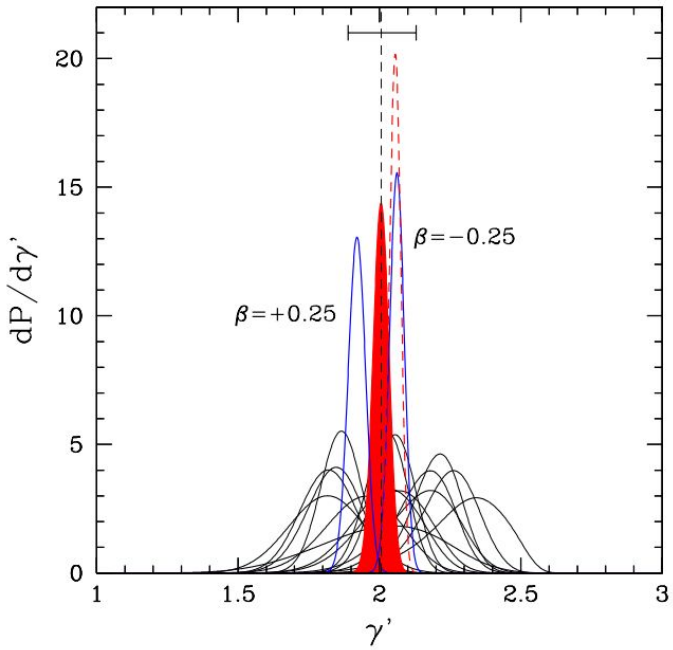


FIG. 3.— Posterior probability distribution functions of the logarithmic total density slope (γ'; see text). The shaded region (red) indicates the joint probability for γ', assuming isotropic stellar orbits and a Hernquist (1990) luminosity density profile. The thin solid curves refer to the 15 individual lens systems. The dashed (red) curve assumes a Jaffe (1983) luminosity density profile, leading to a several percent increase in the maximum-likelihood value of γ'. The two solid (blue) curves, indicated by $\beta = \pm 0.25$, show the probability functions for radially and tangentially anisotropic stellar orbits respectively (assuming a Hernquist profile for the stellar component). The horizontal bar indicates the 1σ intrinsic spread in γ', corrected for the spread due to measurement errors on the stellar velocity dispersions.

Traditional Method

For 10 high S/N 1D Gaussian pulses I infer where:

μ = 50.0, σ = 10.0

The traditional method gives:

μ = 46.47 +- 13.2 [1σ]

σ = 8.78 +- 6.89 [1σ]

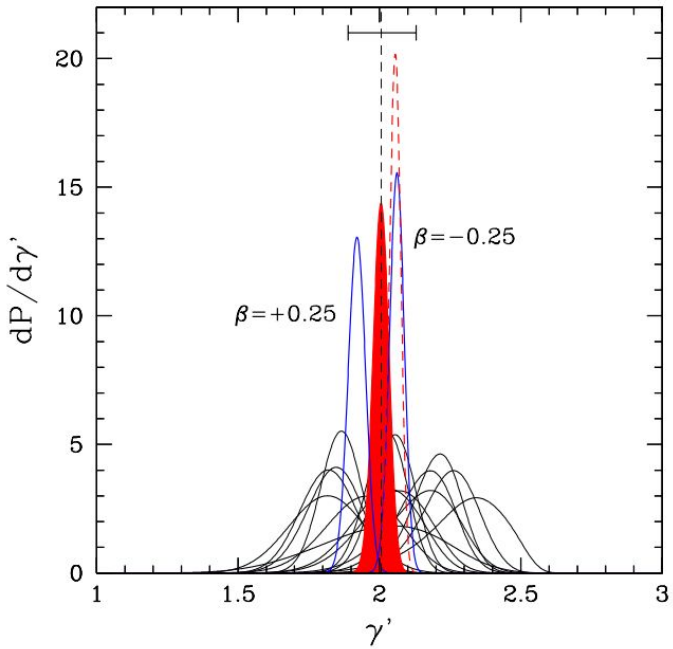


FIG. 3.— Posterior probability distribution functions of the logarithmic total density slope (γ'; see text). The shaded region (red) indicates the joint probability for γ', assuming isotropic stellar orbits and a Hernquist (1990) luminosity density profile. The thin solid curves refer to the 15 individual lens systems. The dashed (red) curve assumes a Jaffe (1983) luminosity density profile, leading to a several percent increase in the maximum-likelihood value of γ'. The two solid (blue) curves, indicated by $\beta = \pm 0.25$, show the probability functions for radially and tangentially anisotropic stellar orbits respectively (assuming a Hernquist profile for the stellar component). The horizontal bar indicates the $1\,\sigma$ intrinsic spread in γ', corrected for the spread due to measurement errors on the stellar velocity dispersions.

Traditional Method

For 10 high S/N 1D Gaussian pulses I infer where:

μ = 50.0, σ = 10.0

The traditional method gives:

μ = 46.47 +- 13.20 [1σ]
σ = 8.78 +- 6.89 [1σ]

The graphical model gives:

μ = 48.45 +- 5.31 [1σ]
σ = 11.56 +- 3.64 [1σ]

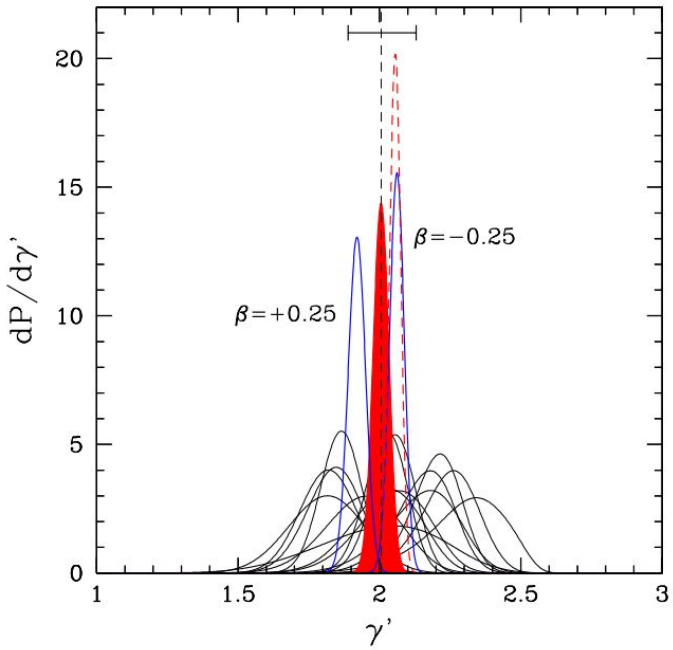


FIG. 3.— Posterior probability distribution functions of the logarithmic total density slope (γ'; see text). The shaded region (red) indicates the joint probability for γ', assuming isotropic stellar orbits and a Hernquist (1990) luminosity density profile. The thin solid curves refer to the 15 individual lens systems. The dashed (red) curve assumes a Jaffe (1983) luminosity density profile, leading to a several percent increase in the maximum-likelihood value of γ'. The two solid (blue) curves, indicated by $\beta = \pm 0.25$, show the probability functions for radially and tangentially anisotropic stellar orbits respectively (assuming a Hernquist profile for the stellar component). The horizontal bar indicates the $1\,\sigma$ intrinsic spread in γ', corrected for the spread due to measurement errors on the stellar velocity dispersions.

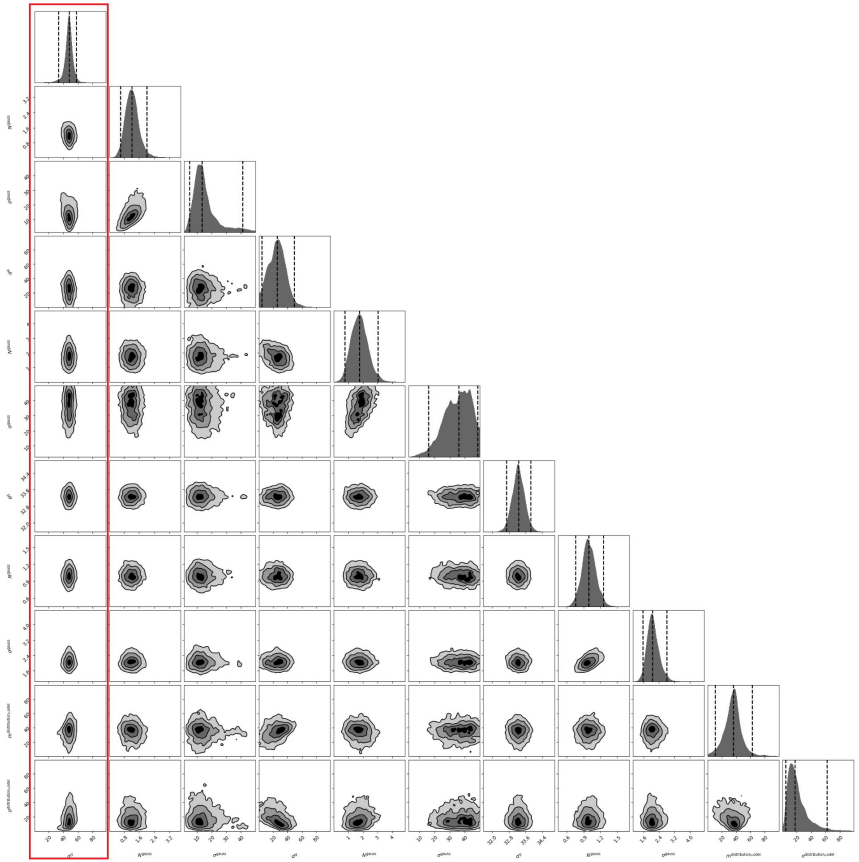
Why?

For the graphical model, the centre of every dataset constrains the centres of the other dataset!

- Certain solutions for dataset 0 are ruled out based on how datasets 1-9 update the hierarchical distribution constraints.

This occurs *through the hierarchical parent distribution.*

For the traditional fit, the centres of every dataset are fitted independently and this does not happen.

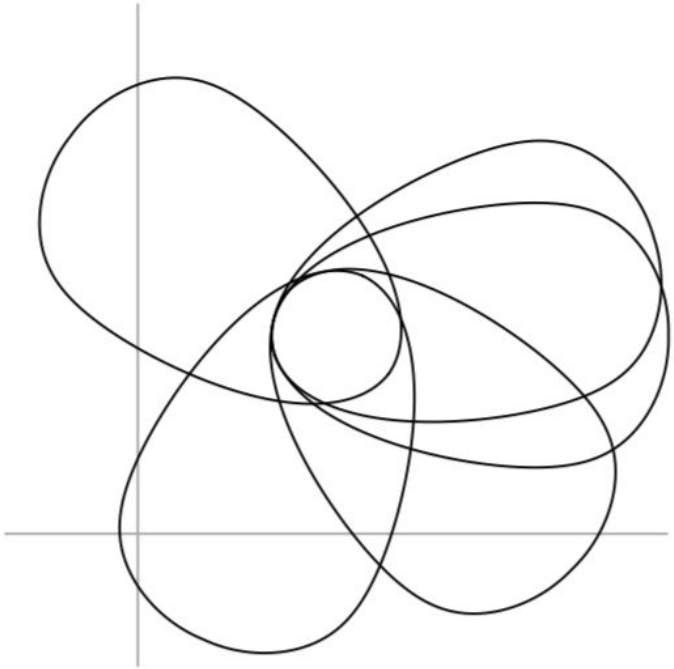


Hierarchical Models (“Bayesian Hierarchical Analysis”)

Fits to individual lenses give poor constraints on DM halo masses.

Inference over entire sample will break degeneracies.

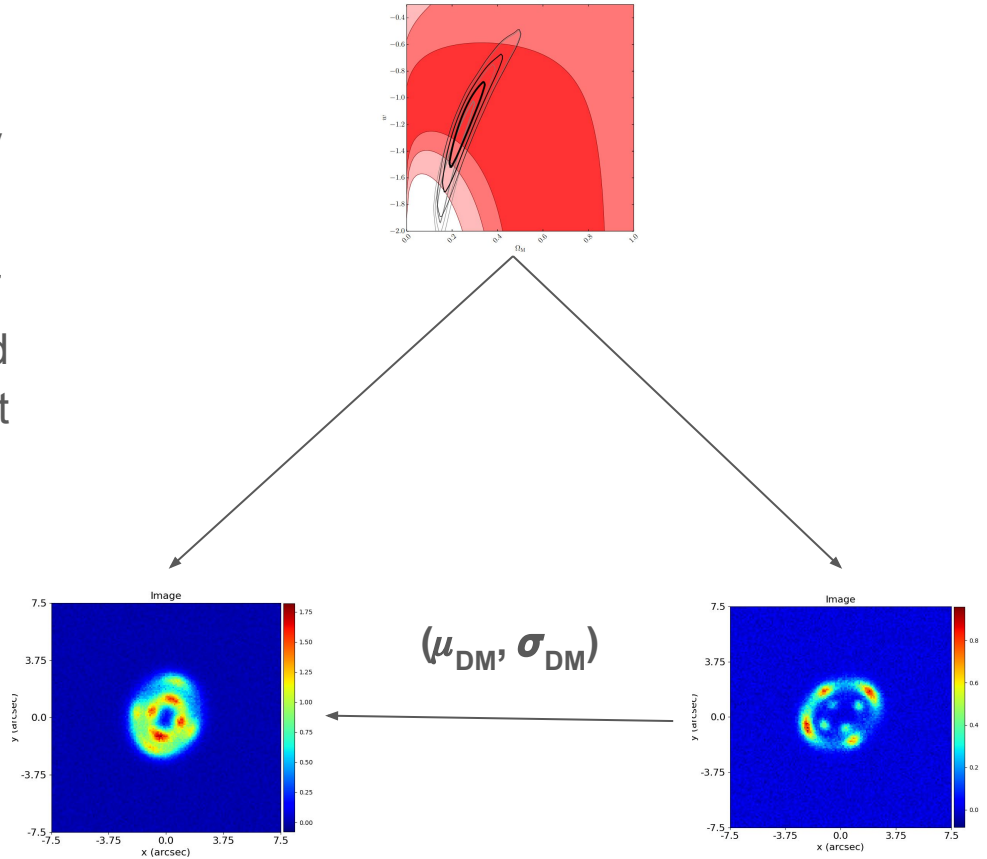
The datasets “talk to one another”.



Hierarchical Models (“Bayesian Hierarchical Analysis”)

Link lens model parameters across different datasets that were previously not shared.

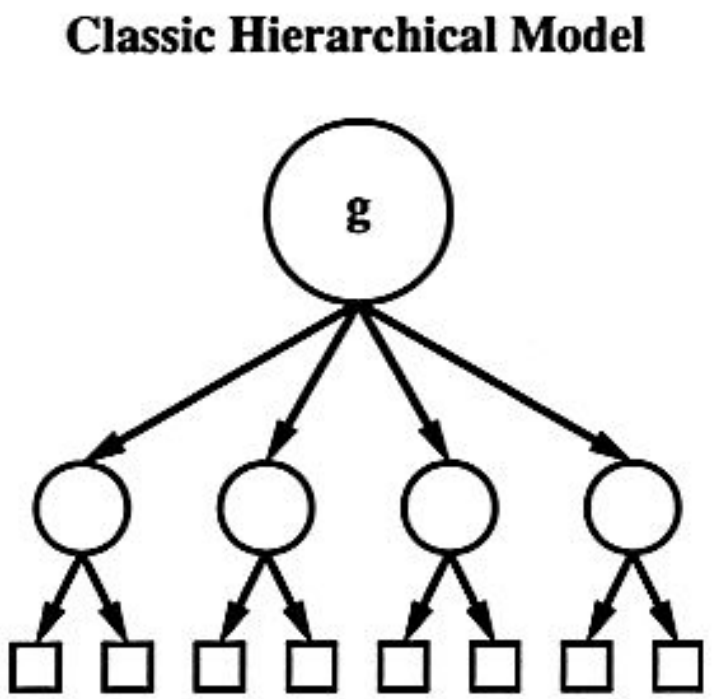
- Assume dark matter halo masses of lenses (which cannot be constrained individually) are drawn from a parent distribution.



Hierarchical Models (“Bayesian Hierarchical Analysis”)

Why did I assume a parent Gaussian distribution?

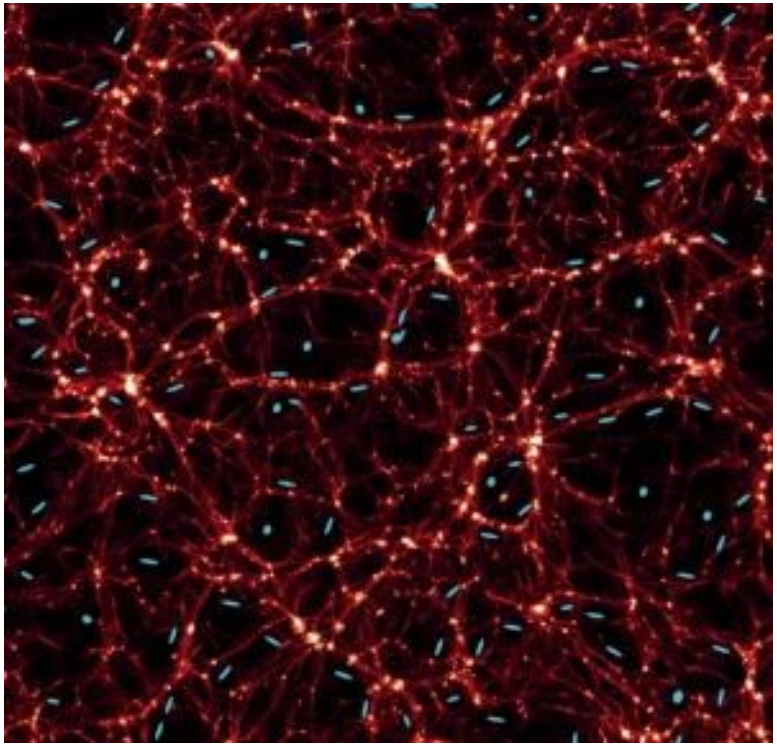
- Could be the parameter is genuinely defined by this underlying distribution (often true in Cosmology).
- Could simply be that if you measure that parameter independently from 1000 datasets, the measurement reduces to a Gaussian as is often the case in statistics.



Graphical and Hierarchical Models

Cosmology is going in the direction of combining graphical & hierarchical models:

- Graphical model is the Cosmology parameters.
- Hierarchical components marginalization over galaxies.



Expectation Propagation (EP)

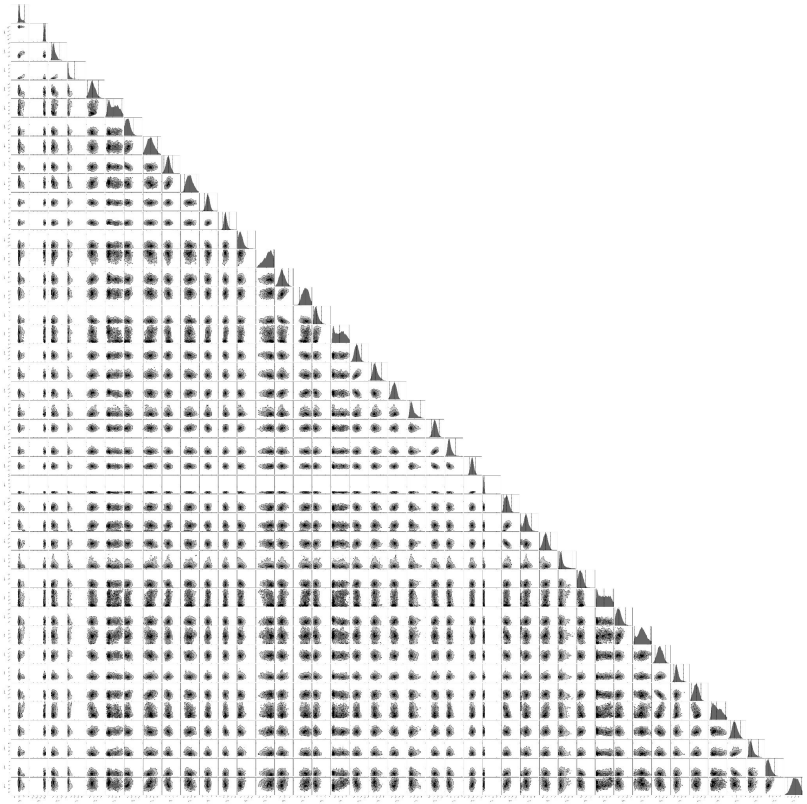
Curse of Dimensionality

Obvious limitation of graphical models -> **dimensionality.**

Every time we add a new dataset, we add **N additional free parameters**.

For 1D Gaussian's 25 datasets creates a model with **N=75** parameters.

For strong lens modeling we have **20+ free parameters per dataset**.



Expectation Propagation (EP)

Framework which fits every dataset one-by-one, passing information between model-fits to replicate graphical model.

Cavity Distribution: Represents posterior of parameters of all other datasets.

Titled Distribution: Represents posterior of parameter of dataset we are fitting.

https://arxiv.org/abs/1412.4869

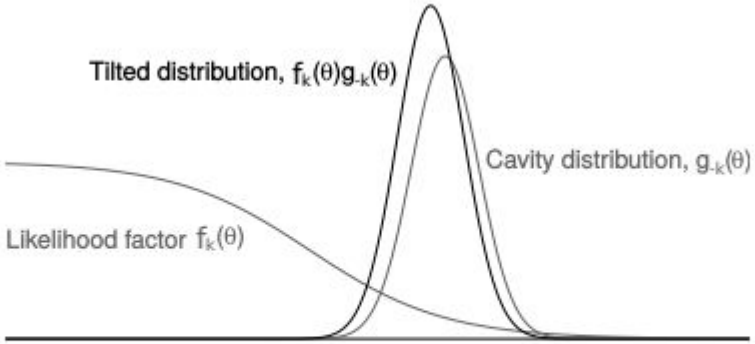
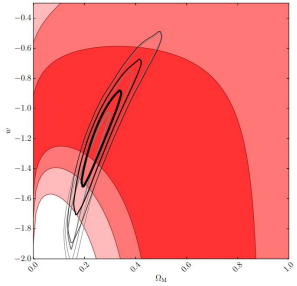
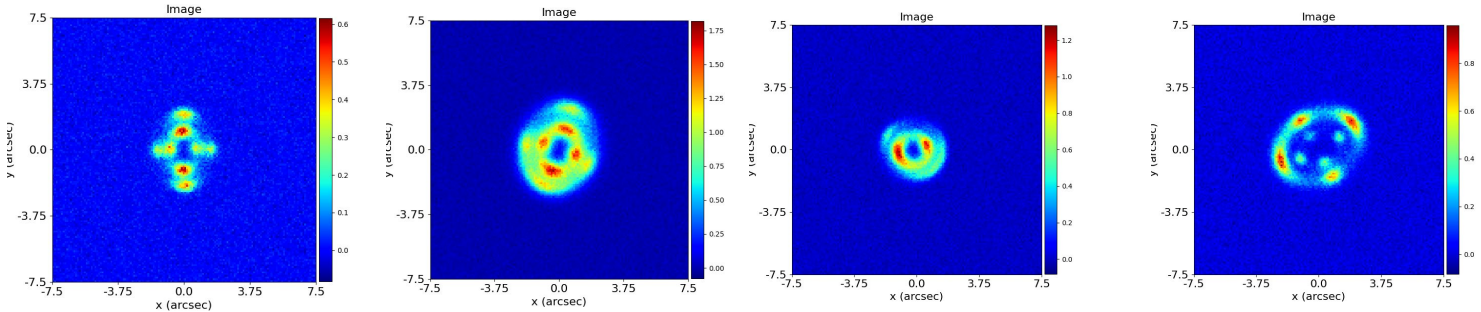


Figure 3: Example of a step of an EP algorithm in a simple one-dimensional example, illustrating the stability of the computation even when part of the likelihood is far from Gaussian. When performing inference on the likelihood factor $p(y_k|\theta)$, the algorithm uses the cavity distribution $g_{-k}(\theta)$ as a prior.

Bayesian Graph

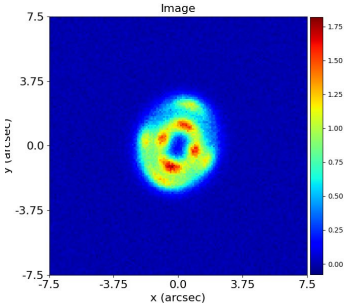
Exploit partitioning of data and model.





EP

1) **Fit dataset 1.**

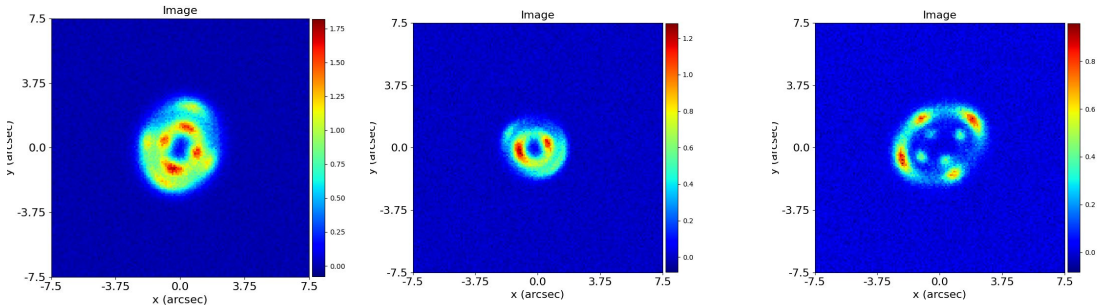


EP

1) Fit dataset 1.
2) Use result to create a message updating our beliefs about the model parameters.

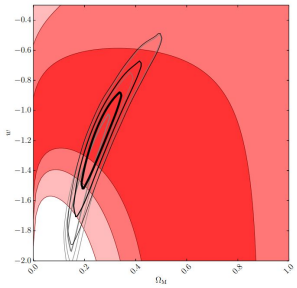
$M(\Omega_m, w, \ldots)$

. . .



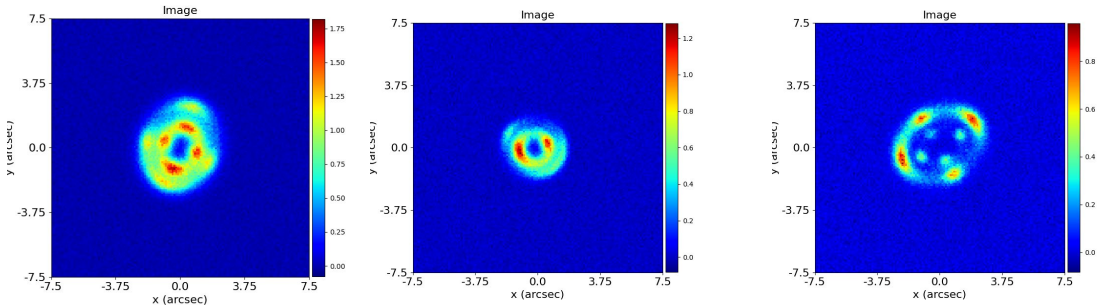
EP

1) Fit dataset 1.
2) Use result to create a message updating our beliefs about the model parameters.
3) Broadcast message across all shared parameters in graph.



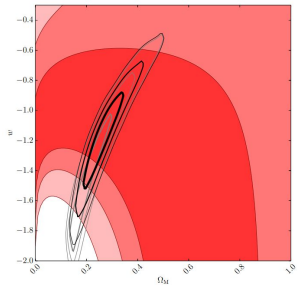
$M(\Omega_m, w, \ldots)$ $M(\Omega_m, w, \ldots)$ $M(\Omega_m, w, \ldots)$

. . .

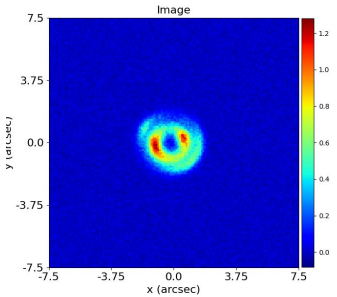


EP

1) Fit dataset 1.
2) Use result to create a message updating our beliefs about the model parameters.
3) Broadcast message across all shared parameters in graph.
4) Fit dataset 2 using updated model (priors on shared parameters updated).



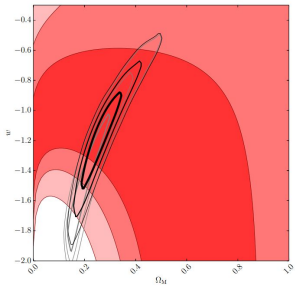
$M(\Omega_m, w, \ldots)$



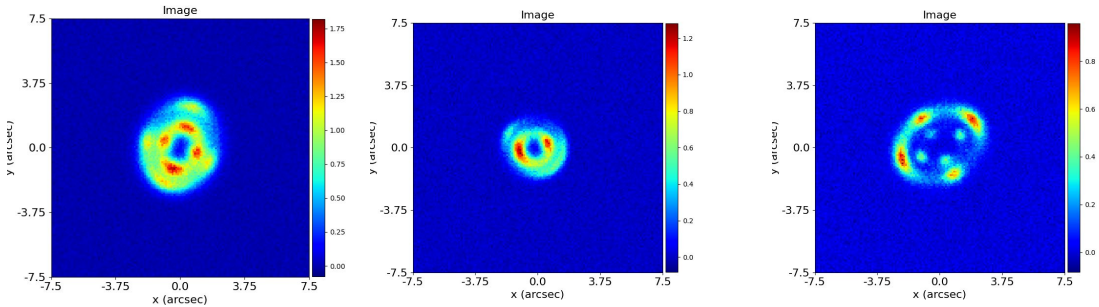
EP

1) Fit dataset 1.
2) Use result to create a message updating our beliefs about the model parameters.
3) Broadcast message across all shared parameters in graph.
4) Fit dataset 2 using updated model (priors on shared parameters updated).
5) Create message from result.
6) Broadcast across graph.

. . .



$M(\Omega_m, w, \ldots)$ $M(\Omega_m, w, \ldots)$ $M(\Omega_m, w, \ldots)$



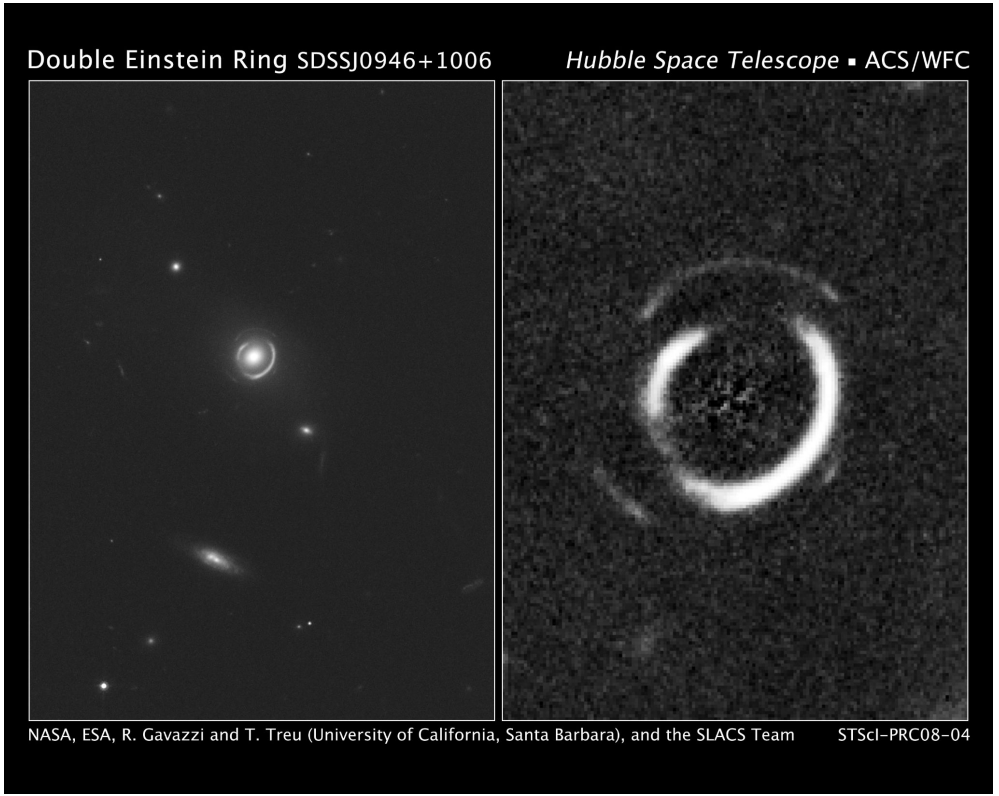
Cosmology / Galaxy Formation Models

Strong Lensing (PyAutoLens)

Model 100 (simulated) double source plane lenses:

- 27 parameters per lens.
- 2700 parameters in total!

Aim: infer ω and Ω_m.

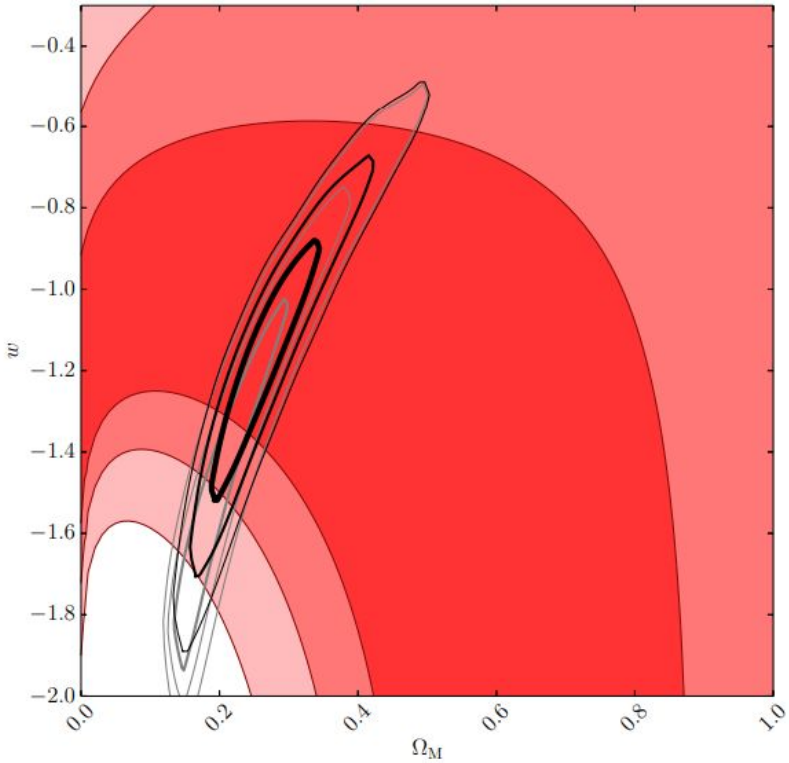


Strong Lensing (PyAutoLens)

Model 100 double source plane lenses:

- 27 parameters per lens.
- 2700 parameters in total!

Aim: infer ω and Ω_m.



Galaxy Formation & Evolution (PyAutoGalaxy)

Hierarchical model which infers parent distribution of Sersic indexes of the bulge and disk in a sample of 1000 simulated galaxies.

- 14 parameters per galaxy.
- 14000 parameters in total!

Aim: Infer underlying parent distribution of galaxy structures.

PyAutoFit

PyAutoFit

GitHub:
https://github.com/rhayes777/PyAutoFit

Readthedocs:
https://pyautofit.readthedocs.io/en/latest/

JOSS Paper:
https://joss.theoj.org/papers/10.21105/joss.02550

Binder:
https://mybinder.org/v2/gh/Jammy2211/autofit_workspace/HEAD

We can also use it to get a model instance of the `median_pdf` model, which is the model where each parameter is the value estimated from the probability distribution of parameter space.

In [14]:
```
mp_instance = result.samples.median_pdf_instance
print()
print("Median PDF Model:\n")
print("Centre = ", mp_instance.centre)
print("Intensity = ", mp_instance.intensity)
print("Sigma = ", mp_instance.sigma)
```

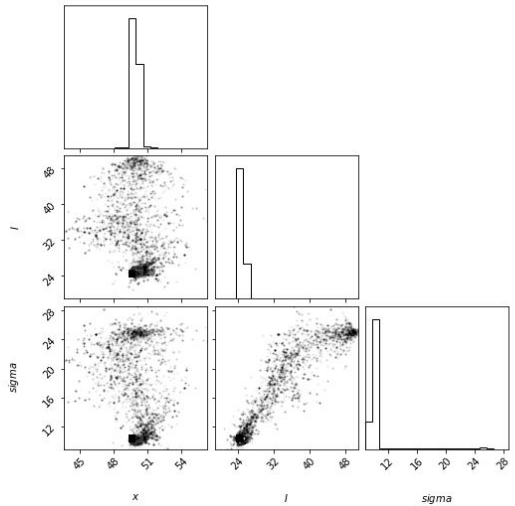
```
Median PDF Model:

Centre =  49.92285569756167
Intensity =  24.974961843717058
Sigma =  9.969794911012947
```

The Probability Density Functions (PDF's) of the results can be plotted using the Emcee's visualization tool `corner.py`, which is wrapped via the `EmceePlotter` object.

The PDF shows the 1D and 2D probabilities estimated for every parameter after the model-fit. The two dimensional figures can show the degeneracies between different parameters, for example how increasing σ and decreasing the intensity I can lead to similar likelihoods and probabilities.

In [15]:
```
emcee_plotter = aplt.EmceePlotter(samples=result.samples)
emcee_plotter.corner()
```

```
2021-07-26 16:42:47,675 - root - WARNING - Too few points to create valid contours
2021-07-26 16:42:47,712 - root - WARNING - Too few points to create valid contours
2021-07-26 16:42:47,737 - root - WARNING - Too few points to create valid contours
```



PyAutoLens: Open Source Strong Gravitational Lensing

All code publically available.

GitHub: https://github.com/Jammy2211/PyAutoLens

Readthedocs: https://pyautolens.readthedocs.io/en/latest/

JOSS paper:
https://github.com/Jammy2211/PyAutoLens/blob/master/paper/paper.md

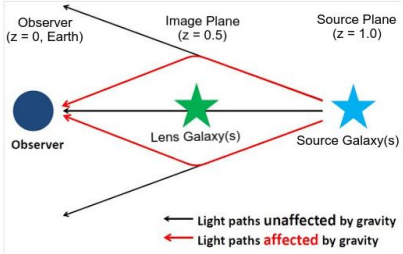
Have a professional software engineer developing PyAutoLens with me – high quality software!

Tutorial 4: Planes

So far, we have learnt how to combine light profiles, mass profiles and galaxies to perform various calculations. In this tutorial we'll use these objects to perform our first ray-tracing calculations!

A strong gravitational lens is a system where two (or more) galaxies align perfectly down our line of sight from Earth such that the foreground galaxy's mass (represented as mass profiles) deflects the light (represented as light profiles) of a background source galaxy(s).

When the alignment is just right and the lens is massive enough, the background source galaxy appears multiple times. The schematic below shows such a system, where light-rays from the source are deflected around the lens galaxy to the observer following multiple distinct paths.



As an observer, we don't see the source's true appearance (e.g. a round blob of light). Instead, we only observe its light after it has been deflected and lensed by the foreground galaxies.

In the schematic above, we used the terms 'image-plane' and 'source-plane'. In lensing, a 'plane' is a collection of galaxies at the same redshift (meaning that they are physically parallel to one another). In this tutorial, we'll use the `Plane` object to create a strong lensing system like the one pictured above. Whilst a plane can contain any number of galaxies, in this tutorial we'll stick to just one lens galaxy and one source galaxy.

In [ ]:
```
%matplotlib inline
from pyprojroot import here
workspace_path = str(here())
%cd $workspace_path
print(f"Working Directory has been set to `{workspace_path}`")

import autolens as al
import autolens.plot as aplt
```

Initial Setup

As always, we need a 2D grid of (y, x) coordinates.

However, we can now think of our grid as the coordinates that we are going to 'trace' from the image-plane to the source-plane. We name our grid the `image_plane_grid` to reflect this.

In [ ]:
```
image_plane_grid = al.Grid2D.uniform(shape_native=(100, 100), pixel_scales=0.05)
```

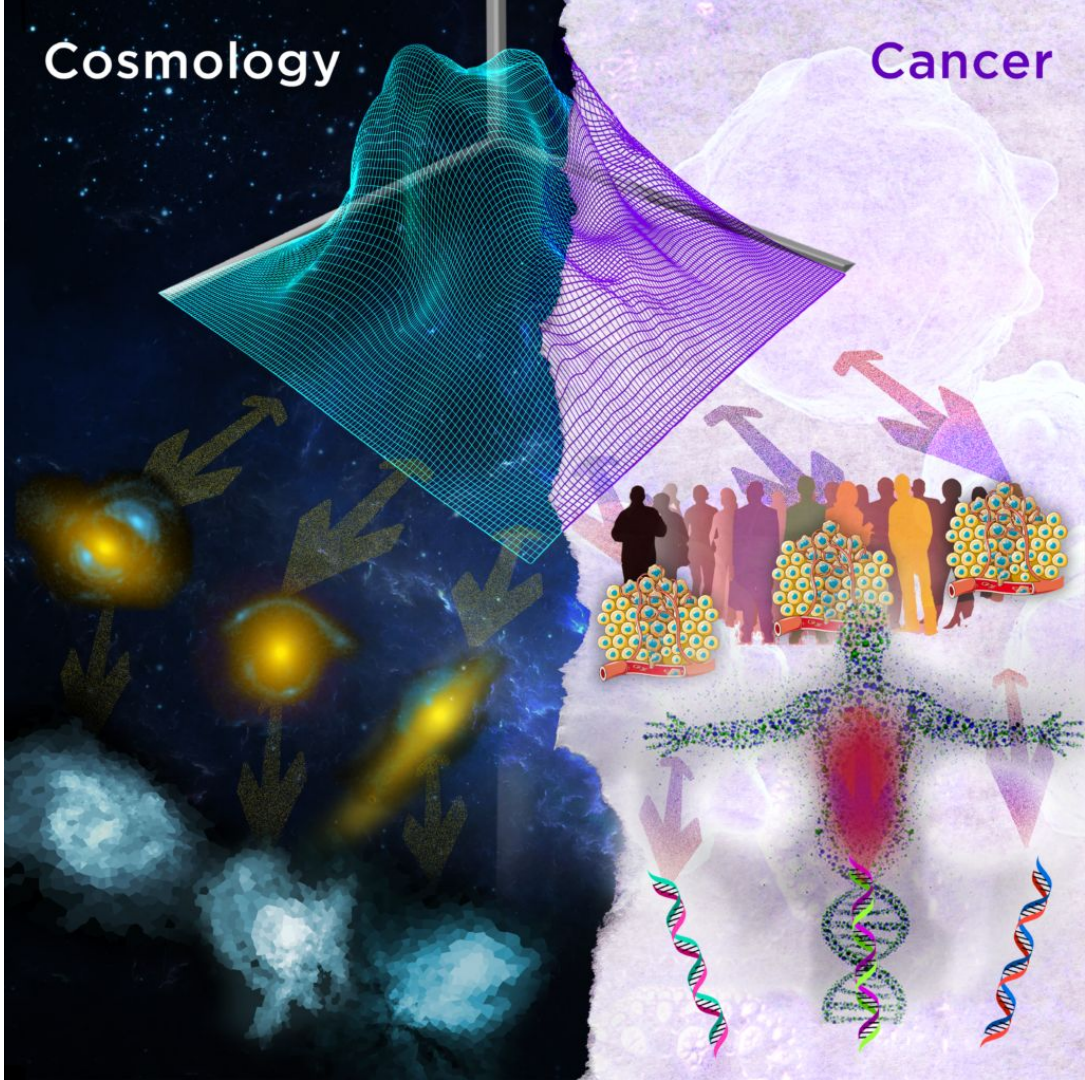
We will also name our `Galaxy` objects `lens_galaxy` and `source_galaxy`, to reflect their role in the schematic above.

Cosmology & Cancer

PyAutoFit and this statistical framework was developed in collaboration with an industry partner.

Concr: Biotech company that aim to improve cancer therapy via big data analysis.

They are now fitting models with over 100 million parameters!



Summary

Graphical Models: Infer Cosmology by fit large datasets with a single model.

Hierarchical Models: Study galaxy formation & evolution by determining parent distribution of galaxies.

Expectation Propagation: Overcome **curse of dimensionality** to make this feasible on large datasets.

PyAutoFit: Software that means anyone can do this!

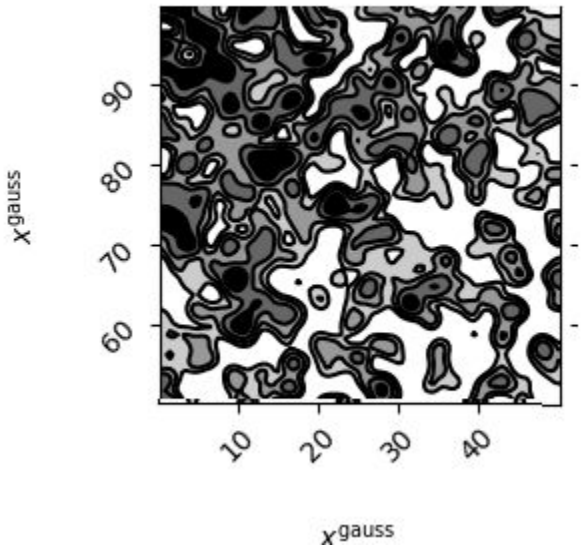
Bayesian Methods

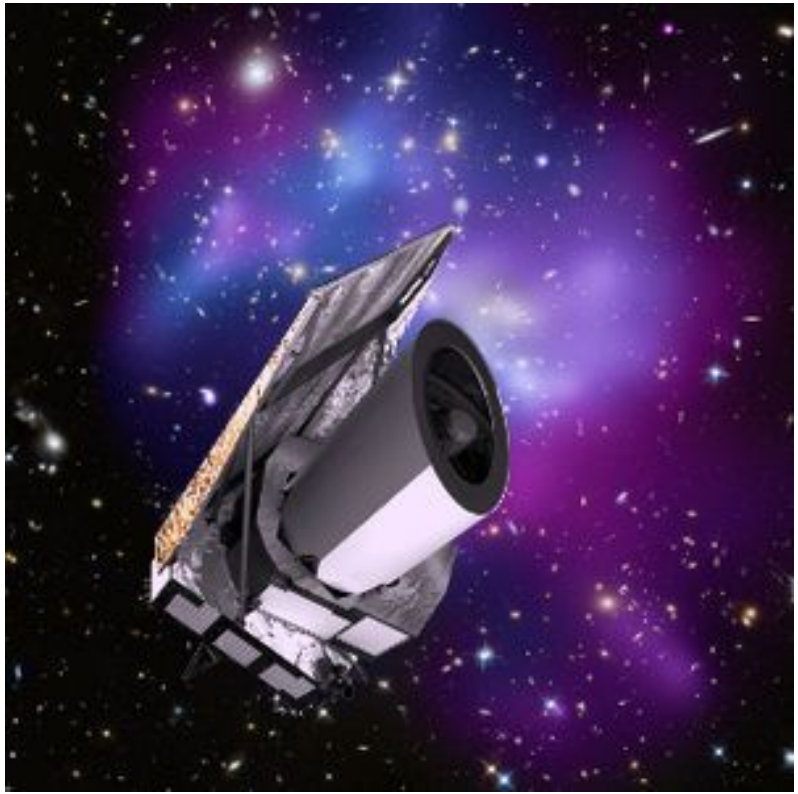
Graphical Model

In theory, posterior multiplication via a 2D KDE would give the same result.

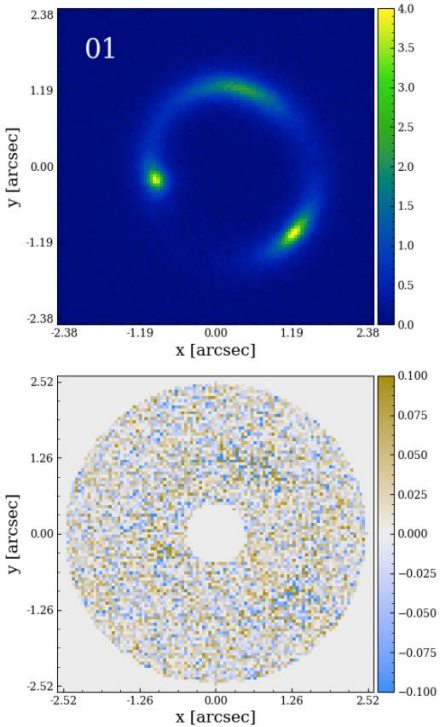
In practise, this depends on the signal to noise of the data.

Does not scale up to high dimensions!





Strong Lensing Substructure



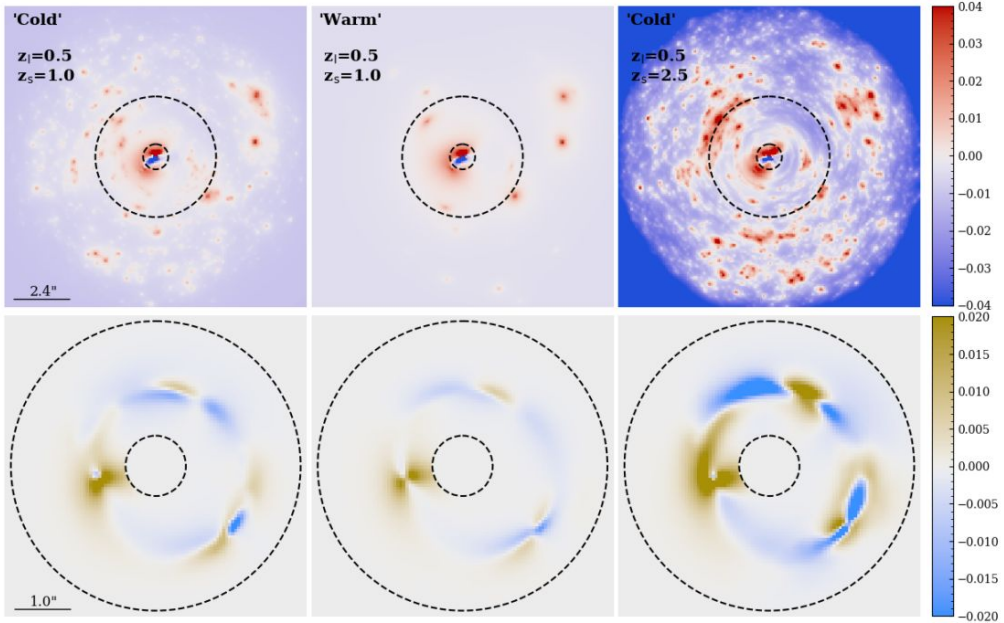


Figure 1. Upper panels: the effective convergence of low mass perturbers. Lower panels: the corresponding best-fit image residuals. The colour scale is in units of e^- pix^{-1} s^{-1}. The parameters of the lens and source galaxies shown in the three columns are the same as those used in Fig. 4, except for the source redshifts. The lensing systems in the left and middle columns have a source at $z = 1$, while that in the right column has a source at $z = 2.5$. The systems in both the left and right columns have a cutoff in the mass function at 10^7 $M_\odot$, while the one in middle column has the cutoff at 10^9 $M_\odot$. The inner and outer dashed circles in each panel have radii, 0.5″ and 2.4″, respectively.