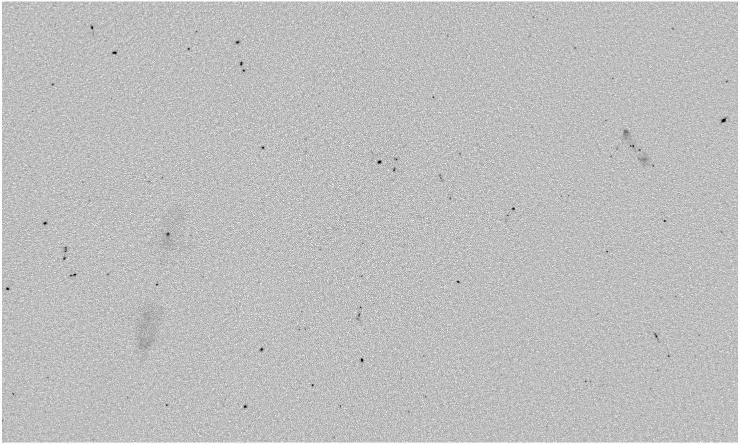
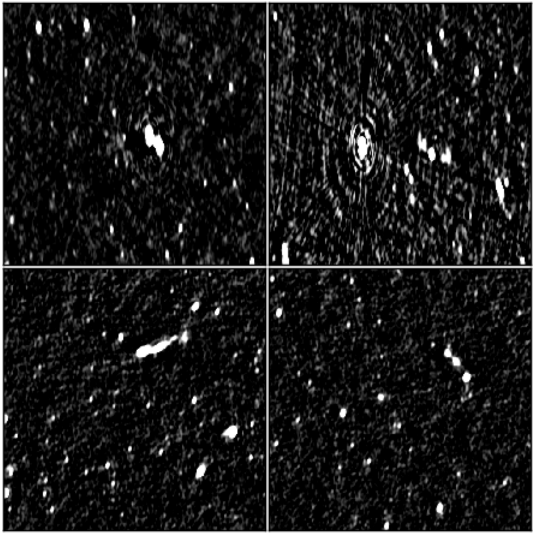
Source finding by model comparison

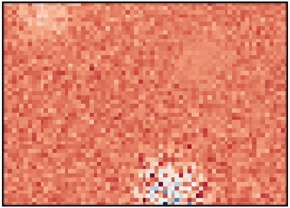
Marcus Frean, Anna Friedlander,
Melanie Johnston-Hollitt and Chris Hollitt

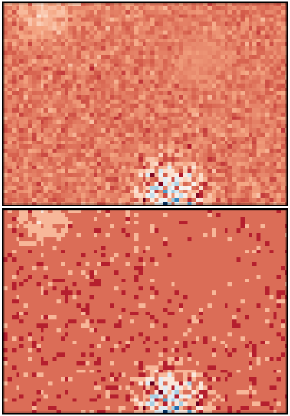
Victoria University of Wellington
Wellington, New Zealand

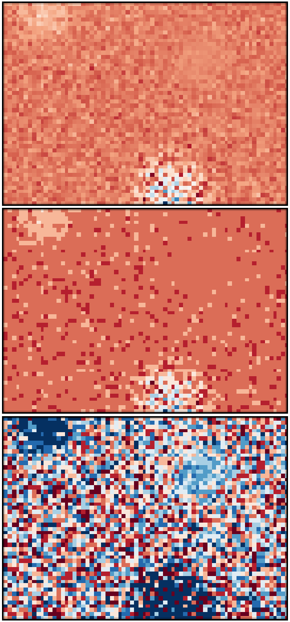
NEED: detect very faint objects embedded in noise, robustly, efficiently, with a minimum of manual tuning.







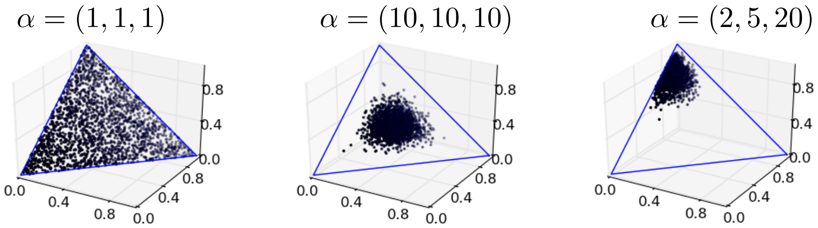




dirichlet borders

There are no “correct” borders, so we generate *lots of them*.

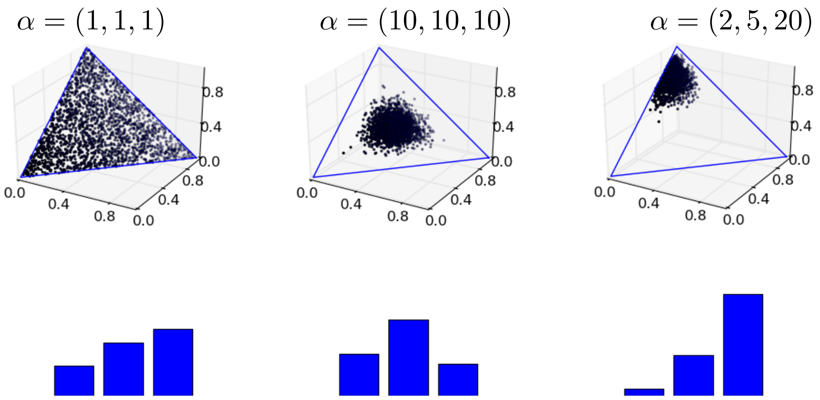
Use the Dirichlet distribution to make the bin borders:



dirichlet borders

There are no “correct” borders, so we generate *lots of them*.

Use the Dirichlet distribution to make the bin borders:

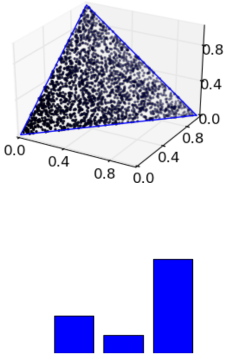


dirichlet borders

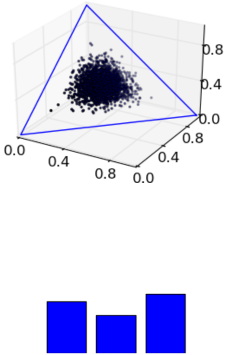
There are no “correct” borders: so we generate *lots of them*.

Use the Dirichlet distribution to make the bin borders:

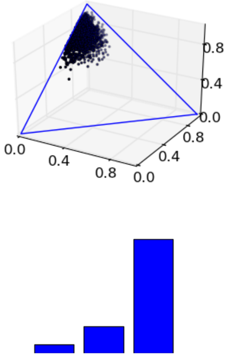
$\alpha = (1, 1, 1)$



$\alpha = (10, 10, 10)$



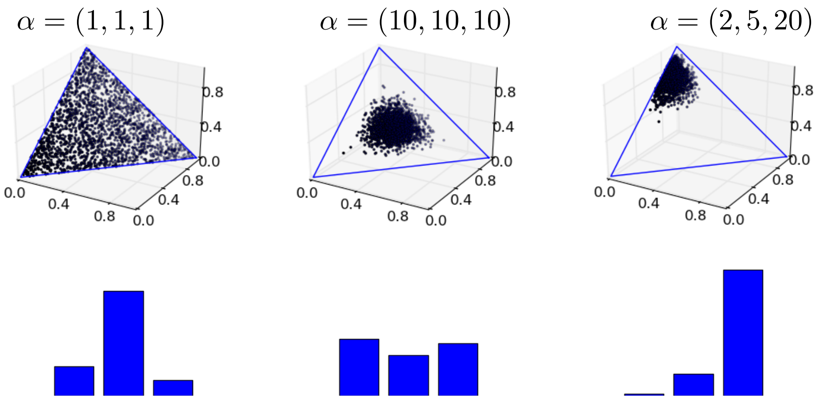
$\alpha = (2, 5, 20)$



dirichlet borders

There are no “correct” borders: so we generate *lots of them*.

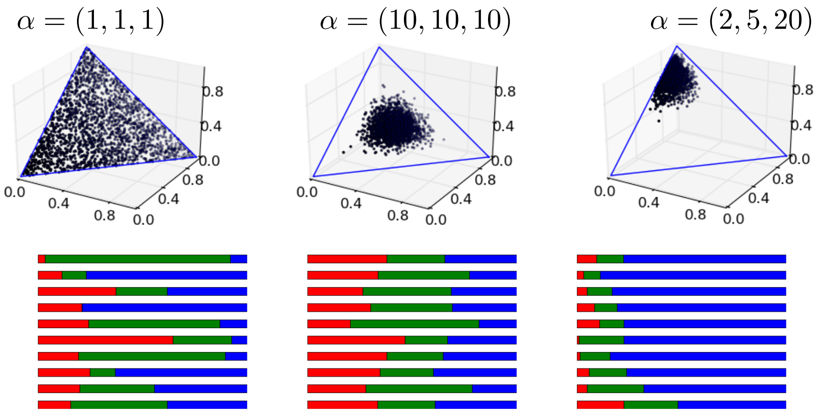
Use the Dirichlet distribution to make the bin borders:



dirichlet borders

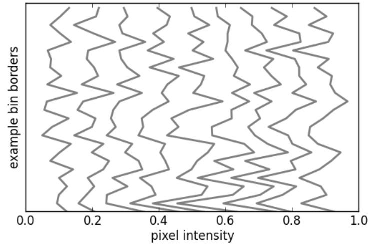
There are no “correct” borders: we generate *lots of them*.

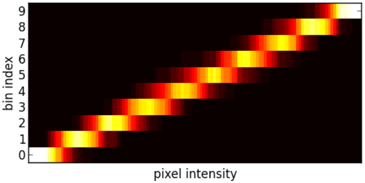
Use the Dirichlet distribution to make the bin borders:



so we don’t have to believe in just one binning scheme

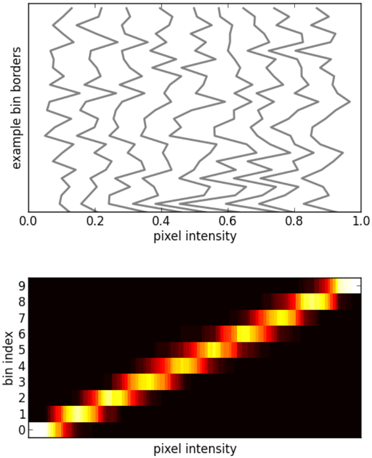
$\sim$ Equal bins



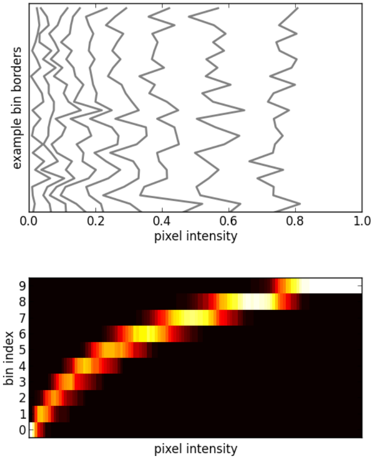


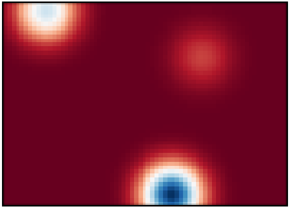
so we don't have to believe in just one binning scheme

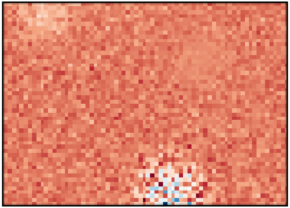
$\sim$ Equal bins

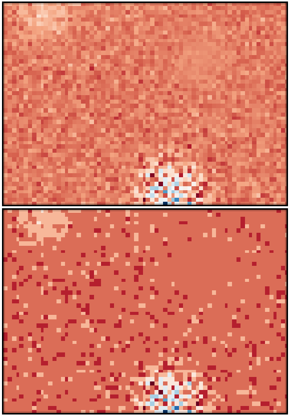


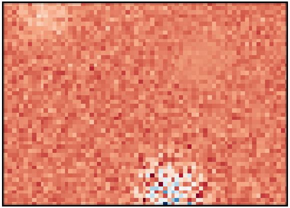
$\sim$ Exponential bins

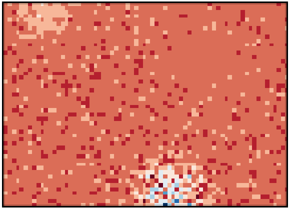


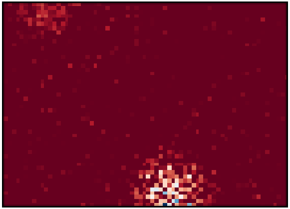


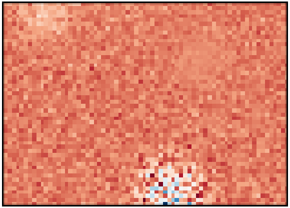


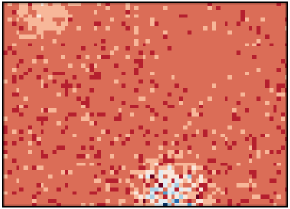


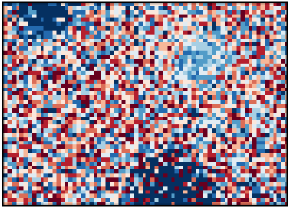


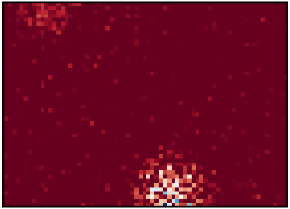


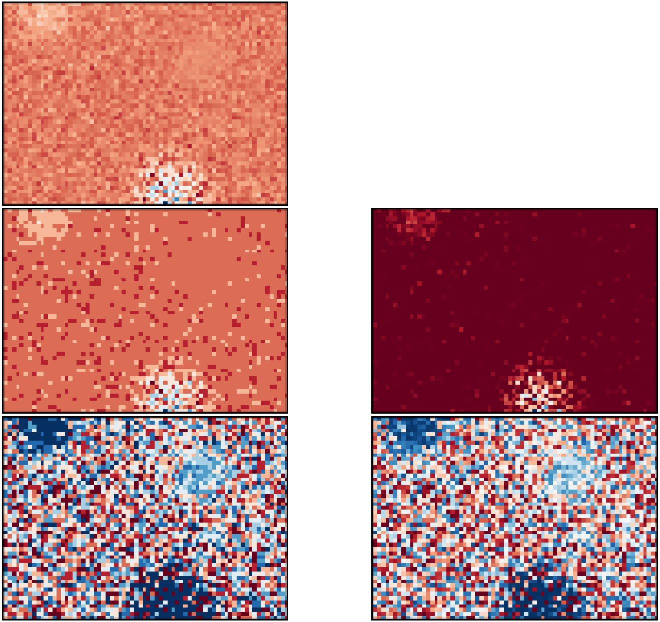




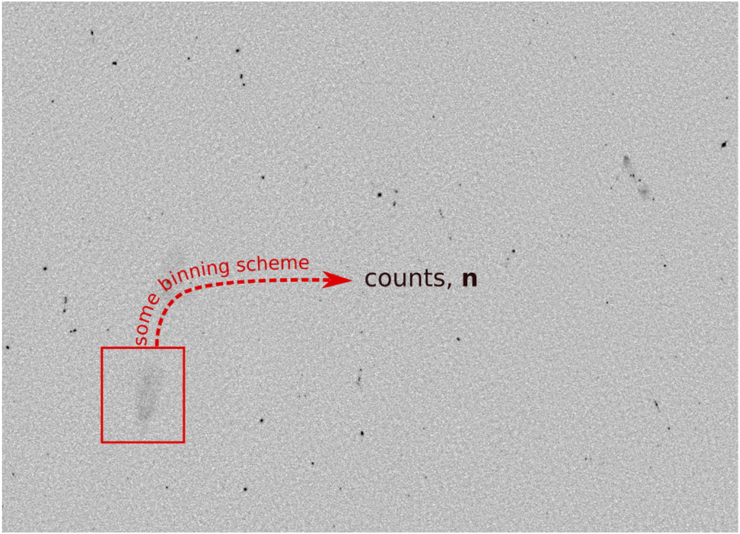






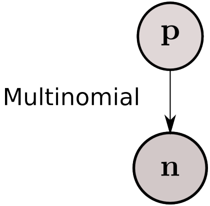


How to score a region in an image?



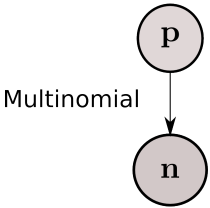
How to score a region in an image?

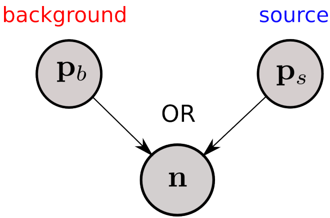
figure out the **categorical** distributions typical of pixels from background and source, and compare the likelihoods under a multinomial model...



How to score a region in an image?

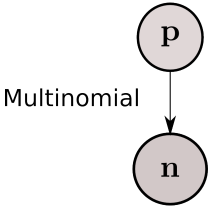
figure out the **categorical** distributions typical of pixels from background and source, and compare the likelihoods under a multinomial model...

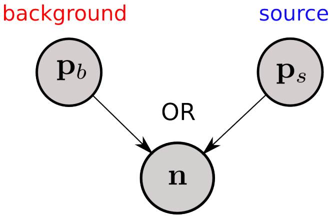




How to score a region in an image?

figure out the **categorical** distributions typical of pixels from background and source, and compare the likelihoods under a multinomial model...





$$\text{score} = \log \frac{P(\mathbf{n} \mid \text{source})}{P(\mathbf{n} \mid \text{background})}$$

One can imagine such a model (histogram) for background, but

One can imagine such a model (histogram) for background, but

- all sources are different - we should not commit to look for "typical" sources

One can imagine such a model (histogram) for background, but

- all sources are different - we should not commit to look for “typical” sources
- small region versus large region? Big region has more evidence: this *should* matter, but it won’t.

One can imagine such a model (histogram) for background, but

- all sources are different - we should not commit to look for “typical” sources
- small region versus large region? Big region has more evidence: this *should* matter, but it won’t.

- Ah! We could treat the background distribution as a (frequentist) null hypothesis to be rejected, or

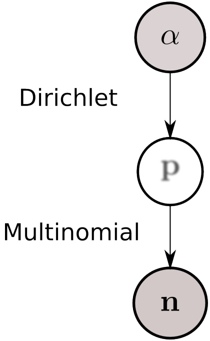
One can imagine such a model (histogram) for background, but

- all sources are different - we should not commit to look for "typical" sources
- small region versus large region? Big region has more evidence: this *should* matter, but it won't.

- Ah! We could treat the background distribution as a (frequentist) null hypothesis to be rejected, or
- do something sensible

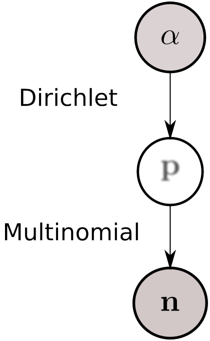
The Dirichlet-multinomial distribution

...a compound distribution, where the parameters of a categorical distribution are drawn from a Dirichlet distribution with parameters $\boldsymbol{\alpha} = \alpha_1..\alpha_K$.



The Dirichlet-multinomial distribution

...a compound distribution, where the parameters of a categorical distribution are drawn from a Dirichlet distribution with parameters $\boldsymbol{\alpha} = \alpha_1..\alpha_K$.

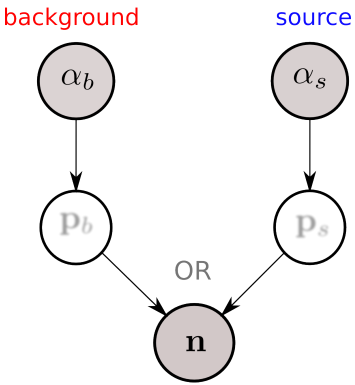


We can integrate out **p** analytically $\longrightarrow$ evidence:

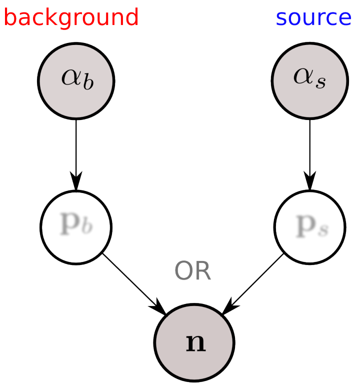
$$P(\mathbf{n} \mid \alpha) = \frac{\Gamma(A)}{\Gamma(N+A)} \prod_k \frac{\Gamma(n_k + \alpha_k)}{\Gamma(\alpha_k)}$$

(No MCMC required)

the Bayes factor

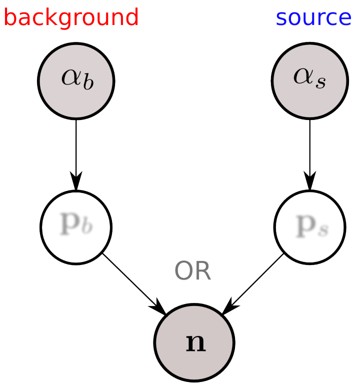


the Bayes factor



$$\text{score} = \log \frac{P(S \mid \mathbf{n})}{P(B \mid \mathbf{n})} = \log \frac{P(\mathbf{n} \mid S)}{P(\mathbf{n} \mid B)} + \log \frac{P(S)}{P(B)}$$

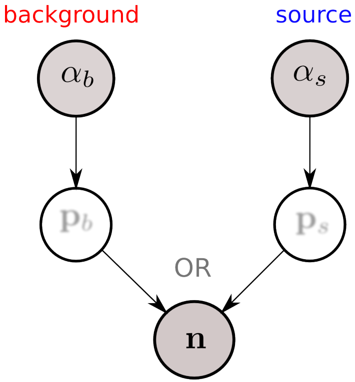
the Bayes factor



$$\text{score} = \log \frac{P(S \mid \mathbf{n})}{P(B \mid \mathbf{n})} = \log \frac{P(\mathbf{n} \mid S)}{P(\mathbf{n} \mid B)} + \log \frac{P(S)}{P(B)}$$

- background: lots of evidence - typical by definition: we used the overall counts as the hyperparameters α_B

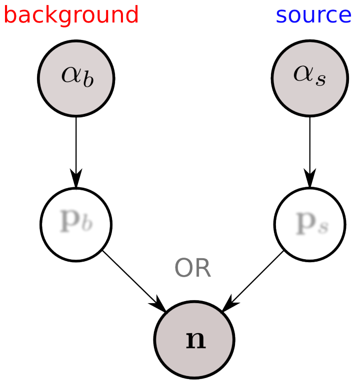
the Bayes factor



$$\text{score} = \log \frac{P(S \mid \mathbf{n})}{P(B \mid \mathbf{n})} = \log \frac{P(\mathbf{n} \mid S)}{P(\mathbf{n} \mid B)} + \log \frac{P(S)}{P(B)}$$

- background: lots of evidence - typical by definition: we used the overall counts as the hyperparameters α_B
- source: much less evidence for “typical”: we set all α_S to 1

the Bayes factor

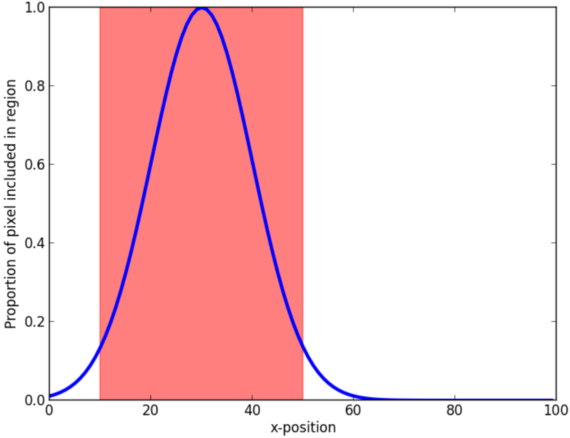


$$\text{score} = \log \frac{P(S \mid \mathbf{n})}{P(B \mid \mathbf{n})} = \log \frac{P(\mathbf{n} \mid S)}{P(\mathbf{n} \mid B)} + \log \frac{P(S)}{P(B)}$$

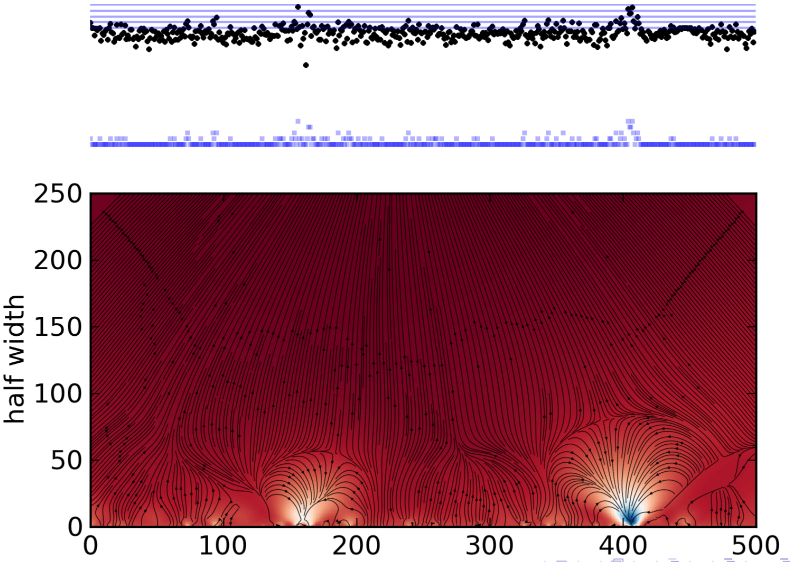
- background: lots of evidence - typical by definition: we used the overall counts as the hyperparameters α_B
- source: much less evidence for "typical": we set all α_S to 1

A score of zero means we're right on the fence for this region.

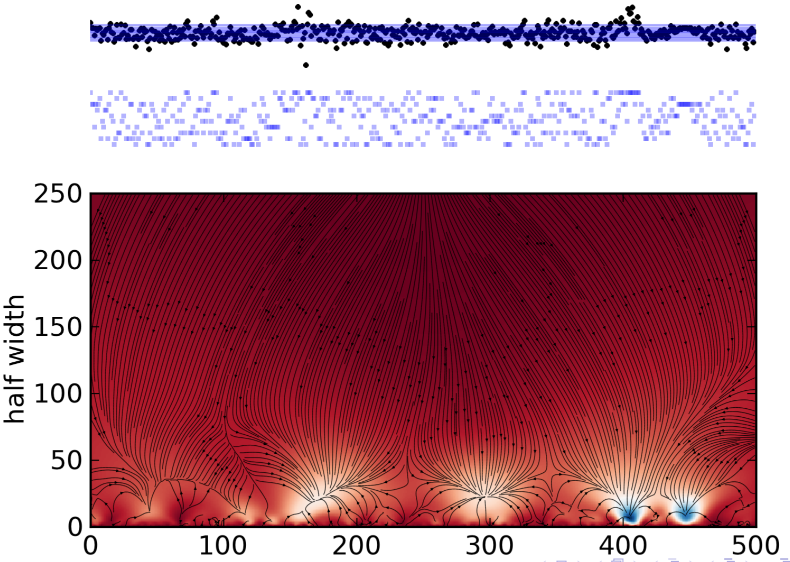
soft boundaries ⟶ partial counts



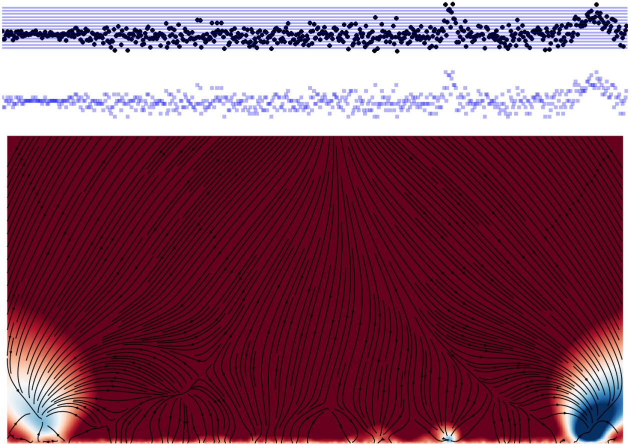
1-D simulated data



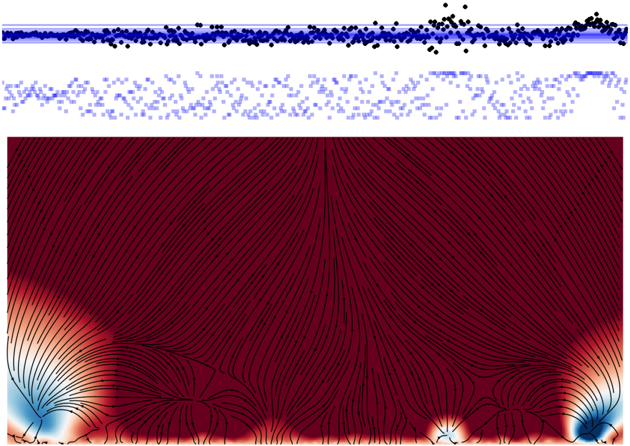
1-D simulated data



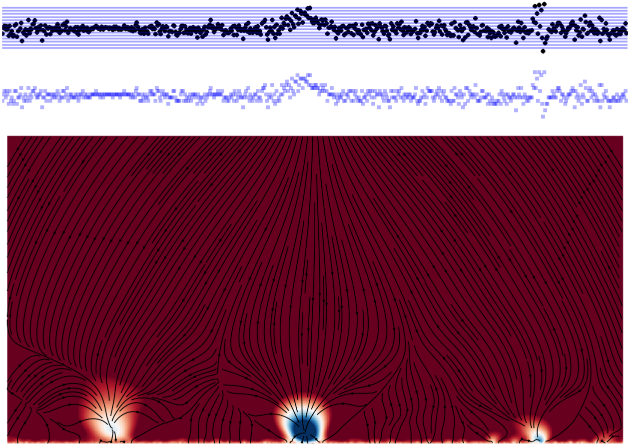
1-D simulated data (eg 2)



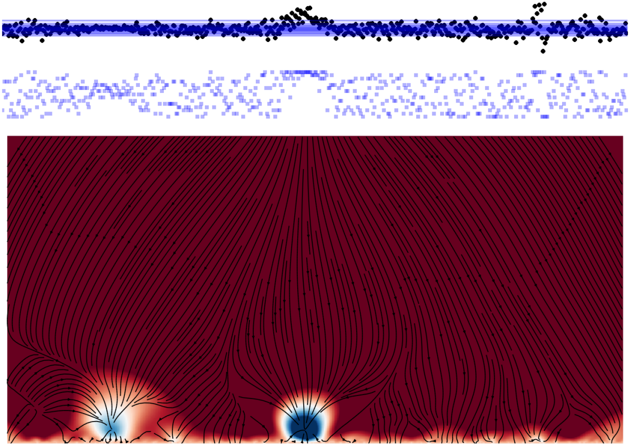
1-D simulated data (eg 2)



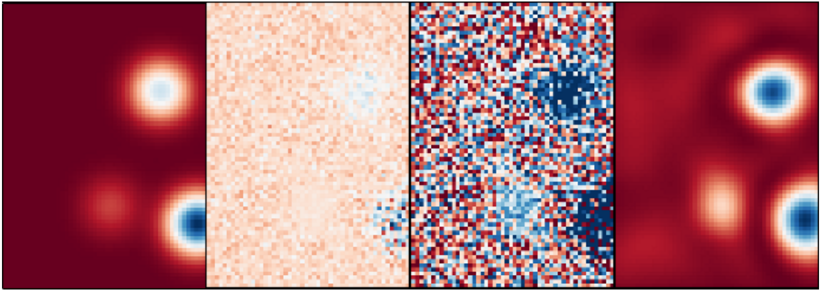
1-D simulated data (eg 3)



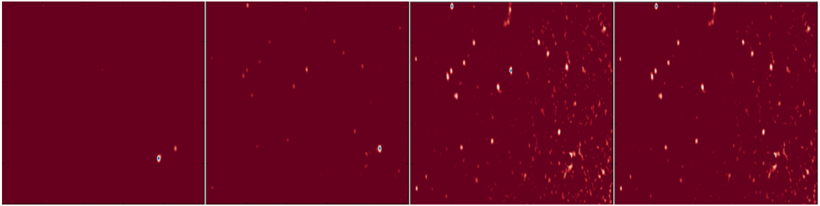
1-D simulated data (eg 3)



DMR score on 2-D simulated data



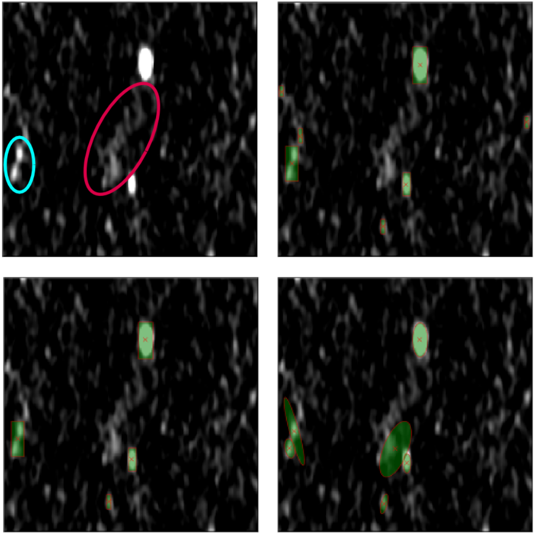
iterative removal of sources

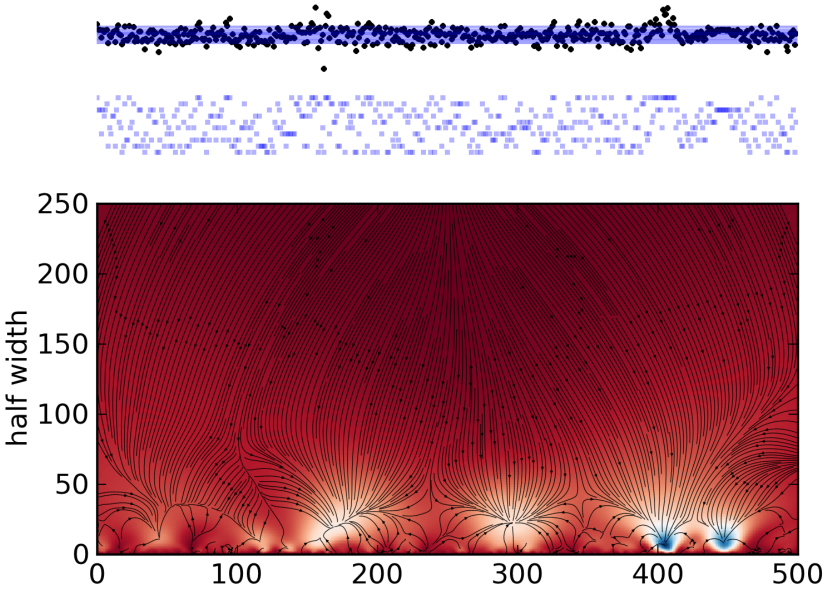


At each round of gradient ascent, a source is removed, and the bin borders and α^B vector recalculated.

With each round, new sources are revealed that were previously hidden in “background” bins.

comparisons




250
200
150
100
50
0
half width
0
100
200
300
400
500