Title: Feature-finding in the cosmic microwave background

Date: Feb 09, 2012 11:40 AM

URL: http://pirsa.org/12020144

Abstract:

Feature finding in the CMB

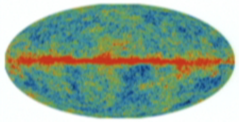
Matthew C. Johnson
Perimeter Institute

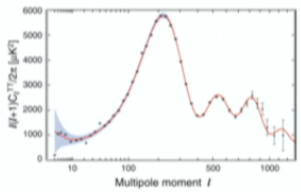
In collaboration with:

Stephen Feeney
Daniel Mortlock
Hiranya Peiris

The data

- Lambda-CDM: very successful at describing the CMB power spectrum.



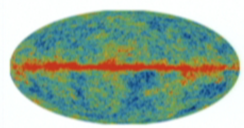


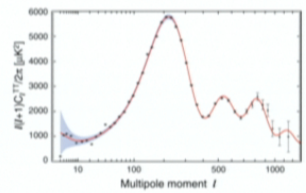
WMAP 7-year data

Is there room for anything else?

The data

- Lambda-CDM: very successful at describing the CMB power spectrum.





WMAP 7-year data

Is there room for anything else?

Theories with features

- A number of theories predict specific features in the CMB on top of those predicted by LCDM:

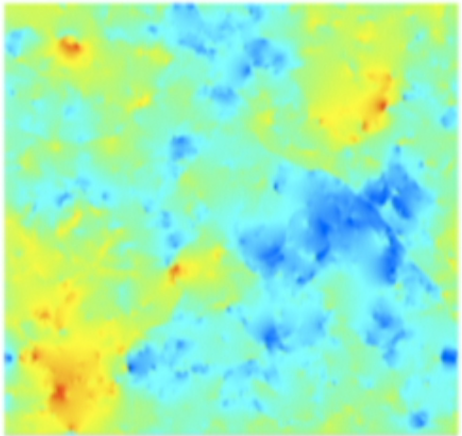
$$\frac{\Delta T(\hat{\mathbf{n}})}{T} = (1 + f(\hat{\mathbf{n}}))\,(1 + \delta_{\rm LCDM}(\hat{\mathbf{n}})) - 1$$
$$\simeq f(\hat{\mathbf{n}}) + \delta_{\rm LCDM}(\hat{\mathbf{n}})$$

Theories with features

- A number of theories predict specific features in the CMB on top of those predicted by LCDM:

$$\frac{\Delta T(\hat{\mathbf{n}})}{T} = (1 + f(\hat{\mathbf{n}}))\,(1 + \delta_{\rm LCDM}(\hat{\mathbf{n}})) - 1$$
$$\simeq f(\hat{\mathbf{n}}) + \delta_{\rm LCDM}(\hat{\mathbf{n}})$$

- Topological defects: cosmic strings

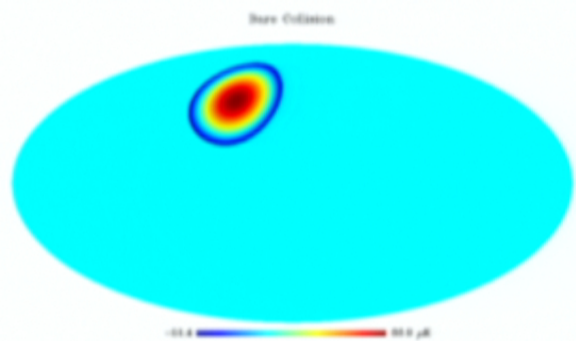


Theories with features

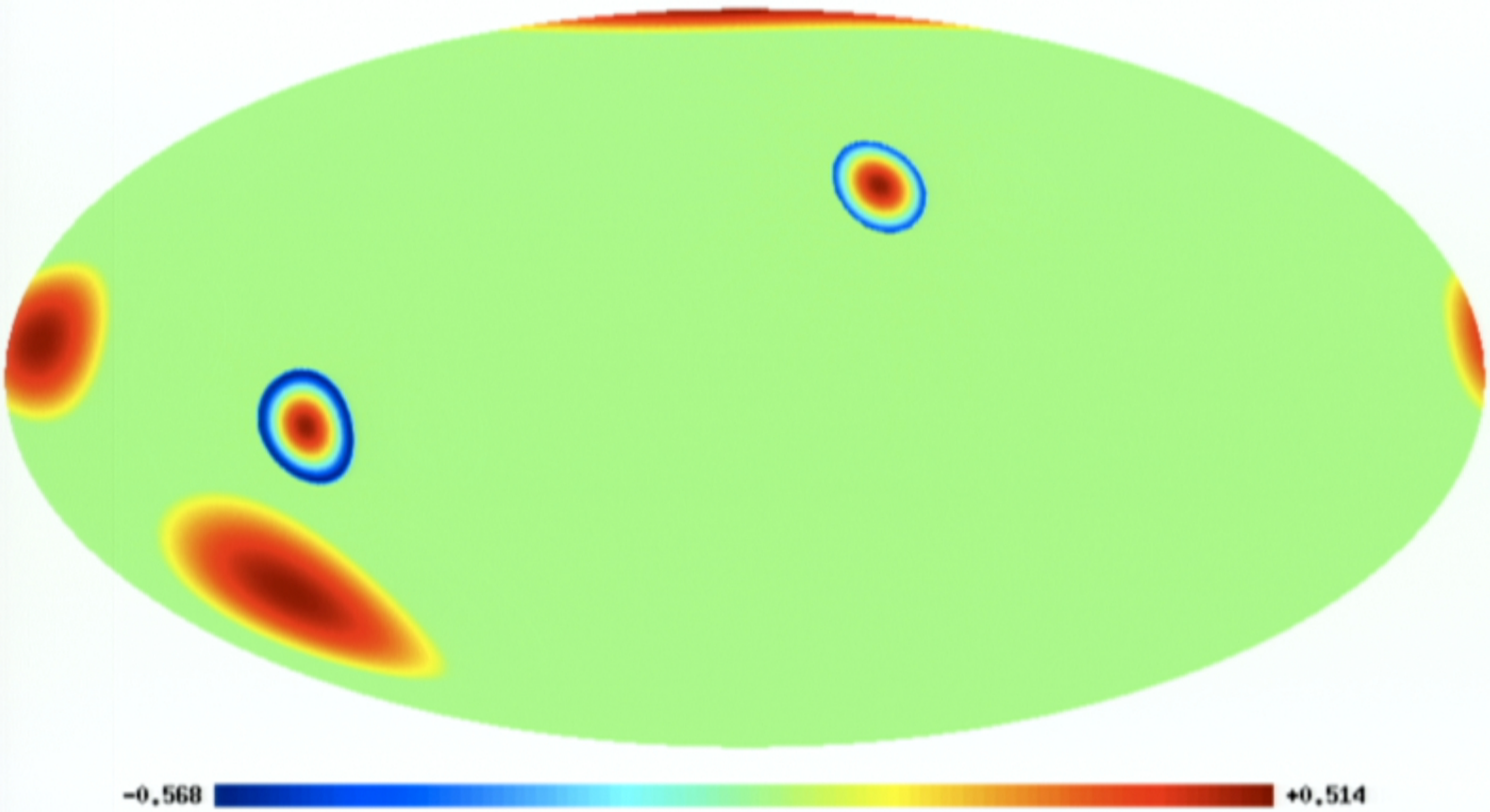
- A number of theories predict specific features in the CMB on top of those predicted by LCDM:

$$\frac{\Delta T(\hat{\mathbf{n}})}{T} = (1 + f(\hat{\mathbf{n}}))\,(1 + \delta_{\rm LCDM}(\hat{\mathbf{n}})) - 1$$
$$\simeq f(\hat{\mathbf{n}}) + \delta_{\rm LCDM}(\hat{\mathbf{n}})$$

- Topological defects: cosmic strings and cosmic textures.
- Cosmic bubble collisions: observational signature of eternal inflation.



Features

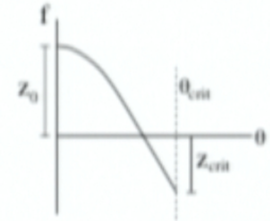


A framework for searching for features

- **A theory that predicts features:**

Template with free parameters $\mathbf{m}$

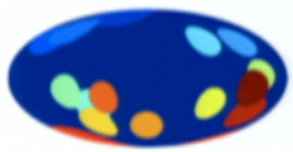
size, location, amplitude....



PDF for the number of features and template parameters

$$\Pr(\bar{\mathrm{N}}_s, \mathbf{m})$$

How many features of each type do I expect to see?



A framework for searching for features

- Bayes' theorem:

$$\Pr(\bar{N}_{\rm s}|\boldsymbol{d}) = \frac{\Pr(\bar{N}_{\rm s})\,\Pr(\boldsymbol{d}|\bar{N}_{\rm s})}{\Pr(\boldsymbol{d})}$$

A framework for searching for features

- Bayes' theorem:

$$\Pr(\bar{N}_s|\boldsymbol{d}) = \frac{\Pr(\bar{N}_s)\Pr(\boldsymbol{d}|\bar{N}_s)}{\Pr(\boldsymbol{d})}$$

- The actual number of features is drawn from a Poisson distribution:

$$\Pr(\boldsymbol{d}|\bar{N}_s) = \sum_{N_s=0}^{\infty} \Pr(N_s|\bar{N}_s)\Pr(\boldsymbol{d}|N_s)$$

$$\Pr(N_s|\bar{N}_s) = \frac{\bar{N}_s^{N_s} e^{-\bar{N}_s}}{N_s!}$$

A framework for searching for features

- Bayes' theorem:

$$\Pr(\bar{N}_s|\boldsymbol{d}) = \frac{\Pr(\bar{N}_s)\Pr(\boldsymbol{d}|\bar{N}_s)}{\Pr(\boldsymbol{d})}$$

- The actual number of features is drawn from a Poisson distribution:

$$\Pr(\boldsymbol{d}|\bar{N}_s) = \sum_{N_s=0}^{\infty} \Pr(N_s|\bar{N}_s)\Pr(\boldsymbol{d}|N_s)$$

$$\Pr(N_s|\bar{N}_s) = \frac{\bar{N}_s^{N_s} e^{-\bar{N}_s}}{N_s!}$$

- The likelihood for N_s templates is given by:

$$\Pr(\boldsymbol{d}|N_s) = \int \mathrm{d}\boldsymbol{m}_1\, \mathrm{d}\boldsymbol{m}_2 \ldots \mathrm{d}\boldsymbol{m}_{N_s} \Pr(\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots \boldsymbol{m}_{N_s})\Pr(\boldsymbol{d}|N_s, \boldsymbol{m}_1, \boldsymbol{m}_2, \ldots \boldsymbol{m}_{N_s})$$

We've marginalized over template parameters

$$\boldsymbol{m}_i = \{\theta_0^{(i)}, \phi_0^{(i)}, \theta_{\mathrm{crit}}^{(i)}, z_0^{(i)}, z_{\mathrm{crit}}^{(i)}\}$$

A framework for searching for features

- The likelihoods are given by:

$$\Pr(\boldsymbol{d}|N_s, \boldsymbol{m}_1, \boldsymbol{m}_2, \ldots \boldsymbol{m}_{N_s}) = \frac{1}{(2\pi)^{N_{px}/2}|\mathsf{C}|} \exp\left(-\frac{1}{2}\left[\boldsymbol{d} - \sum_{i=1}^{N_s} \boldsymbol{t}(\boldsymbol{m}_i)\right] \mathsf{C}^{-1} \left[\boldsymbol{d} - \sum_{i=1}^{N_s} \boldsymbol{t}(\boldsymbol{m}_i)\right]^{\mathrm{T}}\right)$$

$\mathbf{d}$	vector containing the data at each pixel
$\mathbf{t}(\mathbf{m}_i)$	vector containing the value of the template at each pixel
C	covariance matrix

- Neglecting instrumental noise:

$$\mathsf{C}_{ij} = \sum_{\ell} \frac{2\ell+1}{4\pi} C_\ell \, P_\ell(\cos\theta_{ij})$$

A framework for searching for features

- The likelihoods are given by:

$$\Pr(\boldsymbol{d}|N_s, \boldsymbol{m}_1, \boldsymbol{m}_2, \ldots \boldsymbol{m}_{N_s}) = \frac{1}{(2\pi)^{N_{pix}/2}|\mathsf{C}|} \exp\left(-\frac{1}{2}\left[\boldsymbol{d} - \sum_{i=1}^{N_s} \boldsymbol{t}(\boldsymbol{m}_i)\right] \mathsf{C}^{-1} \left[\boldsymbol{d} - \sum_{i=1}^{N_s} \boldsymbol{t}(\boldsymbol{m}_i)\right]^{\mathrm{T}}\right)$$

$\mathbf{d}$	vector containing the data at each pixel
$\mathbf{t}(\mathbf{m}_i)$	vector containing the value of the template at each pixel
C	covariance matrix

- Neglecting instrumental noise:

$$\mathsf{C}_{ij} = \sum_{\ell} \frac{2\ell+1}{4\pi} C_\ell \, P_\ell(\cos\theta_{ij})$$

A framework for searching for features

- Bayes' theorem:

$$\Pr(\bar{N}_s|\boldsymbol{d}) = \frac{\Pr(\bar{N}_s)\Pr(\boldsymbol{d}|\bar{N}_s)}{\Pr(\boldsymbol{d})}$$

- The actual number of features is drawn from a Poisson distribution:

$$\Pr(\boldsymbol{d}|\bar{N}_s) = \sum_{N_s=0}^{\infty} \Pr(N_s|\bar{N}_s)\Pr(\boldsymbol{d}|N_s)$$

$$\Pr(N_s|\bar{N}_s) = \frac{\bar{N}_s^{N_s} e^{-\bar{N}_s}}{N_s!}$$

- The likelihood for N_s templates is given by:

$$\Pr(\boldsymbol{d}|N_s) = \int \mathrm{d}\boldsymbol{m}_1\,\mathrm{d}\boldsymbol{m}_2\ldots\mathrm{d}\boldsymbol{m}_{N_s}\,\Pr(\boldsymbol{m}_1,\boldsymbol{m}_2,\ldots\boldsymbol{m}_{N_s})\Pr(\boldsymbol{d}|N_s,\boldsymbol{m}_1,\boldsymbol{m}_2,\ldots\boldsymbol{m}_{N_s})$$

We've marginalized over template parameters

$$\boldsymbol{m}_i = \{\theta_0^{(i)}, \phi_0^{(i)}, \theta_{\mathrm{crit}}^{(i)}, z_0^{(i)}, z_{\mathrm{crit}}^{(i)}\}$$

A framework for searching for features

- Bayes' theorem:

$$\Pr(\bar{N}_s|\boldsymbol{d}) = \frac{\Pr(\bar{N}_s)\Pr(\boldsymbol{d}|\bar{N}_s)}{\Pr(\boldsymbol{d})}$$

- The actual number of features is drawn from a Poisson distribution:

$$\Pr(\boldsymbol{d}|\bar{N}_s) = \sum_{N_s=0}^{\infty} \Pr(N_s|\bar{N}_s)\Pr(\boldsymbol{d}|N_s)$$

$$\Pr(N_s|\bar{N}_s) = \frac{\bar{N}_s^{N_s} e^{-\bar{N}_s}}{N_s!}$$

- The likelihood for N_s templates is given by:

$$\Pr(\boldsymbol{d}|N_s) = \int \mathrm{d}\boldsymbol{m}_1\,\mathrm{d}\boldsymbol{m}_2\ldots\mathrm{d}\boldsymbol{m}_{N_s}\,\Pr(\boldsymbol{m}_1,\boldsymbol{m}_2,\ldots\boldsymbol{m}_{N_s})\Pr(\boldsymbol{d}|N_s,\boldsymbol{m}_1,\boldsymbol{m}_2,\ldots\boldsymbol{m}_{N_s})$$

We've marginalized over template parameters

$$\boldsymbol{m}_i = \{\theta_0^{(i)},\phi_0^{(i)},\theta_{\mathrm{crit}}^{(i)},z_0^{(i)},z_{\mathrm{crit}}^{(i)}\}$$

A framework for searching for features

- The likelihoods are given by:

$$\Pr(\boldsymbol{d}|N_s, \boldsymbol{m}_1, \boldsymbol{m}_2, \ldots \boldsymbol{m}_{N_s}) = \frac{1}{(2\pi)^{N_{\rm pix}/2}|\mathsf{C}|} \exp\left(-\frac{1}{2}\left[\boldsymbol{d} - \sum_{i=1}^{N_s} \boldsymbol{t}(\boldsymbol{m}_i)\right] \mathsf{C}^{-1} \left[\boldsymbol{d} - \sum_{i=1}^{N_s} \boldsymbol{t}(\boldsymbol{m}_i)\right]^{\rm T}\right)$$

$\mathbf{d}$ vector containing the data at each pixel

$\mathbf{t}(\mathbf{m}_i)$ vector containing the value of the template at each pixel

C covariance matrix

- Neglecting instrumental noise:

$$\mathsf{C}_{ij} = \sum_{\ell} \frac{2\ell+1}{4\pi} C_\ell \; P_\ell(\cos\theta_{ij})$$

A framework for searching for features

- Problems:
  - Covariance matrix is very large and non-diagonal, hard to invert!
  - Need to sample a many-dimensional parameter space!
  - Arbitrary number of templates!

A framework for searching for features

- The likelihoods are given by:

$$\Pr(\boldsymbol{d}|N_s, \boldsymbol{m}_1, \boldsymbol{m}_2, \ldots \boldsymbol{m}_{N_s}) = \frac{1}{(2\pi)^{N_{\rm pix}/2}|\mathsf{C}|} \exp\left(-\frac{1}{2}\left[\boldsymbol{d} - \sum_{i=1}^{N_s} \boldsymbol{t}(\boldsymbol{m}_i)\right] \mathsf{C}^{-1} \left[\boldsymbol{d} - \sum_{i=1}^{N_s} \boldsymbol{t}(\boldsymbol{m}_i)\right]^{\rm T}\right)$$

$\mathbf{d}$	vector containing the data at each pixel
$\mathbf{t}(\mathbf{m}_i)$	vector containing the value of the template at each pixel
C	covariance matrix

- Neglecting instrumental noise:

$$\mathsf{C}_{ij} = \sum_{\ell} \frac{2\ell+1}{4\pi} C_\ell \; P_\ell(\cos\theta_{ij})$$

A framework for searching for features

- Problems:
  - Covariance matrix is very large and non-diagonal, hard to invert!
  - Need to sample a many-dimensional parameter space!
  - Arbitrary number of templates!
- Solution:
  - Locate candidate features with a blind analysis.
  - Find an approximation to the posterior by integrating only over the regions of parameter space where the likelihood is large.

A framework for searching for features

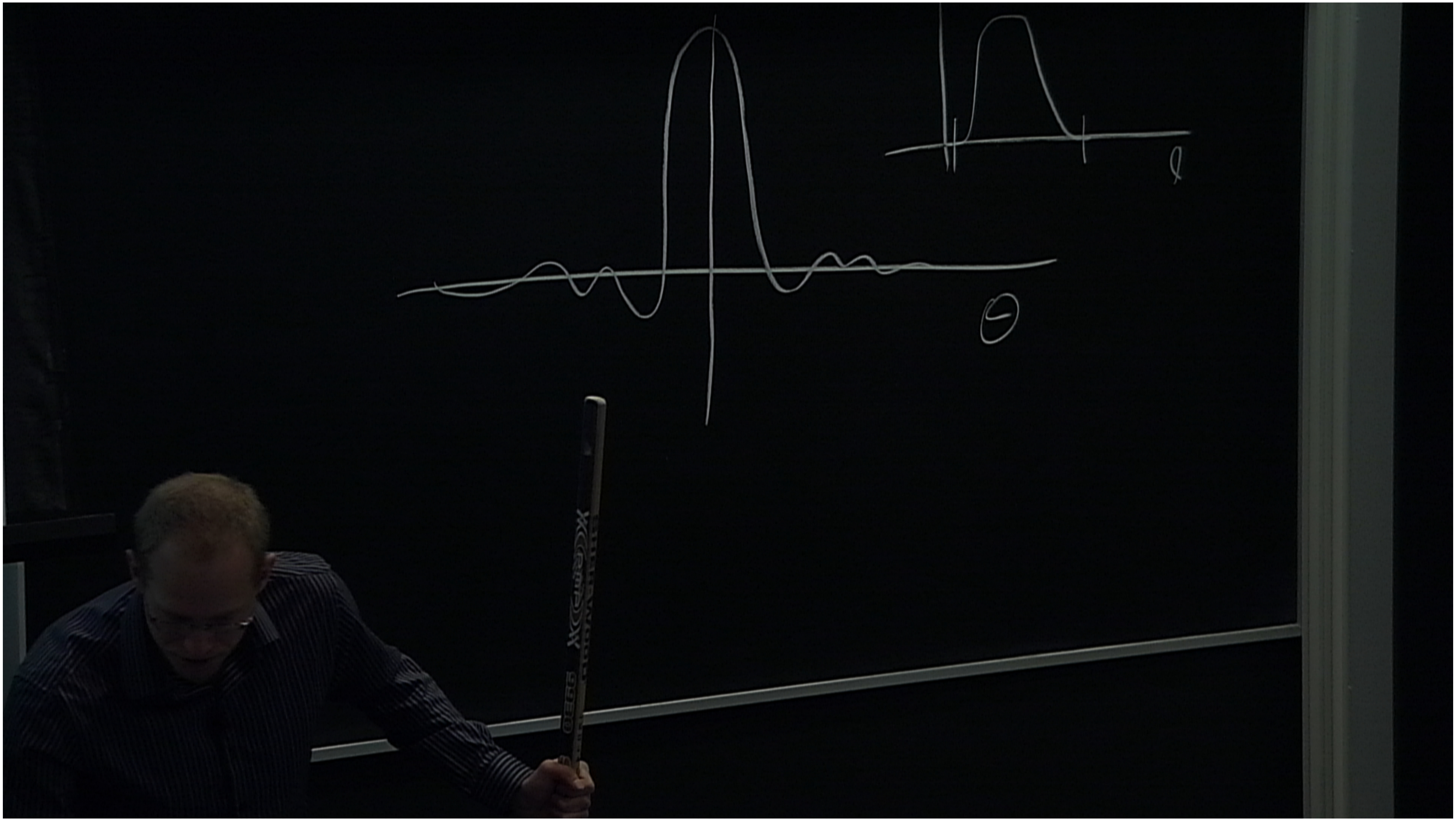
- Problems:
  - Covariance matrix is very large and non-diagonal, hard to invert!
  - Need to sample a many-dimensional parameter space!
  - Arbitrary number of templates!
- Solution:
  - Locate candidate features with a blind analysis.
  - Find an approximation to the posterior by integrating only over the regions of parameter space where the likelihood is large.

A framework for searching for features

- The current pipeline uses needlets (a type of wavelet transform) to identify candidates: sensitive to scale (j) and location (k).

$$T(\hat{\gamma}) = \sum_{j,k} \beta_{jk}\psi_{jk}(\hat{\gamma}) \qquad \beta_{jk} = \int T(\hat{\gamma})\psi_{jk}(\hat{\gamma})d\Omega.$$

- The coefficients are computed in harmonic space.

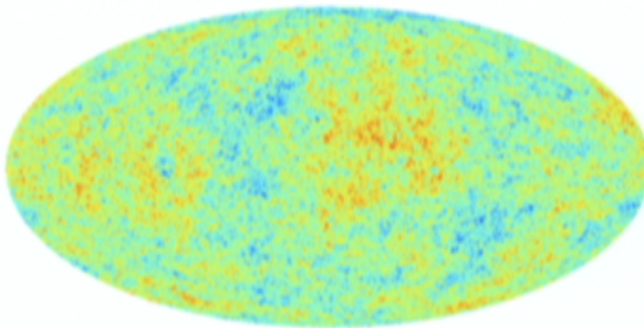


A framework for searching for features

- The current pipeline uses needlets (a type of wavelet transform) to identify candidates: sensitive to scale (j) and location (k).

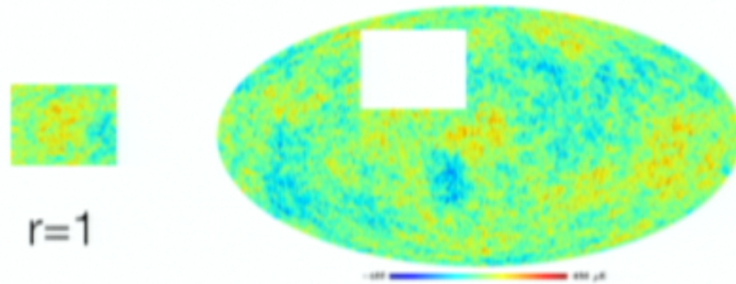
$$T(\hat{\gamma}) = \sum_{j,k} \beta_{jk}\psi_{jk}(\hat{\gamma}) \qquad \beta_{jk} = \int T(\hat{\gamma})\psi_{jk}(\hat{\gamma})d\Omega.$$

- The coefficients are computed in harmonic space.



A framework for searching for features

- **For one candidate:**



r=1

r=2

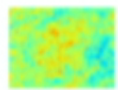
$$\Pr(\bar{N}_s|d) \propto \Pr(\bar{N}_s)\, e^{-\bar{N}_s} \; \left(1 + \bar{N}_s \rho_1\right)$$

$$\rho_1 \equiv \frac{\Pr(\boldsymbol{d}|1)}{\Pr(\boldsymbol{d}|0)} \simeq \frac{\int_{\text{region } 1} \mathrm{d}\boldsymbol{m}\, \Pr(\boldsymbol{m})\; e^{-[\boldsymbol{d}_1 - \boldsymbol{t}_1(\boldsymbol{m})]\mathsf{C}_1^{-1}[\boldsymbol{d}_1 - \boldsymbol{t}_1(\boldsymbol{m})]^{\mathrm{T}}/2}}{e^{-\boldsymbol{d}_1 \mathsf{C}_1^{-1} \boldsymbol{d}_1^{\mathrm{T}}/2}}$$

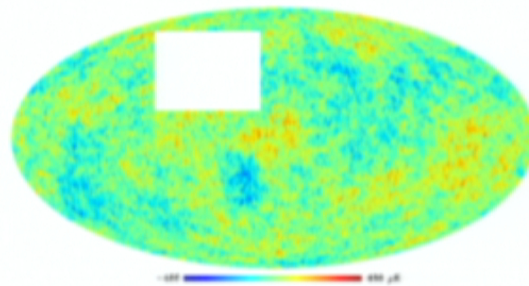
Evidence ratio in the patch: how much better does one describe the data by adding a template

A framework for searching for features

- **For one candidate:**



r=1



r=2

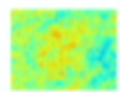
$$\Pr(\bar{N}_s|d) \propto \Pr(\bar{N}_s)\, e^{-\bar{N}_s} \; (1 + \bar{N}_s \rho_1)$$

$$\rho_1 \equiv \frac{\Pr(\boldsymbol{d}|1)}{\Pr(\boldsymbol{d}|0)} \simeq \frac{\int_{\text{region 1}} \mathrm{d}\boldsymbol{m}\, \Pr(\boldsymbol{m})\; e^{-[\boldsymbol{d}_1 - \boldsymbol{t}_1(\boldsymbol{m})]\mathsf{C}_1^{-1}[\boldsymbol{d}_1 - \boldsymbol{t}_1(\boldsymbol{m})]^{\mathrm{T}}/2}}{e^{-\boldsymbol{d}_1 \mathsf{C}_1^{-1} \boldsymbol{d}_1^{\mathrm{T}}/2}}$$

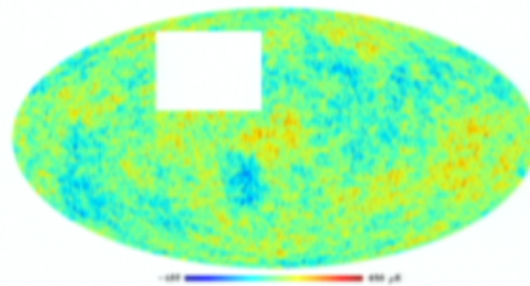
Evidence ratio in the patch: how much better does one describe the data by adding a template

A framework for searching for features

- **For one candidate:**



r=1



r=2

$$\Pr(\bar{N}_s|d) \propto \Pr(\bar{N}_s)\, e^{-\bar{N}_s} \;\left(1 + \bar{N}_s \rho_1\right)$$

$$\rho_1 \equiv \frac{\Pr(\boldsymbol{d}|1)}{\Pr(\boldsymbol{d}|0)} \simeq \frac{\int_{\text{region } 1} \mathrm{d}\boldsymbol{m}\, \Pr(\boldsymbol{m})\; e^{-[\boldsymbol{d}_1 - \boldsymbol{t}_1(\boldsymbol{m})]\mathsf{C}_1^{-1}[\boldsymbol{d}_1 - \boldsymbol{t}_1(\boldsymbol{m})]^{\mathrm{T}}/2}}{e^{-\boldsymbol{d}_1 \mathsf{C}_1^{-1} \boldsymbol{d}_1^{\mathrm{T}}/2}}$$

Evidence ratio in the patch: how much better does one describe the data by adding a template

A framework for searching for features

- The general expression for N_b candidates:

Theory prior

Cosmic variance

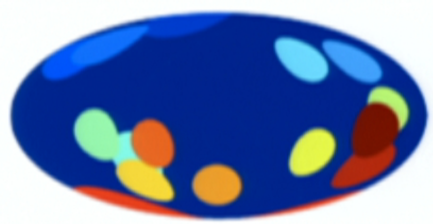
All combos of templates and blobs

$$\Pr(\bar{N}_s|\mathbf{d}, f_{\rm sky}) \propto \Pr(\bar{N}_s)\, e^{-f_{\rm sky}\bar{N}_s} \sum_{N_s=0}^{N_b} \frac{(f_{\rm sky}\bar{N}_s)^{N_s}}{N_s!} \sum_{b_1,b_2,\ldots,b_{N_s}=1}^{N_b} \left[\prod_{s=1}^{N_s} \rho_{b_s} \prod_{i,j=1}^{N_s} (1-\delta_{s_i,s_j})\right]$$

Expected number of features

Poisson process

Evidence ratio in each blob



A framework for searching for features

- The general expression for N_b candidates:

Theory prior

Cosmic variance

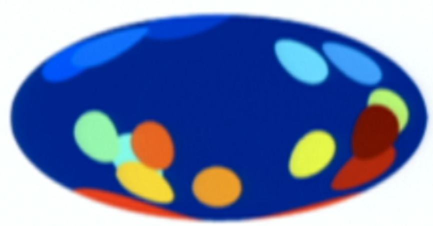
All combos of templates and blobs

How do we evaluate this expression?

Expected number of features

Poisson process

Evidence ratio in each blob



CMB data from WMAP

- Data is publicly available from the LAMBDA web site.
- Full sky data is available at 5 frequencies: useful to remove foregrounds.
- The data is represented using the Healpix pixelization scheme:

CMB data from WMAP

- Data is publicly available from the LAMBDA web site.
- Full sky data is available at 5 frequencies: useful to remove foregrounds.
- The data is represented using the Healpix pixelization scheme:

- Hierarchical, Equal Area, isoLatitude Pixelization.

$$N_{\rm pix} = 12\ N_{\rm side}^2 \qquad N_{\rm side} = 1, 2, \ldots$$

- WMAP data: $N_{\rm side} = 512$, $N_{\rm pix} = 3,145,728$.

Nyquist: 2 pixels per beam width (.2 degrees)

CMB data from WMAP

- Data is publicly available from the LAMBDA web site.
- Full sky data is available at 5 frequencies: useful to remove foregrounds.
- The data is represented using the Healpix pixelization scheme:

- Hierarchical, Equal Area, isoLatitude Pixelization.

$$N_{\rm pix} = 12\ N_{\rm side}^2 \qquad N_{\rm side} = 1, 2, \ldots$$

- WMAP data: $N_{\rm side} = 512$, $N_{\rm pix} = 3,145,728$.

Nyquist: 2 pixels per beam width (.2 degrees)

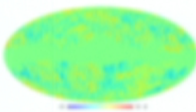
CMB data from WMAP

- HEALPix software: public tools for the analysis and simulation of data.
- Fortran 90, C++, IDL...
- Parallelized.
  - Pixelization allows fast and accurate spherical harmonic transform.
  - Subroutines to manipulate data, perform forward and inverse spherical harmonic transforms, perform file I/O....
  - Facilities to make simulated maps, find power spectra, visualize data....

CMB data from WMAP

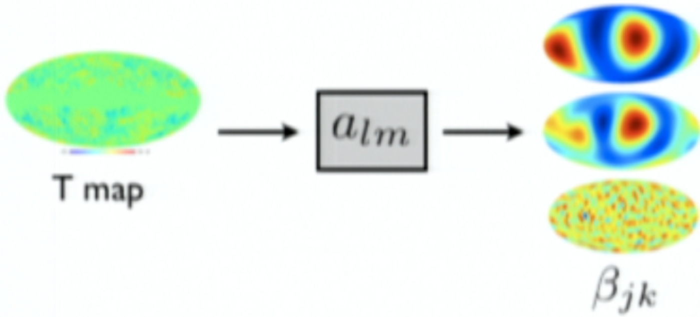
- HEALPix software: public tools for the analysis and simulation of data.
- Fortran 90, C++, IDL...
- Parallelized.
    - Pixelization allows fast and accurate spherical harmonic transform.
    - Subroutines to manipulate data, perform forward and inverse spherical harmonic transforms, perform file I/O....
    - Facilities to make simulated maps, find power spectra, visualize data....

Schematic of the pipeline

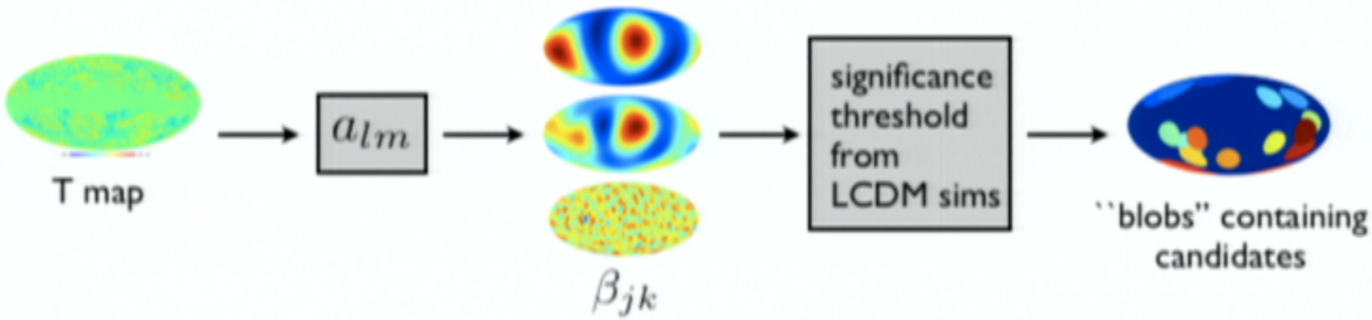


T map

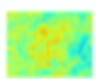
Schematic of the pipeline



Schematic of the pipeline

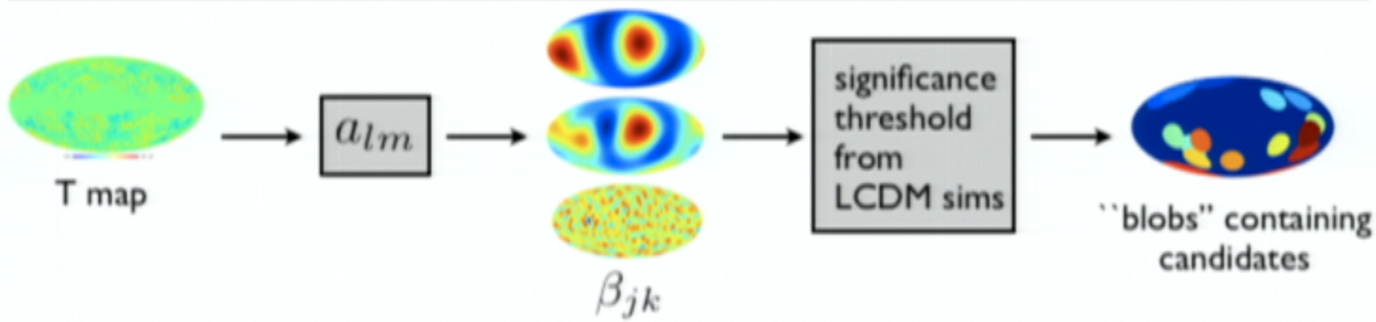


For each blob:

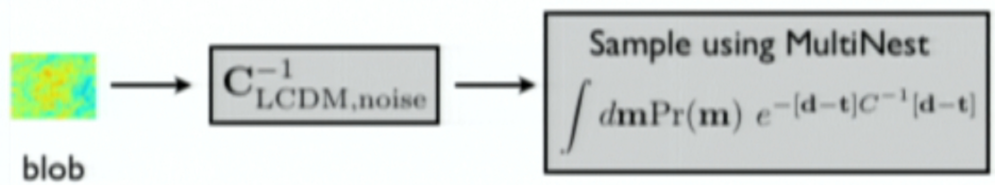


blob

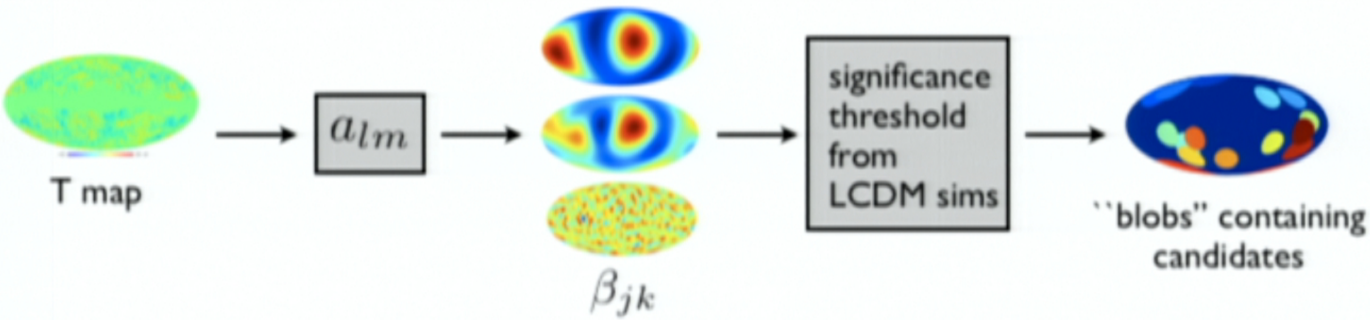
Schematic of the pipeline



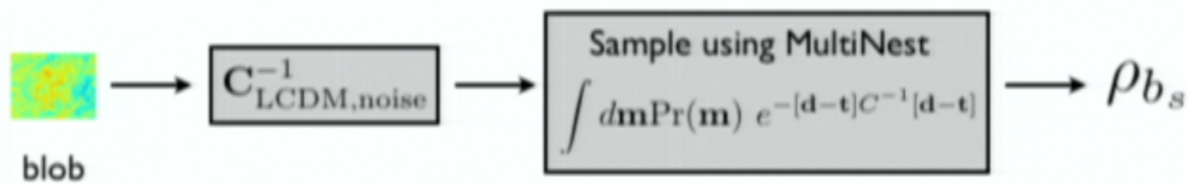
For each blob:



Schematic of the pipeline



For each blob:

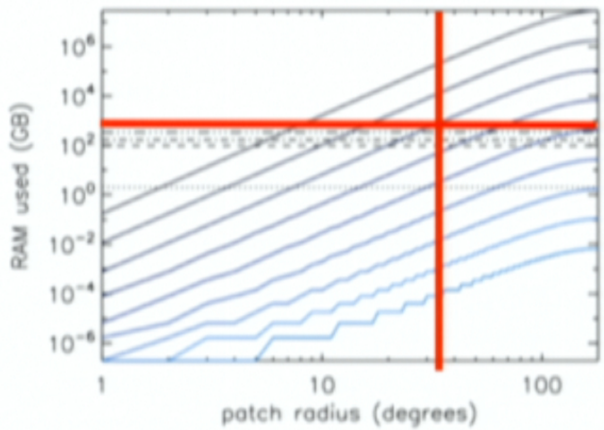


Combine according to:

$$\Pr(\bar{N}_s|\mathbf{d}, f_{\rm sky}) \propto \Pr(\bar{N}_s)\, e^{-f_{\rm sky}\bar{N}_s} \sum_{N_s=0}^{N_b} \frac{(f_{\rm sky}\bar{N}_s)^{N_s}}{N_s!} \sum_{b_1,b_2,\ldots,b_{N_s}=1}^{N_b} \left[\prod_{s=1}^{N_s} \rho_{b_s} \prod_{i,j=1}^{N_s} (1-\delta_{s_i,s_j})\right]$$

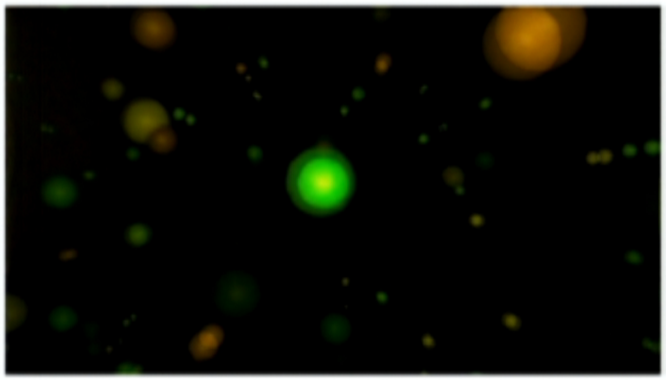
Limitations

- Each blob contains a huge amount of data.
- Storing and inverting the covariance matrix requires lots of RAM:

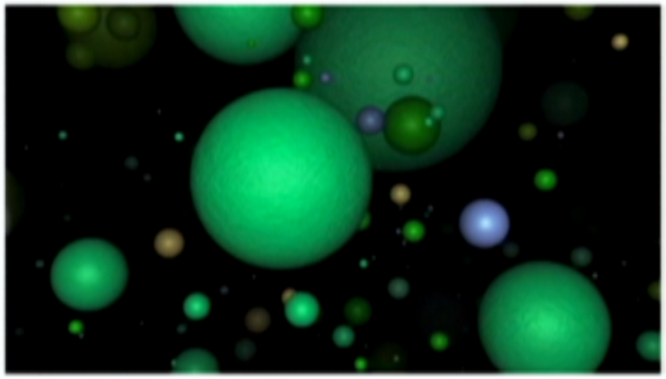


- Currently running on a 672GB shared memory machine.
- The code is parallelized.
- The resolution inside a blob is degraded according to the resources available.

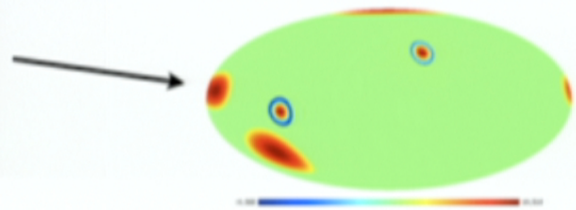
Results: Cosmic bubble collisions



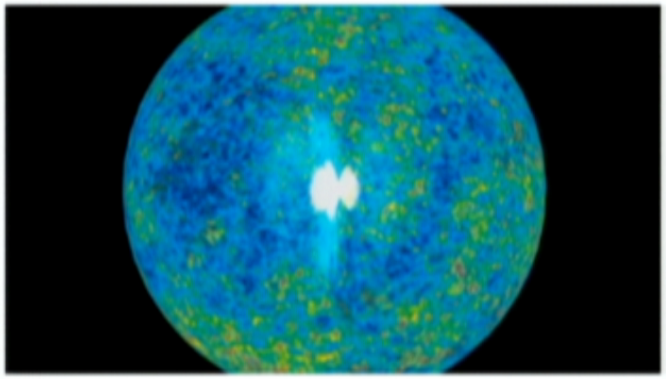
Results: Cosmic bubble collisions



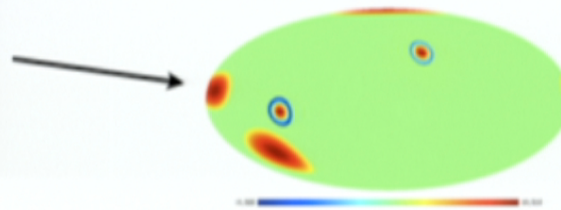
- The collision of our bubble with others provides an observational test of eternal inflation.



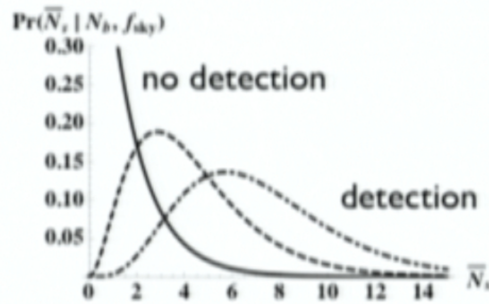
Results: Cosmic bubble collisions



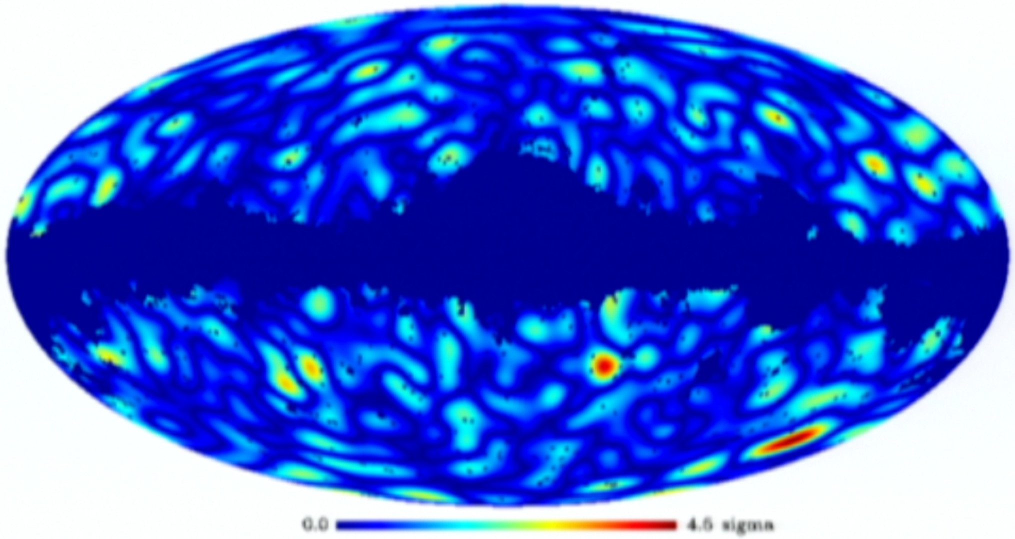
- The collision of our bubble with others provides an observational test of eternal inflation.



- Some possible outcomes:



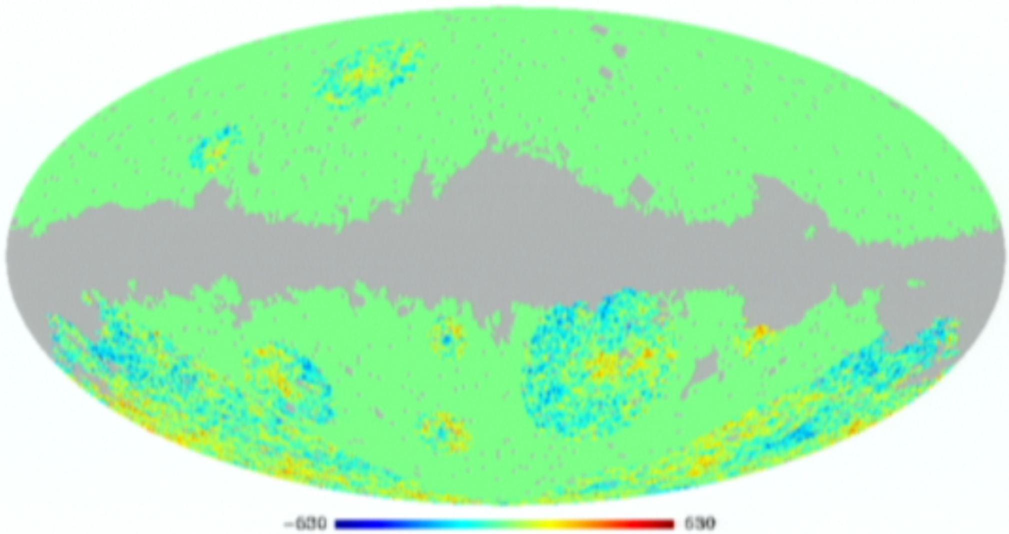
WMAP7 W-Band (94 GHz) : Needlets



(sensitive to 5-14 degree templates)

11 features pass thresholds, with detections in multiple needlet types/frequencies

WMAP7 W-Band (94 GHz) : Blobs



Divide the sky into “blobs” containing each candidate collision.

WMAP7 W-Band (94 GHz) : Posterior

- The posterior is peaked around $\bar{N}_s = 0$

  The data does not support the bubble collision hypothesis.

- From the shape of the posterior, we can rule out

$$\bar{N}_s < 1.6 \text{ at } 68\% \text{ CL}$$

WMAP7 W-Band (94 GHz) : Posterior

- The posterior is peaked around $\bar{N}_s = 0$

  The data does not support the bubble collision hypothesis.

- From the shape of the posterior, we can rule out

$$\bar{N}_s < 1.6 \text{ at } 68\% \text{ CL}$$