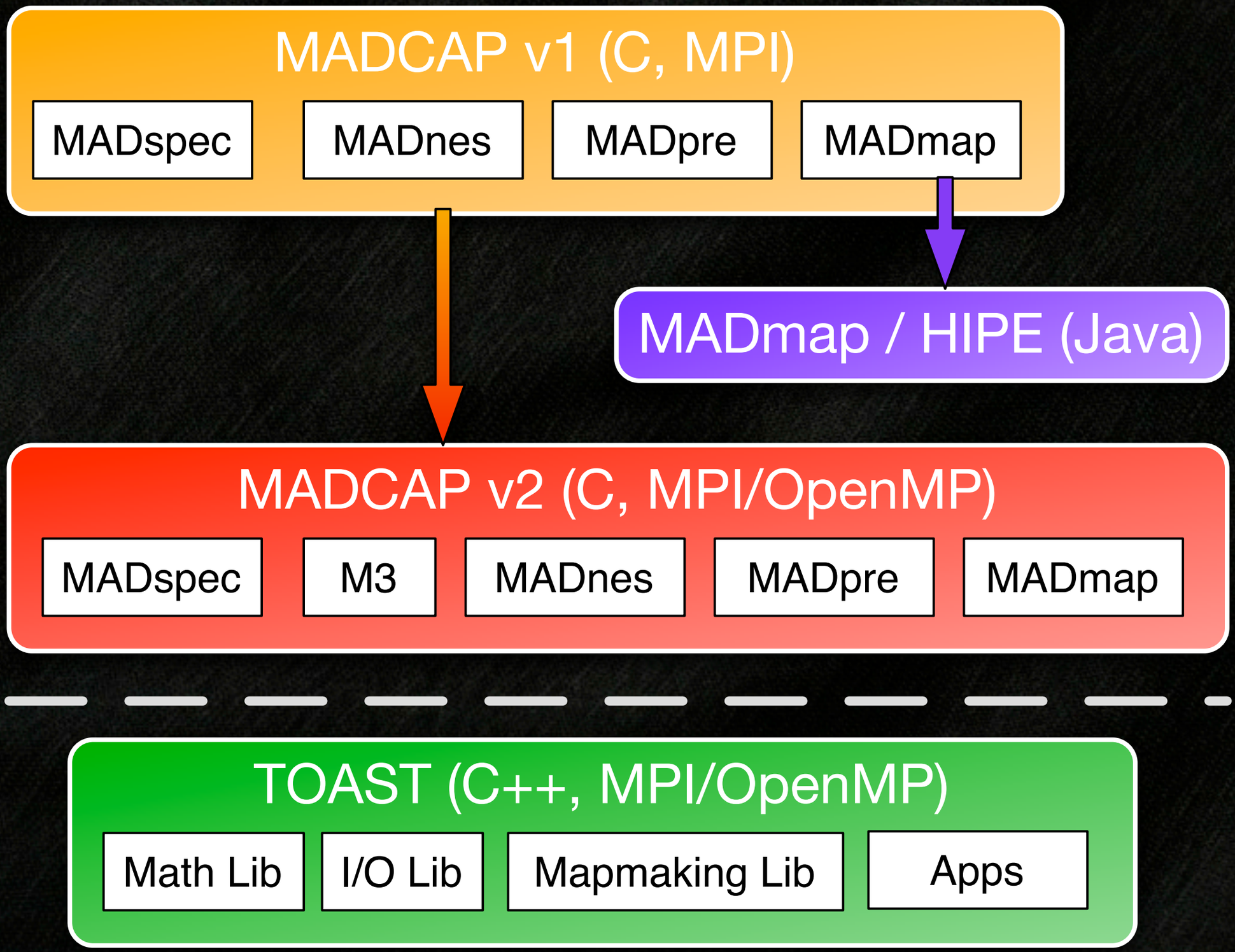
MADmap (and TOAST) for PACS

Theodore Kisner

Computational Cosmology Center, LBNL

2013-01-28

Historical Perspective



Generalized Least Squares Map-making

$b = \mathcal{C}\ x$ But we have (Gaussian) noise...

$d_t = \mathcal{A}_{tp}\ s_p + n_t \qquad \mathcal{N}_{tt'} = \langle n_t\ n_{t'} \rangle$

Algebra: plug into maximum likelihood expression...

$$\left(\mathcal{A}^T_{tp}\ \mathcal{N}^{-1}_{tt'}\ \mathcal{A}_{t'p'}\right)\ s_p = \mathcal{A}^T_{tp}\ \mathcal{N}^{-1}_{tt'}\ d_t$$

...and we have the noise-weighted least squares equation for the maximum likelihood map

Generalized Least Squares Map-making

$$\left(\mathcal{A}^T_{tp}\,\mathcal{N}^{-1}_{tt'}\,\mathcal{A}_{t'p'}\right)\; s_p = \mathcal{A}^T_{tp}\,\mathcal{N}^{-1}_{tt'}\,d_t$$

Inverse noise covariance
(Toeplitz - use FFT for product)

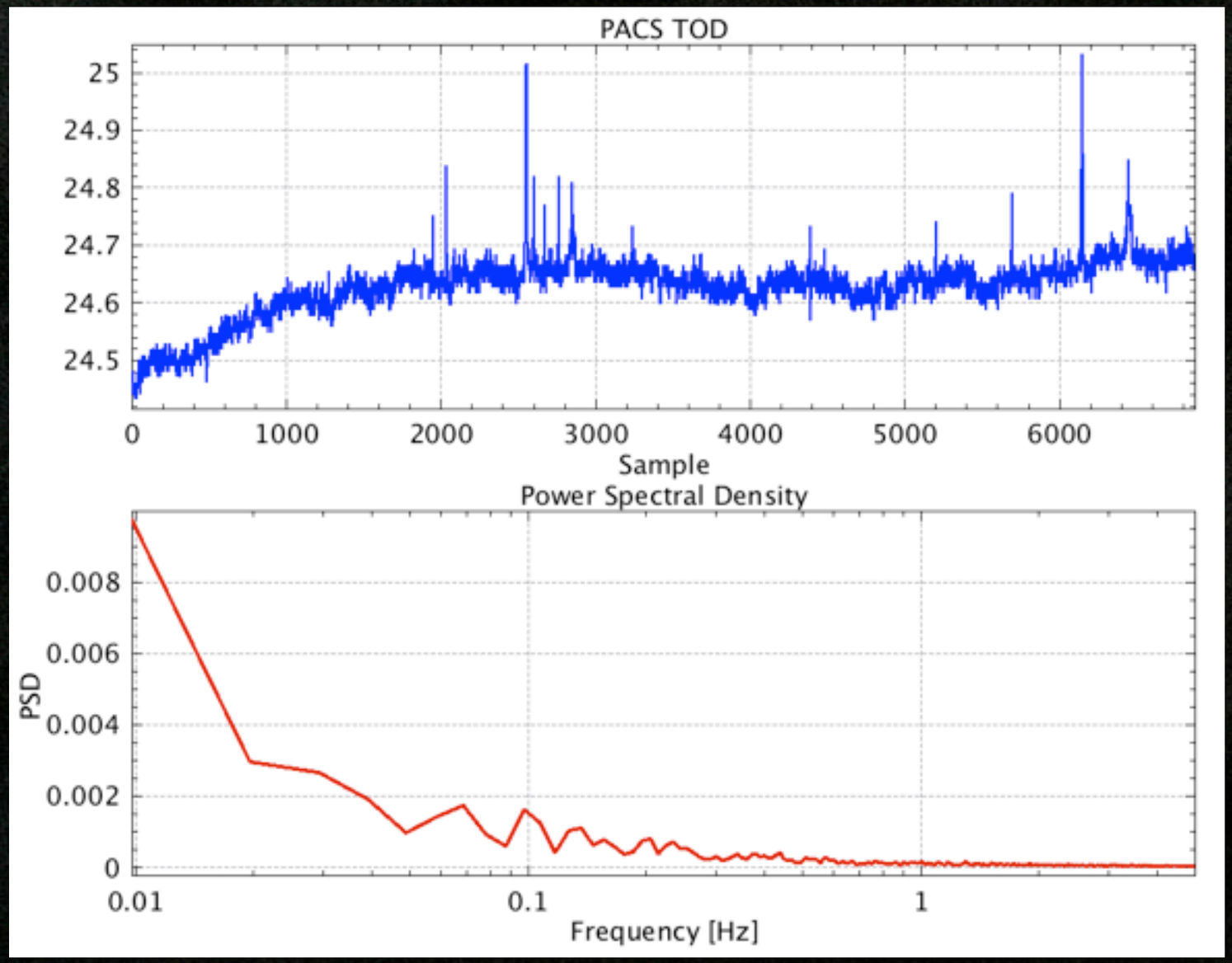
RHS /
“noise weighted map”

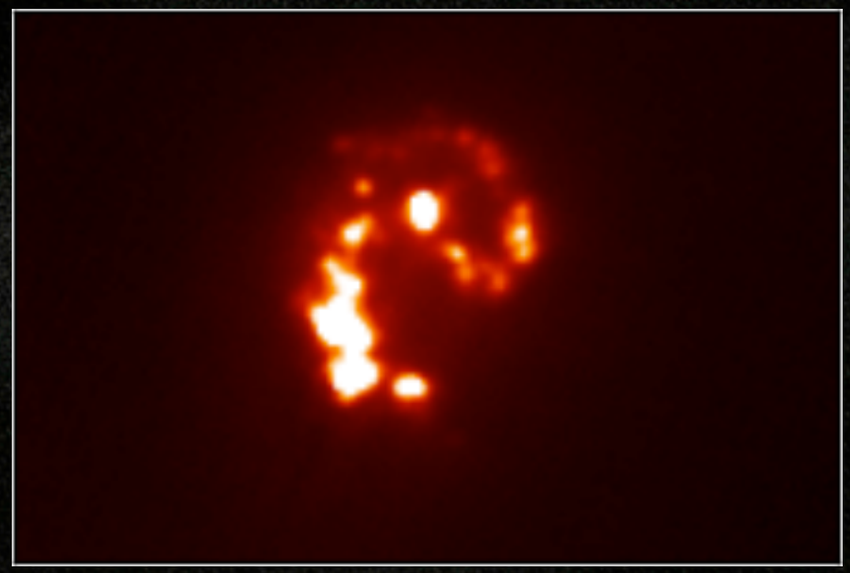
Solve this iteratively with a PCG...

GLS Map-makers do not Stand Alone...

- (Except for simulations)
- Real data requires some way to estimate the noise covariance: need a “noise dominated” timestream with the same noise properties as your dataset.
  - If S/N is bad, just use the data!
  - Find some other data without signal which has the same noise properties
  - Subtract estimate of the sky signal (e.g. destriped map) from the TOD and use that to build covariance

PACS TOD


PACS TOD
25
24.9
24.8
24.7
24.6
24.5
0
1000
2000
3000
4000
5000
6000
Sample
Power Spectral Density
0.008
0.006
0.004
0.002
0
PSD
0.01
0.1
1
Frequency [Hz]



PACS Noise Estimation

How to get a noise TOD for covariance estimation?

- How stable is noise? Can “dark” observations be trusted?
- Make a first guess at signal map by destriping or filter-and-binning

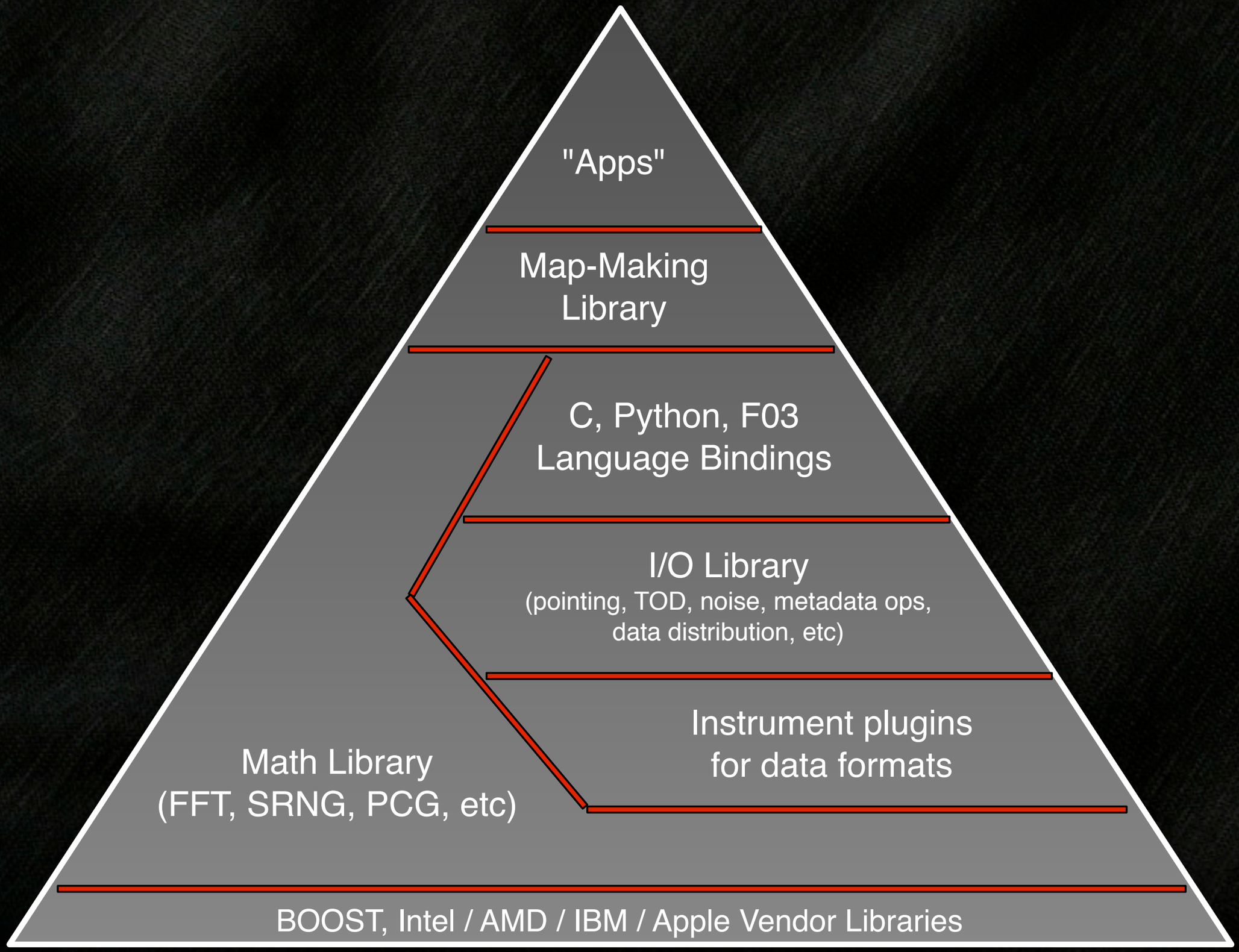
Are common-mode drifts “signal” or “noise”?

- If Signal: subtract it from the beginning, BUT this breaks linear data model, so need simulations to verify that the effect is small
- If Noise: need to handle full cross spectra between detectors...

Time Ordered Astrophysics Scalable Tools (TOAST)

- Multiple instruments supported through use of I/O plugins (basically C++ derived classes)
- Instrument-specific data selection (in python) used to build up a “run” file containing which data to process
- Built in data distribution and caching across MPI tasks
- OpenMP and MPI/OpenMP versions of map-making tools: diagonal pixel noise covariance, GLS mapmaker, simple destriper, noise estimation, etc
- Hi-level apps for “usual” map-making tasks

TOAST Code Organization



Status of TOAST with PACS

Task	Status
Class for PACS TOD reading	IN PROGRESS
Class for reading PACS boresight pointing	IN PROGRESS
WCSLIB integration	IN PROGRESS
Plan for (spherical) focalplane geometry	DONE
Format for focalplane geometry	NOT STARTED
Python class for building runs	NOT STARTED
PSF model / partial deconvolution?	R & D
Testing!	NEVER DONE...

Future

- Our group looks forward to more engagement with Herschel people and data!
- Techniques and experience from CMB map-making may be useful for Herschel (and vice-versa)
- We are working on TOAST documentation and trying to release the code “soon”. Code is currently in a private github repo, and internal LBNL paperwork is proceeding...