NHSC/PACS Webinar

PACS Extended Emission Data Processing

Data Processing Tools Outside HIPE: UNIMAP (http://w3.uniroma1.it/unimap/)

Roberta Paladini

(with a significant inputs from Lorenzo Piazzo's presentation at Herschel Map-Making Worskhop)

April 2013

PART I:
The UNIMAP package - Overview

The Developer Team

- L. Piazzo, D. Ikhenaode (University of Rome – DIET)
- M. Pestalozzi, S. Pezzuto, D. Elia, E. Schisano (IAPS-INAF)
- L. Calzoletti, F. Faustini (ASDC-ASI)

History

- Early development took place within Hi-GAL Key Program
- Originally, code was known as ROMAGAL (Traficante et al., 2011)
- UNIMAP is a *user-friendly* version of ROMAGAL
- It is now a standalone project funded by the Italian Space Agency (ASI)

Features

Environment:

- Code is in MATLAB
- It is compatible with Linux, Windows, Mac environments
- Compiled version can run w/o MATLAB
- Compiled version currently distributed for Linux and Mac (64-bit machines)

Interactivity:

- execution is controlled with parameter file
- automatic execution uses default values of parameters
- typically, default parameters values provide good quality results

Flexibility:

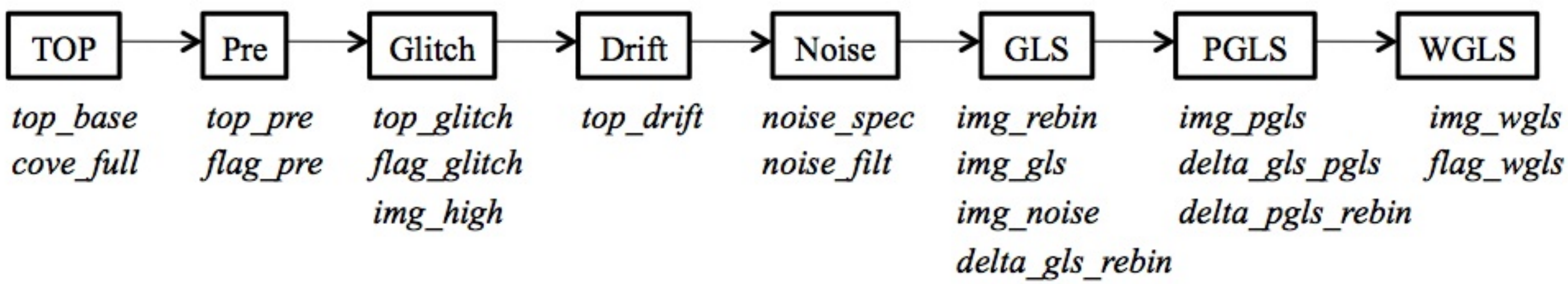
- Each pipeline step can be run separately
- Input data can be downsampled for fast-execution tests
- Intermediate products are saved so execution can be re-started

Run Time:

- 10^6 frame data set for blue channel only requires ~12 GB RAM

PART II:
The UNIMAP package – How it Works

UNIMAP Pipeline



TOP:	astrometry and offset removal	**Pre-processing**
Pre:	jumps and onset detection and removal	
Glitch:	glitch detection and removal	
Drift:	drift estimation and removal	
Noise:	noise spectrum estimate	
GLS:	noise removal	**GLS**
PGLS:	removal of GLS distortions	
WGLS:	minimization of PGLS noise	

Subspace Least Square (SLS) Drift Removal

Data Model : d = s + y + n

signal

drift

(= bolometers common mode)

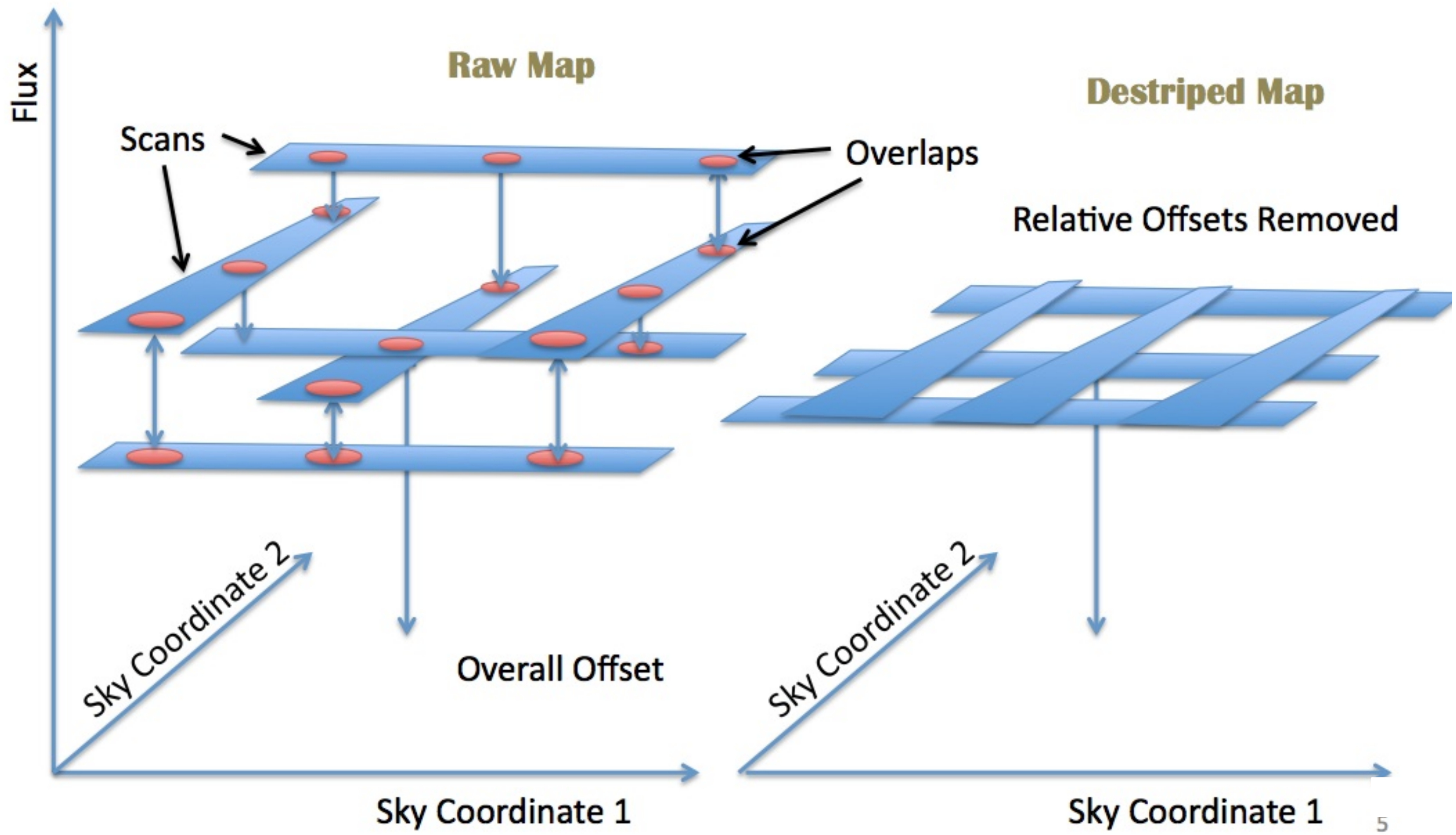
noise

Assumptions:

- scans of different bolometers across the sky typically overlap in some positions
- each readout in a signal timeline S(t) is associated to a specific position on the sky
- positions in the sky where timelines of different detectors and scan directions cross constrain the solution

Subspace Least Square (SLS) Drift Removal

This is basically the SPIRE destriper !



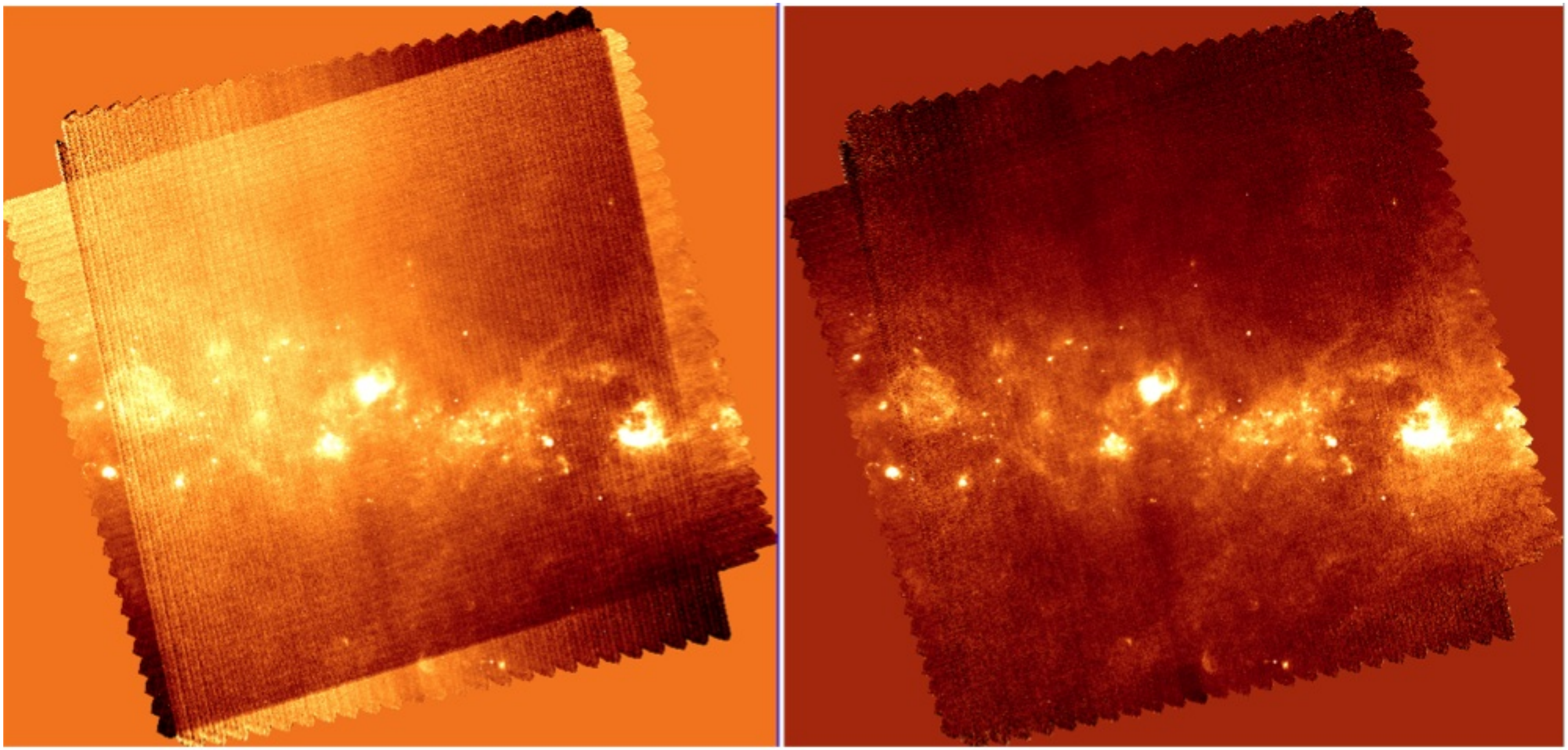
Subspace Least Square (SLS) Drift Removal

Basic Algorithm

- Make first naïve map
- Re-sample readouts within map
- Compare each re-sampled signal timeline with the corresponding original timeline
- For each timeline: fit offset function to difference (→ simplest case is zero-order polynomial)
- Subtract fitted offset function from original timeline and make another map
- Calculate χ^2 and continue with re-sampling step while difference between consecutive χ^2 is above threshold
- Stop at convergence

NOTE: drift can be estimated for each timeline of for a sub-array

Drift Removal - Example



L004 blue. *top_pre* *top_drift*

Generalized Least Square (GLS)

Data Model : $d = s + n = Pm + n$

- d → ...
- s → signal
- n → noise
- P → pointing matrix
- m → sky signal
- n → noise (= white noise + uncorrelated 1/f)

Assumptions:

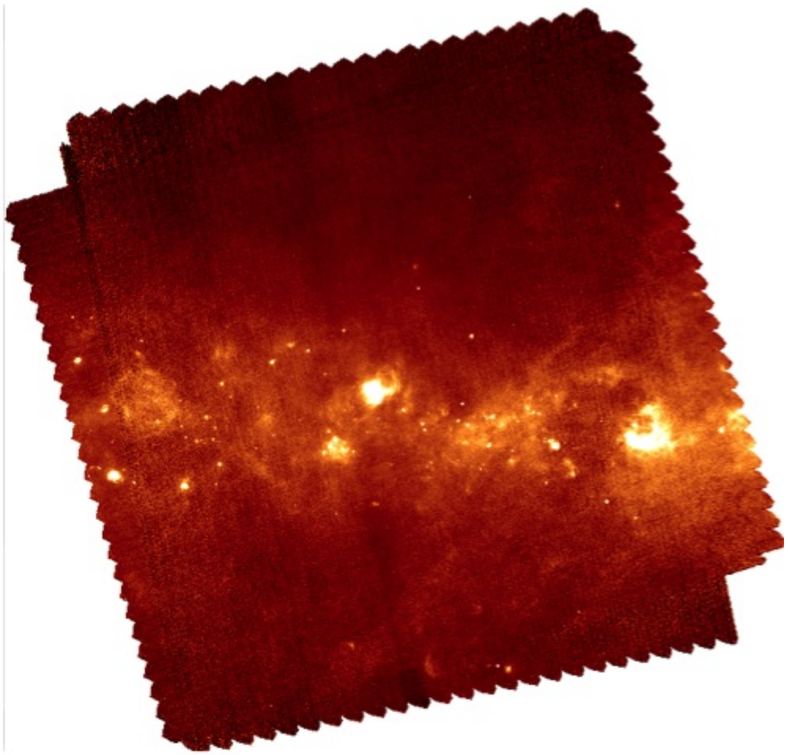
- noise is Gaussian and with zero mean
- noise is piecewise stationary
- sky signal does not vary over time

$$m^* = (P^T N^{-1} P)^{-1} P^T N^{-1} d$$

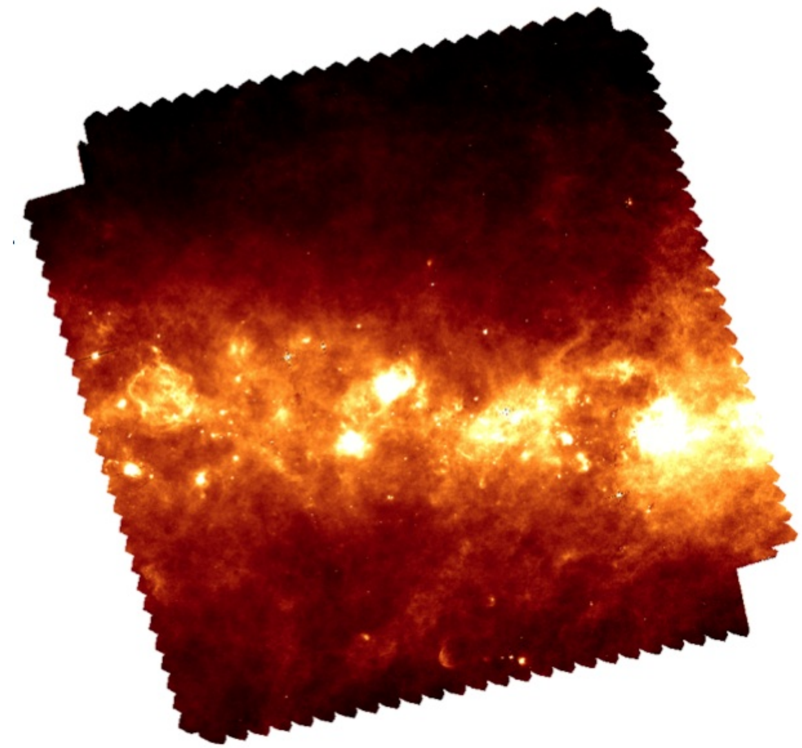
N = covariance noise matrix

Generalized Least Square (GLS)

Naïve map



GLS map

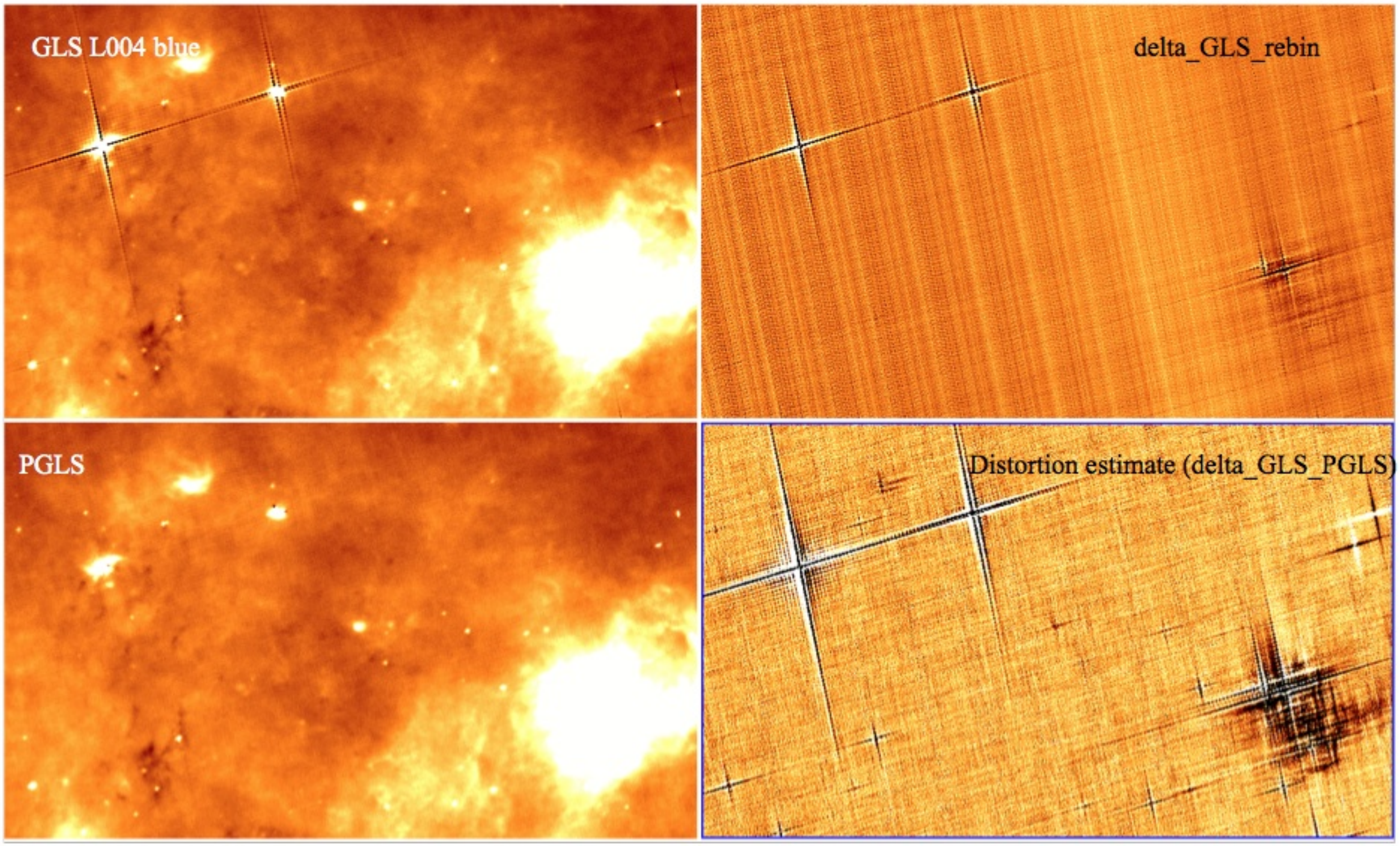


Post Processing for GLS (PGLS)

- The GLS processing step can introduce distortions
- These distortions can be: cross-like, diffuse or absent
- GLS-introduced distortions can be identified and subtracted from the GLS map
- This operation is performed by means of median filtering and naïve projection
- As a side effect of distortion corrections, noise level in the map is increased. Such a noise increase is negligible if the map has high SNR
- The noise increase can be corrected for through the WGLS processing step

NOTE: PGLS is not needed if GLS did not introduce distortions !

Post Processing for GLS (PGLS)



PART III:
The UNIMAP package – How to Use It

Download

The code can be downloaded from:

http://w3.uniroma1.it/unimap/

→ Download and unzip "Unimap" package, e.g:
"Unimap_5.4.1_maci64_package.zip"

Installation

Installation steps (ex: Mac):

- Unzip the "MCRinstaller.zip" file for the installation of the MATLAB executable
- Run ./install to install the MATLAB executable
- Write down directory where executable is going to be installed (e.g /Applications/MATLAB/MATLAB_Compiler_Runtime/v716)
- Insert in .cshrc (or equivalent for bash/tshell) the lines as instructed during MATLAB installation
- Create "example_data" dir containing the data sample provided with the distribution
- Edit "unimap_par.txt" file specifying path of "example_data" dir
- Run Unimap on example data set:

./run_unimap.sh /Applications/MATLAB/MATLAB_Compiler_Runtime/v716

UniHIPE

- UniHIPE is developed by ASI and can be downloaded at http://herschel.asdc.asi.it/index.php?page=unimap.html
- UniHIPE allows the conversion of Level 1 data from HIPE format to UNIMAP format
- It needs to be called from the same directory where HIPE is installed
- The data (i.e. HIPE Level 1 products) can be loaded from locally saved files

Inputs & Outputs

- **input data**

unimap_obsid_1.fits

...

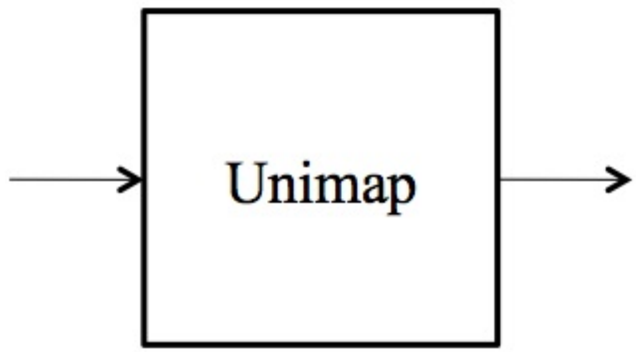
unimap_obsid_n.fits

- **par file**

unimap_par.txt

- **fits header**

unimap_header.txt



- **output images**

img_wgls.fits

img_pgls.fits

img_gls.fits

img_rebin.fits

img_high.fits

- **noise map**

img_noise.fits

- **text output**

unimap_log.txt

Parameter File

```
=============== % Params for Unimap 5.2.0 - Delete this file and run Unimap to produce defaults ======
../data/1004_16/        % data_path - working directory
250                     % max_ite_par - positive integer - global iteration limit
0                       % start_module - positive integer - first module to execute (1 = top, 2 = pre etc)
8                       % stop_module - positive integer - last module to execute (1 = top, 2 = pre, ..., 8 wgls)
0                       % save_eval_data - 0/1 - if 1 save evaluation data
0                       % save_tops - 0/1 - if 1 saves the intermediate tops
=============== % 1: Top ============================================================
1                       % top_use_galactic - 0/1 - if 1 use galactic coords, if 0 keep equatorial
1                       % top_use_gnomonic      - 0 = no projection (CAR), 1 = gnomonic (TAN), 2 = cyl eq area (CEA)
1                       % top_bolo_sub          - positive integer - bolometers subsampling
30                      % top_max_bad           - real in [0,100] - max percent of flagged samples to accept bolo
0                       % top_unit  - if 0 MJy/sr, if 1 Jy/pixel, if 2 Jy/beam (SPIRE only)
............
0                       % top_cpi2 - real - ref point coord2 in the pixel plane  (pixel)
0                       % top_nax1 - Positive integer - number of pixels on the first axis - if zero use minimum
0                       % top_nax2 - Positive integer - number of pixels on the second axis
=============== % 2: Pre ============================================================
0                       % pre_threshold - positive real - threshold for calibration detection 2
                        % pre_jump_threshold - positive real - Threshold for jump detection (0 suppress detection,.
25                      % pre_jump_hfwin - positive real - Window length for hcb (e.g. 25).
0                       % pre_onset_len - positive integer - len of onset (samples) - if 0 suppress onset removal
=============== % 3: Glitch ============================================================
25                      %glitch_hfwin - positive integer - half len of the highpass filter (samples)
0                       % glitch_sub - positive integer - subsampling for glitch search (pixels).
0                       % glitch_max_dev - positve real - threshold to declare a readout a glitch
=============== % 4: Drift ============================================================
............
```

Thank You !