

## ctools - Feature #2713

### implementation of a time-variability search tool

11/05/2018 04:30 PM - Bonnefoy Simon

<b>Status:</b>	Closed	<b>Start date:</b>	11/05/2018
<b>Priority:</b>	Normal	<b>Due date:</b>	11/11/2018
<b>Assigned To:</b>	Bonnefoy Simon	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.6.0		
<b>Description</b> Implementation of a new tool to search for time variability. The tool is intended to be used to search for variability over an event list. The energy range and the time scale can be directly set by the user. The tool is based on the algorithms used in HEGS and FAVA.  See here for more details: <a href="https://cta-redmine.irap.omp.eu/attachments/download/2963/variability_presentation.pdf">https://cta-redmine.irap.omp.eu/attachments/download/2963/variability_presentation.pdf</a>			

#### History

##### #1 - 11/21/2018 11:13 AM - Bonnefoy Simon

- File *skymap\_three\_sources.png* added
- File *skymap\_variability.png* added
- File *variability\_evolution.png* added

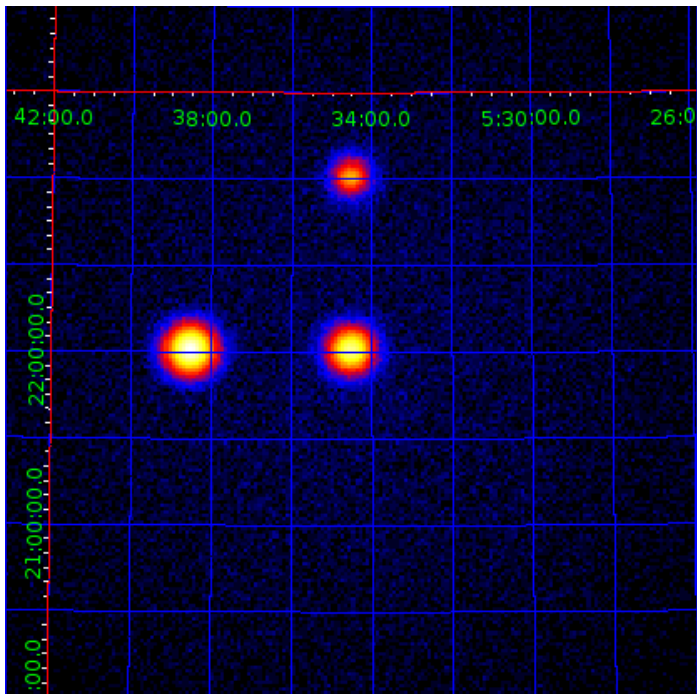
The ctfindvar tool has been implemented.

The tool searches for variability over a given field of view.  
The user can provide some observation files or an event list.  
The position of several sources can be specified in an xml file, or only one source coordinate can be set as parameter.  
In order to search for variability, the tool bins the events in time for each pixel, and compares the number of events in each bin with respect to the others bins, set as background.  
Only bins with a significance lower than 4.5, with respect to the background sample, are kept for the estimation of the background.  
This significance threshold is a parameter of the tool and can be changed by the user.  
The energy boundaries and the time scale can be set by the user as parameters.

The tool produces one FITS file as output containing:  
-a skymap with the highest significance for each pixel  
-the significance evolution of the source(s)  
-the max significance in the field of view for each time bin.

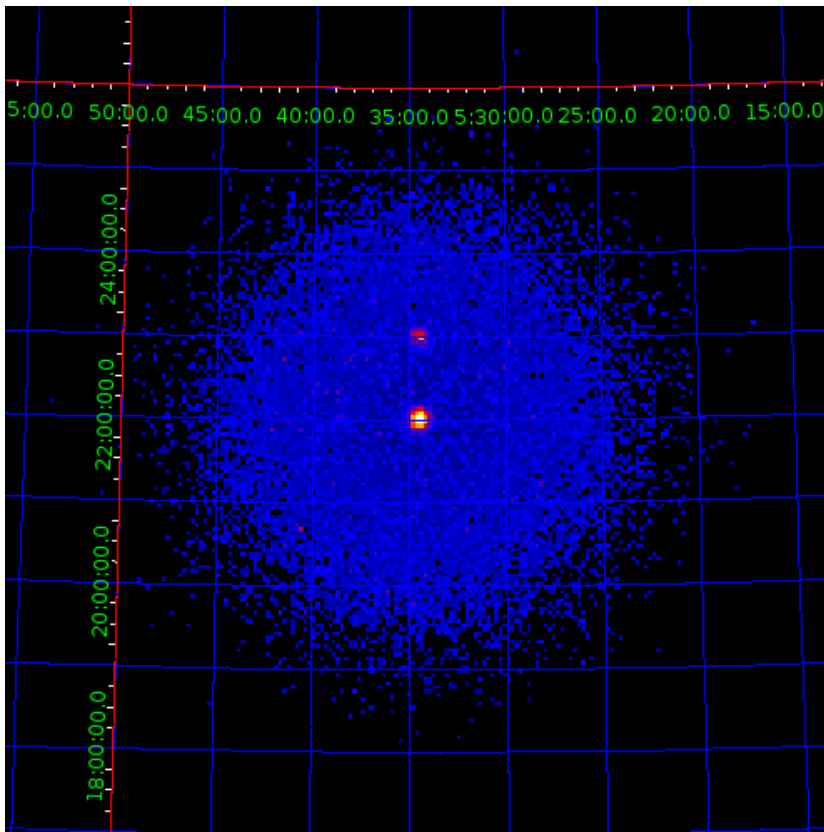
As an example, we simulated three sources:  
-two sources with different level of variability  
-one source is not variable

Here is the skymap obtained running ctskymap on the input files (count map).



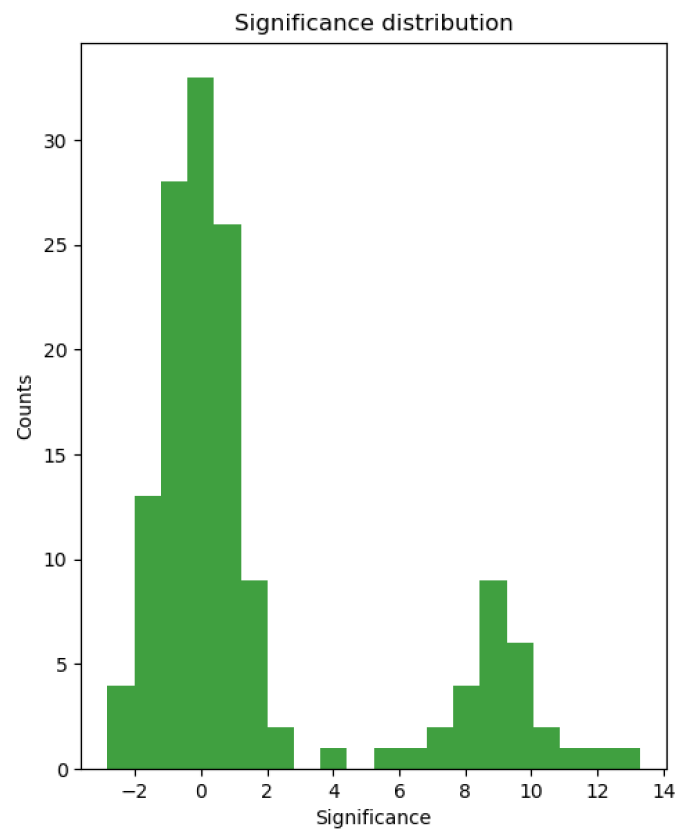
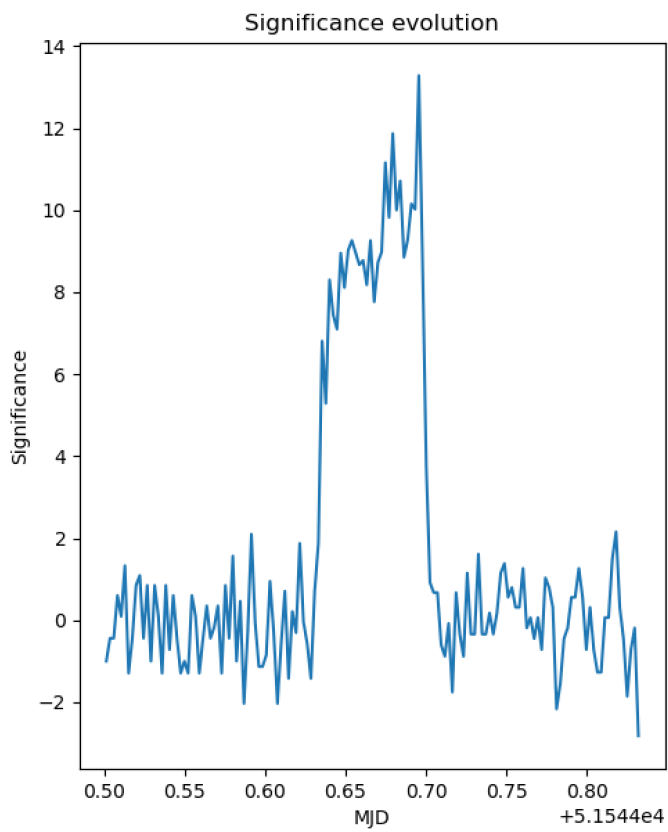
The three sources are visible with different intensities.

Below is the skymap representing the significance of the variability for each pixel



Here, only the two variable sources are detected, at different level, given the distinct variability set for the two sources.

For the sources set in a xml file, a list of the significance evolution is contained in the output file. From this list, a lightcurve can be built together with the significance distribution (see `exmaples/show_variability_evolution.py`).



## #2 - 11/21/2018 03:42 PM - Bonnefoy Simon

- Status changed from New to Pull request
- % Done changed from 0 to 100

The final tool can be found here:

Pull repo: [simonbonnefoy/ctools](#)  
Pull branch: 2713-variability-tool

## #3 - 12/11/2018 02:30 PM - Knödlseider Jürgen

- Status changed from Pull request to Feedback
- Target version set to 1.6.0

I merged the code into the devel branch.

I did a bit of restructuring of the code to make it more modular. Specifically:

- I changed `ctfindvar::init_cube` to `ctfindvar::create_cube` that also includes the filling of the cube and the optional smoothing
- I added `ctfindvar::analyse_cube` that contains the analysis code that previously was in the `ctfindvar::run` method
- I added `ctfindvar::get_pixels` that returns a vector of sky map pixel indices
- I added `ctfindvar::sky_model` that builds a sky model for a given sky direction
- I changed the interface to `ctfindvar::get_variability_sig` that now returns the results in a `GNdarray`
- I changed `ctfindvar::fill_alpha_vector` to `ctfindvar::get_alphas` that now returns a vector of alpha values
- I added `ctfindvar::get_tstart` and `ctfindvar::get_tstop` methods to determine start and stop time
- I merged `ctfindvar::write_srchist` and `ctfindvar::write_srchist_fits` into a single `ctfindvar::write_source_histograms` method. The format of the binary table extensions was also changed, since it was not very clear what the content of the second position extension was. Now, the third extension contains three columns, a first with a source name, and two others with Right Ascension and Declination. One could imagine to store also TSTART and TSTOP in seconds, and to have the time reference saved in the header file, which would make the extension more OGIP compliant.

The interface of the class now looks as follows:

```
/*
 * @class ctfindvar
 *
 * @brief Time variability search tool
 *
 * @todo Add tool description.
 */
class ctfindvar : public ctobservation {
public:
    // Constructors and destructors
    ctfindvar(void);
    explicit ctfindvar(const GObservations& obs);
    ctfindvar(int argc, char *argv[]);
    ctfindvar(const ctfindvar& app);
    virtual ~ctfindvar(void);

    // Operators
    ctfindvar& operator=(const ctfindvar& app);

    // Methods
    void clear(void);
    void run(void);
    void save(void);

    // Get the information on the time interval from
    int time2inx(const GTime& time);
    GGti inx2gti(const int& indx);
    const GSkyMap& counts(void);

protected:
    // Protected methods
    void init_members(void);
    void copy_members(const ctfindvar& app);
    void free_members(void);
```

```

void      get_parameters(void);
void      init_gtis(void);
void      create_cube(void);
void      fill_cube(GCTAObservation* obs);
void      analyse_cube(void);
std::vector<int> get_pixels(void);
GNDarray  get_variability_sig(const int& ipix);
std::vector<double> get_alphas(const int& ipix) const;
double    gti_overlap(const GGti& gti1, const GGti& gti2) const;
GTime     get_tstart(void);
GTime     get_tstop(void);
GModelSky sky_model(const GSkyDir& dir) const;
void      write_source_histograms(GFits& fits);

// Protected members
GSkyMap    m_counts;      //!< Counts for each time interval
std::vector<GGti> m_gti;   //!< List of time intervals
GModels    m_inmodel;     //!< List of models for source positions
GSkyDir     m_max_sig_dir; //!< Sky direction associated with maximum significance
double      m_minoff;     //!< Minimum counts for use in significance calculation
double      m_sig_threshold; //!< Minimum significance required to set source as variable
GSkyMap     m_peaksigmap;  //!< Skymap holding the maximum significance
GNDarray    m_pixsigsrc;   //!< Store distributions of the source significances
GNDarray    m_pixsigmax;   //!< Store distribution for pixel with max significance
GTime       m_tstart;     //!< Start time for variability study
GTime       m_tstop;      //!< Stop time for variability study
GEnergy     m_emin;       //!< Minimum energy for events
GEnergy     m_emax;       //!< Maximum energy for events
GModels     m_model_above_thr; //!< Model storing position with significance above thr
};

```

Furthermore, I changed some of the parameter names for compliance with the other ctools. Specifically:

- I removed the prefix parameter and replaced it by the explicit filename parameters outmap and outmodel
- I renamed sig\_thr to threshold
- I renamed smoothkrnl to smooth\_kernel
- I renamed smoothpar to smooth\_rad

In addition, the emin and emax parameters can no longer be zero, since the energy limits are needed in the alpha computation. This should be changed in the future so that general background models can be supported, and not only the IRF background model. I created a new feature for that: #2769.

```

#
# General parameters
#=====
inobs,    f, a, events.fits,,, "Input event list or observation definition XML file"
inmodel,   f, h, NONE,,, "Input model definition file for extracting source positions"
outcube,   f, h, NONE,,, "Output counts cube file"
outmap,    f, a, map.fits,,, "Output map file"
outmodel,  f, a, model.xml,,, "Output model definition file"
caldb,     s, a, prod2,,, "Calibration database"
irf,       s, a, South_0.5h,,, "Instrument response function"

#
# Variability search parameters
#=====
coordsys,  s, a, CEL|CEL|GAL,, "Coordinate system (CEL - celestial, GAL - galactic)"
xsrc,      r, a, 83.63,0.360, "First coordinate of the source of interest (RA or galactic l)"
ysrc,      r, a, 22.01,-90.90, "Second coordinate of the source of interest (DEC or galactic b)"
emin,      r, a, 0.1,,, "Lower energy limit (TeV)"
emax,      r, a, 100.0,,, "Upper energy limit (TeV)"
threshold, r, h, 4.5,,, "Significance threshold for variability detection"
minoff,    r, h, 0,,, "Minimum required events in off bin for computing Noff and alpha"
smooth_kernel, s, h, NONE, GAUSSIAN|DISK|NONE,, "Smoothing kernel for counts maps"
smooth_rad, r, h, 0.05,,, "Smoothing kernel radius (deg)"

#
# Time binning parameters
#=====
tinterval, r, a, 300,,, "Time interval for each bin (seconds)"
tmin,      t, a, NONE,,, "Start time (UTC string, JD, MJD or MET in seconds)"
tmax,      t, a, NONE,,, "Stop time (UTC string, JD, MJD or MET in seconds)"

```

```
#
# Spatial binning parameters
#=====
usepnt, b, h, no,,, "Use pointing instead of xref/yref parameters?"
nxpix, i, a, 200,,, "Size of the X axis in pixels"
nypix, i, a, 200,,, "Size of the Y axis in pixels"
binsz, r, a, 0.02,,, "Image scale (in degrees/pixel)"
proj, s, a, CAR|AIT|AZP|CAR|GLS|MER|MOL|SFL|SIN|STG|TAN,, "Projection method"
xref, r, a, 83.63,0,360, "First coordinate of image center in degrees (RA or galactic l)"
yref, r, a, 22.51,-90,90, "Second coordinate of image center in degrees (DEC or galactic b)"

#
# Standard parameters
#=====
nthreads, i, h, 0,0,1000, "Number of parallel processes (0=use all available CPUs)"
chatter, i, h, 2,0,4, "Chattiness of output"
clobber, b, h, yes,,, "Overwrite existing output files with new output files?"
debug, b, h, no,,, "Activate debugging mode?"
mode, s, h, ql,,, "Mode of automatic parameters"
logfile, f, h, ctfindvar.log,,, "Log filename"
```

From my side things look okay. Could you please double check that the code is also okay from your side.

#4 - 03/06/2019 01:56 PM - Knödlseeder Jürgen  
- Status changed from Feedback to Closed

Code was merged into devel.

Files			
skymap_variability.png	56 KB	11/21/2018	Bonnefoy Simon
skymap_three_sources.png	74.2 KB	11/21/2018	Bonnefoy Simon
variability_evolution.png	189 KB	11/21/2018	Bonnefoy Simon