General thinks
There are three main methods of the study of the solar wind with using scintillating sources.

The first method is to work with the individual sources. We can work with sources if integral flux density is at least 3-5 times the level of confusion of the sources at this frequency and at the antenna. Regular assessment of the index of scintillation gives us the ability to track a CME at the sharp change of index (g-maps). The power spectrum allows to estimate the velocity of the solar wind. Another way to estimate the solar wind velocity based on simultaneous observations of the scintillation source on different telescopes on the same frequency and crosscorrelate scintillations. Time delay, reflected as an offset of the peak of the CCF allow us to estimate the speed.

This method was tested by us and it is implemented in the processing program.

The second method is to use the statistical ensemble of sources. If there is a lot of weak scintillating sources, very often in the observation of the BSA it is impossible to estimate the scintillation index of these weak radio sources, and can be estimated only fluctuations of flux density. We do not know the characteristics of week scintillating sources, but if our processing area on the sky has a sufficient number of compact radio sources, then averaging all scintillating sources in the square (in bin; in pixel) we can consider this area as new compact source, which has the same characteristics as other pixels. The pixels on the sky are in the same places all the time, but the real sources are shifted by approximately 1 degree per day and therefore each pixel is updated every day. It every time we have a new set of sources. We can compare this pixel with itself, but for the previous day. If there is an increased scintillation due to the CME, it will be seen on the map. The question is, what size of pixel must be to considered statistically significant for using. We tested different areas and we believe that pixel with sizes 3o*3o is the minimum possible.

This method was tested by us and really close to full automatization.

The third method is an attempt to combine the advantages of the first and second method. The main parameter that can be estimated in the course of processing the observations is a fluctuation of the flux density of the source. These fluctuations are expressed in arbitrary units. If we can connect these fluctuations with Jansky (to calibrate the observations) and will be able to evaluate daily all sources on the sky then the strong changes of the flux density fluctuations in comparison with the previous day to talk about the perturbation in the solar wind.

The third method is implemented partially, part of the program has not yet been tested, the part of programme works non correctly. Below I will talk more about the third method.

The use of all the observed compact radio sources in the program "Space Weather"

The density of the scintillating sources on the sky for the antenna (BSA LPI) is approximately 1 source per 1 square degrees on elongations from 20o to 40o and falls about 10 times at elongation near 90o. In this case fluctuations of the flux density of the weakest sources detected by the search program is approximately 100-150 mJy. We can see eyes at good elongations quantity of scintillating sources is two times less (1 source per 2 square degrees).

For BSA there are two levels of confusion signal. The first level is the confusion extended (not scintillating) sources. It is 1 Jansky. This means that if the integral flux density of the source is less than 3 Jansky, the real flux density is determined with large errors (in a result large mistakes in estimation of scintillation index). Additionally, this means that if scintillation is determined by the source with an integral flux density of less than 3 Jansky, we have a problem with the estimation scintillation index, because we do not know of average intensity. The second level of confusion is a confusion of scintillating sources. The median esimation for this level of confusion gives 150 mJy. Sample of extended and compact radio sources coincide only partly, and therefore any estimates of the scintillation index for weak sources can have errors. In any case, the border for search of a week scintillating radio sources close to the limits of level confusion because the size of beamwidth of BSA LPI is approximately 0.5*1o, and the density of detectable scintillating radio sources is approximately 1 source per 1 square degrees.

If we want to work with a complete sample of sources, then you need to have a catalogue of these sources. We do this work in Pushchino, but it is not yet finished.

The niceties of work with BSA LPI
BSA LPI is a phased array. The beams of the antenna have fixed positions in the sky. However, there is the ionosphere, which can change the visible coordinates of the sources. Typical offset of the visible coordinates of 3-5 arcmin. Since the distance between beams of approximately 30 arcmin these displacements are not important, but in the case of the disturbed ionosphere typical offset can be 10-15 minutes, and this will lead to incorrect estimates of the density fluctuations of the flux density and therefore must be taken into account.

Another point is that the phased array beams can have a full band only in a limited set of angles on the sky. For beams far from these angles the effective band is narrow. It also must be considered.

About processing programme
- programme is working, but not finish yet
- I tested some part of this programme, but not all parts
- some part of programme tested by students, but have not full automatization
- there are a few buttons which have not real programme code
- programme have very regular falling and I have not any information about reasons of this falling

What is possible changing (steps) for improving programme?
1) Translation of shell from russian to english (we can do a lot of action after this)
2) It is very usefull to found the reasons of programme falling (now programme can fall if you have too much sources (what is “too much” I don't know because this number change each time) and if you take too much quantity of processing days (I think both reason works simultaneously))
3) It is important to do realization functions which have buttons but don't work (for example if we will be have correct search of f_break in power spectra, we will be have correct velocity of solar wind)
4) Additional... (for example automatic removing bad records, removing bad power spectra; full automatization of processing; also some part of good spectra remove, some part of bad spectra have mark “good”, to do output data which can used Bernie programme without additional actions, and so on)

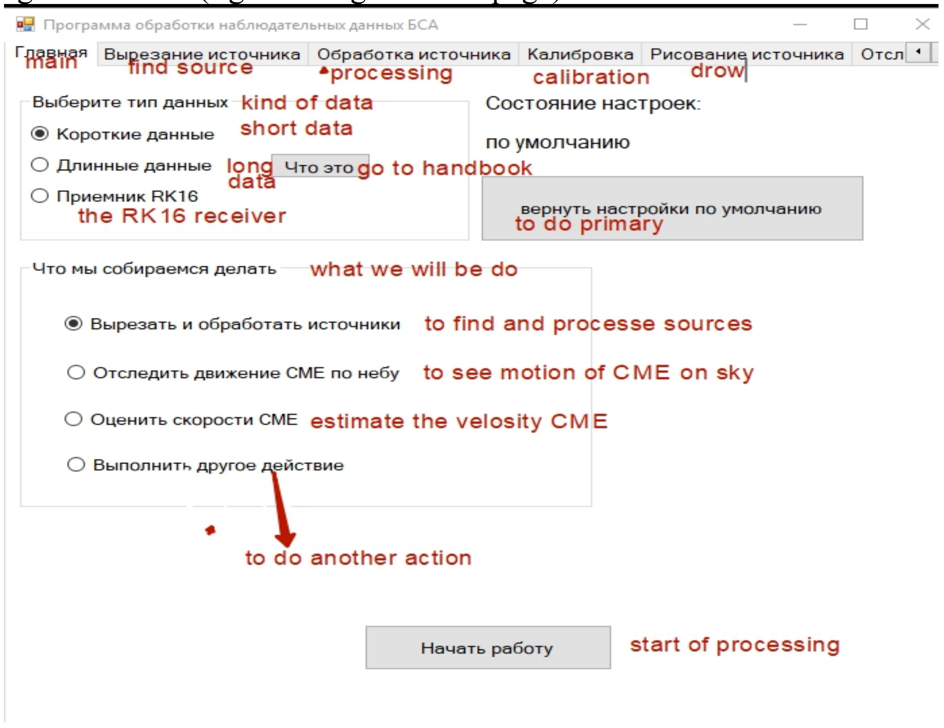
I think it will be very usefull if each step will be have some idea for paper

In table below the possibility processing programme and what is testing

What we do	Tested me	Tested students	No testing
Find source in record (via convolution with largest integral flux, via convolution with scintillation, using exact coordinates) Find sources by hand via mask	Yes Yes No	Yes Yes Yes	- - -
Calculation of power spectra (short data, long data	Yes No	Yes Yes	-
Draw of source, power spectra and others	Yes	Yes	-
Calibration of data via noise generator via calibrate sources	No No	Yes No	- made without testing
Ionosphere shift coordinates (realized???)	No	Yes (partially)	Yes
Other functions: calculation elongation coordinates BSA culmination helio latitude of source helio latitude of aiming point	Yes Yes Yes No No	Yes Yes Yes Yes (partially) Yes (partially)	- - - non correct (small ang.) non correct (small ang.)
To see motion CME on the sky	No	Yes	-
To estimate velocity of CME from shift of scintillating sources (or from X-ray burst)	No	No	Yes
To calculate during processing of data elongation m_classical m_fragment sigma_noise(left or right “0”, top of source) angular size of source turbulence index f_break Velocity of solar wind from power spectra	 Yes Yes Yes Yes No No No No	 Yes Yes Yes Yes Yes (partially) Yes (partially) No Yes (partially)	 - - - - only for strongest sourc only for strong. sources non correct method estimation is work if you have correct f_break and elongation less 60o (mistake 15% for elong. 60o).
Output data: power spectra points, data for preparing power spectra	Yes	Yes	-

Next text for Victor and Hsiu-Shan

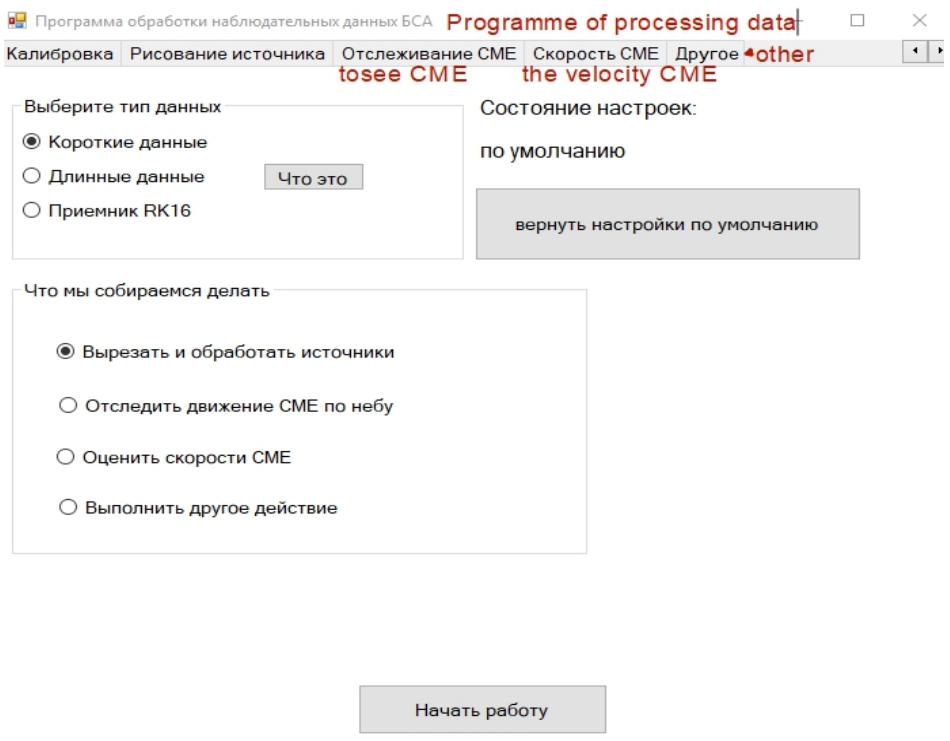
Pictures of programme shell (fig 1 and fig2 is main page)



Выберите тип данных: [✓ Короткие данные] Длинные данные Приемник RK16

Что мы собираемся делать: [✓ Вырезать и обработать источники] Отследить движение CME по небу Оценить скорости CME Выполнить другое действие

fig 1

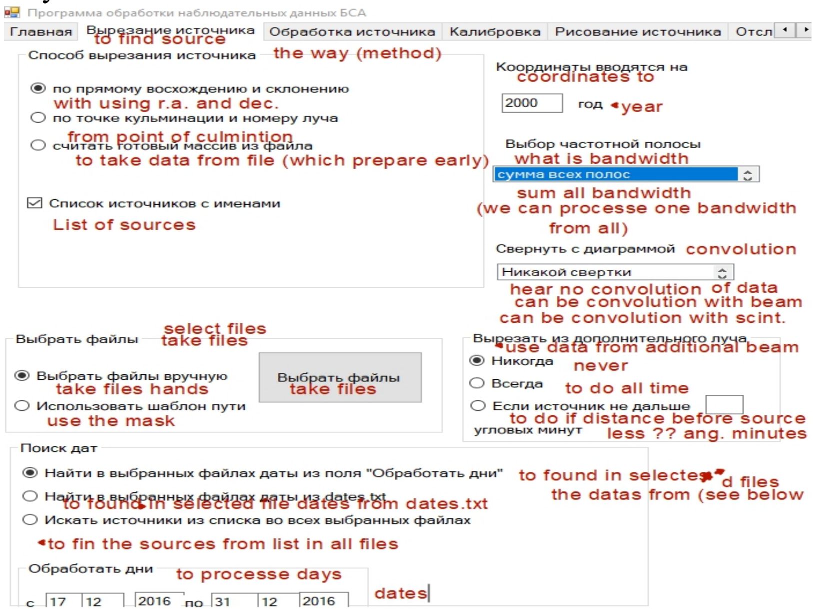


Выберите тип данных: [✓ Короткие данные] Длинные данные Приемник RK16

Что мы собираемся делать: [✓ Вырезать и обработать источники] Отследить движение CME по небу Оценить скорости CME Выполнить другое действие

fig2

Find source in primary record



Программа обработки наблюдательных данных БСА

Главная | Вырезание источника (to find source) | Обработка источника | Калибровка | Рисование источника | Отсл

Способ вырезания источника — the way (method)

● по прямому восхождению и склонению — with using r.a. and dec.
○ по точке кульминации и номеру луча — from point of culmintion
○ считать готовый массив из файла — to take data from file (which prepare early)

☑ Список источников с именами — List of sources

Координаты вводятся на — coordinates to
2000 год — year

Выбор частотной полосы — what is bandwidth
сумма всех полос — sum all bandwidth (we can processe one bandwidth from all)

Свернуть с диаграммой — convolution
Никакой свертки — hear no convolution of data, can be convolution with beam, can be convolution with scint.

Выбрать файлы — select files, take files
● Выбрать файлы вручную — take files hands
○ Использовать шаблон пути — use the mask
Выбрать файлы — take files

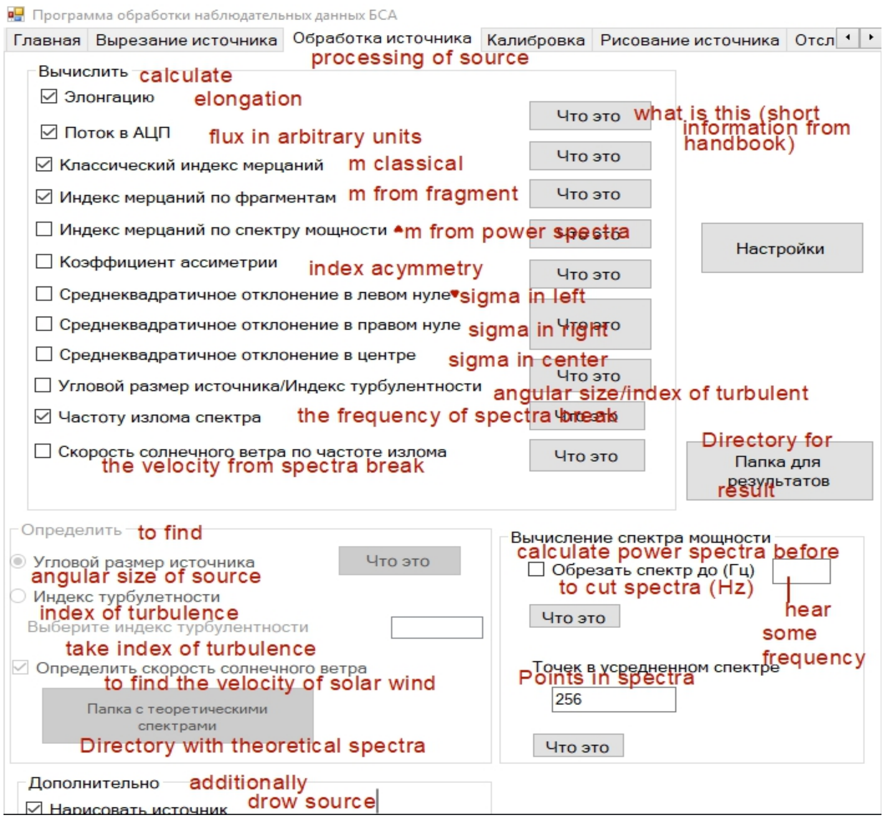
Вырезать из дополнительного луча — use data from additional beam
● Никогда — never
○ Всегда — to do all time
○ Если источник не дальше ___ угловых минут — to do if distance before source less ?? ang. minutes

Поиск дат
● Найти в выбранных файлах даты из поля "Обработать дни" — to found in selected files the datas from (see below
○ Найти в выбранных файлах даты из dates.txt — to found in selected file dates from dates.txt
○ Искать источники из списка во всех выбранных файлах — to fin the sources from list in all files

Обработать дни — to processe days
с 17 12 2016 по 31 12 2016 — dates

fig3

Processing of sources



Программа обработки наблюдательных данных БСА

Главная | Вырезание источника | Обработка источника (processing of source) | Калибровка | Рисование источника | Отсл

Вычислить — calculate
☑ Элонгацию — elongation
☑ Поток в АЦП — flux in arbitrary units
☑ Классический индекс мерцаний — m classical
☑ Индекс мерцаний по фрагментам — m from fragment
☐ Индекс мерцаний по спектру мощности — m from power spectra
☐ Коэффициент ассиметрии — index acymmetry
☐ Среднеквадратичное отклонение в левом нуле — sigma in left
☐ Среднеквадратичное отклонение в правом нуле — sigma in right
☐ Среднеквадратичное отклонение в центре — sigma in center
☐ Угловой размер источника/Индекс турбулентности — angular size/index of turbulent
☑ Частоту излома спектра — the frequency of spectra break
☐ Скорость солнечного ветра по частоте излома — the velocity from spectra break

Что это — what is this (short information from handbook)

Настройки

Папка для результатов — Directory for result

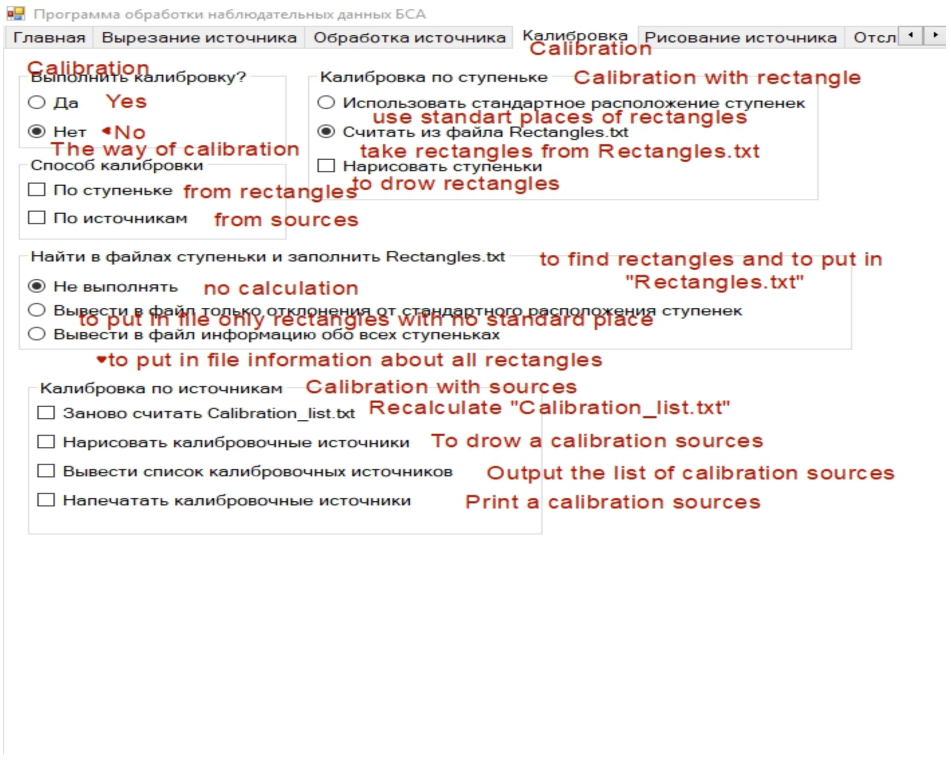
Определить — to find
○ Угловой размер источника — angular size of source
○ Индекс турбулентности — index of turbulence
Выберите индекс турбулентности — take index of turbulence
☑ Определить скорость солнечного ветра — to find the velocity of solar wind
Папка с теоретическими спектрами — Directory with theoretical spectra

Вычисление спектра мощности — calculate power spectra before
☐ Обрезать спектр до (Гц) — to cut spectra (Hz) — hear some frequency
Что это
Точек в усредненном спектре — Points in spectra
256
Что это

Дополнительно — additionally
☑ Нарисовать источник — drow source

fig 4

Calibration of sources


Программа обработки наблюдательных данных БСА

Главная | Вырезание источника | Обработка источника | Калибровка Calibration | Рисование источника | Отсл

Выполнить калибровку? Calibration
- Да Yes
- [✓ Нет] No

Способ калибровки The way of calibration
- ☐ По ступеньке from rectangles
- ☐ По источникам from sources

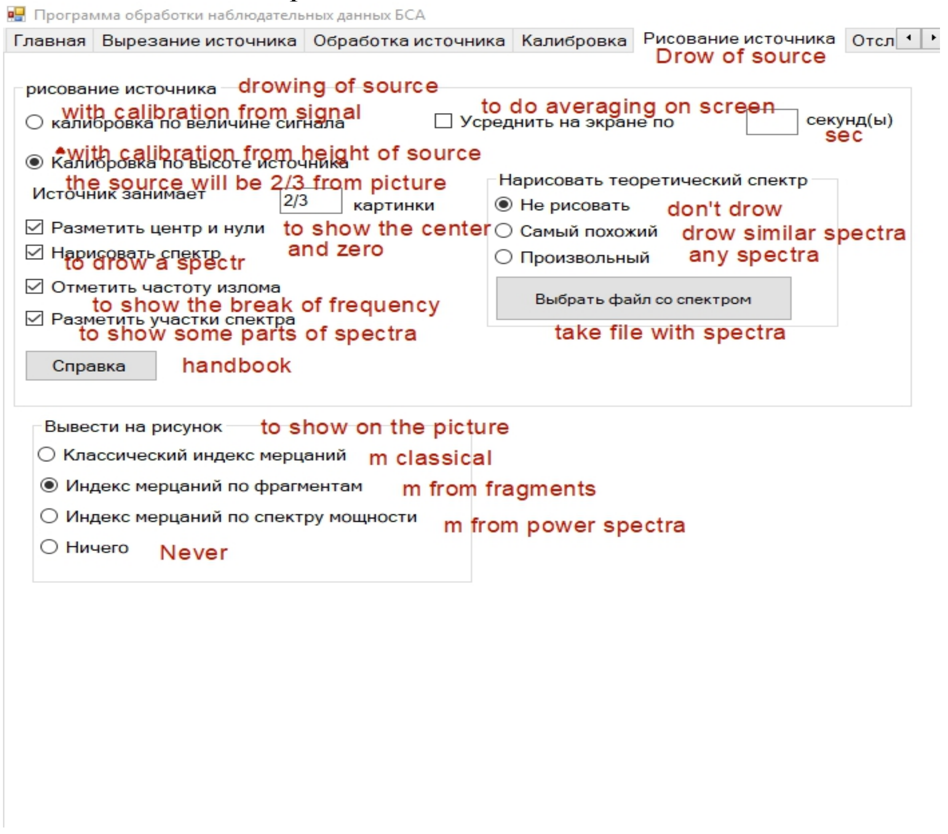
Калибровка по ступеньке Calibration with rectangle
- Использовать стандартное расположение ступенек use standart places of rectangles
- [✓ Считать из файла Rectangles.txt] take rectangles from Rectangles.txt
- ☐ Нарисовать ступеньки to drow rectangles

Найти в файлах ступеньки и заполнить Rectangles.txt to find rectangles and to put in "Rectangles.txt"
- [✓ Не выполнять] no calculation
- Вывести в файл только отклонения от стандартного расположения ступенек to put in file only rectangles with no standard place
- Вывести в файл информацию обо всех ступеньках to put in file information about all rectangles

Калибровка по источникам Calibration with sources
- ☐ Заново считать Calibration_list.txt Recalculate "Calibration_list.txt"
- ☐ Нарисовать калибровочные источники To drow a calibration sources
- ☐ Вывести список калибровочных источников Output the list of calibration sources
- ☐ Напечатать калибровочные источники Print a calibration sources

fig5

Draw pictures and some text on the pictures


Программа обработки наблюдательных данных БСА

Главная | Вырезание источника | Обработка источника | Калибровка | Рисование источника Drow of source | Отсл

рисование источника drowing of source
- калибровка по величине сигнала with calibration from signal
- [✓ Калибровка по высоте источника] with calibration from height of source

Источник занимает 2/3 картинки the source will be 2/3 from picture

- ✓ Разметить центр и нули to show the center and zero
- ✓ Нарисовать спектр to drow a spectr
- ✓ Отметить частоту излома to show the break of frequency
- ✓ Разметить участки спектра to show some parts of spectra

Справка handbook

☐ Усреднить на экране по ___ секунд(ы) to do averaging on screen sec

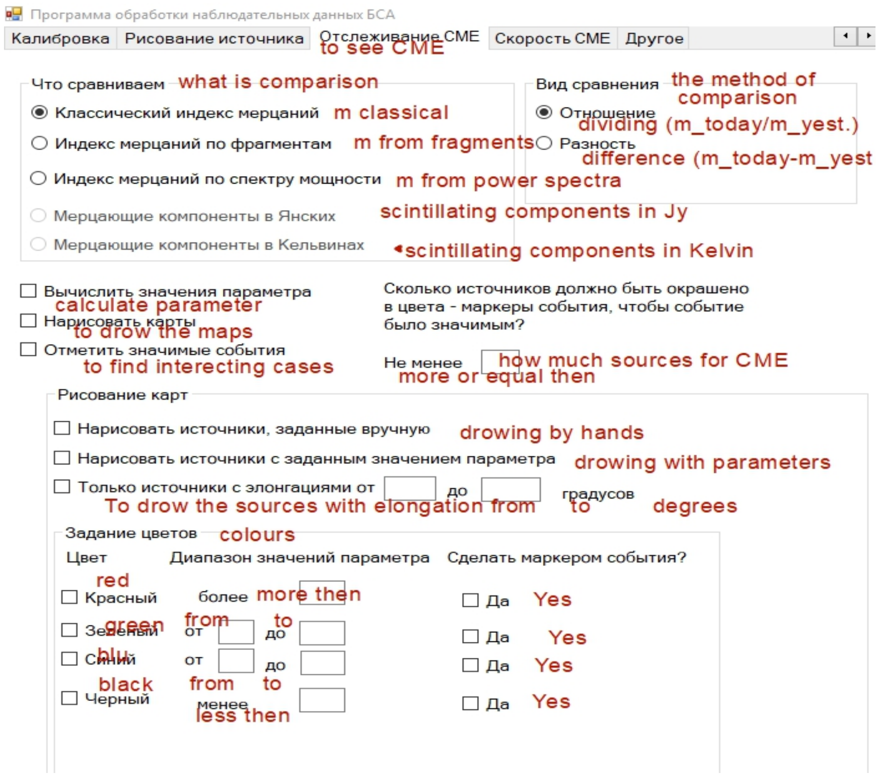
Нарисовать теоретический спектр
- [✓ Не рисовать] don't drow
- Самый похожий drow similar spectra
- Произвольный any spectra

Выбрать файл со спектром take file with spectra

Вывести на рисунок to show on the picture
- Классический индекс мерцаний m classical
- [✓ Индекс мерцаний по фрагментам] m from fragments
- Индекс мерцаний по спектру мощности m from power spectra
- Ничего Never

fig6

to show CME



Программа обработки наблюдательных данных БСА

Калибровка | Рисование источника | Отслеживание CME *to see CME* | Скорость CME | Другое

Что сравниваем *what is comparison*
- ● Классический индекс мерцаний *m classical*
- ○ Индекс мерцаний по фрагментам *m from fragments*
- ○ Индекс мерцаний по спектру мощности *m from power spectra*
- ○ Мерцающие компоненты в Янских *scintillating components in Jy*
- ○ Мерцающие компоненты в Кельвинах *scintillating components in Kelvin*

Вид сравнения *the method of comparison*
- ● Отношение *dividing (m_today/m_yest.)*
- ○ Разность *difference (m_today-m_yest*

- ☐ Вычислить значения параметра *calculate parameter*
- ☐ Нарисовать карты *to drow the maps*
- ☐ Отметить значимые события *to find interecting cases*

Сколько источников должно быть окрашено в цвета - маркеры события, чтобы событие было значимым? *how much sources for CME*

Не менее *more or equal then* ☐

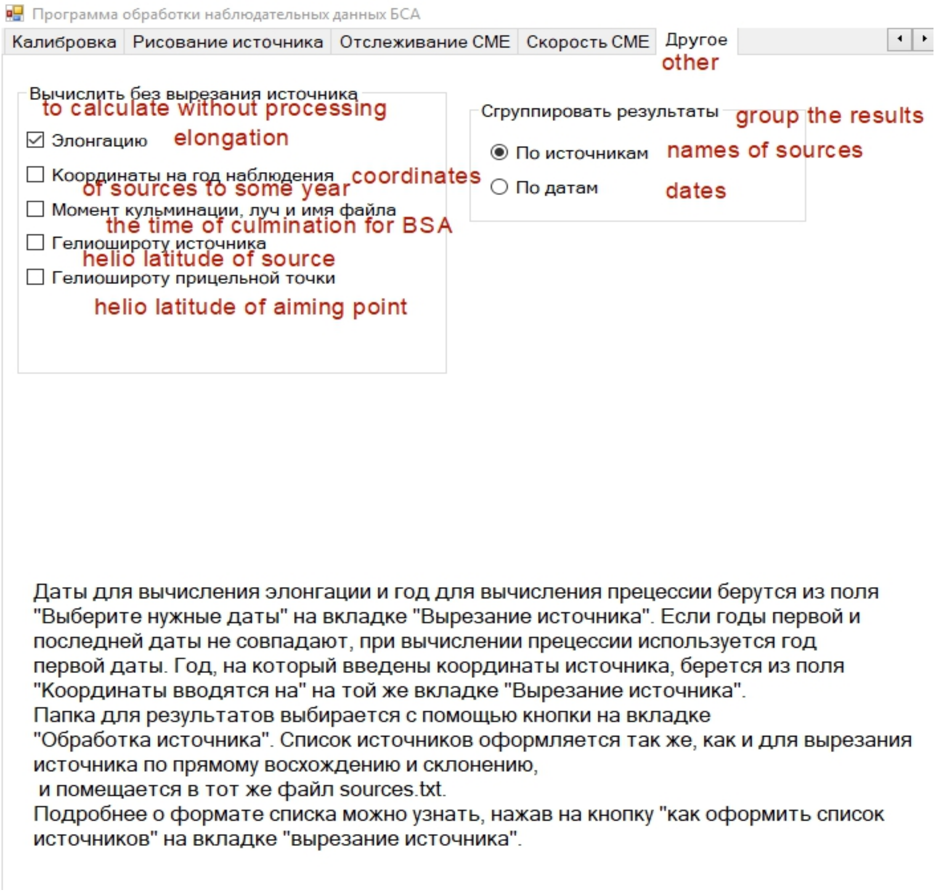
Рисование карт
- ☐ Нарисовать источники, заданные вручную *drowing by hands*
- ☐ Нарисовать источники с заданным значением параметра *drowing with parameters*
- ☐ Только источники с элонгациями от ☐ до ☐ градусов *To drow the sources with elongation from to degrees*

Задание цветов *colours*

Цвет	Диапазон значений параметра	Сделать маркером события?
☐ Красный *red*	более *more then* ☐	☐ Да *Yes*
☐ Зеленый *green*	от *from* ☐ до *to* ☐	☐ Да *Yes*
☐ Синий *blu*	от ☐ до ☐	☐ Да *Yes*
☐ Черный *black*	*from to* менее *less then* ☐	☐ Да *Yes*

fig7

other calculations



Программа обработки наблюдательных данных БСА

Калибровка | Рисование источника | Отслеживание CME | Скорость CME | Другое *other*

Вычислить без вырезания источника *to calculate without processing*
- ☑ Элонгацию *elongation*
- ☐ Координаты на год наблюдения *coordinates of sources to some year*
- ☐ Момент кульминации, луч и имя файла *the time of culmination for BSA*
- ☐ Гелиоширота источника *helio latitude of source*
- ☐ Гелиошироту прицельной точки *helio latitude of aiming point*

Сгруппировать результаты *group the results*
- ● По источникам *names of sources*
- ○ По датам *dates*

Даты для вычисления элонгации и год для вычисления прецессии берутся из поля "Выберите нужные даты" на вкладке "Вырезание источника". Если годы первой и последней даты не совпадают, при вычислении прецессии используется год первой даты. Год, на который введены координаты источника, берется из поля "Координаты вводятся на" на той же вкладке "Вырезание источника".
Папка для результатов выбирается с помощью кнопки на вкладке "Обработка источника". Список источников оформляется так же, как и для вырезания источника по прямому восхождению и склонению,
 и помещается в тот же файл sources.txt.
Подробнее о формате списка можно узнать, нажав на кнопку "как оформить список источников" на вкладке "вырезание источника".

fig8

Functions, classes and other in processing programme (sorry I don't know the standard translation of program terms and therefore some words may be with non correct translation). Below it is translation the letter from my postgraduate student with my small adding.

About program
This program use C#3.0 with Visual Studio 2008, .NET Framework 3.5. There is only one external library alglibnet2.dll. From this library used only one function (Fast furie transform).

About data
Digital receivers BSA can work in two different formats. The first format is the "short data" (time constant 0.0999424 s., 6 frequency channels, the file extension .pnt) and "long data" time constant (0.0124928 s., 32 frequency channels, the file extension .pnthr). At the time, when I finished the program, antenna have two digital receivers. Each receiver had 6 modules with 8 rays in each modulus (it means one receiver written 48 beams of BSA). Each beam is 6 or 32 frequency bands depending on, the "short" data is written or "long". Each band of each beam has its own declination. Thus, when you write "short" data in sight 2 * 6 * 8 * 6 = 576 directions in the sky, and recording "long" – 2 * 6 * 8 * 32 = 3072 directions on the sky. In fact, however, neighboring bands have very similar declination (and sometimes do coincide), and only for the lowest declination the difference in the declinations of adjacent strips substantially.

In the primary record for each beam is also recorded additional "band" (number 7 for short data and number 33 for long). Actually it's not a real band, but simply the pointwise sum of the signal in all bands of a given beam. Often it is convenient to cut the source, since the addition signal of the closest of the bands increases the ratio "signal/noise".

In addition, the program can work with files of the receiver RK16, writing from 2006 to 2013, but there are only 16 beams, low sensitivity, there is no breakdown of the beams on the strip etc.

The type of data to be operated on (long data, short data or RK16) is selected on the main page of shell (switch to the "Select data type").

Classes:
Astrometry
Data_processing

These are static classes, libraries, methods. Note that the **Astrometry** have an array of *Months* that contains the names of the months on English, and *Coordinate_for_print* method that receives the coordinates in seconds and converts it to a string "hh mm ss", suitable for printing to a file.

Classes
BSA_parameters
Structure
BSA3_band

BSA_parameters class contains methods, variables, and constants describing the radio telescope: the declination of the rays, the time constant for short and long data etc. mostly constant, but there are important exceptions. For example, *choosed_data_type* is a variable that stores the user-selected type of data (long data, short data or RK16).

Before you begin working with sources, it should be considered a declination of the rays. The declination is read by the method *Reading_of_beam_declinations_BSA3* (after clicking "Start" this method is called one of the first). Method refers to the file “declination_short_data”, when the short data, and "declination_long_data" if long. These files were prepared in the required format, and to change them is not necessary until you connect an optional recorder (we plan to do additional reconstruction of BSA and add 32 beams to our 96 beams which we have now) .

Method that reads the declination bands, keeps them in two arrays: *bands_sorted_by_order* and *bands_sorted_by_declination*. Each element of both arrays is an sample of the structure **BSA3_band**. In this structure you can see the register number, modulus number, beam, band, and declination of strip.

In the array *bands_sorted_by_order* the bands were arranged in the following ascending order of numbers: the element at index 0 is "receivor 1 module 1 beam 1, the band 1", the element with index 1 is "receivor 1 module 1 beam 1 strip 2", etc. This procedure approximates the descending order of the declinations, but only approximately. For example, 01 band of 5th beam, 3rd module 1st of receivor has a declination 33 40 27, and the band 02 - a greater decline 33 40 48. With the array *bands_sorted_by_order* used *BSA3_band_return* method that returns the band numbers and numbers of the receivor, modulus, beam and band.

bands_sorted_by_declination is an array in which the bands are arranged in order of increasing declinations. With this array have used *Nearest_bands* method that returns a band that is closest to this declination (and other thinks which will be below).

For security purposes, arrays *bands_sorted_by_order* and *bands_sorted_by_declination* are available only from class **BSA_parameters**. Throughout the rest of the programme code, in order to know the declination bands from number of receivor, module, etc., it is necessary to call the method *BSA3_band_return*.

To find what is band closest to this declination, call the *Nearest_bands* that returns a collection of instances of the structure **BSA3_band**. Flexible interface of method provides a bunch of possibilities: output *n* bands closest to some declination, output all bands lying less *m* angular seconds from the declination, etc. See details in comments to the method.

BSA_parameters class has another nested structure: **Declination_diapason_and_bands**. It provides the ability to read instructions on founding the sources from a given set of bands from file "declination_and_bands" (as to make such instructions written in the file itself). The flexibility of these instructions is very high, and therefore, reading them from a file *Declination_and_bands_file_reading* method is very cumbersome and therefore to be better not to read and not rewritten this part of code.

Classes:
Current_day
Current_hour

Digital receivers each hour of observation recorded in a separate file. In addition to observational data, the file contains a header in which are recorded the observational parameters: the exact time of the start of observations, the list of active modules, etc. instance of the class **Current_hour** is a summary of all the information we can get from the passport and the name of the file of the observations, including the data type, the register number, etc. Themselves observational data in the class instance **Current_hour** not keeped because have large size (46 mb per file of short data). For binary data we can take more easy the data directly from a file with using possibilities of C# (starting with an arbitrary byte; no need to first read all preceding data like text header (passport)).

The class contains a method *Signal_clone* copying a signal of a predetermined band from the first to the last specified index. It is called, for example, in the method of the same class *Rectangle_search* which do the search of rectangles (calibrate signals from noise generater) by which calibrated the BSA.

To read files receiver RK16 there is a class **Current_day**, and that his instance, unlike **Current_hour**, stores all the observational data, as a file receiver RK16 is text file.

Class:
Source

The program is designed to work with cosmic radio sources. It is calculated the flux density in arbitrary units, in Kelvin and in Jansky, and for scintillating sources of various parameters of the scintillation (scintillation index, power spectrum, etc.) All these parameters are calculated based on observational data and therefore may vary from day to day. Accordingly, the main information about the source is a piece of record from a file with data in which this source is recorded in a given day. All other parameters is the result of processing this record. Finding of the piece, which was recorded, for example, the source 3C48 in the data for January 1, 2017, and to retain the fragment in memory in the jargon is called "cut source 3C48 for January 1, 2017".

The “cut source” is an instance of class **Source**. Its variables are kept the cut piece of the signal and various computed source parameters (flux density, sigma of the noise in the zeros of the source, etc.). The class **Source** is the heir class **Scintillating_source**. It adds new variables that have meaning only for scintillating sources: the index of the scintillation, etc.

The class constructor cuts the source, and fills in some of the related variables: keeps the original signal in the array, generates a name in the format B1950 etc. All the variables used in the constructor, the program listed in the list entitled: "variables populated in the constructor". All other variables are filled in when calling non-static methods of a class.

Cutting source

As you know, each frequency band of each pattern beam BSA has its own declination. The source is most clearly seen in nearest to declination band. Before you cut the source, you need to decide what bands to use. Of course, better suited to the declination of nearest bands. But in the upper beams of the declinations relates to the BSA bands are so close that it makes sense to fold the signal point of multiple bands. For example, we decided to fold the first and second bands. On the basis of the right ascension of the source, we know the entry containing the source starts at the n-th time, and ends at the k-th (one point is equal to the time constant (integral time; sampling) for your data in about 0.1 s (short data), about 0.01 s (long data) in length). N-th moment of entries in the first band is formed of N-th moment of recording on the second track, $N+1$-th moment of entries in the second band is formed with $N+1$-th entry point of the second band, etc. Instead of two arrays of length K-N is the result. Due to the proximity of the band of the original signal, they can be considered identical, i.e., the signal added is increased by 2 times. The noise, according to the summation of independent random processes will only increase by $\sqrt{2}$ times. Thus, the signal-to-noise ratio will increase by $2 / \sqrt{2} = \sqrt{2}$.

For the upper part of the observed declinations it makes sense to fold all bands of the beam. Fortunately, the receiver does this automatically, recording each ray of "7th band" (for short data) or "the 33rd band" (for long data) is equal to the point value of all the bands of the beam. For the lower declinations, perhaps it is best to put only a pair of adjacent strips or even cut out the only source of the nearest band. It is no testing this part of programme, therefore, the choise of this band taken from shell of programme (this information it was received from the external code). One version of the constructor simply takes a set of ranges (a set of instances **BSA_parameters.BSA3_band**) as a parameter, the other accepts a delegate **BSA_parameters.Beams_from_declination_delegate** that you can pass methods, in accordance with the declination refers to the source, to determine from a set of bands to cut. Currently the program is only one such method is a static method of the class **Beams_from_declination BSA_parameters**.

The primary and additional set of bands

For the upper declinations a declinations of bands of one beam have small differencies from band to band, so for simplicity we can assume that the beam has a uniform decline. Let the source located almost in the middle between the two rays. Then its contribution to both beams will be almost identical. Cutting it of these two rays, we get two independent measurements (signal in both beams is the same, and noises are independent) and, thus, increase the amount of information.

For these reasons, the program provides the ability to cut not only the source from the main set of bands (*general_band_set*), but also of the additional (*add_band_set*). Any method which estimates the source parameters (flux density in arbitrary units, scintillation index, etc.), receives the input variable of the enumeration **General_or_add_band_set**. The value of this variable (or **General_or_add_band_set.general_band_set** or **General_or_add_band_set.add_band_set**) indicates a core set of bands to calculate the or for additional. Depending on this, the result is stored in the corresponding variable source. For example, the flux density of the arbitrary units in the main beam is stored in *flux_ADC_in_general_band_set*, and additional beam in *flux_ADC_in_add_band_set*.

Class constructors
Source.

The class contains several constructors. The most useful of these two:

1) <u>Constructor which the clip source coordinates.</u>
It gets the coordinates of the source in the year indicated on the form (the column "year to which the entered coordinates"), date of observation and the path to the directory where the files. Right ascension and the date of the observation, it determines which file should be cut by the source and which point from file is the culmination of source.

2) <u>The constructor which finding of source if we have the file name and number of a point of culmination.</u>

Both constructors get the primary and secondary sets of bands from the external code (from shell of programme). The standard used for this method **BSA_parameters. Bands_from_declination**, which is the declination of the source parameter specifies which bands which beams to cut it out, based on the value of the variable **Settings.band_choice_regime**.

Calibrate signal from noise generator (“Rectangle”)

From time to time, the antenna on the receiver instead of the signal from the antenna put the signal from the noise generator (rectangle). It is needed for the calibration sources in Kelvin. The rectangle is also considered as an object of class **Source**. This is because during the processing rectangles and sources have to solve similar problems (to cut a piece of the signal to determine the height in arbitrary units, etc). In this regard, the **Source** class has two constructors specially for cutting rectagles.

Class
Scintillating_source

Inherits from the **Source** class, but has additional variables that only have meaning for the scintillating source (the index of the scintillation, etc).

Only in the class **Scintillati ng_source** is a method of drawing source. This is the technical difficulties related to the fact that the method that draws the source and draws its power spectrum which is defined only for scintillating sources. Therefore, even the calibration source must be determined as an object of class **Scintillating_source** if its supposed to draw.

Class
Band_calibration_by_sources

The class is designed to calculate the amount of Jansky in the arbitrary unit for calibration sources in a given band. For more details see below paragraph "what happens during calibration".

Methods of
Form1

Source_processing method receives an instance of the class **Scintillating_source** and perform other actions, prescribed by the form on the shell "processing source" and "Drawing source" (to calculate the scintillation index, elongation, etc.). To do this, it calls the appropriate methods usually in the class **Scintillating_source** (e.g. *Count_classic_scint_index* for computing a classic index of the scintillation).

Class
Servis

The class contains several usefull methods. For example, the method *Reading_of_files* reads a text file line by line and converts it into an array of strings (the overloaded version is in the collection of numbers of type ***double***). Method *Date_for_print* represents a date as a string "dd_mm_yyyy".

What happens when you run the program (main stages)?

1. The first step of programme is read and stored in memory declination bands (from file *declination_short_data*, when the short data from a file *declination_long_data*, if long data). This makes the method *Reading_of_beam_declinations_BSA3* of the class **BSA_parameters**.

2. **If** on the shell "Find source" in the menu "choosing a frequency bandwidth" to select "read from file declination_and_bands" **then** triggered method **BSA_parameters.Declination_and_bands_file_reading** reading information from this file.

3. The user selects the data file of observations. **If** the option "Find in selected files desired date" **then** you can select any files. The program will only work with the correct dates.

4. **If** in the shell we take "Calibration" in the part "Find in files the rectangles and fill **Rectangles.txt**" **then** you selected something other than "Not run", *Rectangle_Search method* is invoked, which searches in the selected file and the selected dates, rectangles, and outputs them to the file Rectangles.txt.

5. **If** we want to do the calibration our observations **then** information about rectangles is read from the file **Rectangles.txt**.

6. **If** we want to do the calibration with using calibration sources, **then**:

6.1. Start method **Band_calibration_by_sources.Correction_reading** that reads from a file "corrections_short_data" (or "corrections_long_data") special corrections that need to multiply the flux calibration sources to divide the flow of the calibrated sources (this thinks connected with fixe direction of beam BSA on the sky and coordinates of sources which not coincide with directions of beams (see theory of phased arrays)).

6.2. If you select "re-calculate **Calibration_list.txt**" runs method **Band_calibration_by_sources.Calibration_list_by_bands_create** that reads from this file the information about the calibration sources, distributes them through the bands and multiplies the flow on the correction coefficients. This information is used to calibrate the sources and also is saved in a file **Calibration_list_by_bands.txt**.

6.3. **If** this option is not selected, **then** the information is simply read from a file **Calibration_list_by_bands.txt** by method **Band_calibration_by_sources.Calibration_list_by_bands_reading.**

Actions which have made for each source:

7. The method is called *BSA_parameters.Bands_from_declination* that from the coordinates of the source and on the basis of the selected menu item "selecting a frequency band" parameter specifies which bands to cut out the source.

8. The selected source is cut and used for processing.

9. *Source_processing* method is called, which executes on the source of the actions, prescribed by the form on the shell "edit source" and "Drawing source" (to calculate the scintillation index, elongation, etc.)

10. **If** we want to do the calibration with using rectangles **then** used the method is called *Calibration_by_rectanle*.

11. **If** we want to do the calibration with using calibration sources **then** used the method is called Calibration_by_sources.