region

August 16, 2005

Abstract

Creates source/background region definition files for extraction of products from event lists.

1 Instruments/Modes

Instrument	Mode
EPIC	Imaging

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

The **region** task takes a FITS-table list of source positions (which must be RA and dec in decimal degrees) as input and creates one or more FITS-format region tables as output. The spatial regions which the task writes to output are intended to be used to filter the original event list for the exposure in question so as to isolate those events corresponding to either (i) a single source in the input source list; (ii) the local background around such a source; or (iii) the 'global' background consisting of all events sufficiently far from all sources in the input list.

The region tables which the task creates conform to the ASC region specification [1].

3.1 Modes

The three possible modes in which the task can work are set via the `operationstyle` parameter. These are described as follows:

- `operationstyle`='single': in this mode, the task constructs just two regions: a source region and a local-background region, both for the same single source. The source region is stored in one file (specified via parameter `regionset`) and the background region in another (specified via parameter `bkgregionset`). The source is selected from the list of sources via the parameter `usersrcid`. This specifies the row number (beginning at 1) of the source as it occurs in the source list *after* the list is filtered via the selection expression provided in parameter `expression`. Because it can be difficult to identify a priori which row number a source will have in the filtered source list, it is not recommended to use the source-table filtering capability when **region** is invoked in this mode.

- `operationstyle`='batch': in this mode, a source region/local background region pair is constructed for each source in the (filtered) source list. The source regions are stored in one dataset (specified via parameter `regionset`) and the background regions in another (specified via parameter `bkgregionset`). Each region is stored in a different table in its respective dataset.
- `operationstyle`='global': in this mode, a single region is constructed which is intended for filtering all but background events from the event list. This global background region is the logical negation of the logical OR of all source regions. This single region is stored in the dataset specified via parameter `bkgregionset`.

NOTE! In NO case does **region** construct its output regions so as to exclude events on bad pixels or outside the instrument field of view. Such events must be independently rejected if this is desired.

3.2 Spatial coordinate system of the region(s)

The regions can be specified via the `outunit` parameter in either of two projection-plane coordinate systems, namely 'detxy' and 'xy'. (This parameter is not case-sensitive.) The first choice corresponds to columns `DETX` and `DETY` of the standard XMM EPIC event list, the second to columns `X` and `Y`. The DETXY coordinate system is fixed with respect to the EPIC instrument and is equivalent to the CAMCOORD2 system described in the **cal** task documentation, except that the units are 0.05 arcseconds instead of mm. Hence the DETXY projection plane is normal to the optic axis of the telescope[1]. The XY system is fixed with respect to the sky (apart from any errors in the attitude correction); its units (= the size of its 'pixels') happen also at present to be 0.05 arcseconds, but could in principle take any value since **region** makes no assumptions about them.

Note that in either case the task needs to know information about the EPIC instrument and the epoch of the observation. This information is obtained from keywords in the header of the supplied event list.

If the user specifies `outunit`='xy' then the task also needs WCS keywords to define the connection between the source RAs and decs and their positions in the XY coordinate system. These keywords are obtained from the `X` and `Y` columns of the event list.

If `outunit` is given as 'detxy' and if the supplied source list is a standard XMM EPIC product then in principle the event list is unnecessary. However it has been decided for simplicity to make it mandatory to supply both the event list and the source list to **region**.

Note that if the event list to be filtered by the regions is different from that used to supply the WCS and other keywords, it is important that at least the `INSTRUME` keyword in the headers and (if `outunit`='xy') the WCS keywords in the `X` and `Y` columns of the two event lists are the same.

[1]Although it is not necessarily true that the telescope optic axis coincides with the origin of DETXY!

3.3 Source selection

At times it may be desirable to filter the source list before calculating regions. An example of such a case is if the list contains many rows for the same source position (as do eg the standard XMM source-list products). There is usually not much point in constructing more than one region for the same source position, so in this case it would be desirable to filter out all the excess rows before constructing regions. This can be achieved via the parameter `expression`. This takes any selection expression which obeys the **selectlib** grammatical rules and which involves only those columns present in the source list.

3.4 Specifying source region shapes and sizes

The basic source region shape is a circle. There are three ways to assign the radius, specified via the parameter `radiusstyle`:

- `radiusstyle`='userfixed'. The user should supply the radius via the parameter `fixedradius`, at present in units of arcseconds.

  If `outunit`='xy', the smaller of the XY-system pixel dimensions in arcseconds provides the conversion factor to convert this radius to sky-image pixel coordinates.

- `radiusstyle`='enfrac'. In this choice, the task uses for the source region radius that value which encloses `energyfraction` of the total flux under the Point Spread Function (PSF).

  Note that in this case, regardless of the value of `outunit`, the initial circle must be drawn in the DETXY system, since the PSF is returned in this system. However if `outunit`='xy' this involves a reprojection onto another image plane. In theory, locii which appear circular in the DETXY projection plane appear in general as ellipses in another projection plane (and pixellization scheme). At present the effect of this reprojection on the source region contours is ignored and circles are retained. The scalar conversion between circle radii in the two coordinate systems is performed by taking the smaller of the XY-system pixel dimensions in DETXY coordinates as the conversion factor. In most cases the effect of this approximation is expected to be insignificant. This issue may be dealt with more rigorously in a later version of the task.

- `radiusstyle`='contour'. In this choice, the task calculates an elliptical locus which approximates the contour of the PSF at a given flux value. This procedure is described more fully in the following subsection (3.4.1). Note that in this choice the distortion caused by transform from the instrument-centric PSF frame to a sky-centric XY frame *is* taken into account.

3.4.1 The contour method of calculating source region shape

When `radiusstyle`='contour', the task calculates the source region shape as an elliptical locus which approximates the contour of the PSF at a given flux value. The procedure uses an analytical model of the PSF which has the following form:

$$PSF(x,y) = A\,exp\Big(-\frac{x^2+[y/s]^2}{2\sigma^2}\Big) + \frac{(1-A)}{(x^2+[y/s]^2)^{\alpha}}$$

where A, σ, s and α depend on the source position in the FOV. The (x,y) coordinate system employed here is transformed into DETXY by addition of $(r_0 = \sqrt{x_0^2+y_0^2}, 0)$ followed by rotation through an angle $arctan(y_0/x_0)$ where (x_0, y_0) is the centre of the source in DETXY.

Flux contours of the model given above have elliptical shapes. Clearly, in order to calculate the shape of such a flux contour, it is necessary firstly to normalize the PSF to the source brightness and secondly to determine the flux level (in 'brightness' units per pixel) at which to find the contour. Note that the units must be compatible: if the source brightness is specified as total counts, for example, then the flux level must be in counts per pixel.

Ideally the source brightness would be specified by allowing the user to supply (via a command-line parameter) the name of the column of the source list which contains such information. However this facility is not yet in place: at present the task is 'hard-wired' to look for source brightness in a column of the source list named `SCTS`. If there is no column of this name in the source list then the `radiusstyle`='contour' style cannot be used.

Matters are more flexible as far as specifying the flux level is concerned. The flux level is given in terms of a local background flux level and a multiplying constant. There are several ways in which the user can specify the first of these, the selection being governed by parameter `bkgratestyle`. The multiplying constant is specified via parameter `bkgfraction`.

`bkgratestyle` will eventually provide four choices ('col', 'kwdset', 'image' and 'user') but at present this functionality is only partially realized and the only choices the task will accept are 'col' or 'user'. These are described as follows:

- `bkgratestyle`='col'. In this choice, the task looks for a column in the source list named `BG_MAP`. In future the user will be able to specify a different name via the parameter `bkgratecol`, but at present this parameter is not read by the task.
- `bkgratestyle`='user'. In tbis choice, the user must supply the background flux (at present the units must be counts per pixel so as to match the source brightness which is tied to column `SCTS`) via the parameter `bkgrateuser`.

Notes:

- The background flux × `bkgfraction` must evaluate to greater than zero. If not, a warning will be issued and no regions constructed for that source.
- It is of course possible for the specified flux level to exceed the maximum value of the normalized PSF (for instance if the source is faint and background bright). In this case the region is of zero size.
- If it is desired to set the region size in this way, but retain the circular shape, then the user should select `nosrcellipse`='yes'. This gives circular source regions with radii equal to the long semi-axes of the respective ellipses.
- Advantages of the choice `radiusstyle`='contour' are:
  1. The size of the source region follows the brightness of the source - ie, brighter sources have larger extraction regions.
  2. The shape of the source region follows (approximately) the distortion of the off-axis PSF. NOTE however that this mapping of the PSF shape distortion IS at present only approximate. In future it is hoped to be able to determine it in a more systematic way.
- This choice of `radiusstyle` cannot as yet be used unless `outunit`='xy'.

3.5 Source non-overlap algorithm

Sources which lie close to each other may generate event-extraction regions which overlap. It is obviously undesirable to include in a subset of events associated with one source events which are also associated with another. **region** provides a mechanism for avoiding this overlap. If the user selects `shrinkconfused`='yes', and provided that the source region shape is circular not elliptical (true if `radiusstyle` is not equal to 'contour', or if `radiusstyle`='contour' but `nosrcellipse`='yes'), the task will reduce the radii of pairs of overlapping source regions until they no longer overlap. Each source of the pair suffers the same fractional reduction.

Elliptical regions are not at present shrunk because the complexity of the geometry required does not seem to be compensated by the advantage gained.

Note that, in general, a source has a non-zero chance of being contaminated by events from every other source in the field of view (FOV). It is only possible to correct for this cross-contamination on a statistical basis, and even this is difficult to do without introducing systematic errors greater than the contamination one is attempting to remove. The shrinkage of extraction regions performed by **region** must be considered only a very coarse first step towards such correction.

Note that source-region cutouts from background regions are not shrunk.

3.6 Local-background region shapes and sizes

Local-background regions are in all cases annuli. The fundamental choice here is between annuli concentric about the optic axis of the particular EPIC telescope or annuli concentric about the source location itself. This choice is made via the parameter `fovbkgannulus` and is described as follows.

- `fovbkgannulus`='yes'. The local-background annulus for each source is made concentric about the telescope optic axis (OA). The central radius of the annulus passes through the source location, and the thickness of the annulus is equal to the maximum diameter of the (unshrunk) source region. Background regions of this shape should be chosen when there is no extended emission in the FOV, because OA-concentric regions are in general larger than source-concentric ones (thus offer better statisitics), yet sample everywhere approximately the same background level, because the value of the vignetting function is approximately the same at a given radius from the OA.
- `fovbkgannulus`='no'. The local-background annulus for each source is made concentric about the source position itself. The inner radius of the annulus is equal to the maximum semiaxis of the (unshrunk) source region, and the outer radius is set to three times that value. Background regions of this shape should be chosen if there is extended emission in the FOV.

So far it has been assumed that the annuli are circular. However, if the region units are selected to be 'xy', then locii of equal radius (in DETXY coordinates) from the OA will in general appear as ellipses in the XY projection and pixellization scheme. If `fovbkgannulus`='yes' in this case then the annuli must follow this distortion in order to preserve the approximate uniformity of background around the annulus. The task therefore distorts the circular DETXY annuli into appropriate ellipses if `outunit`='xy' (not the default), `fovbkgannulus`='yes' (not the default) and `nobkgellipse`='no' (default), but not otherwise.

All source-extraction regions are excised from any given local-background region: thus in most general terms a local-background extraction region looks like an elliptical annulus with holes at source locations.

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints

eventset	yes	dataset	dummy_default	

Reference event file used to define the coordinate system for the source and background regions. In general this should be the event list which is to be filtered by the regions.

srclisttab	yes	table	dummy_default	

The name (in <dataset name>:<table name> format) of the FITS table containing the sources.

expression	no	string		

A selection expression used to filter the source list. The selection expression should adhere to the **selectlib** grammar and should contain as arguments only keywords and column names actually present in the source list table.

operationstyle	no	string	single	single—batch—global

See section 3.1 for a detailed description.

usersrcid	yes	integer	1	1 ≤usersrcid≤ 9999

If `operationstyle`='single', the task looks in this parameter for the row number of the source list (*after* the list is filtered via `expression`) for which to construct source and local-background event-extraction regions.

srcidcol	yes	string	dummy_default	

If `operationstyle`='batch', the task looks in this parameter for the name of the column in which to look for a (unique) source id integer. Source and local-background event-extraction regions are constructed for each source in the (`expression`-filtered) list and stored in separate binary table extensions in two datasets, one dataset for the source regions, the other for the local-background regions. The names of the extensions are constructed using the unique id integers (see section 7).

shrinkconfused	no	bool	yes	

Whether to shrink extraction regions until they don't overlap. See section 3.5.

radiusstyle	no	string	userfixed	userfixed—enfrac—contour

This governs the way in which the size and shape of the source extraction region is calculated. See section 3.4 for a detailed description.

fixedradius	no	real	12.0	0.0 ≤fixedradius

If radiusstyle=‘userfixed’, the task looks in this parameter for the radius (in arcseconds) to be used for all source extraction regions.

energyfraction	no	real	0.9	0 ≤fixedradius< 1

If radiusstyle=‘enfrac’, the task looks in this parameter for an energy fraction between 0 and 1. This is used to calculate a radius for the source extraction regions via the **cal** library call CAL_psfEncircledEnergy.

Note that the actual PSF energy fraction contained within the region circle may (because the **cal** returns the PSF in pixellized form) differ slightly from the value specified via the present parameter.

bkgratestyle	no	string	col	user—col—kwdset—image

If radiusstyle=‘contour’, the source extraction regions are calculated by a ‘contour’ method, described in section 3.4.1. The task first obtains an analytical model of the PSF, for which contours of equal brightness are elliptical in shape. Thus, given the normalization of the PSF (ie, source brightness) and a brightness contour level equal to the local background times a multiplying constant (provided via the parameter bkgfraction), an elliptical shape can be calculated for the source extraction region. In order to calculate such a contour there must be some way to specify the local background level. bkgratestyle provides four alternatives (however at present only two of these are implemented), each of which requires a different additional parameter. These alternatives are described under the appropriate parameter name.

bkgratecol	no	string	BG_MAP	

If radiusstyle=‘contour’ and bkgratestyle=‘col’, in theory the column name containing the background flux values is read from the present parameter. However this is not yet implemented - ‘BG_MAP’ is hard-wired into the code.

bkgrateuser	no	real	1.0	0.0 <bkgrateuser

If radiusstyle=‘contour’ and bkgratestyle=‘user’, the background flux may be entered here. Note that it should be in a unit compatible with the source brightness. Since source brightness is at present ‘hard-wired’ to come from SCTS column of the source list, this ‘compatible unit’ must at present necessarily be counts/pixel.

bkgfraction	no	real	1.0	0.0 <bkgfraction

As described in section 3.4.1 and also in the bkgratestyle parameter description, the contour method of obtaining a source region shape requires a flux level to be specified via two values: a value of the local background flux ϕ (which can be specified in a number of ways, switched via parameter bkgratestyle), and a number by which this value is to be multiplied (given via the present parameter). Thus the source region is returned as that shape for which the PSF function equals $\phi\times$ bkgfraction.

nosrcellipse	no	bool	no	

If radiusstyle=‘contour’, the shapes of the source extraction regions are in general elliptical. If this is not desired, they may be converted to circles of radii equal to the greater semiaxis of the respective ellipses.

fovbkgannulus	no	bool	no	

Local-background annuli may be concentric about either the optic axis or the source location (see section 3.6). The present parameter provides the switch for this behaviour.

outunit	no	string	xy	detxy—xy

Specifies the spatial coordinate system for the output event-extraction regions. DETXY is equivalent to the CAMCOORD2 system described in the **cal** task documentation, except that the units are 0.05 arcseconds instead of mm. The standard XMM EPIC event lists contain event coordinates in `DETX` and `DETY` columns. The XY system refers to `X` and `Y` column values of the standard event lists. These pairs of values give the location of the event on a tangential projection plane and are calculated (using task **attcalc**) shortly after the event list is created in order to save time when the product images are extracted. (Images in several energy bands are generally extracted but they all share the same projection plane.) The units of the XY system are 0.05 arcseconds, same as DETXY.

Note that this parameter is not case-sensitive.

nobkgellipse	no	bool	no	

The DETXY and XY coordinate systems are not coplanar. Thus circles in DETXY become transformed to ellipses in XY. As described in section 3.6, it may be desirable to construct local-background annuli as circular annuli in the DETXY system, concentric about the optic axis (OA), because contours of the vignetting function follow (broadly speaking) a similar form. If `outunit`='xy', such contours become distorted to ellipses, so normally if `fovbkgannulus`='yes' the background annuli are distorted to ellipses in the same way. However in the (possibly unlikely) event that this distortion may not be desired, the present parameter has been provided. If `nobkgellipse`='yes', local-background annuli will not be transformed to ellipses in XY but will remain circular. The radius conversion factor is taken from the smaller of the XY-system pixel dimensions expressed in DETXY coordinates.

regionset	no	dataset	src_region.ds	

The dataset in which source extraction regions are stored.

bkgregionset	no	dataset	bkg_region.ds	

The dataset in which background (either local or global) extraction regions are stored.

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

badOutUnit *(error)*
The value of `outunit` was not recognized.

badInstrum *(error)*
The value of the `INSTRUME` keyword of the event list was not one of the XMM EPIC instruments.

noSourcesSelected *(error)*
None of the sources in the input event list passed the filter specified by `expression`.

tooFewRows *(error)*
When `operationstyle`='single' a single source is specified via the `usersrcid` parameter. This refers to the row number of the filtered source list (note the first row is assumed to be numbered 1 not 0). The present error occurs if the number of rows is less than `usersrcid`.

nonUniqueSrcId *(error)*
When `operationstyle`='batch', it is necessary to assign a unique identification integer to each source, because the extension names in which the regions are stored contain this integer. The integer is taken from a column specified via the `srcidcol` parameter. The present error occurs if any value of the integer specified in this column occurs more than once (in the filtered source list).

badIdBand *(error)*
At present it is necessary that the source list must contain an `ID_BAND` column. This is to allow the energy limits of the source detection band to be determined. `ID_BAND` numbers must be between 0 and 9 inclusive, otherwise the task does not know how to decode them.

badEnergyBandEdges *(error)*
The 'lower' energy was found to be higher than the 'higher' energy for this band! This is probably a bug.

notSupported *(error)*
Flags a mode which is not yet supported.

badBkgFluxStyle *(error)*
The value of `bkgratestyle` was not recognized.

badRadiusStyle *(error)*
The value of `radiusstyle` was not recognized.

badOperationsStyle *(error)*
The value of `operationstyle` was not recognized.

energyThetaOrPhiOutOfRange *(error)*
This occurs as the task attempts to access the PSF stored in the XMM calibration files via a **cal** library call. This call returns the PSF only for a limited range of source energy and position. The position is specified in the polar coordinates (theta,phi) of the TELCOORD system (see the **cal** documentation for details). Actually only one out-of-range condition is accepted: if theta (the off-axis angle) exceeds its **cal** limit, a warning rather than an error is generated and the limiting value is substituted. This difference is because the theta range does not quite cover the actual instrument extent, thus the theta-over-range condition occurs rather often in practice; whereas the other out-of-range conditions should not occur at all.

nonRealSrcRadii *(error)*
Indicates a bug in the confused-source-radius shrinking routine. Report this to the code developer.

doubledSource *(error)*

This occurs when the task is attempting to eliminate overlap between source-regions of confused sources. However, many source lists contain more than one entry for the same position (say, to record the source flux at different energies). This is clearly not the same thing as a confused source and it is not desirable that the regions for all such duplicated source positions be shrunk to zero in an attempt to remove such spurious 'confusion overlap'. The task attempts to detect such duplicated source positions and fails with the present error if it finds any. the solution is to filter these duplicated sources from the list and then resubmit the list to **region**.

tooManyRows *(warning)*

When `operationstyle`='batch', regions are stored in FITS binary table extensions which have names containing an identifying integer. The maximum value of this (from length considerations of the extension name string) is presently 9999. If there are more than 9999 source in the (filtered) input list, the present warning will be generated.
corrective action: The task only stores regions for the first 9999 sources.

nonUniqueIdBand *(warning)*

More than one value of ID_BAND was detected in the source list. This implies that the list contains more than one row for one or more of the source positions. In this case several concentric regions will be produced for these sources. If you don't want this to happen you should filter the source list (the task can do this for you via its `expression` parameter) so that only one value of ID_BAND remains.
corrective action: The task takes no corrective action.

noBandEnergy *(warning)*

The task couldn't work out the source energy. This may be a bug and should be reported to the code developer.
corrective action: The task uses a default energy value.

srcThetaTooLarge *(warning)*

This occurs as the task attempts to access the PSF stored in the XMM calibration files via a **cal** library call. This call returns the PSF only for a limited range of source energy and position. The position is specified in the polar coordinates (theta,phi) of the TELCOORD system (see the **cal** documentation for details). However, the theta range does not quite cover the actual instrument extent, thus the theta-over-range condition is probably going to happen sooner or later. There's no cause for alarm.
corrective action: The task uses the maximum value of theta instead.

6 Input Files

1. A source list, which at minimum must contain a table extension named `SRCLIST` with the following columns:

   - (mandatory) `RA`: right ascension of the source in decimal degrees.
   - (mandatory) `DEC`: declination of the source in decimal degrees.
   - (mandatory) `SCTS`: total source counts.
   - (mandatory) `BG_MAP`: total background counts per pixel.
   - (mandatory) `ID_INST`: integer id of the EPIC instrument.
   - (mandatory) `ID_BAND`: integer id of the energy band.
   - (optional) a further column, specified by `srcidcol`, which should contain a unique source identifying integer. Note that **region** only attempts to access this column if `operationstyle`='batch'.

If `radiusstyle`='contour', the source list should also contain WCS keywords so as to enable one to convert the flux-per-pixel background values to flux-per-arcsec. XMM EPIC source lists currently store these in a slightly non-standard way, as keywords in the table header, instead of as image-position column keywords. Since the major grist to **region**'s mill is EPIC source lists, the task has been 'hard-wired' to look for these header keywords. These keywords are described as follows:

- `TCTYPEn`: ='RA—TAN' for n=1 and 'DEC–TAN' for n=2.
- `TCRPIXn`: the position in pixel units of the reference point.
- `TCRVALn`: the position in world (ie, RA or dec) units of the reference direction.
- `TCDELTn`: the size of the pixels in world (ie, RA or dec) units. Note that for RA/dec as world units, the n=1 pixel size is usually made negative, so that RA increases to the left.
- `TCROTA2`: the rotation angle of the world frame. Usually this rotation is zero and in this case the keyword can be omitted.

'n' in the above should run from 1 to 2.

I realize that this is a very narrow and restrictive source-list format definition. In future it is hoped to be able to make **region** accept much more generic lists of sources.

2. An XMM EPIC event list against which the region filter will be applied.

7 Output Files

The task in general writes two files, one specified by parameter `regionset` for the source region(s) and the other, specified by parameter `bkgregionset`, for the background region(s). However the exact details depend on the value of the `operationstyle` parameter as follows:

- `operationstyle`='single': task writes both source and background files; each contains a single extension called `REGION` in both cases.
- `operationstyle`='batch': task writes both source and background files; each file contains one extension per source, these being named `REG_nnnn`, where the nnnn are consecutive integers starting at 1.
- `operationstyle`='global': task writes background file only; extension name is `REGION`; this contains the global background region, which is the entire projection plane minus source cutouts.

8 Algorithm

```
module region_mod

  subroutine region

    Get parameters;

    Read instrument, exposure time etc from event list and set up cal;
```

```
if (--outunit eq 'xy') {read WCS kwds from X and Y columns;}

Filter source list and load the result into memory;

if (--radiusstyle ne 'userfixed') {calculate source mean energies;}

Calculate the position of each source in both DETXY and TELCOORD;

if (--outunit eq 'xy') {
  Calculate the position of each source in the XY scheme;
}

# Calculate the shapes and sizes of the source regions:

if (--radiusstyle eq 'contour') {
  if (--bkgratestyle eq 'col') {
    get bkg_rates from BG_MAP column of source list;

  } elsif(--bkgratestyle eq 'user') {
    set all bkg_rates to --bkgrateuser;
  }

  Calculate elliptical cutouts (uses ssclib/source_cutouts routine);

  if (--nosrcellipse) {change ellipses to circles of r = 1;}

} elsif(--radiusstyle eq 'userfixed') {

  Calculate region radius by changing --fixedradius to the appropriate
    coordinate system (either detxy or xy);

} elsif(--radiusstyle eq 'enfrac') {

  Calculate (using cal call) the radius of a circle which encloses
    --energyfraction of the flux of the PSF;

  Convert this radius to the appropriate coordinate system (either
    detxy or xy);
}

if ((--radiusstyle ne 'contour' || --nosrcellipse)
  && --operationstyle ne 'global' && --shrinkconfused) {

  Find sets of confused sources (uses a routine in ssclib/confusion);

  Shrink radii of confused regions until they don't overlap;
}

if (--operationstyle ne 'global') {
# Construct the background annuli:

#### fill in some details??
}
```

```
    Construct the region specifications and save them to file.

  end subroutine region

end module region_mod
```

9 Comments

10 Future developments

- The task will in future be modified to accept much more generic source lists.

References

[1] J. McDowell and A. Rots. FITS REGION Binary Table Design. Technical Report ASC-FITS-REGION-1.0, Chandra Science Center, March 1998. Found at the URL: `http://hea-www.harvard.edu/~jcm/asc/docs/asc/region.ps`.