PIPELINE - PHOTOMETRY REDUCTION
Date 10/03/2009

################ FIELD CALIBRATION

This part should be run every times the field is not calibrated.

1) CREATE THE LIST FILE and TEMPLATE IMAGE OF THE FIELD

a) Chose one good image of your field that you want to use as template
(an image large enough with enough stars)

b) Rename this image: "cp image.fits -> name_field_templ.fits"
(e.g.name_filed = sn07gr or sn07rt or sn08D)
c) **qubaref**:
> qubaref name_field_templ.fits

(create the list file of the sequence of local stars)
output => file_coordinate.list
(note: at this stage the file will only include coordinate, while the magnitudes column are filled with: 'V BV UB VR RI '
eg. 13.239 0.234 1 V BV UB VR RI
13.211 0.232 2 V BV UB VR RI)
the magnitude will be replaced once the magnitudes of local stars are known (see point **4c**))

d) Rename the output file 'file_coordinate.list':
"mv file_coordinate.list name_field.list"

e) Copy "name_field.list" and "name_field_templ.fits" in
/home/sne/QUBA/site-packages/pipeline/coordinate_std/optical/
to upload the files use the web-page available:
http://star.pst.qub.ac.uk/sne/cgi-bin/template_upload.py
or better contact Stefano Valenti.

2) SELECTION OF GOOD NIGHTS FOR CALIBRATE THE LOCAL STAR'S SEQUENCE
Select a set of nights you think are photometric
among the night with standard fields observed.

For each night make 2 lists:

sn_U.fits	std_U.fits
sn_B.fits	std_B.fits
sn_V.fits	std_V.fits
sn_R.fits	std_R.fits
sn_I.fits => listasn	std_I.fits => listastd

3) COMPUTE FOR EACH NIGHT THE ZERO POINT AND COLOR TERM

a) **qubastd**:
> qubastd @listastd -l namefieldstandard

output program => lp_ru149_20080403
(lp or NOT or ekar or TNG or NTT or ….)
(note: namefieldstandard = rui149 or rui152 ....)

Warning: write "qubastd" without option to know which standard fields are available. If your field is not in the list.
Send a good image of the field and a table with the magnitudes to Stefano and ask him to add that standard field to the list.
The table with magnitude should include these magnitudes:
(NAMESTANDARD V BV UB VR RI) optical
or (NAMESTANDARD g ug gr ri iz) sloan
or (NAMESTANDARD J JH HK) infrared

b) **qubaph**: (input is the output of the previous program)
> qubaph lp_ru149_20080403 -r @listasn

Compute the zero points and color terms
and uploaded header of SN's images

c) **qubapsf**:
> qubapsf @listasn -l name_field –i

output files => .psf.fits and .ec
(name_field is the same of step 1b and step 1d)
This program compute the psf of the images

d) **resnoopy**:
> resnoopy @listasn"
output file = name_files_night_doc
(name_files_night_ is the root of the image's name for each night, e.g. 'sn_' in step 2)

compute the apparent magnitudes of sequence stars

REPEAT STEP 3 FOR EACH NIGHT OF THE SUBSET

4) **CALIBRATION OF THE FIELD**

a) Step 3 on each night will produce a set of 'doc' files
Collect all these files in a list:

name_files_night_1_doc

name_files_night_2_doc
name_files_night_3_doc
name_files_night_4_doc
name_files_night_5_doc
name_files_night_6_doc > listadoc

b) **qubaexa**:
> qubaexa @listadoc

output -> Magnitude of the sequence stars
(eg. 1 17.4 0.3 0.23 0.3 0.5
2 V BV UB VR RI
3 ……………………….)

(if you follow the suggestions..for all the bands
each sequence star will be calibrated from
the brightest to the faintest)

c) copy the sequence star's magnitudes in the 'name_field.list'
and update the file 'name_field.list'
(sequence stars magnitude computed with qubaexa)
to upload the file use the web-page available:
"http://star.pst.qub.ac.uk/sne/cgi-bin/template_upload.py"

#################### **SN MEASUREMENTS** ####################

5) This step include:
-zero point and color term (all the nights)
-compute the PSF (only for the nights not in the previous set)
-measure the sn instrumental magnitude (all the nights)
-compute the sn apparent magnitude (all the nights)

a) zero point and color term (all the nights)

- **qubastd**: (again, but this time using the sn_field)
> qubastd @listasn -l name_field

output program => lp_sn07gr_20080403
(lp or NOT or ekar or TNG)
(name_field = sn07gr or sn07rt or sn08D ....)

- **qubaph**: (again, but this time using the sn_field)
(input is the output of the previous program)
> qubaph lp_sn07gr_20080403 -r @listasn

Compute the zeropoints and color terms
and uploaded header of SN's images

b) compute the PSF (only for the nights not in the previous set)

qubapsf:
> qubapsf @listasn -l name_field -i

(name_field is the same of step 1b and step 1d)
This program compute the psf of the images
output files => .psf.fits and .ec

c) measure the sn instrumental magnitude (all the nights)

qubasn:
> qubasn @listasn -i

update both .ec files and headers with instrumental magnitudes

d) Compute the sn apparent magnitude (all the nights)
qubaresnoopy:
> qubaresnoopy @listasn

output = apparent magnitude e !!!!
##
EXPLANATION OF EACH PROGRAM IN DITAIL:

> QUBASTD
running the program without option, the help is shown together with the list of field for which the magnitudes of local stars are already known (standards fields and recent SNe)

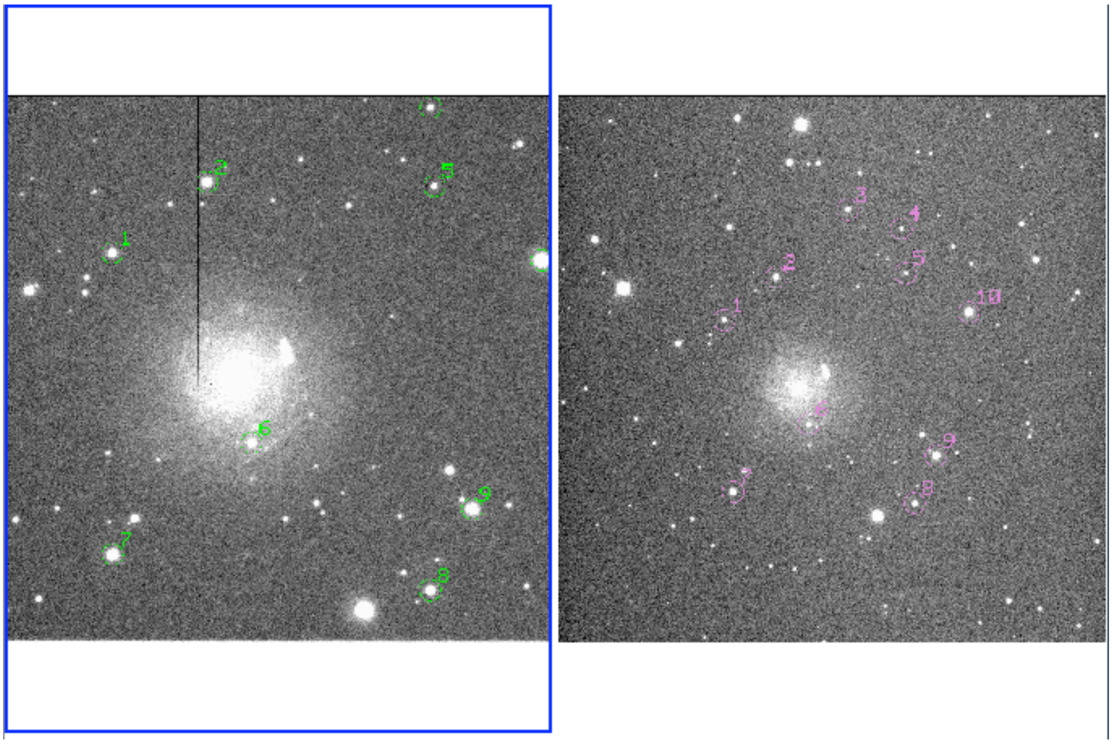
```
help = Usage:   svstandard.py filename  -l coordinatelist
             input        filelist iraf format
             -l,--list   file coordinate
             [-o,--output filename] output file
             available list:
             ['marka', 'ngc300', 'ngc300_1', 'pg0231', 'pg0918', 'pg0942', 'pg1047', 'pg1323', 'pg1525',
'pg1528', 'pg1530', 'pg1633', 'pg1657', 'pg2213', 'pg2331', 'pg2336', 'ps1test', 'ps1test2', 'rui149', 'rui1
52', 's2008ha', 'sn05ip', 'sn07bi', 'sn07gr', 'sn07j', 'sn07rt', 'sn08D', 'ss08ha']
```

> QUBASTD @listname –l coordinatelist
(example ‘qubastd @listastd –l rui194’)

```
### TELESCOPE= lp
#################################
System optical: UBVRI
#################################
#### Header cheak ......ok
##########  B  ##############
#########60.0
z1=157.1459 z2=243.1196
z1=40.39753 z2=100.4921
is the astrometry of the field good [y/n] ? [y]
```

after the identification of the telescope and of the photometric system
the procedure will ask if the astrometry of the image is corrected.

If the astrometry is good, on the frame 1 of DS9 the local stars will be selected in green, while the reference image will be plotted on the frame 2 with the local stars selected in violet.



If no stars are well marked on frame 1 answer no and the procedure
Will ask to identify 3 stars on frame 1 (among the stars selected on frame 2).
Repeat the identification until the stars are well identified on frame 1.
And repeat the operation for all the images in the list. If also the astrometry of the other images is not good, the procedure will try to use the previous identification marking the stars in blue and asking:

```
AND NOW, is the astrometry of the field good [y/n] ? [y]
```

After the astrometry check the procedure will measure the magnitudes of local stars with imexam iterating 3 times on the radius and producing a calibration file:

```
##############################
band U -> ok
band B -> ok
band V -> ok
band R -> ok
band I -> ok
##################################
```

output file example:

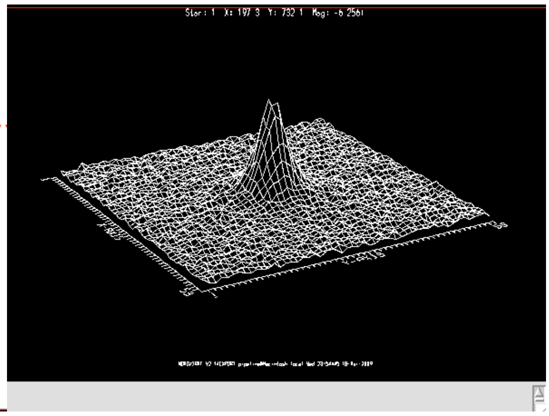
```
RATCam 20071102
*** sn07gr 6
 240.0    60.0    60.0    40.0    40.0
1.3596  1.3804  1.3881  1.3958  1.4021
 -8.22  -10.65  -11.79  -12.24  -12.48      4   16.042   1.19    1.299   0.765   0.622
-11.57   -12.8  -13.34  -13.41  -13.32      5   14.514   0.538  -0.013   0.33    0.308
  9999    9999    9999    9999    9999      6   15.544   0.856   0.504   0.497   0.433
 -9.41  -10.64  -11.26  -11.37   -11.3      7   16.578   0.652   0.012   0.378   0.368
 -8.58  -10.17  -10.99  -11.17  -11.18      9   16.857   0.827   0.474   0.473   0.423
 -8.21   -10.9  -12.02   -12.4  -12.65     12   15.804   1.109   1.007   0.693   0.635
##################################################################
```

###
QUBAPSF   (python version of ecpsf)
> qubapsf @listasn –l sn07gr -i

```
#################################################################
help = Usage:   svpsf.py filename coordinatelist
        input           filelist iraf format
        -l --list filename  list coordinate sequence stars
        [-i,--interactive]     interactive mode (check the stars)
        available list:
        ['infrared/ngc300', 'infrared/sn07gr', 'infrared/sn08D', 'optical/marka', 'optical/ngc300', 'optical/ngc300_1',
'optical/pg0231', 'optical/pg0918', 'optical/pg0942', 'optical/pg1047', 'optical/pg1323', 'optical/pg1525', 'optical/pg1528',
'optical/pg1530', 'optical/pg1633', 'optical/pg1657', 'optical/pg2213', 'optical/pg2331', 'optical/pg2336', 'optical/ps1test',
'optical/ps1test2', 'optical/rui149', 'optical/rui152', 'optical/s2008ha', 'optical/sn05ip', 'optical/sn07bi', 'optical/sn07gr',
'optical/sn07j', 'optical/sn07rt', 'optical/sn08D', 'optical/sn08HA', 'optical/ss08ha', 'sloan/ngc300', 'sloan/sn05ip']
#################################################################
```

the procedure will check the astrometry (see qubastd) and (in interactive way) will ask to accept <a> or delete <d> the stars to compute the psf. Fit <f>, write <w> and quit <q> the interactive psf computation.

```
################ FWHM(median) = 5.87
################ FWHM(mean) = 5.86
################ FWHM = [5.87] ?
add sn07gr_20071101_B,qubafwhm = 5.87
sn07gr_20071101_B updated
sn07gr_20071101_B   197.30   732.10  58.19321   -6.256  ok
sn07gr_20071101_B   375.00   864.60  58.86838   -8.362  ok
sn07gr_20071101_B   793.10  1004.50  56.99038   -6.215  ok
sn07gr_20071101_B   800.10   857.30  57.21449   -5.743
```



```
------------------------------------------------------------------------------
 on GRAPHIC display  > a <  to accept star,  > d <  to delete
on IMAGE   display  > f <  to fit PSF, > w <  to write PSF fit,  > q <  to quit
------------------------------------------------------------------------------
Fitting function gauss    norm scatter: 0.03242829
Analytic PSF fit
   Function: gauss  X: 511.5  Y: 511.5  Height: 3268.591  Psfmag: -6.256
   Par1: 3.394158  Par2: 3.0734
Computed 1 lookup table(s)
Writing PSF image sn07gr_20071101_B.psf.fits
Writing output PSF star list sn07gr_20071101_B.pst
Writing output PSF star group file sn07gr_20071101_B.psg

*****************************************************
id radius  ap_mag   ph_mag  fit_mag   ap_cor  ph_cor
         4        -6.256        -6.262       0.057378         0.006
         5        -8.362        -8.376       0.021378         0.014
         6        9999.0        9999.0          9999          9999
         7        -6.215        -6.198       0.003378        -0.017
         9        -5.743        -5.701       -0.02362        -0.042
        12         -6.523        -6.603       0.148378          0.08
        13         -7.905        -7.939       0.034378          0.034
*****************************************************
APERTURE CORRECTION --> imex   (ap_mag) = 0.0351       +/-     0.0454
APERTURE CORRECTION --> phot   (ph_mag) = 0.0152       +/-     0.0318
number of stars = 9
*******************************************************************************
APERTURE CORRECTION --> imex   (ap_mag) = 0.0273       +/-     0.0164
APERTURE CORRECTION --> phot   (ph_mag) = 0.0193       +/-     0.0090
number of stars= 7       (after 1 sigma rejection)
aperture correction ?  [0.0273]
```

For each image the procedure end with the aperture correction estimation (to be confirmed manually by the reducer. Sometimes it's better to don't use that value but the aperture of one good star close to the SN.
The procedure end once the psf is computed for all images in the list.
###

QUBASN (python version of ecsn)

> qubasn @lista –i

```
###############################################################
help = Usage:   svsn.py filename
           input          filelist (iraf format)
        [-p --psf psffile or psflist]  (iraf format)
        [-i,--interactive]     interactive mode (check the measurement)
        [-s,--system]     Specific photometric system
        [-z,--size size_value] half size of stamp around the object
        [-c,--coordinate]      position of the object from list
        [-b,--background_region]  region of background interpolation
                           (until the border of the stamp)
        [-x --xorder  value]   xorder for background
        [-y --yorder  value]   yorder for background
        [-n,--iteration]     number of background iteration
########################################################
```

strongly suggested to use the interactive option (-i) to check all the steps:
- identification of the object
- cutting a stamp around the object
- selecting again the object on the stamp
- selecting region where the background is fitted (using the outer region)
- chose the order to fit the background
- iterate the background fit
- compute the psf-fit of the object
- manual adjustment of the fit
- artificial stars experiment

other options:
-c if the coordinate of the object are known (in the supernovalist in the archive) the position of the SN is marked to help the identification. In NOT interactive way the identification is done automatically using the coordinate (if –c is selected)
[-z value] the size of the stamp is fixed to value and the program skip the question.
[- b value] the size of the region where the background is fitted. The program will skip the question.
[-x value] and [-y value] x order and order of the background fit. The program will skip the question.
[-n number] number of the iteration on the background. The program will stop iterating the background N times.

```
########## I ##############
### psf check ......ok
### FWHM header check ......ok
### FWHM = 4.03
### aperture correction header check ......ok
### APCO = 0.009649978
>>> Cuts OK ? [y/n] ? [y]
```

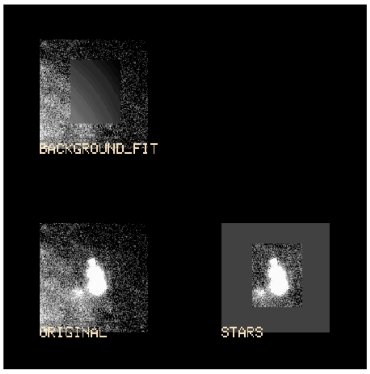
```
 MARK SN REGION WITH - x -, EXIT  - q -
Warning: Attempt to delete a nonexistent file (tmplabel)
### repeat selection ? [y/n] ? [n]
sn07gr_20071101_I[485:566,505:586] -> original
>>>>> Cuts OK [y/n] [y]?
```

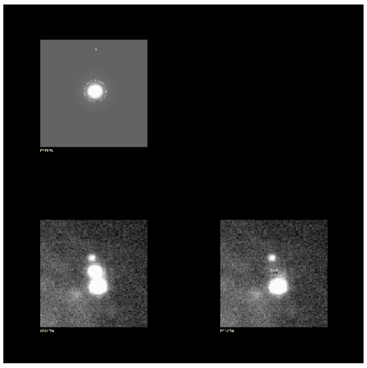
z1=139.6785 z2=335.4828
139.6785 335.4828

IDENTIFY SN AND CO-STARS(S) WITH - x -, EXIT - q -

---

1 1 'ID. SN AND CO-STAR(S) WITH -x- EXIT -q-'
>>>>> SN AND CO-STARS(S) IDENTIFICATIONS OK [y/n] [y]?





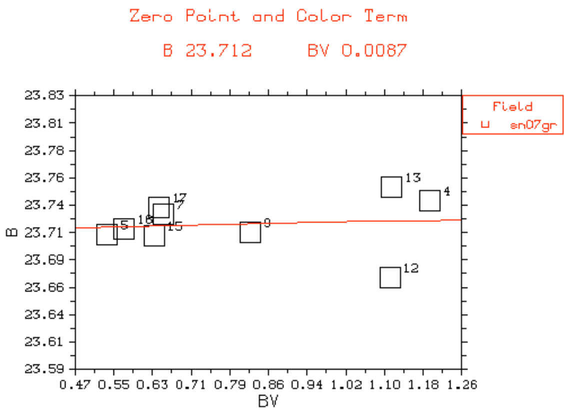
****
ID <apmag on original> <apmag on bgsubt> fitmag truemag err_fit
5 12 18 5 12 18
0 -6.715 -7.477 -8.377 -6.587 -6.948 -6.947 -7.128 -7.10062187 0.007
****
z1=49.99054 z2=150.6663
z1=-33.43946 z2=67.2363
z1=49.99054 z2=150.6663
>>> Iterate on background [y/n] [y] ?
****
#id x_ori y_ori x y ap_ori ap_bgsub fit_mag err_art err_fit
SN 0 522.368 551.915 61.368 61.915 -8.377 -8.289 -7.16862187 0 0.006
****
>>> Not yet happy ? Wish to adjust manually stellar peak ? [y/n] [n]
>>> Errors estimate (through artificial star experiment ?) [y/n] [y] n
****
#id x_ori y_ori x y ap_ori ap_bgsub fit_mag err_art err_fit
 # id ap_original ap_bgsub fit_mag err_art err_fit
 # SN_FIT
 # id ap_ori ap-bg fit_mag
SN 0 522.368 551.915 61.368 61.915 -8.377 -8.289 -7.16862187 0 0.006
****
add sn07gr_20071101_B,QUBASN1 = "-8.377 -8.289 -7.168 0 0.006"
sn07gr_20071101_B updated

##
QUBAPH (python version of ph.m)
> qubaph NOT_20070103_sn07xx –r @listasn

##
help = Usage: ph_new.py filename
input format: the output file of svstandars.py (eg lp_ru149_20080212)
[-r list] list format: the input file of svstandard.py (eg lista_lp_20080212)
-f fix color term
##

the procedure ask the site of the telescope and the zero point and color term to calculate:
TELESCOPE= lp
##################################
System optical: UBVRI
##################################
Header cheak ......ok
site: e-so lasilla, p-aranal, a-siago, c-alar alto, s-iding spring, m-auna kea, r-oque, k-ait or b-ao ? [] r
##################################
[0.46000000000000002, 0.22, 0.12, 0.080000000000000002, 0.040000000000000001]
##################################
how many standard fields [1]?
define band [B]

define color [BV]
<<<< Warning 42: More legend lines than calls to curve in legend! (1)
<<<< (1.272800 / 23.723381) out of axis scaling!
<<<< (1.272800 / 23.723381) out of axis scaling!
do you want to cancel a point ? [y/n] [n]



after the rejection of bad points, the program give zero point and color term of the night (both on the screen as uploading the header of listasn

add sn07gr_20071101_B.fits,qubacons = "B   23.712   BV 0.0087"
sn07gr_20071101_B.fits updated
####################################
#
 # lp 20071102 54406.193119 1.38042199612   B 23.712   BV 0.0087
#
####################################

##
QUBARESNOOPY

qubaresnoopy @listasn

output => SN magnitudes

the program ask first for the site of the telescope, then try to take all the information from the header of the images:
qubaSN1 and **qubacons** but also **exptime,JD,airmass.**
If the previous programs were use correctly a sequence of ‘return’ will give you the output (Zeropoint and color term have to be confirmed interactively by the reducer).
If the constant are not in the header the program will ask to introduce those interactively.

TELESCOPE= lp
####################################
System optical: UBVRI
####################################
site: e-so lasilla, p-aranal, a-siago, c-alar alto, s-iding spring, m-auna kea, r-oque, k-ait or b-ao ? [] r
###################################
[0.46000000000000002, 0.22, 0.12, 0.080000000000000002, 0.040000000000000001]
###################################
Header cheak ......ok
FILTER = B

```
### AIRMASS header check ......ok
### AIRMASS = 1.38042199612
### exptime header check ......ok
### EXPTIME = 60.0
zero point (with filter = B)?[23.712]
color term [BV,VR,RI] (with filter B)?[BV]
color term value (with filter B)?[0.0087]
### CONSTANTS header check ......ok
### CONSTANT = "B 23.712     BV 0.0087"
### instrumental magnitude header check ......ok
### MAG = -7.21
### JD header check ......ok
### JD = 54406.693119
### FILTER = I
### AIRMASS header check ......ok
### AIRMASS = 1.40218794346
### exptime header check ......ok
### EXPTIME = 40.0
WARNING: NO CONSTANT FOR THE FRAME !!!
do you want to give the constant interactively  [y/n] ? [y]
zero point (with filter = I)? 23.5
color term [BV,VR,RI] (with filter I)? RI
color term value (with filter I)? 0.04
16.2011125688 B
14.8314387661 I
14.3364237012 R
16.2985402403 U
15.8786518882 V
################################################################################
# B   BV
#  23.712    0.0087
# I   RI
#  23.5    0.04
.....
################################################################################
#      JD          U            B            V            R           I
# 54406.69      16.299 0.017    16.201 0.006    15.879 0.003    14.336 0.004    14.831 0.005
################################################################################
```

##

RESNOOPY (python version of resnoopy.m)

resnoopy @listasn

output => local stars and SN magnitudes (using .ec files)

the program ask first for the site of the telescope, and then ask which magnitude (phot or fit), then try to take the instrumental magnitudes from .ec files and the constant from the header **qubacons.** If the constant are not in the header the program will ask to introduce those interactively. (Zeropoint and color term have to be confirmed interactively by the reducer).

```
### TELESCOPE= lp
###################################
System optical: UBVRI
###################################
site: e-so lasilla, p-aranal, a-siago, c-alar alto, s-iding spring, m-auna kea, r-oque, k-ait or b-ao ? [] r
##################################
[0.46000000000000002, 0.22, 0.12, 0.080000000000000002, 0.040000000000000001]
##################################
#### Header cheak ......ok
phot or fit (p/f) [f]? f

...
################################################################################
#
#      name         U            B            V            R           I
    4 18.028 0.078    17.150 0.018    16.770 0.011    15.662 0.010    14.484 0.007
```

```
   5 14.734 0.010    15.031 0.006    15.212 0.005    14.482 0.006    13.623 0.007
   6 9999.0 0.000    9999.0 0.000    9999.0 0.000    9999.0 0.000    9999.0 0.000
   7 16.937 0.037    17.209 0.020    17.333 0.013    16.559 0.014    15.669 0.015
   9 17.697 0.061    17.708 0.023    17.632 0.016    16.758 0.015    15.799 0.014
  12 17.394 0.053    16.808 0.014    16.535 0.008    15.478 0.008    14.311 0.011
  13 16.116 0.025    15.472 0.007    15.151 0.005    14.180 0.008    13.113 0.008
  15 15.184 0.014    15.412 0.007    15.518 0.005    14.746 0.006    13.858 0.008
  16 13.860 0.008    14.203 0.003    14.389 0.003    13.656 0.004    12.801 0.006
  17 13.560 0.005    13.837 0.003    13.944 0.003    13.166 0.005    12.291 0.004
 SN1 16.297 0.017    16.241 0.006    16.134 0.003    15.313 0.004    14.372 0.005
#
###################################################################################
do you kow is the night was photometric ? (p) photometric or (n) not photometric [n]
```

the program ask then if the night was photometric or not and store all the information in the DOC file:

```
*** 54406.6    lp   ph
#     name        U          B          V          R          I
#
   4 18.028 0.078    17.150 0.018    16.770 0.011    15.662 0.010    14.484 0.007
   5 14.734 0.010    15.031 0.006    15.212 0.005    14.482 0.006    13.623 0.007
   6 9999.0 0.000    9999.0 0.000    9999.0 0.000    9999.0 0.000    9999.0 0.000
   7 16.937 0.037    17.209 0.020    17.333 0.013    16.559 0.014    15.669 0.015
   9 17.697 0.061    17.708 0.023    17.632 0.016    16.758 0.015    15.799 0.014
  12 17.394 0.053    16.808 0.014    16.535 0.008    15.478 0.008    14.311 0.011
  13 16.116 0.025    15.472 0.007    15.151 0.005    14.180 0.008    13.113 0.008
  15 15.184 0.014    15.412 0.007    15.518 0.005    14.746 0.006    13.858 0.008
  16 13.860 0.008    14.203 0.003    14.389 0.003    13.656 0.004    12.801 0.006
  17 13.560 0.005    13.837 0.003    13.944 0.003    13.166 0.005    12.291 0.004
 SN1 16.297 0.017    16.241 0.006    16.134 0.003    15.313 0.004    14.372 0.005
```

All the doc files are then used for the photometric calibration of the local star sequence (see QUBAEXA.PY)

##

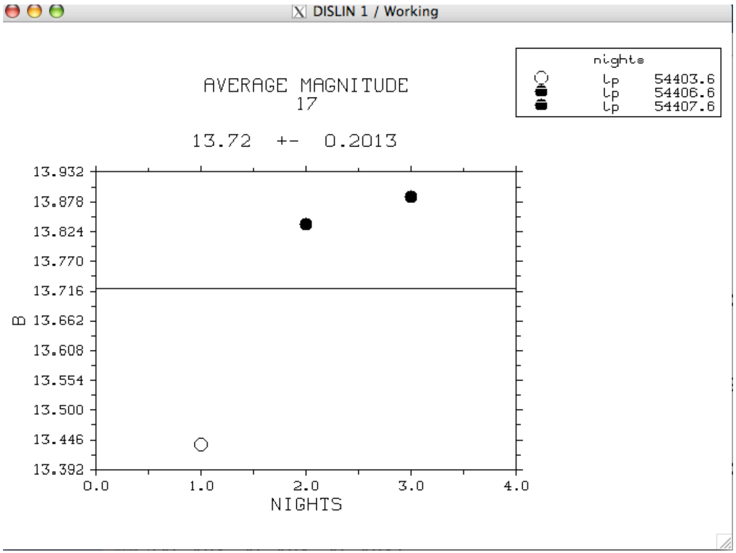
QUBAEXA (python version of exa)

Qubaexa @listadoc

The input is a list of doc files (produced by resnoopy.py). The program can use also doc file of different length. Ask the photometric system and shows the name of all sequence stars and the number of measurements for each star. After that, the pipeline will sort the stars from the brightest to the faintest and **suggest** the order to compute the magnitude. If one follows the order all the stars in all filters will be measured.

```
which photometric system UBVRI [0], JHK [1], ugriz [2] 0

SEQUENCE STARS
['4', '5', '6', '7', '9', '12', '13', '15', '16', '17', 'SN1']
Number of nights the star was measured
[3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]
which sequence star [17] ?
which BAND [U] ? B
```



the program ends showing the magnitudes of sequence stars and the color and producing 2 files **sequence_star.list** and **sequence_star_mag.list**

SEQUENCE STARS: MAGNITUDE

17	12.45	0.03	13.720	0.201	14.000	0.03	13.033	0.03	14.020	0.01
16	14.245	0.02	15.234	0.302	15.34	0.05	14.030	0.4	15.050	0.3
13	15.345	0.04	16.345	0.02	13.444	0.03	12.440	0.4	13.234	0.04

SEQUENCE STARS: LIST FORM

17	13.720	0.034	0.234	0.534	0.434
16	12.453	0.100	-0.333	0.0034	0.143
13	14.564	0.345	-0.433	-0.434	0.032

the values in **sequence_star.list** have to copied in the field.list in the directory ~/pipeline/coordinate_std/optical/.
Now the field is photometric calibrate and the reducer can calibrate also the not photometric night using the local sequence stars. Also the photometric nights can be recalibrated using the list of objects (listasn) instead of the standard and computing againt the zero points.
##

END