

Data Reduction Procedure

Enter the directory with your data and type *mkiraf* and then input *xterm* when prompted for your terminal type. This will generate a *login.cl* file that is required for iraf to run correctly. Now you need to add the following to the end (before where it says keep at the end):

```
    imred
    ccdred
    xdimsum
*****
```

```
//Separates out the log into the lists and files used later in the process//
//for the CPAPIR data the following lists were used
```

```
grep dark BD* > dark
awk '{print ""$1""}' dark > dark.list
grep dither BD* > dither
grep 9.4710 dither > dither.list
grep 5.4120 dither > ksci.list
grep J dither.list > jsci.list
grep domeflat_J BD* > jflat.list
awk '{print ""$1""}' jsci.list > J.list
awk '{print ""$1""}' jflat.list > domeflat_J.list
grep H dither.list > hsci.list
grep domeflat_H\ BD* > hflat.list
awk '{print ""$1""}' hsci.list > H.list
awk '{print ""$1""}' hflat.list > domeflat_H.list
awk '{print ""$1""}' ksci.list > K.list
cp K.list skyflat_K.list
mkdir uparm
chmod u+w *
```

```
//Combine the dark images to make a master dark frame
```

Dark frames are median combined to produce a master dark, a histogram of the counts in each pixels is produced.

Every pixels outside of approximately where the Gaussian starts and ends is considered a “bad” pixel.

A good pixel mask is produced by setting good pixels to 1 and bad pixels to 0, this mask is multiplied by every flat and science image.

A bad pixel mask is produced by setting good pixels to 0 and bad pixels to 1, this is used by *ccdproc* to interpolate what the value of the bad pixels should be.//

```
cl>unlearn darkcombine
cl>hedit @dark.list IMAGETYP 'dark' add+ verify-
```

```
cl>darkcombine @dark.list output=dark.fits combine=median reject=minmax process=no
scale="exposure"
```

```
cl>hedit dark.fits IMAGETYP 'dark' add+ verify-
```

```
cl>hedit @J.list IMAGETYP 'object' add+ verify-
```

```
cl>hedit @H.list IMAGETYP 'object' add+ verify-
```

```
cl>hedit @K.list IMAGETYP 'object' add+ verify-
```

```
//Create the mask frame and bpm frame using the dark frame//
```

```
cl>unlearn imreplace
```

```
cl>imcopy dark.fits mask.fits
```

```
cl>imreplace mask.fits 0 lower=INDEF upper=50
```

```
cl>imreplace mask.fits 0 lower=3000 upper=INDEF
```

```
cl>imreplace mask.fits 1 lower=50 upper=3000
```

```
cl>hedit mask.fits IMAGETYP 'mask' add+ verify-
```

```
cl>imcopy dark.fits bpm.fits
```

```
cl>imreplace bpm.fits 1 lower=INDEF upper=50
```

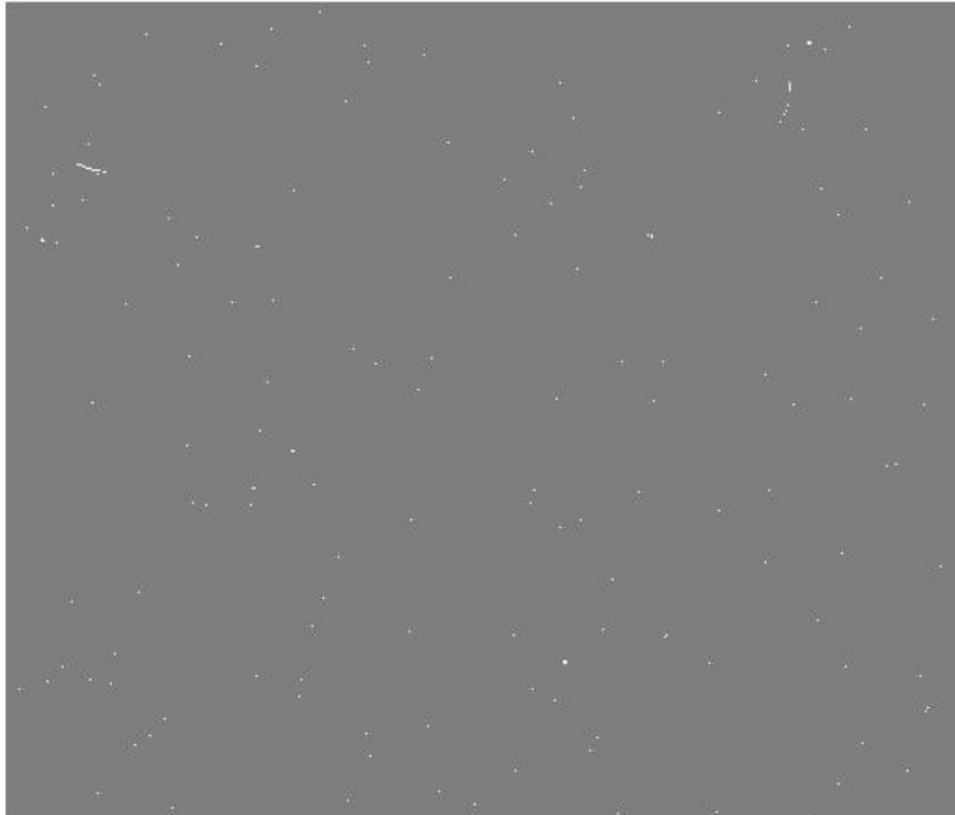
```
cl>imreplace bpm.fits 1 lower=3000 upper=INDEF
```

```
cl>imreplace bpm.fits 0 lower=50 upper=3000
```

```
cl>hedit bpm.fits IMAGETYP 'bpm' add+ verify-
```

```
//This is an example of what a mask.fits should look similar to//
```

```
*****To display this open a DS9 window and in iraf type display image_name*****
```



```

//Multiply all the other images by the mask//
cl>ls *_*.fits > all.list
cl>ls dark.fits >> all.list
cl>imarith @all.list * mask.fits @all.list

//Also if there is substantial dark current it should be subtracted from every image, but in
the case of CPAPIR at CTIO 1.5m this is not necessary//

//Create master flat images in JH band using flatcombine
cl>unlearn flatcombine
//Running flatcombine for J band//
cl>hedit @domeflat_J.list IMAGETYP 'flat' add+ verify-
cl>flatcombine @domeflat_J.list output=Jflat.fits combine=median reject=minmax
process=no scale="median"
//Running flatcombine for H band//
cl>hedit @domeflat_H.list IMAGETYP 'flat' add+ verify-
cl>flatcombine @domeflat_H.list output=Hflat.fits combine=median reject=minmax
process=no scale="median"

//Create master K flat image from either sky K flats or science K images (given in
skyflat_K.list)
cl>unlearn xslm
cl>unlearn makemask
cl>unlearn imcombine

//Copy the K images (either K science or K skyflat) to new images called *.Kflat.fits//
//Due to turbulence in the atmosphere the value of the general sky varies//
//Using xslm in IRAF we use a running mean across ~15 frames to determine the value of
the sky in each pixel and subtract it from the value of the pixel in each frame//
cl>!sed s/^/SKY/ skyflat_K.list > skyflat_new.list
cl>!cp skyflat_K.list skyflat_temp.list
cl>!awk '{print "imcopy \"$1\".fits SKY\"$1\".fits}' skyflat_temp.list > imcopy.cl
cl>cl < imcopy.cl
cl>!mv skyflat_new.list skyflat_K.list

//First pass of xslm and makemask for flats//
//After the first pass of xslm the output images are run through makemask to find object
and create a mask//
cl>xslm @skyflat_K.list "" 15 ".xslm.fits" forcescale=yes useomask=no
statsec="[30:2018,30:2018]" nskymin=15 del_ssmasks=yes del_hmasks=yes nreject=3
cl>ls *.xslm.fits > inmask.list
cl>!sed s/.xslm.fits/_obm.pl/ inmask.list > outmask.list
cl>makemask @inmask.list @outmask.list zmin=-1000 zmax=62000 nsigrej=3 nsigthr=3

// Second pass of xslm and makemask for flats//

```

```
//Second pass of xslm uses the object masks to ignore the pixels that are representing
objects in a frame when calculating the mean sky value for a pixel//
```

```
cl>xslm @skyflat_K.list @outmask.list 15 ".xslm.fits" forcescale=yes useomask=yes
statsec="[30:2018,30:2018]" nskymin=15 del_ssmasks=yes del_hmasks=yes nreject=2
cl>!sed s/.$/.pl/ skyflat_K.list > outmask_Kflat.list
makemask @inmask.list outlist="@outmask_Kflat.list" zmin=-1000 zmax=62000
nsigrej=3 nsigthr=2
```

```
// Imcombine the K images with masks from previous step//
```

```
cl>!awk '{print "hedit SKY"$1".fits BPM SKY"$1".pl add+ verify-"}' skyflat_temp.list >
addbpm.cl
```

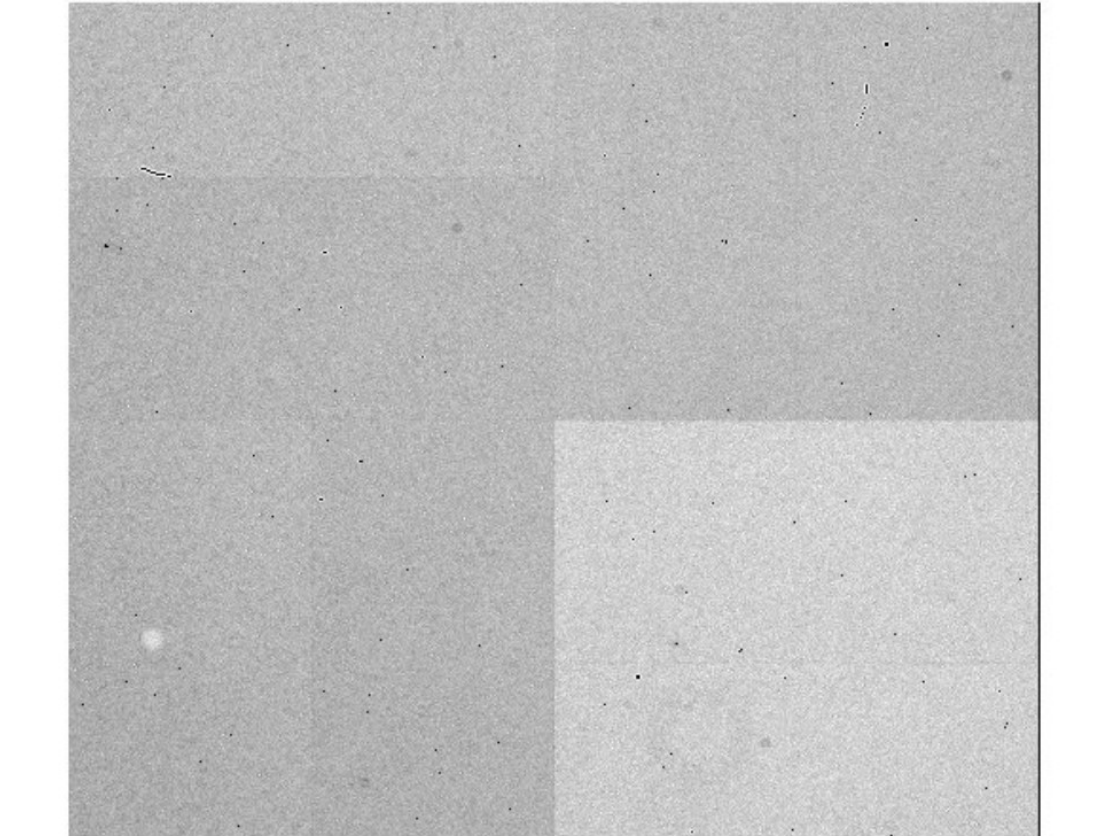
```
cl>cl < addbpm.cl
```

```
cl>!sed s/.$/.fits/ skyflat_K.list > skyflat_K2.list
```

```
cl>imcombine @skyflat_K2.list Kflat.fits combine=median reject=minmax scale=median
masktyp=goodvalue maskvalue=0 nlow=10 nhigh=10
```

```
cl>hedit *flat.fits IMAGETYP 'flat' add+ verify-
```

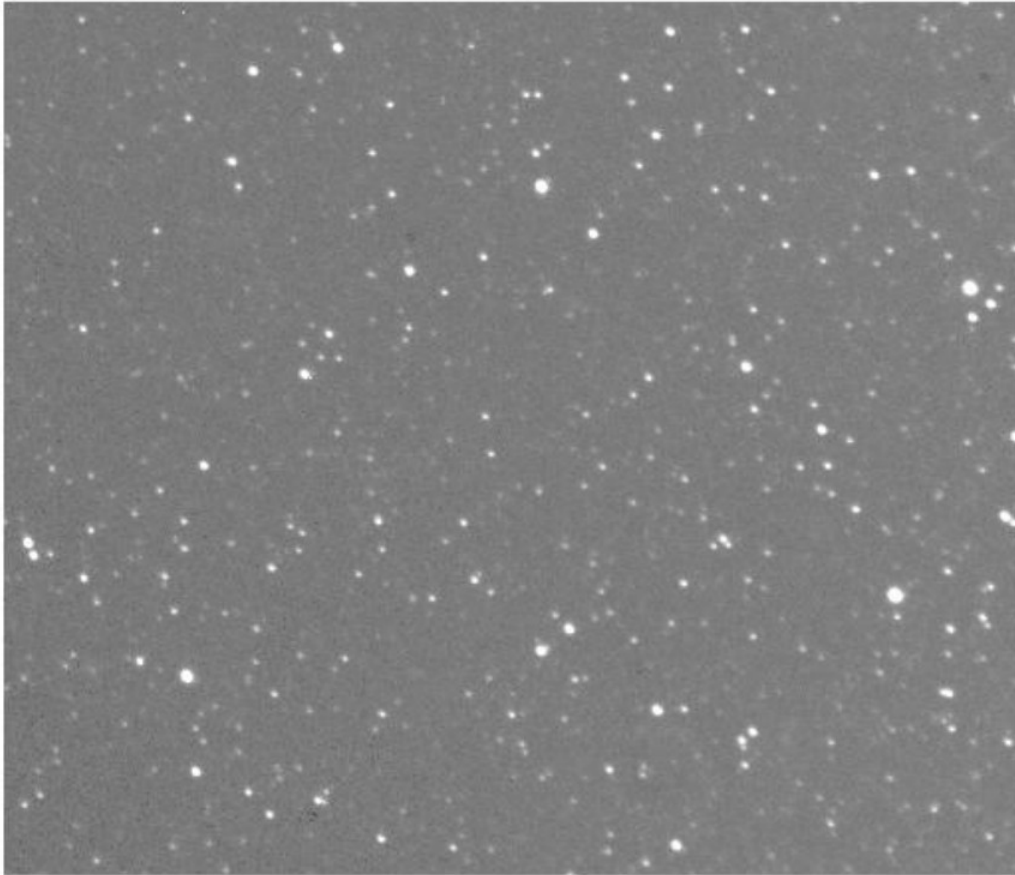
```
//Should have an image somewhat similar to this for your flats//
```



```
//Process the science images using CCDproc//
```

```
//CCDproc will interpolate the values of the pixels in the bad pixel mask and apply
(divide) each JHK dithered frame by the normalized flat//
cl>unlearn ccdproc
cl>ccdproc @J.list fixpix=yes overscan=no trim=no zerocor=no darkcor=no flatcor=yes
illumcor=no fringecon=no readcor=no scancor=no fixfile="bpm.fits" flat="Jflat.fits"
cl>ccdproc @H.list fixpix=yes overscan=no trim=no zerocor=no darkcor=no flatcor=yes
illumcor=no fringecon=no readcor=no scancor=no fixfile="bpm.fits" flat="Hflat.fits"
cl>ccdproc @K.list fixpix=yes overscan=no trim=no zerocor=no darkcor=no flatcor=yes
illumcor=no fringecon=no readcor=no scancor=no fixfile="bpm.fits" flat="Kflat.fits"

//Should have an image similar to this after running CCDproc//
```



```
//First pass of xslm and makemask//
//After the first pass of xslm the output images are run through makemask to find object
and create a mask//
cl>unlearn xslm
cl>xslm @J.list "" 15 ".red.fits" forcescale=yes useomask=no statsec="[30:2018,30:2018]"
nskymin=15 del_ssmasks=no del_hmasks=no nreject=3
cl>xslm @H.list "" 15 ".red.fits" forcescale=yes useomask=no
statsec="[30:2018,30:2018]" nskymin=15 del_ssmasks=no del_hmasks=no nreject=3
cl>xslm @K.list "" 15 ".red.fits" forcescale=yes useomask=no
statsec="[30:2018,30:2018]" nskymin=15 del_ssmasks=no del_hmasks=no nreject=3
```

```
//Create mask lists and run makemask//
cl>unlearn makemask
cl>ls *.red.fits > inmask.list
cl>!sed s/./_obm.pl/ inmask.list > outmask.list
cl>makemask @inmask.list @outmask.list zmin=-1000 zmax=62000 nsigrej=3 nsigthr=2
```

```
//Second pass of xslm and makemask for the science images//
//Second pass of xslm uses the object masks to ignore the pixels that are representing
objects in a frame when calculating the mean sky value for a pixel//
cl>unlearn xslm
cl>!sed s/./_obm.pl/ J.list > Jmask.list
cl>!sed s/./_obm.pl/ H.list > Hmask.list
cl>!sed s/./_obm.pl/ K.list > Kmask.list
cl> xslm @J.list @Jmask.list 15 ".final.fits" forcescale=yes useomask=yes
statsec="[30:2018,30:2018]" nskymin=15 del_ssmasks=yes del_hmasks=yes nreject=2
cl>xslm @H.list @Hmask.list 15 ".final.fits" forcescale=yes useomask=yes
statsec="[30:2018,30:2018]" nskymin=15 del_ssmasks=yes del_hmasks=yes nreject=2
cl>xslm @K.list @Kmask.list 15 ".final.fits" forcescale=yes useomask=yes
statsec="[30:2018,30:2018]" nskymin=15 del_ssmasks=yes del_hmasks=yes nreject=2
```

```
//Add the SKYMED value back to the images
cl>ls *.final.fits > allfinal.list
cl>imheader @allfinal.list l+ grep SKYMED > skymed.input
cl>!awk '{print "'1.0/$3''"}' skymed.input > skymed.output
cl>imarith @allfinal.list + @skymed.output @allfinal.list
```

```
//Remove all unnecessary images and files//
cl>!rm all.list imgcopy.cl addbpm.cl skyflat_temp.list *.xslm.fits *.Kflat.fits *.pl
normalize.cl *mask*.list logfile
```

```
//ALL THE PROCESSED SCIENCE IMAGES ARE *.final.fits//
```

```
//This should be similar to what one of your final images should like after running the
whole data reduction procedure//
```

