EUROPEAN SOUTHERN OBSERVATORY

Organisation Européene pour des Recherches Astronomiques dans l'Hémisphère Austral
Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

ESO - European Southern Observatory
Karl-Schwarzschild Str. 2, D-85748 Garching bei München

Very Large Telescope

Paranal Science Operations

SINFONI data reduction cookbook

Doc. No. VLT-MAN-ESO-14700-4037

Issue 1.0, Date 26/10/2007

Prepared .... `C. Dumas` .... Date Signature

Approved .... `A. Kaufer` .... Date Signature

Released .... `O. Hainaut` .... Date Signature

This page was intentionally left blank

Change Record

Issue/Rev.	Date	Section/Parag. affected	Reason/Initiation/Documents/Remarks
0.0	12/11/2006	All	First draft provided by P. Amico
1.0	26/10/2007	All	First version release by C. Dumas

This page was intentionally left blank

Contents

1 Introduction

You will find in this document all the basic information to guide you through the various steps of reduction of SINFONI data, which ultimately produce the reconstructed 3D cube needed to extract the spectral information corresponding to the region of interest of the SINFONI field-of-view.

1.1 Purpose

The purpose of this document is to provide a brief introduction and quick help to the reduction of SINFONI data. A more detailed explanation of the data reduction process and use of the pipeline can be found in the SINFONI Pipeline User Manual (VLT-MAN-ESO-19500-3600), whose latest version can be downloaded from the ESO-VLT instrument pipelines webpage: http://www.eso.org/projects/dfs/dfs-shared/web/vlt/vlt-instrument-pipelines.html

It is also important to keep in mind that, as all reduction software aiming at processing a wide variety of astronomical data, the tools described in this document will certainly be found of limited use in certain complex cases, either due to the observing strategy adopted, or simply to the nature of the science target.

Finally, it is useful to keep in mind that this document is the first version of the SINFONI reduction cookbook, which is expected to evolve considerably over time, particularly while the users, including myself, are gaining experience with the subtleties of SINFONI data reduction. For this reason, all users are welcomed to send useful comments and suggestions regarding improved ways to reduce SINFONI data to the instrument user account:"sinfoni@eso.org".

1.2 Reference documents

Please refer to the documents listed below to best optimize both the use of SINFONI to carry-out your research program, and the use the ESO pipeline to reduce your data:

1.2.1 Main documents

1 SINFONI Pipeline User Manual - VLT-MAN-ESO-19500-3600
http://www.eso.org/projects/dfs/dfs-shared/web/vlt/vlt-instrument-pipelines.html

2 SINFONI User Manual - VLT-MAN-ESO-14700-3517
http://www.eso.org/instruments/sinfoni/doc

1.2.2 Additional reading:

3 Common Pipeline Library (CPL)
http://www.eso.org/sci/data-processing/software/cpl

4 Gasgano File Organizer
http://www.eso.org/observing/gasgano

5 ESORex command line tool
http://www.eso.org/sci/data-processing/software/cpl/esorex.html

6 QFITSView datacube viewer
http://www.mpe.mpg.de/~ott/QFitsView

7 Euro3D cube viewer
http://www.aip.de/Euro3D/E3D

8 SINFONI quality control and data flow operations web pages
http://www.eso.org/observing/dfo/quality/index_sinfoni.html

9 Euro3D data format
http://www.aip.de/Euro3D/E3D/E3D_distr_files/Documentation/Euro3D_format.pdf

10 Pipeline recipes, download links, installation instructions
http://www.eso.org/projects/dfs/dfs-shared/web/sinfoni/sinfo-pipe-recipes.html

1.3 Pipeline installation

Gasgano is a Data File Organizer developed and maintained by ESO to help its user community to manage and organize in a systematic way the astronomical data observed and produced by all VLT compliant telescopes, i.e. by telescopes which are being operated through Observation Blocks. The tool also supports FITS files which are not generated by those telescopes but with a limited number of functionalities. Gasgano provides an interactive interface to the data reduction recipes of the pipeline and also allows users to run custom data reduction scripts.

ESOrex is a command-line utility for running pipeline recipes. It may be embedded by users at their home institute into data reduction scripts for the automation of processing tasks.

Both Gasgano and ESOrex are included in the public release packages of the pipeline. Typically, mainstream data reduction can be done using GASGANO. Both command line recipes, and GASGANO, will be used for reducing more complex data sets. Currently, the SINFONI pipeline (ESOREX and GASGANO) is offered for platforms such as Linux (glibc 2.1 or later), Sun Solaris 2.8 or later and MacOSX. Please refer to the Pipeline User Manual about pipeline installation (http://www.eso.org/projects/dfs/dfs-shared/web/vlt/vlt-instrument-pipelines.html).

Most of the data reduction description contained in this document refers to the use of GASGANO, except when ESOREX command lines are explicitly needed.

IMPORTANT: You should read the “Quick start” section on how to use Gasgano and Esorex, which is provided in the SINFONI pipeline User Manual.

1.4 Abbreviations and acronyms

The following abbreviations and acronyms are used in this document:

SciOps	Science Operations
ESO	European Southern Observatory
Dec	Declination
eclipse	ESO C Library Image Processing Software Environment
ESO-MIDAS	ESO's Munich Image Data Analysis System
FITS	Flexible Image Transport System
IRAF	Image Reduction and Analysis Facility
PAF	Parameter File
RA	Right Ascension
UT	Unit Telecope
VLT	Very Large Telescope

1.5 Stylistic conventions

The following styles are used:

bold	in the text, for commands, etc., as they have to be typed.
italic	for parts that have to be substituted with real content.
box	for buttons to click on.
`teletype`	for examples and filenames with path in the text.

Bold and *italic* are also used to highlight words.

2 An overview of the SINFONI data product

SINFONI is an Integral Field Spectrograph fed by the MACAO Adaptive Optics (AO) Module. Observations can be done either using AO in Natural Guide Star (NGS) or Laser Guide Star (LGS) modes, or in seeing limited conditions (i.e. no AO correction). SINFONI uses an image slicer to sample a square patch of the focal plane, cut it into 32 stripes. These stripes are dispersed spectrally and reimaged on a 2048x2048 detector. The instrument operates in the J, H, K and H+K bands (i.e. from 1 to 2.5 microns) with average resolutions between ∼2000-4000 depending on the setting. SINFONI has plate scales of 0.25, 0.1 and 0.025 per pixel, which correspond to 8, 3 and 0.8 squared FOVs (large, medium and small).

An Integral Field Spectrograph such as SINFONI, produces 2-dimensional raw image files that contain both spatial and spectral information. It is therefore possible to reconstruct the science frame into a 3D data cube. Conventionally the X and Y represent the spatial directions on sky, while the corresponding spectrum is along the Z axis.

2.1 File viewers

Regular fits files are best viewed with the usual suites of tools such as saoimage, ds9, fv, skycat, the two later allowing also to display the content of fits tables. Links to download these tools can be found at: http://tdc-www.harvard.edu/software/saoimage.html.

An efficient tool to display SINFONI 3-D cube has been developed by our colleagues at MPE and can be accessed at : http://www.mpe.mpg.de/~ott/QFitsView.

More information can be found in the user manual, as well as on the instrument webpages:
- http://www.eso.org/instruments/sinfoni/overview.html
- http://www.eso.org/instruments/sinfoni/inst

2.2 Calibration and science raw files

The science and calibration files for SINFONI consist of the following frames below. ESO uses custom FITS keywords for file classifications. Every raw frame produced by the instrument is uniquely classified by means of the DPR.CATG, DPR.TYPE and DPR.TECH keywords. The table below lists al the available combinations for SINFONIs calibration frames.
A complete description can be found at: http://www.eso.org/observing/dfo/quality/SINFONI/pipeline/recipe_calib.html

DPR.CATG	DPR.TYPE	DPR.TECH	Type of frame
CALIB	LINEARITY, LAMP	IFU	Detector Linearity
CALIB	DARK	IMAGE	Dark
CALIB	FLAT,LAMP	IFU	Flat
CALIB	DISTORTION,FIBER,NS	IFU	Distortion North-South
CALIB	DISTORTION,FLAT,NS	IFU	Distortion Flats
CALIB	DISTORTION,WAVE,NS	IFU	Distortion Wave
CALIB	WAVE,LAMP	IFU	Wave
CALIB	PSF-CALIBRATOR	IFU	PSF Star
CALIB	SKY,PSF-CALIBRATOR	IFU	Sky for PSF Star
CALIB	STD	IFU	Telluric Standard
CALIB	SKY,STD	IFU	Sky for Telluric
SCIENCE	OBJECT	IFU,NODDING	Science
SCIENCE	SKY	IFU,NODDING	Sky for Science

2.3 Calibration products

The ESO pipeline recipes produce files which can be classified according to their PRO.CATG keywords. In the table below we list some of those files. Again, more information about the pipeline recipe outputs files can be found in the Pipeline User Manual, as well as at: http://www.eso.org/observing/dfo/quality/SINFONI/pipeline/recipe_calib.html

PRO.CATG	Type of frame
BP_MAP_NL	Non linear bad pixels map
BP_MAP_HP	Hot pixels map
MASTER_BP_MAP	Map of all bad pixels
MASTER_DARK	Reduced dark frame
MASTER_FLAT_LAMP	Reduced lamp flat field
DISTORTION	Distortion coefficients
SLITLETS_DISTANCE	Table with relative slitlets distances
WAVE_MAP	Wavelength calibration map
SLIT_POS	Pixel position for start/end of each slitlet (static also provided)
DRS_SETUP_WAVE	Static: wavelength calibration
FIRST_COL	Static: table with initial guess of pixel position of first slitlet
REF_LINE_ARC	Static: table with wavelength and line intensity
REF_BP_MAP	Static bad pixel map
STD_STAR_SPECTRUM	Extracted spectrum 1D image
COADD_STD	Coadded cube for standard
COADD_OBJ	Coadded cube for science object

3 Reducing SINFONI data

Please, for more information, refer to the "Reduction cascade" description, available in the Chapter "Data reduction" of the Pipeline User Manual. This Chapter guides you though the main steps of data reduction, which are summarized below.

3.1 Bad lines removal

Prior to use the pipeline (either via gasgano or esorex or a combination of both) to reduce SINFONI data, it is recommended to clean the raw frames, both science and calibrations, from the bad lines that are created by the data processing hardcoded at the detector level. The 4 pixels surrounding the detector are not illuminated and are used by the hardcoded processing to estimate the "bias" level for each frame. The mean of each line is then subtracted from the frame prior to write the data onto the instrument workstation. The longer the exposure, the larger the numbers of hot pixels present on the frame. When one of this hot pixel is located among the non-illuminated "edge" pixels, the mean value of the corresponding line is then over-estimated, which creates some dark stripes in the raw frames when it is subtracted (see Fig. 1 below).
You are thus encouraging to correct each frame from this effect, which is particularly important when handling data obtained on a faint science source requiring long integrations. You will find in Annex A an IDL code whose purpose is to clean the frames from the contaminated lines. Of course, you are free to write your own script using the language of your choice, or modify the IDL code provided within this document.

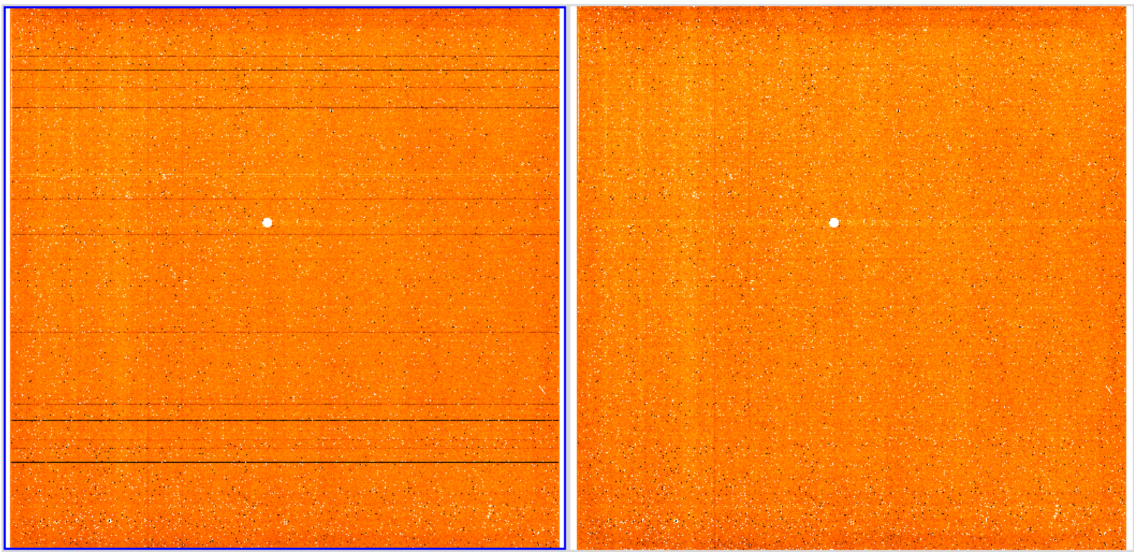

Figure 1: Illustration of a raw 300s dark frame before (left) and after (right) correcting from the dark line effect.

3.2 Bad Pixel Mask ,Master Dark and Master Flat

The first step of the reduction consists to build the mask of badpixels, which is made of the dead, hot, and non-linear pixels. The following will guide you through the reduction steps to identify bad pixels, and create the "master dark" and "master flat" frames.

3.2.1 Linearity frames

You will need to run the **sinfo_rec_detlin** recipe, either with Gasgano or Esorex, on the set of 24 Linearity frames (DPR.TYPE=LINEARITY, LAMP) provided as part of the SINFONI calibration.

The main output file will be the map of non-linear pixels (PRO.CATG=BP_MAP_NL).

The identification of the non-linear pixels is made using the linearity frames obtained for each grism setting. Since these bad pixels are independent of the pre-optics used, the linearity frames are obtained by illuminating the array for several exposure times (1, 9, 18, 27, 36, 45s), with the grism and filter used for your science observations, and using the 25mas/pix scale. If you want to experiment with the input parameter adjustments, the threshold parameters, which define the range between identifying a pixel as good or bad, are the most sensitive parameter.

3.2.2 Dark frames

You will need to run the **sinfo_rec_mdark** recipe, on your set of dark frames (DPR.TYPE=DARK).

The main output file will be the map of hot pixels and master dark (PRO.CATG=BP_MAP_HP and MASTER_DARK).

Note that it is recommended that you carefully look at your set of dark frames in order to discard any frames contaminated with cosmic rays, which would degrade the quality of your master dark frame and hot pixel frame. Similarly, you will need to adjust the threshold parameters as needed to best identify the bad pixels. You can also adjust the percentage of the rejection value for the low- and high- intensity pixels.

3.2.3 Flat frames

You will need to run the **sinfo_rec_mflat** recipe, on your set of flat frames (DPR.TYPE=FLAT, LAMP) and on the non-linear pixel map (BP_MAP_NL), hot pixel map (BP_MAP_HP), and reference bad pixel map (REF_BP_MAP).

The main output file will be the master map of bad pixels and master flat (PRO.CATG=MASTER_BP_MAP and MASTER_FLAT_LAMP).

3.3 Distortion

You will need to run the **sinfo_rec_distortion** recipe, on your set of 80 distortion frames (DPR.TYPE=DISTORTION,FIBER,NS, DISTORTION,FLAT,NS , and DISTORTION,WAVE, NS). Other inputs of the recipe are (REF_LINE_ARC) and .

The main output files will be the distortion coefficients and the distance between the individual slitlets (PRO.CATG=SLITLET_DISTANCES and DISTORTION).

These files will be used to create the wavelength map (i.e. to assign a wavelength value to each pixel on your detector), as well as to reconstruct the final 3D cubes.

3.4 Wavelength calibration

You will need to run the **sinfo_rec_wavecal** recipe, on your set of wavelength frames (DPR.TYPE=WAVE,LA using also as input the following frames: MASTER_BP_MAP, MASTER_FLAT_LAMP, DISTORTION, REF_LINE_ARC and DRS_SETUP_WAVE.

The main output files will be the wavelength map of the detector and the slitlet edge position table (PRO.CATG=WAVE_MAP and SLIT_POS).

At this stage you have created all the calibration files you need to reduce your science data and reconstruct their corresponding 3D cubes.

3.5 Cube reconstruction

You will need to run the **sinfo_rec_jitter** recipe, on your set of:
- Telluric standard frames (STD, and SKY,STD) to estimate the image quality or AO correction, or
- PSF standard frames (PSF, and SKY,PSF) to measure the instrumental and atmospheric responses, or
- Science frames (OBJ, and SKY) to reduce your science data,
in addition to the following input frames:MASTER_BP_MAP, MASTER_FLAT_LAMP, WAVE_MAP, DISTORTION, SLIT_POS, SLITLET_DISTANCES.

The main output file will be the coadded reconstructed cube (PRO.CATG=COADD_STD or COADD_OBJ).

At this stage, you can now use your preferred reduction package to extract the spectrum of interest for your science. A useful way to both visualize the 3D cube and extract the spectrum of any part of the cube, is to use QFitsView.

3.6 Tips to improve sky subtraction for faint targets

You should refer to section (11.1.19) and (11.1.20) of the SINFONI Pipeline User manual in order to investigate how to best optimize the sky subtraction step in your SINFONI data. By default, the "auto_jitter" parameter in the **sinfo_rec_jitter** recipe is set to "1", which means that the sky closest in time will be subtracted from the individual science frames. If you prefer to subtract the sky independently using your own reduction techniques, then you can set this parameter to "0" and no sky will be subtracted. If you have no sky frames available, but your object is not extended, you can try to estimate the sky by taking the median of all your science frames (as long as you have jittered your object sufficiently within the field) and will then need to set "auto_jitter" to "2".

If your target is faint and best sky subtraction is a requirement, you should consider activating the "objnod_scale_sky" parameter when running the **sinfo_rec_jitter** recipe. This parameter allows you to subtract the median value of each slice of your cube (i.e. for each wavelength in your cube) in order to remove any sky subtraction residual. This is useful in the case your object is not extended. Otherwise, you will need to estimate the sky residual differently prior to remove it.

Note that if you want to use the benefit of the extended sky residual correction provided by the pipeline, which models the level of emission from the atmospheric OH lines, you will need to set the parameter "product_density" to "2". Otherwise this feature is ignore by the pipeline. Please, refer to section (11.1.20) of the SINFONI Pipeline User manual for a full description of the sky residual correction.

3.7 Moving targets

In the case your target has a proper motion, which is not compensated by the telescope tracking or AO system, you will will need to co-register your individual spectral cubes using the Esorex command line **"sinfo_utl_cube_combine"** to take into account the proper motion of your object. This is typically the case for the observation of a faint solar system object in AO mode, while the AO system use a different source (nearby bright source or other brighter moving target within the vicinity of the main science target) to sense the wavefront. You can follow the instruction provided in the section (10.13) of the SINFONI Pipeline User manual to combine cubes obtained in this particular observing case.

A Bad line removal

You will find below an example of a script written in IDL, whose purpose is to identify the detector lines in the original data array where correction of the hardware processing needs to be applied before starting reducing the data.

```
; *************************************************
; input paramaters
dir='./'
dir_in=dir+'raw/'
dir_out=dir+'cleaned/'
infile='list.dat' ; File list should be in the dir_in "raw directory" where the raw files
dim=2048
width=4
file=''

; Reasonnably good value to identify most of the deviant background pixels
nsigmaback=18

; build the template mask for non illuminated edge pixels (background pixels)
mask_back=replicate (0, dim, dim)
mask_back(0:width-1,width:dim-width-1)=1
mask_back(dim-width:dim-1,width:dim-width-1)=1
; define subscripts of background pixels
backpos=where(mask_back eq 1)
; define y pos of back pixels
ybackpix=backpos/dim

; read raw data files and produce corresponding cleaned image
openr,unit,dir_in+infile,/get_lun
while not eof(unit) do begin
; read input file
    readf,unit,file
    im=readfits(dir_in+file,head)
; read input file; save unmodified images
    im1=im
; reset the background mask
    backpix=im(backpos)
; search for back pixels too deviant
    diffbackpix=backpix-median(backpix)
    sigmaback=stdev(median(diffbackpix,3))
    bad=where(abs(diffbackpix) gt nsigmaback*sigmaback,cntbad)
; if no bad pixel, do nothing
    if cntbad le 0 then goto,jump
; define background median value of good back pixels
    ybad=ybackpix(bad)
    good=where(abs(diffbackpix) le nsigmaback*sigmaback)
```

```
    medvalue=median(backpix(good))
    yprev=-1
; cycle through rows containing bad back pixels and correct them
    for k=0,cntbad-1 do begin
        yval=ybad(k)
        if yval eq yprev then goto,skip
        yprev=yval
        kline=[im(0:width-1,yval),im(dim-width:dim-1,yval)]
        kline_mean=total(kline)/(2.*width)
        im1(width:dim-width-1,yval)=im(width:dim-width-1,yval)+kline_mean-medvalue
        skip:
    endfor
    jump:
; save cleaned file
    writefits,dir_out+file,im1,head
endwhile
free_lun,unit

end
; *****************************************************
```

___oOo___