STATISTICAL IMAGE PROCESSING IN THE VIRTUAL OBSERVATORY CONTEXT

Mireille Louys[1], Francois Bonnarel[2], Andre Schaaff[2], and Cyril Pestel[2]

[1]Laboratoire des Sciences de l'Informatique, de l'Image et de la Teledetection, UMR 7005
[2]Centre de Donnees astronomiques de Strasbourg, Universite de Strasbourg, UMR 7550

ABSTRACT

In an inter-disciplinary collaborative project, we have designed a framework to execute statistical image analysis techniques for multiwavelength astronomical images. This paper describes an interactive tool, `AIDA_WF` , which helps the astronomer to design and describe image processing workflows. This tool allows designing and executing processing steps arranged in a workflow. Blocks can be either local or remote distributed computations via web services built according to the UWS (Universal Worker Service) currently defined in the VO domain. Processing blocks are modelled with input and output parameters. Validation of input images content and parameters is included and performed using the VO Characterisation Data model. This allows first checking of inputs prior to sending the job on remote computing nodes in a distributed or grid context. The workflows can be saved and documented, and collected as well for further re-use.

Key words: methods: statistical data analysis, data analysis workflows; Virtual Observatory protocols.

1. INTRODUCTION

There are various use cases in the astronomical data analysis domain where workflows can be successfully used in order to properly assemble processing blocks on massive observational data, launch them on distributed architecture and save the resulting graph and associated information for further re-use.

As an example , Fig. 1 shows a very simple 3-steps chain designed for the detection of Low Surface Brightness galaxies in a large field of the Virgo Cluster by the INT survey.

- Pixel value classification applied on a bunch of multiband images provides a segmentation map; The segmentation algorithm, based on statistical markovian analysis is well adapted to detection of low signal to noise ratios.

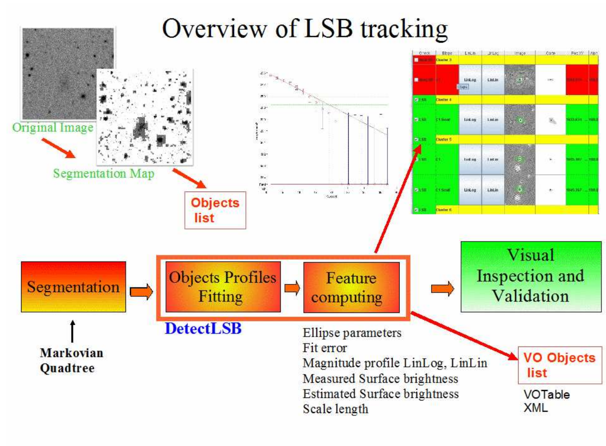

Figure 1. Example of a pipeline for image processing aiming at detection of Low Surface Brightness galaxies.

- this map and the input images set are fed in the *DetectLSB* processing block ; for each region of the map, the program fits a best ellipse shape and derives an averaged luminosity profile (Louys, 2008). LSB galaxy candidates are then sorted out and listed in LSB and non LSB sets of sources.

- The results lists can be browsed using a java program the '*LSBExplorer*' browser that shows for each candidate a thumbnail image, the various profiles and measurements derived.

This scenario can be modified , by replacing the last block by a filtering script, for instance.

2. GRAPHICAL DESIGN OF THE WORKFLOW

`AIDA_WF` stands for Astronomical Image Data Analysis Workflow Framework. This tool helps saving experience acquired about image data analysis and allow complex iterative methods to be fully described, saved and then become reproducible. The user can define its own

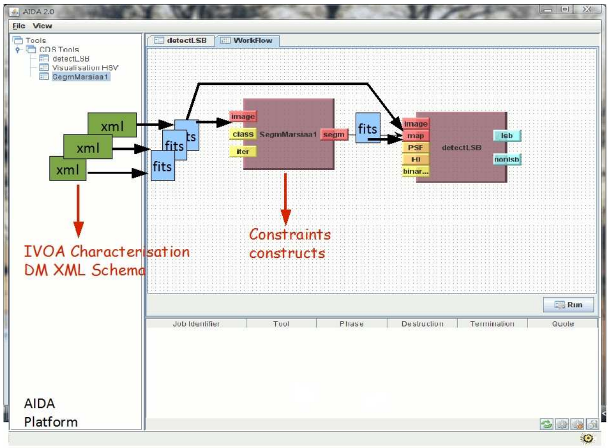

Figure 2. A possible AIDA-WF workflow example for Low Surface Brightness galaxies detection with 2 blocks: 'SegmentationMarsiaa' and 'DetectLSB'. Input images are provided along with their physical axis description as XML files compliant to the IVOA Characterisation Data Model. Rules corresponding to the Segmentation block are applied to these XML files in order to check the validity and compatibility of input images' content.

processing blocks and connections interactively via the `AIDA_WF` client. The workflow is represented as a set of tasks, chained and/or piped together via logical connections as shown in Fig. 2. Each processing block used in the system is described by a small set of metadata: the goal of the tool, input files types and names, simple parameter values, output results as values and files etc. When observation files are used as input , the physical characterisation of the observation content can be given too. For every observation data set published in the Virtual Observatory (VO), a minimal set of metadata expressing the physical properties of the observed signal can be provided. Such metadata lists are organised according to standardised VO data models like the Characterisation Data Model (IvoaDataModelWG, 2007), the Spectrum Data Model (McDowell, 2007), etc., using XML serialisation or "(Utypes,values)" lists of metadata records. Utypes are standard identifiers derived from the data model structure that can be easily combined in table or list structures.

3. PARAMETER VALIDATION

The execution of each processing block is organised and checked by the workflow job execution server (AIDA server). For very long pipelines or iterations, checking for the validity of input observation files helps to detect inadequate parameters or incompatible input data prior to workflow launching. In addition to simple parameters complex observation metadata can be checked too, allowing for instance in the LSB galaxy detection case referenced above, to check that input multiband images have compatible reference position, size and spatial resolution.

We have defined a simple language to express constraints on metadata values, using the utype syntax derived from the characterisation DM. (cf. (Mantelet, 2007)). Fig. 3 shows a set of constraints to be verified for the workflow described in Fig. 1 for the execution of the segmentation step prior to LSB detection.

4. EXECUTION

The execution of the graph tasks is passed by `AIDA_WF` client to the `AIDA_WF` server (JOB execution server) which performs it either locally or remotely on a PC cluster. In our specific LSB galaxy detection example, the segmentation step and the DetectLSB steps are performed remotely on the CDS PC cluster (10 CPU). The task launching, the execution follow-up, logs and results files are made available by a web service compliant to the UWS VO protocol.

5. INTERACTION WITH VO SERVICES VIA VO PORTALS

While building up the workflow graph, and feeding input files to various processing blocks, the user can select input observation files via VO portals. Such portals, like ALADIN (Bonnarel, 2000), (Fernique, 2009) or **VOExplorer** offer access to VO services containing large data sets or archives to the common user.

Data selected via ALADIN, can be checked and visualised at first in ALADIN, then passed on to `AIDA_WF` design tool using the PLASTIC protocol (IvoaApplicationWG, 2006). This mechanism allows to simply propagate data files pointers from one VO tool to another, according to a standardised message passing method. The metadata associated to selected observations fed into the workflow as shown in Fig. 2 is browsable via the CAMEA visualisation tool (Gassmann, 2007). At the other end, results produced by the workflow like images, spectra, tables, etc... can be browsed and compared with VO tools in the same manner. Figure 4 shows the selected images in the ALADIN pile (at the right) and the segmentation map (on the left). The file references are passed to the `AIDA_WF` tool in a broadcast/receive mechanism via the PLASTIC hub. The current tool in the AIDA session, here SegmentationMarsiaa, can reuse the file names as input. AIDA's bottom panel represents the status of the tasks with the metadata associated following the UWS protocol.

The results can be accessed via VO portal too as shown in Fig. 4.

```
#Constraints on input image files content prior to the Marsiaa
# Segmentation processing block
#
AxisShortcut SPATIAL: Axis[ucd="pos"]
AxisShortcut SPECTRAL: Axis[ucd="em"]
AxisShortcut FLUX: Axis[independentAxis="false"]
IF (NOT EQUAL(:SPATIAL.Coverage.location.coord_system_id)
 AND NOT EQUAL(:SPATIAL.coordsystem))
WARNING ("Input images do not have identical coordinate systems")
FI
#0.check units on all spatial axis
 IF (EXISTS(:SPATIAL.Coverage))
EQUAL(:SPATIAL.Coverage.location.unit) OR
EQUAL(:SPATIAL.Coverage.unit) OR
EQUAL(:SPATIAL.unit)
 FI
# 1. check for same resolution or sampling among input images
IF (EXISTS(:SPATIAL.Resolution))
NEAR(:SPATIAL.Resolution.resRefVal.Resolution, 0.3)
# NEAR(:SPATIAL.Resolution.resRefVal.Resolution2.C2, 0.3)
EQUAL(:SPATIAL.Resolution.unit) OR EQUAL(:SPATIAL.unit)
ELSE
NEAR(:SPATIAL.SamplingPrecision.sampRefVal.sampPeriod.C1, 0.3)
NEAR(:SPATIAL.SamplingPrecision.sampRefVal.sampPeriod.C2, 0.3)
EQUAL(:SPATIAL.SamplingPrecision.unit) OR EQUAL(:SPATIAL.unit)
FI
# 2. Check for identical sizes for images
IF (EXISTS(:SPATIAL.numbins))
EQUAL(:SPATIAL.numbins)
ELSIF (EXISTS(:SPATIAL.numbins2))
EQUAL(:SPATIAL.numbins2.i1) AND EQUAL(:SPATIAL.numbins2.i2)
ELSIF (EXISTS(:SPATIAL.numbins3))
EQUAL(:SPATIAL.numbins3.i1) AND EQUAL(:SPATIAL.numbins3.i2) AND
EQUAL(:SPATIAL.numbins3.i3)
ELSE
ERROR( "Input images do not have the same sizes" )
FI
# 3. check that all input images overlap
EQUAL(:SPATIAL.Coverage.location.unit) OR
EQUAL(:SPATIAL.Coverage.unit) OR
EQUAL(:SPATIAL.unit)

EQUAL(:SPATIAL.coordsystem)
EQUAL(:SPATIAL.Coverage.location.coord)
# 4. Check for range of observed signal
EQUAL(1[]:FLUX.coverage.bounds.unit) OR
EQUAL(:SPATIAL.Coverage.unit) OR
EQUAL(:SPATIAL.unit)
IF (1[]:FLUX.bounds.limitHi - 1[]:FLUX.bounds.limitLo
>= 100)
WARNING("(Observables: max-min <100)
Normalisation required before processing ")
FI

# 5.Check for spectral bounds overlap between input images
#IF ((1[1]:SPECTRAL.bounds.limitHi > 1[2]:SPECTRAL.bounds.limitLo) OR
(1[2]:SPECTRAL.bounds.limitHi > 1[1]:SPECTRAL.bounds.limitLo))
IF (1[1]:SPECTRAL.coverage.bounds.limits.CoordScalarInterval.HiLimit >
1[2]:SPECTRAL.coverage.bounds.limits.CoordScalarInterval.LoLimit AND
1[1]:SPECTRAL.coverage.location.coord.Spectral.Value <
1[2]:SPECTRAL.coverage.location.coord.Spectral.Value )
WARNING("(Waveband overlap between input images :
check for correlation )")
FI

IF ( 1[2]:SPECTRAL.coverage.bounds.limits.CoordScalarInterval.HiLimit >
1[1]:SPECTRAL.coverage.bounds.limits.CoordScalarInterval.LoLimit AND
1[2]:SPECTRAL.coverage.location.coord.Spectral.Value <
1[1]:SPECTRAL.coverage.location.coord.Spectral.Value )
WARNING("(Waveband overlap between input images : check for correlation)")
FI

#EQUAL(:FLUX.ucd)
#1[]:FLUX.bounds.extent < 100

# MASK: check for the second entry , that is the segmentation mask
IF (EXISTS(2))
WARNING("Mask entry is provided !")
# 1. ObservableAxis: min=0, max=1:
2:FLUX.ucd = "meta.code.class"
2:FLUX.unit = "unitless"
2:FLUX.coverage.numbins = 2 AND 2:FLUX.bounds.limitLo = 0 AND
2:FLUX.bounds.limitHi >= 0

# 2. SpatialAxis: numbins = numbins INPUT:
# line 8 !
FI
```

Figure 3. An example of a constraints file attached to the Segmentation block. It checks the content of the XML characterisation files attached to input data files. Here these rules verify that for all input images, size and resolution values match properly. Overlap in wavelength coverage is checked too in order to discard cases where the bands used present too much correlation which is not appropriate for the segmentation algorithm in place.

6. EXTENDING THE PROCESSING UNITS COLLECTION

The collection of available blocks tools in the `AIDA_WF` graph designer is still small but can be customized and extended by any developer. Adding new processing blocks only requires to reference a directory containing the tool executable binary file and a tool description ASCII file using a very simple syntax.

7. CONCLUSION

Defining workflows helps for computer demanding applications with massive and/or repetitive computations. Tuning and testing the processing blocks arrangements, running them on distributed architectures and finally saving the design for further re-use, helps to trace back and integrate experience. Such prototypes like `AIDA_WF` offers a platform to explore the development of workflow managing tools in the interoperability framework of the Virtual Observatory.

REFERENCES

Fernique P. et al 2009, Proceedings of Astronomical Data Analysis Software and Systems XIX, ASP Conference Series, in press

Mantelet G. 2007, CDS internship report: Processing blocks description

Gassmann B. 2007, CAMEA, http://eurovotech.org/twiki/bin/view/VOTech/CharacEditorTool

IVOA Data Model WG 2007, Data Model for Astronomical DataSet Characterisation, http://www.ivoa.net/Documents/latest/CharacterisationDM.html

IVOA Note: PLASTIC - a protocol for desktop application interoperability, http://www.ivoa.net/Documents/latest/PlasticDesktopInterop.html

McDowell J. 2007, IVOA Spectral Data Model, http://www.ivoa.net/Documents/latest/SpectrumDM.html

Bonnarel F. et al. 2000, Astron. Astrophys., Suppl. Ser., 143, 33-40.

Louys M. et al. 2008, Proceedings of Astronomical Data Analysis Software and Systems XVII, ASP Conference Series, Vol 394,R.W Argyle, P.S. Bunclark, J.Lewis , eds.

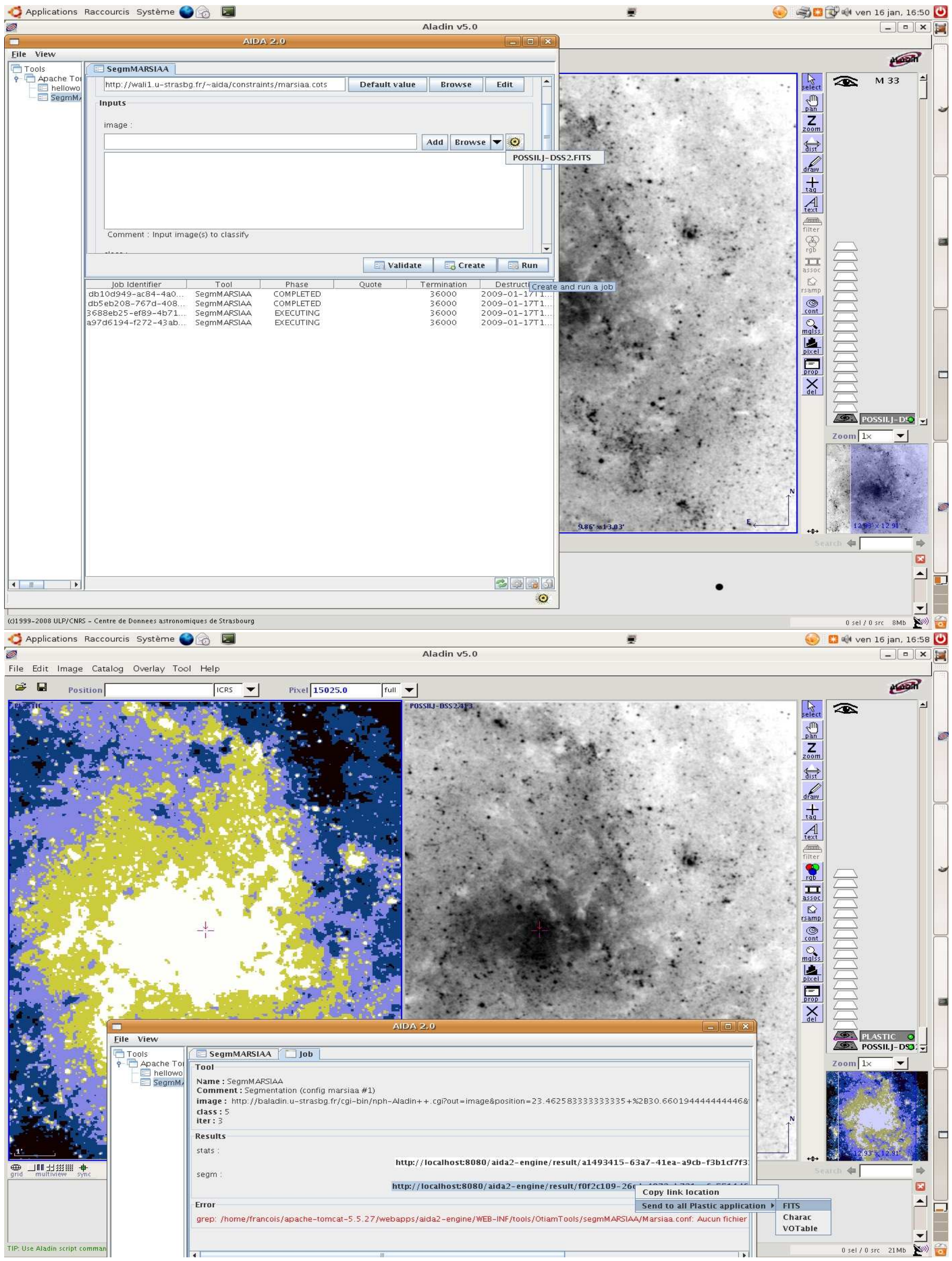

Figure 4. *Top screen: AIDA-WF workflow execution example. Data are selected within the ALADIN portal and broadcasted via PLASTIC to the AIDA interface. The execution's follow-up is shown in the central bottom panel with the tasks identifier and status, and all metadata provided within the UWS protocol. Bottom screen: Exploring results during the workflow execution. ALADIN gets the url of the results file from AIDA-WF via Plastic (blue highlighted line) and displays both the original image, on the right panel, and the segmentation map on the left one.*