Visualisation Interface
to the
Virtual Observatory

VisIVO


*Claudio Gheller (CINECA)[1],*

*Ugo Becciani (OACt)[2], Marco Comparato (OACt)[3] Alessandro Costa (OACt)[4]*


# VisIVO, an open source, interoperable visualization tool for the Virtual Observatory


*1: c.gheller@cineca.it*
*2: ube@ct.astro.it*
*3: mcomp@ct.astro.it*
*4: acosta@ct.astro.it*
*http://visivo.cineca.it*


INAF

CINECA
Consorzio Interuniversitario

The VOTech Project

# Part 1. Overview

# *VisIVO* first steps

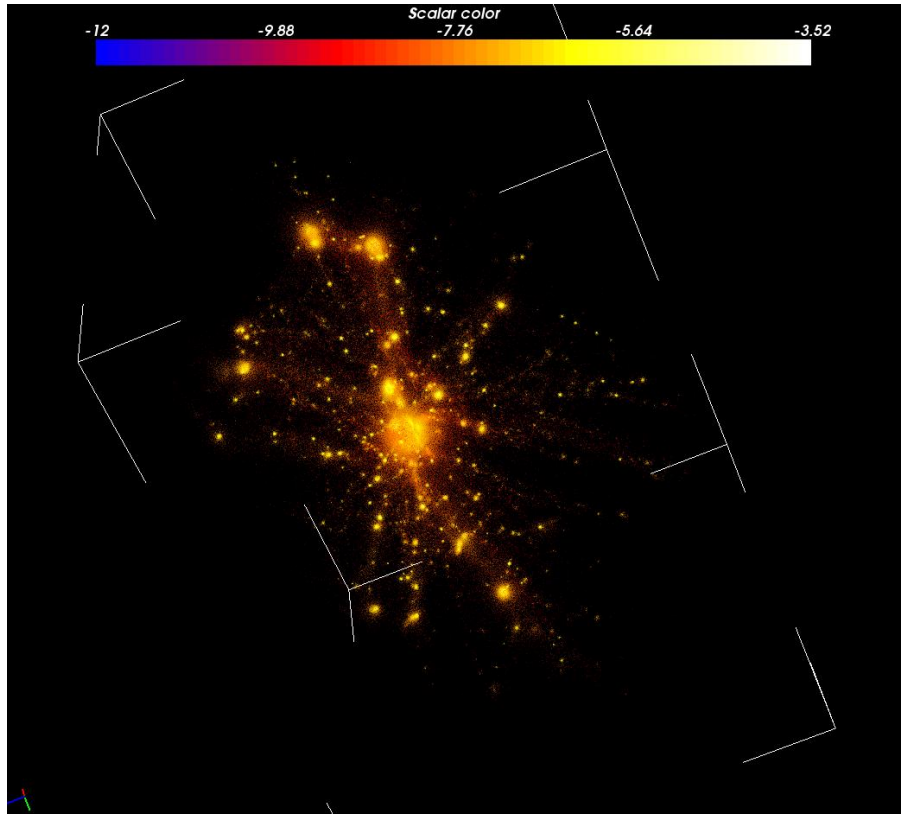
**VisIVO** was born as a visualisation package developed in collaboration between **INAF** (Catania Astrophysical Observatory) and **CINECA** (the largest italian academic supercomputing center) with the specific object of supporting the visualization of astrophysical simulations data.

**Run simulations**

**Download files**

```
Snapshot_z0.0_Lambda.hdf
Snapshot_z0.1_Lambda.hdf
…
Snapshot_z1.0_Lambda.hdf
```

**Visualise results**



# *VisIVO*: Visualisation Interface for the Virtual Observatory

Now it is a tool to interact both with **simulated** and with **observational** data, stored both **locally** and on the **Virtual Observatory**

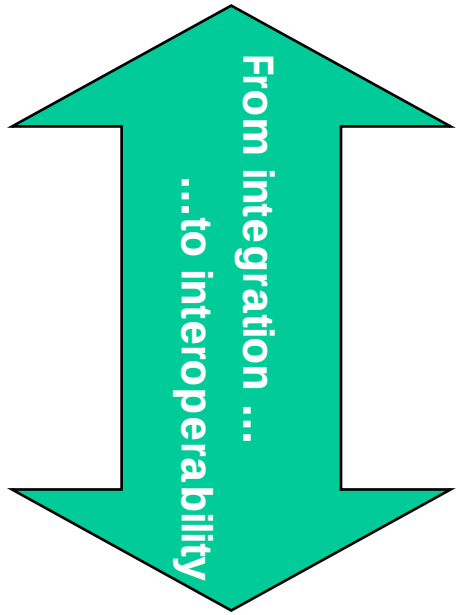
The evolutionary path (last year work):

Integration of data analysis algorithms

Implementation of FITS and VOTables support

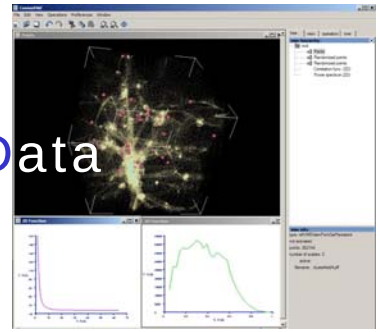
Access via Web Services to remote databases (VizieR)

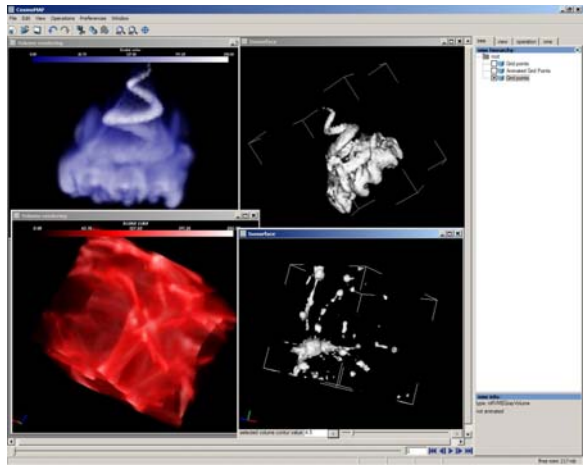
Interaction with external tools (Aladin, AstroNeural)



# *VisIVO* a **interoperable** tool

Integrated Data Analysis Tools





Standard formats (FITS, VOTables, HDF, Raw Binaries) integrated support

AstroNeural data mining neural network tool

## *VisIVO Core* Team

**INAF – Astrophysical Observatory of Catania**

*Becciani Ugo* (permanent staff)

Comparato Marco (young fellow)

*Alessandro Costa* (permanent staff)

**CINECA**

*Claudio Gheller* (permanent staff)

*Anna Guidi* (grant)

*MAF – STAFF*

## Contributions

***Inaf-OACt and CINECA*** support

***VO-TECH:*** EU – FP6 Specific Support Action (DS6) Integration of new technologies into the European Virtual Observatory

## Collaborations

***University of Napoli and Salerno:*** Data Mining - AstroNeural

***CDS:*** Interaction with VizieR and Aladin

***Trieste Observatory:*** Theoretical data

***OPEN TO ANY POSSIBLE COLLABORATION !!!***

# *VisIVO* in one slide

Visualization of N-dimensional data (N > 2)

Open Source

MS Windows (porting to Linux in progress)

**Present Release**: 0.1

**Downloads**:

From web site (binaries only)

http://visivo.cineca.it

From CVS

CVSROOT =
:pserver:anonymous@sirio.cineca.it:/server/project/cvsrepos

PASSWORD: no password required

Checkout module openMAF

Checkout module cosmoMAF

**Software requirements**:

MS Visual C++ 6

CMake 1.6 (patch 7)

**Libraries**:

VTK 4.2 (graphics)

wxWindows (GUI)

CFITSIO (Fits files)

HDF5 (HDF Files)

VO India C++ lib (VOTables)

Xalan, Xerces (XML)

Axis C++ (WS)

**Documentation**:

User Guide (almost ready)

On-line (almost ready)

Developer Guide (who knows...)

Examples (in the distribution)

**Contributions and collaborations**:

...are welcome!!!

# Part 2. Technical description

# *VisIVO* overview

Completely based on **open source** products.

Exploitation of the results of 2 EU funded projects:

- **Multimod** → **MAF** → **Basic Visualisation technology and software architecture**

  (http://www.tecno.ior.it/research/biomechcomp/projects/multimod/mm_home_page.html)

- **Cosmo.Lab** (http://cosmolab.cineca.it)

**AstroMD** → **Astrophysics specific tools (readers, analysis algorithms…)**

# *The Multimod Application Framework* **(MAF)**

Introduction: (from the web site, http://www.cineca.it/B3C/MAF/)

The Multimod Application Framework (MAF) is an open source freely available framework for the rapid development of applications, based on the ***Visualization Toolkit*** and other specialised libraries. It provides **high level components** that can be easily **combined** to develop a vertical application in different areas of scientific visualisation.

Developed by the Visualization Team of Cineca, the University of Bologna and "Istituti Ortopedici Rizzoli"

Special thanks for the continuous support and help to Cineca Visualization Team experts Cinzia Zannoni, Silvano Imboden, Marco Petrone and Paolo Quadrani.

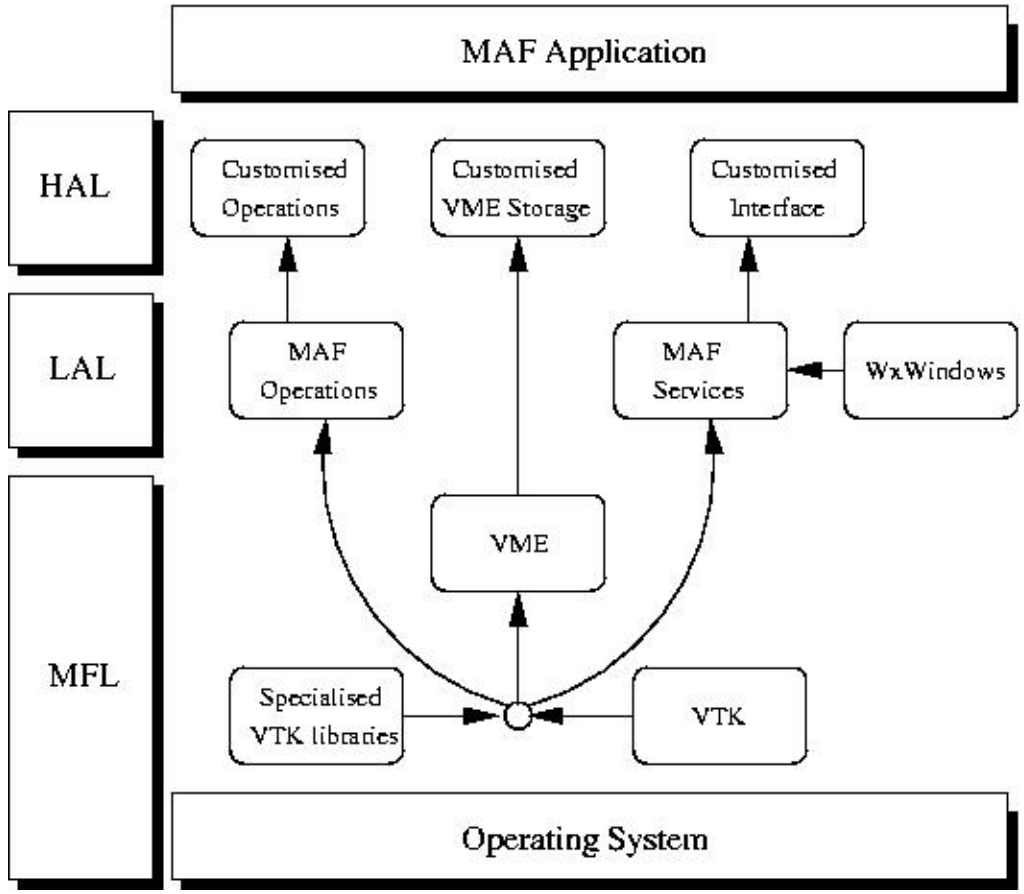
# The MAF architecture

VisIVO has been developed as a MAF application. Its architecture can be summarized according to the following schema:

**High Abstraction Level**: specific of the user application (VisIVO)

**Low Abstraction Level**: general purpose high-level services

**Multimod Foundation Layer**: provides basic software components



# The Visualization ToolKit (VTK)

The graphics functionalities of VisIVO are based on VTK

- VTK is an Open Source tool developed by Kitware inc.(http://www.kitware.com): 3D computer graphics.
- VTK can be used with different languages (Tcl/Tk, Java, Python)
- Can be run on Linux (Unix) and Windows (PC and Mac OSX)
- Largely used to implement scientific tools

## Technical Characteristics

- More than 700 C++ classes
- Extended set of 3D Widgets
- Surface Rendering and Volume Rendering
- Data interaction: good efficiency
- Rendering windows: easy use of lights, cameras and actor properties

## The High Abstraction Level (HAL)

The HAL is the specialized part of the code. It consists in:

**libraries,** which add new and specific functionalities to the code.

- Libraries are developed:
  - According to MFL and LAL rules
  - using MFL components
  - using VTK components
  - integrating other libraries

the **driver code**

- defines and initialize the application

The HAL can be thought as specular to LAL, but **external** to the core application. It can be developed with **no worries** about MAF integrity. Finally, part of HAL functions can be added to LAL or MFL.

# VisIVO on the top of MAF

**VisIVO is our HAL.** Its implementation has actually required changes at all levels (part of our needs were beyond expected requirements). In particular:

**MFL:**

- new implementation of VME to support large particle based dataset with associated properties (scalar fields)

**LAL:**

- integration of the new particle VME
- Specific viewer to visualize particles and properties (transparency, colors, glyphs)
- Introduction of permanent GUIs, to change parameters on the fly
- Introduction of interpolated time evolution (in progress)

# VisIVO HAL

However, most of our work is related to the development of specific functionalities. We have focused on the following main topics

**Importers – classes which allows to read data files**

**Database access – dealing with SQL databases (starting now)**

**Web services interaction – dealing with the Virtual Observatory**

**Data analysis functions – manipulation of data to get derived results**

**Data viewers – how to graphically represent data**

# Importers:
# input = file
# output = VME

Classes which allows to read data files in different formats. At present we support:

- **RAW binary format** (particles and grids)
- **HDF** format (subset for particles)
- **FITS** format
- **VOTables**
- **Tipsy** format
- **VTK** Format

Binary dump of memory (C I/O style). Positions followed by N scalars for particles

Support only for particle files. To be extended. **HDF5 libraries imported**

Fully supported. **CFITSIO library imported**

Fully supported. **XALAN and XERCES libraries (for XML parsing) and VOIndia C VOTables library imported**

Operations:

Input = VME (one or more)

Output = VME (one or more)

Classes which allows to handle data files in order to extract meaningful information. At present we have implemented

**Randomizer:** extraction of random subset of data

**Power Spectrum:** calculates of the spectrum of a periodic particles distribution

**Correlation Function:** computes the correlation function of a particles distribution

**Minkowsky functionals:** estimates the geometry and the topology of a particle distribution

**Points splatter:** distributes a particle field on a regular mesh

## Viewers

Input = VME

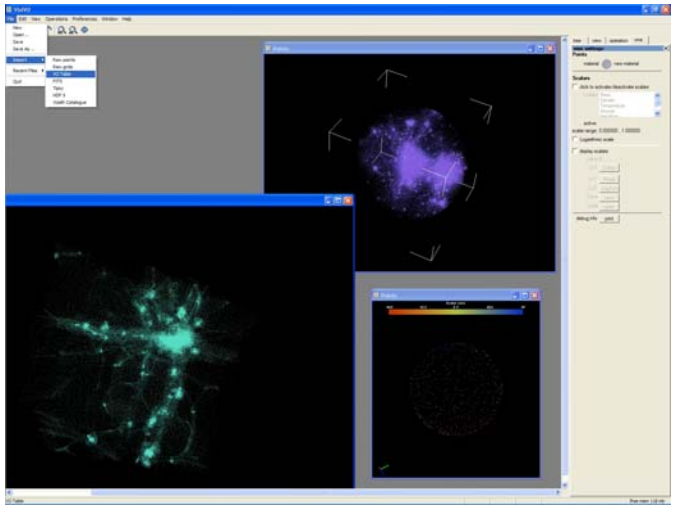
Output = image

Classes which allow to map data to images according to specific algorithms

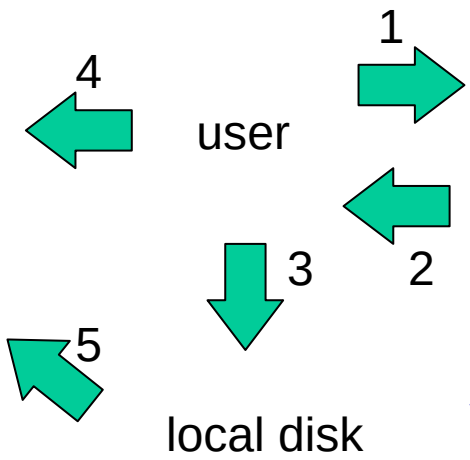
- **Points viewer:** maps **points** to pixels and to glyphs (shapes)
- **Isosurfaces:** draws surfaces that divide **grids** higher and grids lower than a given value
- **Volume rendering:** renders **grids** with colors and tranasparencies (cloud effect)
- **Cutting planes and projections** (in progress)
- **2D graphics:** visualizes curves and plots
- **HTML browser**: for documentation and help pages

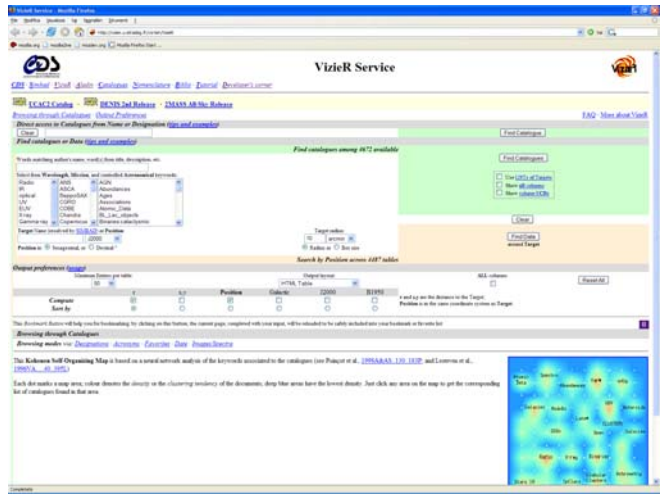
# Part 3. From integration to interoperability

# VisIVO and VizieR: the old way



VisIVO GUI



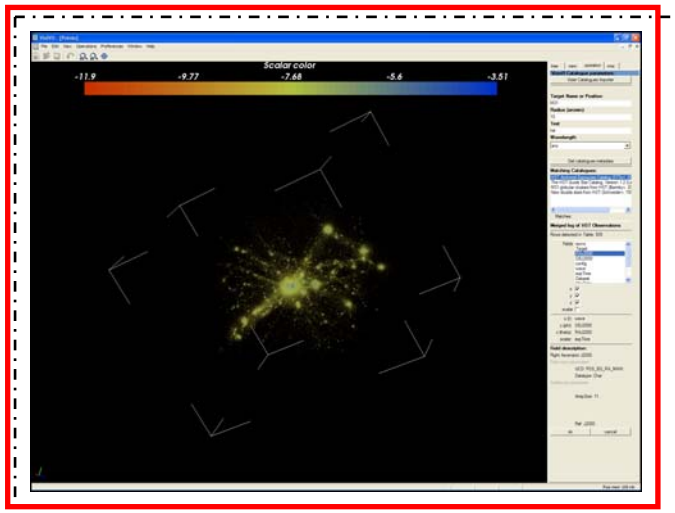




VizieR Service web interface

1. query the service
2. gather data
3. save data to local disk
4. open VisIVO
5. VisIVO loads data from local disk

# VisIVO and VizieR: new way

## Interface




user

1. Query VisIVO for VizieR data

2

3

VizieR Catalogues Web Service

4

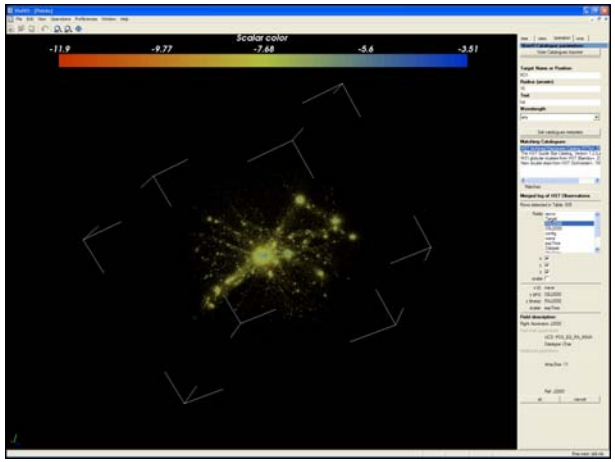
5

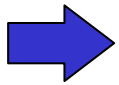
local disk

**Black box**

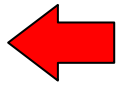
- query the service
- gather data
- save data to local disk
- load data from local disk

# VisIVO and VizieR communication layer




VizieR Catalogues Web Service



Apache Axis is an implementation of the SOAP ("Simple Object Access Protocol") submission to W3C


## why Axis C++?

- C++
- Open Source
- developed for Linux and Windows
- stable
- reliable
- many web services are developed using Axis
- Apache Software Fundation

# VisIVO and VizieR Getting started

- insert the libraries into your project
- obtain the wsdl of the service
- (even if you are on Windows reboot to your favorite Linux distro to …)
- create the stub (classes that describe the service) using the wsdl2ws tool
- (correct errors – probably related to the wsdl)
- create an instance of the service class
- query the service using the methods of the service class

**The WSDL2Ws tool that comes with Axis C++ can be used for generating the following C/C++ source files for a given WSDL file**:

- Server side skeletons and wrappers
- Client side stubs and wrappers

**The generated Wrappers support the following functions**:

- Method invocation
- Serialization
- Deserialization

**Programming Languages**

- WSDL2Ws tool is written in Java and is based on the WSDL tools by Axis Java project
- The tool is capable of generating source code for both C++ and C

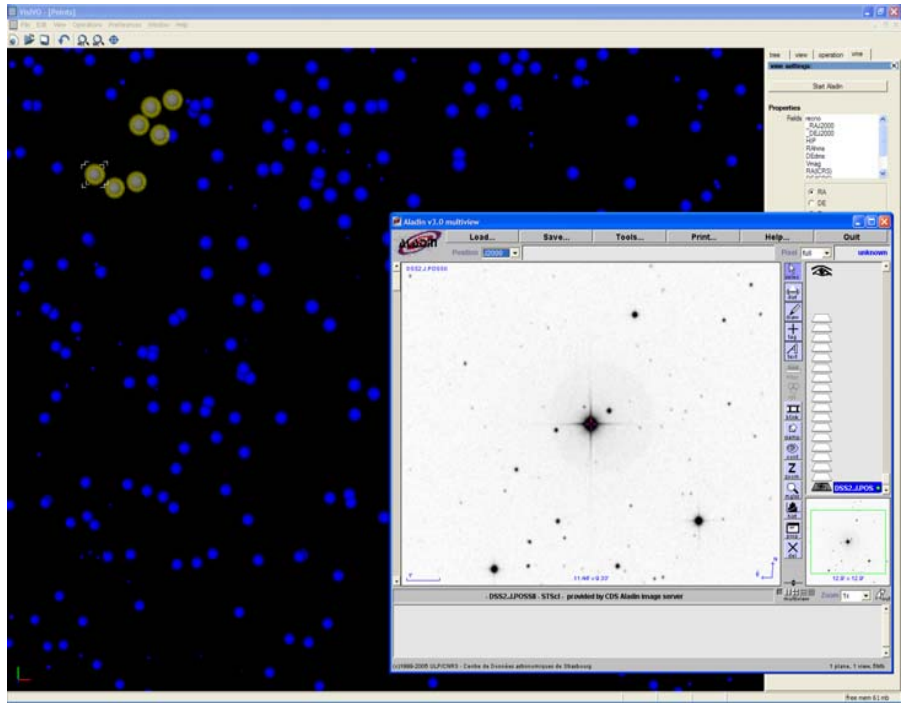
# VisIVO and Aladin

Aladin is an interactive software sky atlas allowing the user to visualize digitized images of any part of the sky, to superimpose entries from astronomical catalogs

Objects can be selected using a picker and passed to Aladin to perform further operations

Aladin can be controlled via commands sent to its standard input

Using a pipe it is possible to interact with it from VisIVO



# AstroNeural

Data mining algorithm, in collaboration with the Universities of Napoli and Salerno, ***AstroNeural*** project.

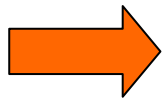
(http://people.na.infn.it/~longo/Ricerca/ASTRONEURAL/astroneural_main.html)

## Aims and applications of AstroNeural:

User friendly tool to perform clustering and data mining in high dimensionality spaces

### Aims

- Clustering & pattern recognition in high dimensionality spaces
- Visualization (VisIVO)
- Classification
- Parametrization of images
- Modeling of massive data sets



### Applications

Astrophysics
Genetics
Geophysics
High energy physics
Atmospheric physics
Etc.

# VisIVO AstroNeural & Aladin: interoperability in action

1 VOTable from sloan catalogue

SDSS

Query

**VOTable data**

2 Astroneural:
photometric redshift estimation
(PPS,NEC clustering)

AstroNeural

**VOTable data**

3 VisIVO: outliers localization

VisIVO

4 Aladin: outliers visualization

Aladin

# VisIVO – Integration in a Data Center