The Radio Astronomer's IT Toolkit

Tara Murphy

ATNF Synthesis Imaging School
30th September 2008

Outline

Scenario: How would you solve this problem?

Your supervisor gives you a data file that they've dug out of the archives. They say that it contains Nobel prize winning data. . . if only you could analyse it. . . you take a look. . .

Scenario: How would you solve this problem?

Your supervisor gives you a data file that they've dug out of the archives. They say that it contains Nobel prize winning data. . . if only you could analyse it. . . you take a look. . .

```
010.1002587 -78.0749976 3.6 4.7 J0000M32 0.000 D
322.2776209 -64.6831876 3.1 3.6 J0000M52 0.108 C
328.0067189 -68.5874347 2.1 2.5 J0000M48 0.193 D
325.6616556 -67.1618942 1.8 2.0 J0000M48 2.539 B
314.1262894 -55.5638273 9.0 2.8 J0000M60 1.266 A
314.6473335 -56.3390663 3.3 3.5 J0000M60 2.522 C
334.7202054 -71.7342040 1.8 2.3 J0000M40 5.693 B
321.3525638 -63.9139186 2.0 2.1 J0000M52 2.080 B
325.7259722 -67.2143813 2.9 3.0 J0000M48 3.311 D
347.0515173 -75.2447016 2.3 3.4 J0000M36 4.345 C
325.5104290 -67.0742957 2.9 3.1 J0000M48 0.105 C
001.9296920 -77.4361767 5.9 6.9 J0000M32 1.530 B
307.3447442 -41.6675454 1.8 2.1 J0000M76 6.400 B
```

. . . and its 10 000 lines long.

What do you do next?

1. Run, screaming, from the room, cursing the astronomers of yesteryear. Then brew a strong coffee and prepare yourself for several days of manual adjustments.

What do you do next?

1. Run, screaming, from the room, cursing the astronomers of yesteryear. Then brew a strong coffee and prepare yourself for several days of manual adjustments.
2. Sigh and take out some old FORTRAN code your supervisor gave you. Comment out 5 lines, uncomment 10 lines, make a few random tweaks, recompile and hope that works.

What do you do next?

1. Run, screaming, from the room, cursing the astronomers of yesteryear. Then brew a strong coffee and prepare yourself for several days of manual adjustments.
2. Sigh and take out some old FORTRAN code your supervisor gave you. Comment out 5 lines, uncomment 10 lines, make a few random tweaks, recompile and hope that works.
3. Spend a few hours remembering Perl and writing a script to reformat it, then sit basking in your own glory, wishing there were other people around to see how brilliant you are.

What do you do next?

1. Run, screaming, from the room, cursing the astronomers of yesteryear. Then brew a strong coffee and prepare yourself for several days of manual adjustments.
2. Sigh and take out some old FORTRAN code your supervisor gave you. Comment out 5 lines, uncomment 10 lines, make a few random tweaks, recompile and hope that works.
3. Spend a few hours remembering Perl and writing a script to reformat it, then sit basking in your own glory, wishing there were other people around to see how brilliant you are.
4. Write a one line Unix script in a couple of minutes, then move on to the Nobel prize winning research. You solve this kind of problem hundreds of times a day.

The solution/s

- Python solution

```python
types = {'A':0, 'B':0, 'C':0, 'D':0}
for line in open('catalogue.txt'):
  cols = line.split()
  dec = float(cols[1])
  type = cols[-1]
  if dec > -70 and dec < -60:
    types[type] += 1

for type in types:
  print type, types[type]
```

- Shell solution

```
> awk '($2>-70 && $2<-60){print $7}' catalogue.txt | sort
| uniq -c | sort
```

The radio astronomer's IT toolkit

- What IT skills do you need to do your research effectively?
  - A data reduction/processing package
    (e.g. Miriad, AIPS++)
  - A FITS visualisation package
    (e.g. kvis, ds9)
  - A range of Un*x tools
    (e.g. cut, paste, grep, sed, awk, for loops)
  - A scripting language
    (e.g. Python, Perl)
  - A plotting package
    (e.g. IDL, matplotlib)
  - Familiarity with accessing large online resources
    (e.g. NED, SIMBAD, Vizier, ADS)
  - Version control software
    (e.g. Subversion)

The radio astronomer's IT toolkit

- Plus sometimes you need to
  - Write software in C, C++, FORTRAN
  - Read other people's code...
  - Query databases using SQL (e.g. SDSS, 6dF)
  - Use a wiki for collaboration
  - Set up a website
  - Set up a website with forms and cgi scripts
  - Set up a database to share/organise your data
  - Use VO Tools for complex queries

{girls, guys} like {girls, guys} who have skills

Why use scripting?

- Advantages vs. manual processing
  - Speed
  - Reproducibility
  - Documentation
  - Collaboration
- Advantages vs. 'real programming'
  - Speed of development
  - Flexibility
  - Easier for a beginner to understand

However. . .


AND A LEAN, SILENT FIGURE SLOWLY FADES INTO THE GATHERING DARKNESS, AWARE AT LAST THAT IN THIS WORLD, WITH GREAT POWER THERE MUST ALSO COME -- GREAT RESPONSIBILITY!

Problem 1

You have reduced your data and now have 200 FITS files sitting on your computer. You want to measure the peak flux in each image.

```
% foreach i (`ls *.fits`)
>   echo $i
>   fits in=$i op=xyin out=$i.xy
>   maxfit in=$i.xy
> end
```

* These examples use tcsh but the principle is the same in other scripting flavours (e.g. bash)

Problem 1

```
1020-5803.fits.xy
Fits: version 1.1 09-Feb-07
There were no blanked pixels in the input
MAXFIT: version 29-Nov-95

Peak pixel    : (65,63,1) = 4.6736E-01

Fitted pixel : (65.21,63.11,1) = 4.5010E-01

...

Coordinate:
  Axis 1: Fitted RA---NCP  =  10:20:15.719
  Axis 2: Fitted DEC--NCP  = -58:03:53.13
  Axis 3:        FREQ-LSR  = 1.86239759E+01 GHz
...
```

Problem 1

You want extract the peak flux and ignore all other output.

```
% foreach i (`ls *.fits`)
>   echo $i
>   fits in=$i op=xyin out=$i.xy
>   maxfit in=$i.xy | grep Peak
> end
1020-5803.fits
Peak pixel   : (65,63,1) = 4.6736E-01
1350-6135.fits
Peak pixel   : (64,66,1) = 8.3779E-01
...
```

Problem 1

You want to fix up the annoying file names (1020-5803.fits.xy).

```
% foreach i (`ls *.fits | sed 's/.fits//g'`)
>   echo $i
>   fits in=$i.fits op=xyin out=$i.xy
>   maxfit in=$i.xy | grep Peak
> end
1020-5803
Peak pixel    : (65,63,1) = 4.6736E-01
1350-6135
Peak pixel    : (64,66,1) = 8.3779E-01
...
```

Problem 1

You want to make a table of your results.

```
% foreach i (`ls *.fits | sed 's/.fits//g'`)
>   fits in=$i.fits op=xyin out=$i.xy
>   echo -n "$i " >> table.txt
>   maxfit in=$i.xy | grep Peak | cut "-d " -f8 >> table.txt
> end
% cat table.txt
1020-5803 4.6736E-01
1350-6135 8.3779E-01
...
```

Everybody stand back, I know regular expressions!

http://xkcd.com

Problem 2

You want to measure the peak flux for each of your sources and print out an annotation file for kvis with circles proportional to the peak flux.

```
import os
for filename in os.listdir('data'):
    # run maxfit on each image
    (rastr, decstr, peak) = maxfit('data/%s' % filename)

    # convert ra and dec to decimal form
    ra = ra2decimal(rastr)
    dec = dec2decimal(decstr)

    # print a kvis annotation file
    print 'circle %f %f %f' % (ra, dec, 0.02*peak)
```

Problem 2

A function to wrap maxfit to run within Python

```python
def maxfit(filename):
    cmd = 'maxfit in=%s' % filename
    (rastr, decstr, cols) = (None, None, None)
    for line in os.popen(cmd):
        cols = line.split()
        if line.startswith('Peak'):
            peak = float(cols[-1])
        elif 'RA' in line:
            rastr = cols[-1]
        elif 'DEC' in line:
            decstr = cols[-1]
    return (rastr, decstr, peak)
```

Problem 2

Some functions to do coordinate conversions.

```
def ra2decimal(rastr):
  r = rastr.split(':')
  ra = (float(r[0]) + float(r[1])/60.0 + float(r[2])/3600.0)*15
  return ra


def dec2decimal(decstr):
  d = decstr.split(':')
  if d[0].startswith('-') or float(d[0]) < 0:
    dec = float(d[0]) - float(d[1])/60.0 - float(d[2])/3600.0
  else:
    dec = float(d[0]) + float(d[1])/60.0 + float(d[2])/3600.0
  return dec
```

Problem 2: putting it altogether

```
import os
def ra2decimal(rastr):
    ...

def dec2decimal(decstr):
    ...

def maxfit(filename):
    ...

for filename in os.listdir():
    (rastr, decstr, peak) = maxfit(filename)

    ra = ra2decimal(rastr)
    dec = dec2decimal(decstr)

    print 'circle %f %f %f' % (ra, dec, 0.01*peak)
```

Problem 2: results

You want to measure the peak flux for each of your sources, calculate the mean for your sample, and print out an annotation file for `kvis` with circles proportional to the peak flux.

```
> python peakflux.py
circle 155.065496 -58.064758 0.009347
circle 207.676467 -61.586053 0.016756
circle 245.046142 -50.888425 0.079280
circle 250.000575 -48.861683 0.065074
circle 260.079975 -35.912969 0.120744
```

Data reduction scripts

- I have written some scripts to go with tutorial 1
  1. `reduce_ngc253_v1.sh`
     The most basic data reduction script.
  2. `reduce_ngc253_v2.sh`
     A slightly more advanced shell script.
  3. `reduce_ngc253.py`
     Python script with more features.
- These are not the only way to do it — they are meant as a guide to get you started
- http://www.physics.usyd.edu.au/~tara/synthesis
- To run them, do

```
> chmod a+x reduce_ngc253_v1.sh
> ./reduce_ngc253_v1.sh
```

- They assume the raw data is in a directory called `data`

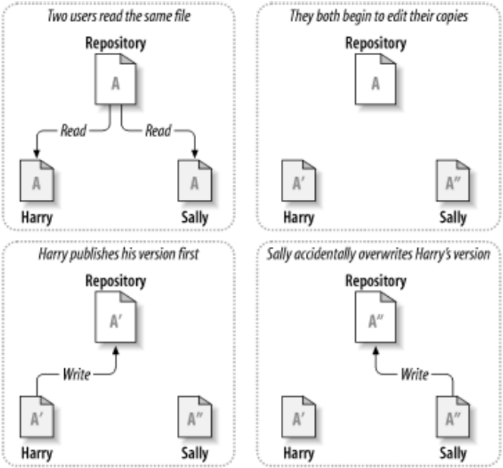
Why should I use version control?

- Works as a constant backup (accessible anywhere at anytime)
- Allows syncing between laptop, work and home desktops
- Allows collaboration on source code, papers, schedule files
- Allows students and supervisors to share code/resources
- Keeps a record of who made changes and why

- Version control works like an e-version of a log book

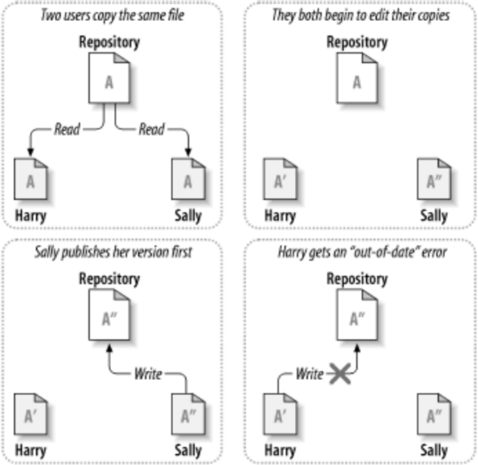
- **It makes you a better coder!**

The problem to avoid



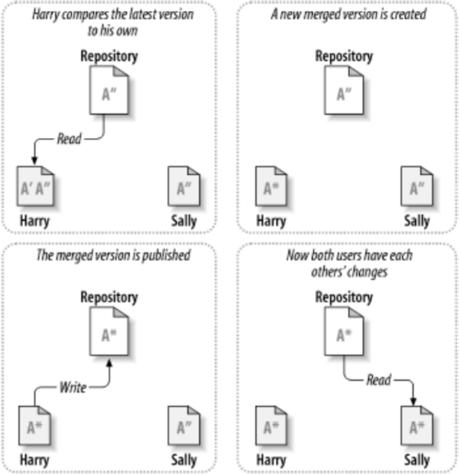
http://svnbook.red-bean.com

The Copy-Modify-Merge solution



http://svnbook.red-bean.com

The Copy-Modify-Merge solution



http://svnbook.red-bean.com

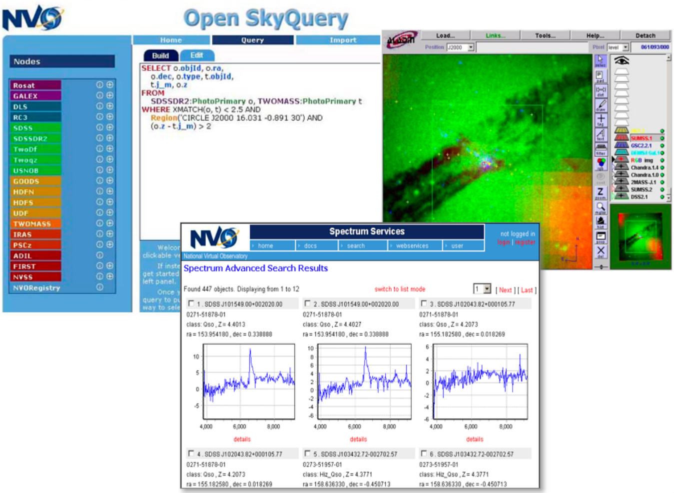
What is the Virtual Observatory?

- We need a way of addressing the data management, analysis, distribution and interoperability challenges of modern astronomy
- The main drivers are
  - Data growth: volume and richness
  - Desire to work online
  - Multi-archive science
  - Large database science

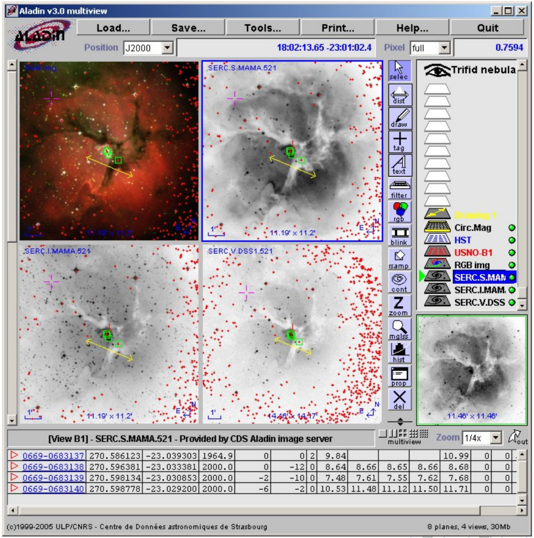
The Virtual Observatory is a distributed collection of

- Data resources
- Software resources
- Computing (grid) resources
- Telescopes

What are VO tools?



An interactive sky atlas: Aladin

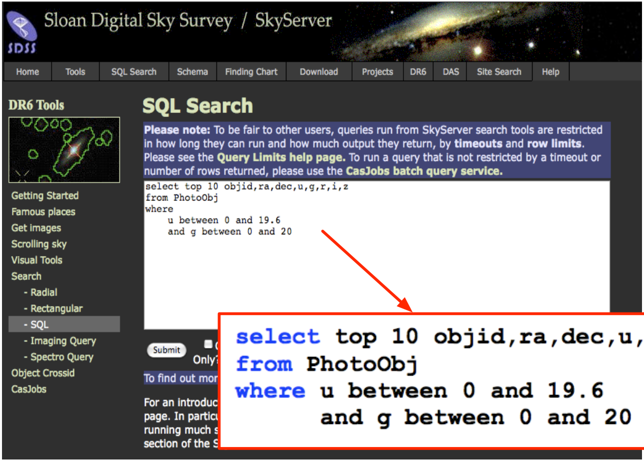


An interactive sky atlas: Aladin

- Visualize digitized astronomical images
- Superimpose entries from catalogues or databases
- Interactively access online data from SIMBAD, NED, Vizier
- Fully VO aware — access other VO resources
- **See demo**
- http://aladin.u-strasbg.fr

- You can also write your own plug-ins (ask Keith Bannister)
- The developers are very keen to get feedback from users — they are happy to make suggested changes!

Querying online databases: SDSS




```sql
select top 10 objid,ra,dec,u,g,r,i,z
from PhotoObj
where u between 0 and 19.6
      and g between 0 and 20
```

http://cas.sdss.org/astrodr6/en/tools/search/sql.asp

Querying online databases: Open Sky Query

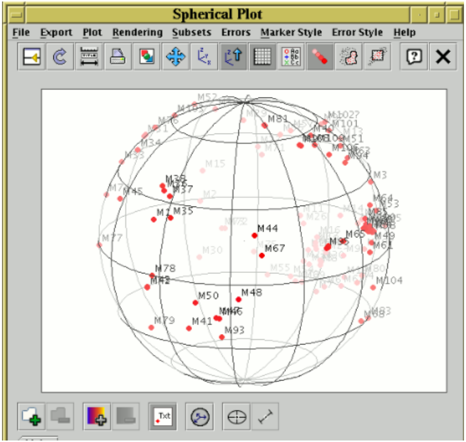
http://openskyquery.net/Sky/SkySite

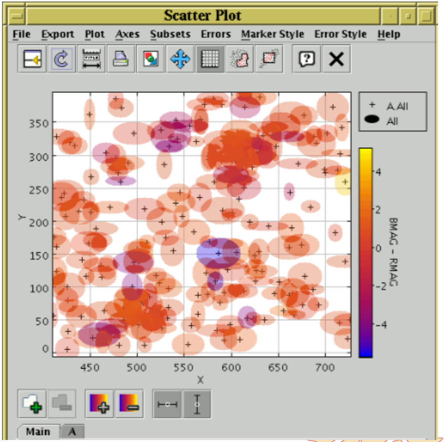
VO enabled plotting

- Many VO tools let you select sources and plot them
- All VO tools allow you to retrieve data as an XML VO table

- TOPCAT is an interactive graphical tool for analysis and manipulation of tabular data
- TOPCAT manifesto: *Does what you want with tables*
- `http://www.star.bris.ac.uk/~mbt/topcat`

VO enabled plotting: TOPCAT





Other tools worth looking at

- DataScope
  heasarc.gsfc.nasa.gov/vo
- SkyView
  http://skyview.gsfc.nasa.gov
- MyADS
  http://myads.harvard.edu
- AstroGrid
  http://www2.astrogrid.org/science
- Google Sky
  http://www.google.com/sky

IT is critical in future astronomy

- IT is becoming increasingly important in ‘everyday’ science
- It is important to learn these skills now!

- Look out for the Astroinformatics Summer School in 2009
- Resources available from the Aus-VO Summer School websites
  http://www.physics.usyd.edu.au/ioa/ausvoss
  http://www.physics.usyd.edu.au/ioa/ausvoss2007