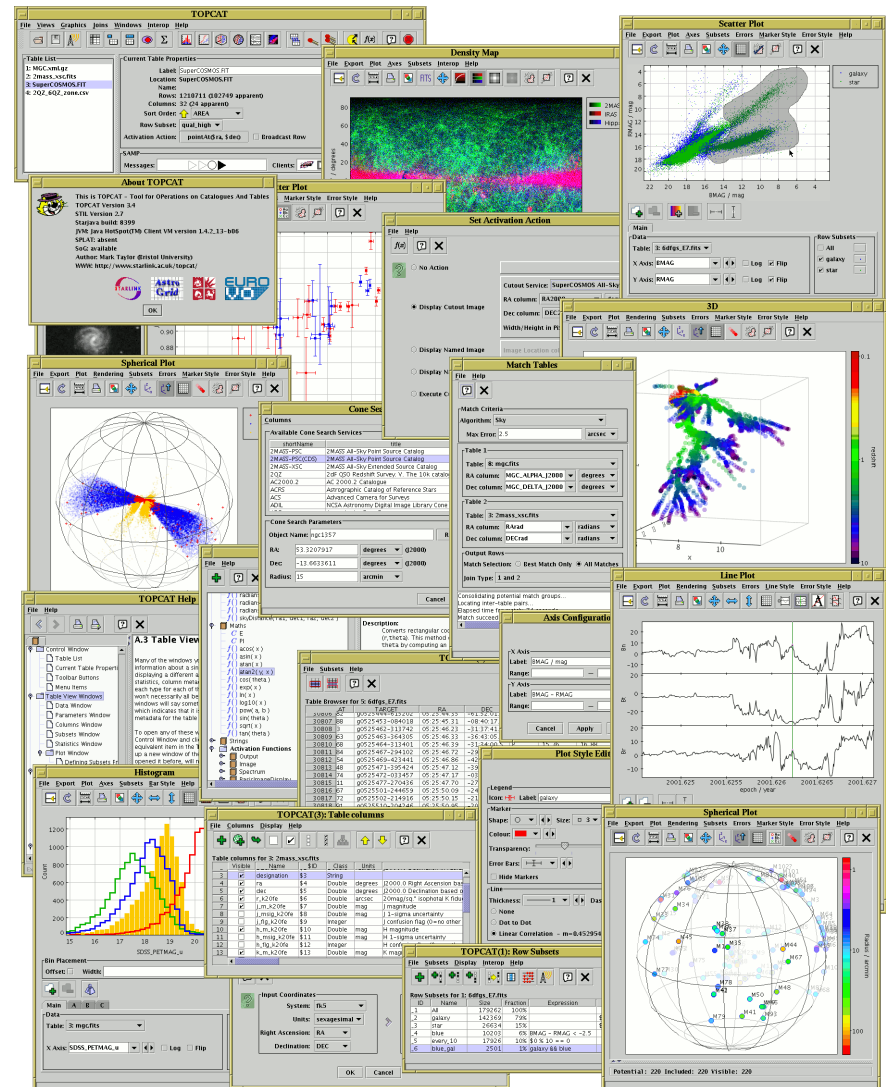# TOPCAT:
Tool for OPerations on Catalogues And Tables

Mark Taylor

Bristol, UK

AG Tagung Short Course

22 September 2011

`$Id: topcat-tagung.tex,v 1.8 2011/09/14 16:24:34 mbt Exp $`

Outline

- Overview
  - What's TOPCAT for?
  - What can it do?
  - High-speed demo

- Details (slides, demos, exercises)
  - Control
  - Load & Save
  - Plotting
  - Crossmatching
  - Virtual Observatory
  - Expression language
  - STILTS (command line)

Web Page: http://www.starlink.ac.uk/topcat/
Tutorial Materials: http://andromeda.star.bris.ac.uk/topcat/tutorial/

Questions welcome

Mission

"Does what you want with tables"

- TOPCAT $=$ Tool for OPerations on Catalogues And Tables

  - Developed in Bristol since $\sim$2003 (Starlink, AstroGrid, Euro-VO, GAVO, . . .)
  - Quite widely used worldwide

- Aims:

  - Easy to use
  - Easy to learn
  - Easy to investigate data — good for exploratory analysis
  - Simple things obvious, complicated things documented
  - Easy to install and run (pure Java — one download file, no library issues)
  - Fast
  - Copes with large data sets ($\sim 10^6$ rows $\times 10^2$ columns)
  - Not format-specific
  - Not necessarily astro-specific
  - User-driven development

Capabilities

It can do:

- Read/write tables in multiple formats
- View/edit data
- View/edit metadata
- Plotting
- Make/combine/display row selections in various ways (linked views)
- Crossmatching — efficient and very flexible
- Access Virtual Observatory (VO) services
- Trigger some event when a row is selected
- Talk to other astro tools (SAMP)
- Calculations

Limitations

It can't do:

- *Really* large tables
- Every ASCII-based format known to man
- Highly configurable publication-quality graphics like IDL/Gnuplot
- Scriptability (though see STILTS)
- Images, spectra, . . . (it's just for tables)

Input/Output

- Table format support:
  - Table file formats:
    - ▷ FITS binary and ASCII tables
    - ▷ ASCII *(but not all ASCII)*
    - ▷ CSV
    - ▷ VOTable
    - ▷ LaTeX *(output only)*
    - ▷ HTML *(output only)*
    - ▷ a few others; extensible
  - . . . making conversion between any of these trivial

- Input sources:
  - Local disk (custom, system or tree browser)
  - Virtual Observatory services (Cone, SIA, SSA, TAP)
  - VizieR
  - Millennium Simulation
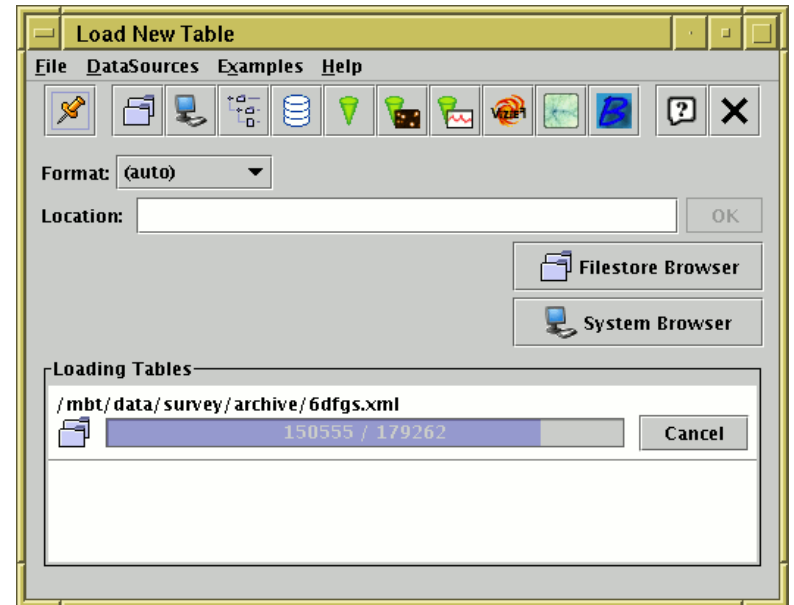 - SQL database
  - a few others; extensible
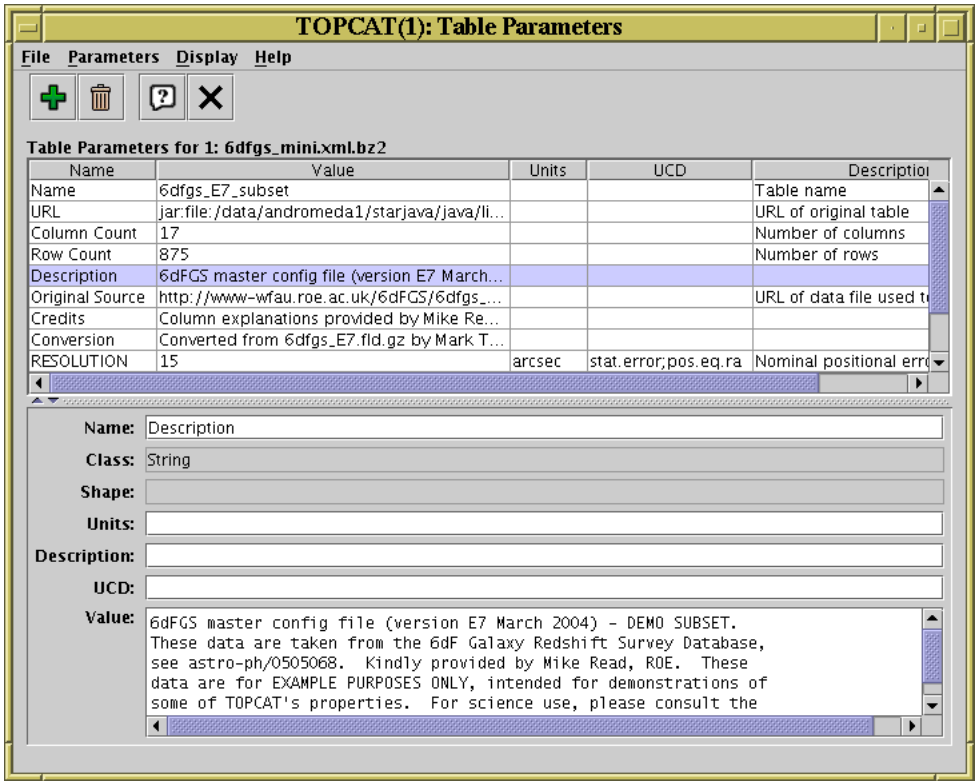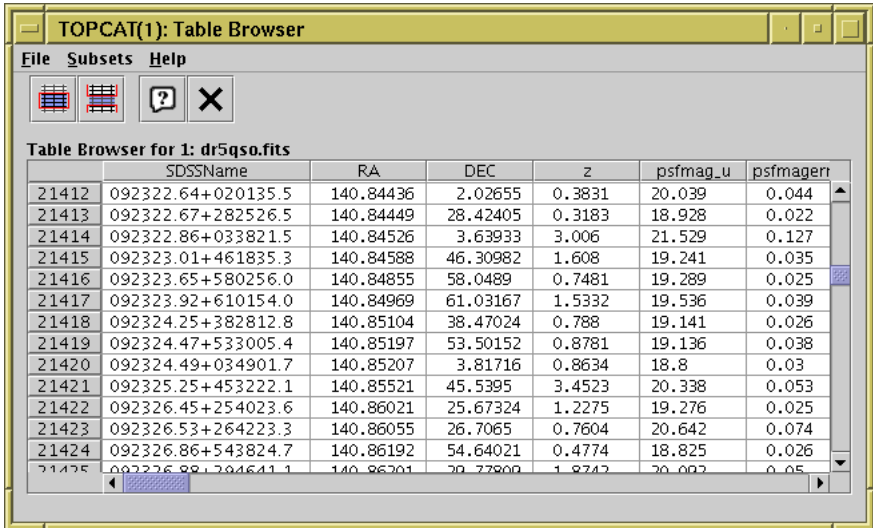
Table Data and Metadata
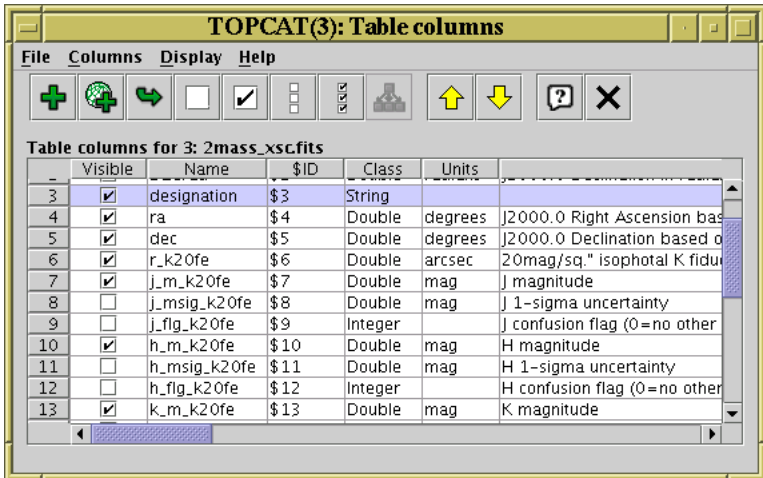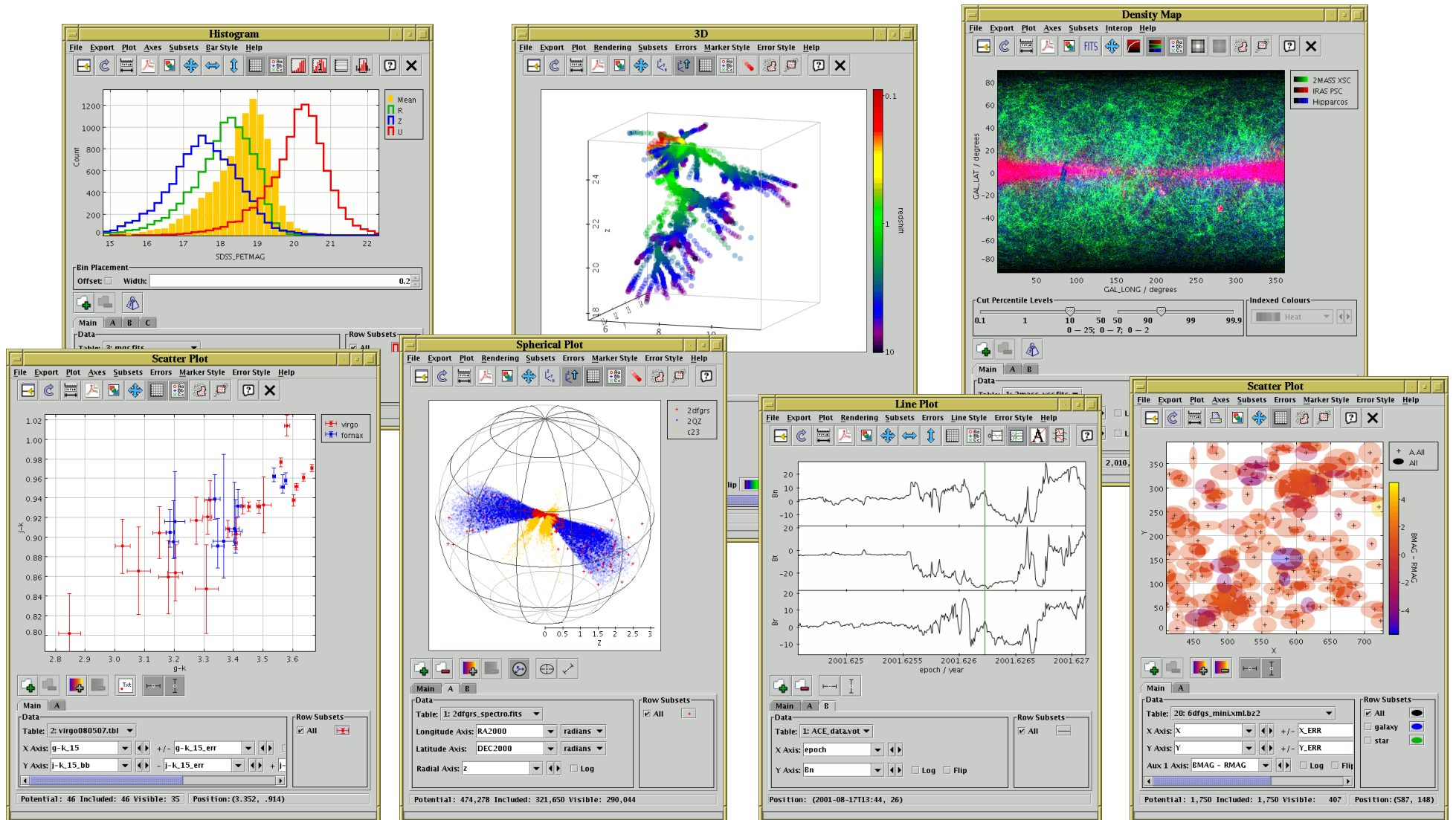
Table data view

Table Metadata view

Column Metadata view

Plotting
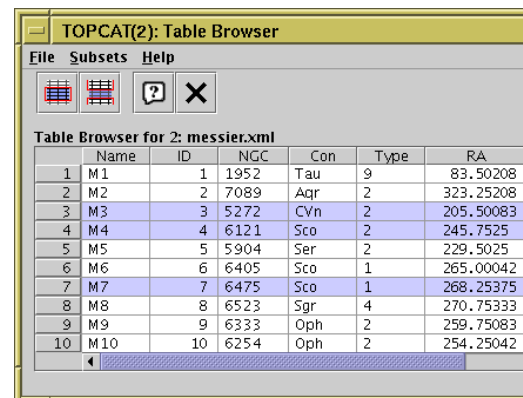
Many options. . .

Row Selections

Different ways to make single or multiple row selections:

- Select point(s) graphically from a plot
- Select row(s) from the table view
- Receive from an external application (SAMP)
- Use an algebraic expression
- Combine existing subsets

Linked views mean a selection made one way is visible in other ways

- Perform crossmatch only on items in red giant branch
- Where on the sky is this colour cut?
- Spot outliers
- Identify objects on ds9 image display

Crossmatching

- Crossmatch tables together:

  - Pair match, Internal match, 3-, 4-, 5-table . . .
  - Flexible match criteria:
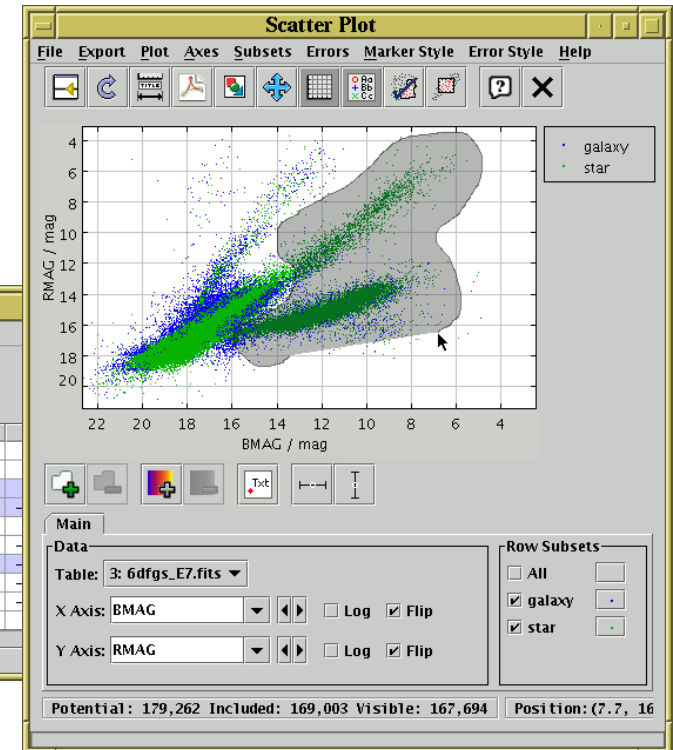   - ▷ RA, Dec
    - ▷ RA, Dec, radius (or redshift)
    - ▷ 2D or 3D (or more) Cartesian positions
    - ▷ Match distance fixed or per-object
    - ▷ Exact (e.g. object ID)
    - ▷ Ellipses (2D or sky)
    - ▷ Combinations of the above

  - Retain closest only or all matches
  - Output matched rows, unmatched rows, union, XOR, . . .
  - Efficient algorithm: usually $<$ a minute or two
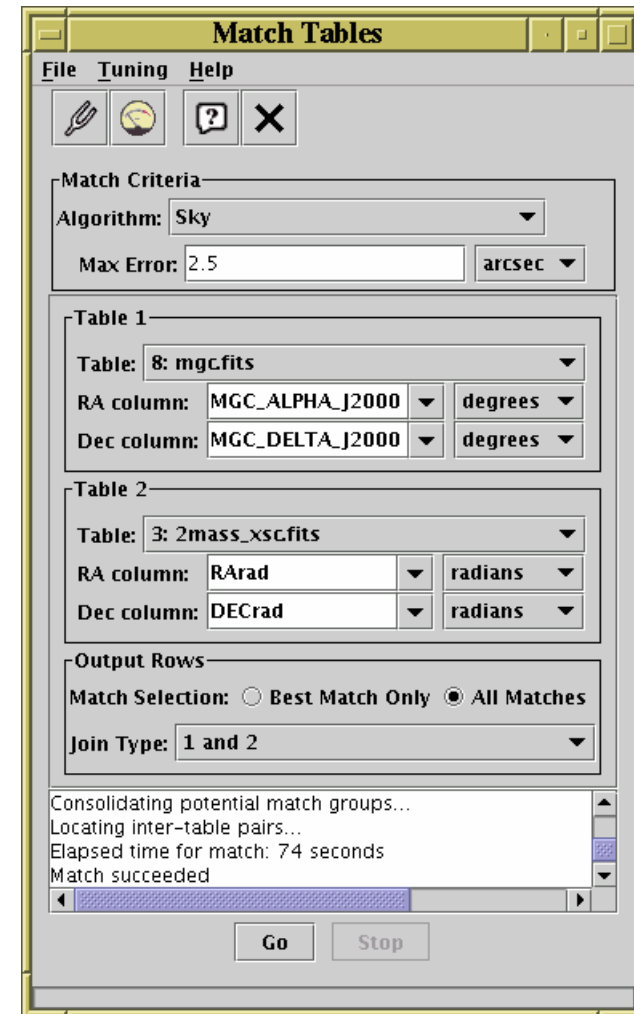
Calculations

- Expression language used for creating columns, defining selections, specifying axes etc:

  - Straightforward arithmetic syntax (C-like)

  - Use column names like variables

  - Standard arithmetic operators (`+`, `-`, `/`, `*`)

  - Standard mathematical functions (`sin`, `cos`, `exp`, `abs`, `max`, `round`, . . .)

  - Sky coordinates (degrees, sexagesimal, sky distances)

  - Time conversions (ISO8601, MJD, Julian, Besselian)

  - Cosmological distances (redshift, luminosity dist, lookback time, . . .)

  - Fluxes (Johnson AB Magnitudes, Jansky)

  - . . . and more (and it's extensible)

- Examples:

  - `psfmag_u - psfmag_g`

  - `janskyToAb(flux)`

  - `skyDistance(ra1,dec1,ra2,dec2) < 0.3*ARC_SECOND`

Virtual Observatory

Virtual Observatory (VO): *"All astro archives in your computer"*

- TOPCAT provides access to three positional data access protocols:
  - ▷ **Cone Search** — Catalogue archives
  - ▷ **Simple Image Access** — Image archives
  - ▷ **Simple Spectral Access** — Spectrum archives

- Use these in two ways:
  - ▷ Single position search:
    - ○ supply RA/Dec or object name, plus radius
    - ○ get back a table with all matched objects/images/spectra
  - ▷ Multi-position search:
    - ○ assign RA/Dec columns from input table, plus radius (column or constant)
    - ○ get back a table with match(es) from each row of input table (i.e. crossmatch with remote table)

- Image/Spectrum outputs — can pass data to external viewers using SAMP
- Other VO services:
  - ▷ **Registry** — Locate archives by keyword
  - ▷ **TAP** — SQL-like table access *(tutorial tomorrow)*

SAMP

Simple Application Messaging Protocol

- Developed within VO

- Many common desktop tools compatible:
  - ▷ TOPCAT, ds9, Aladin, SkyCat/GAIA, SPLAT, tapsh, MS WWT, VirGO, . . .

- Allows tools to exchange data
  - ▷ pass a table
  - ▷ pass a row selection
  - ▷ pass an image (FITS)
  - ▷ pass a sky position

TOPCAT: Help!

- Full tutorial and reference documentation:
  - ▷ HTML/PDF manual on web page `http://www.starlink.ac.uk/topcat/` (or Google it)
  - ▷ **Help for Window** button 🔲 on every window
  - ▷ Help browser includes search tool
  - ▷ More options in Help Menu (including **Help for Window in Browser** item 🔲)
  - ▷ Or print out the 230-page manual

- Support by mail:
  - ▷ on list: topcat-user@bristol.ac.uk
  - ▷ in person: m.b.taylor@bristol.ac.uk
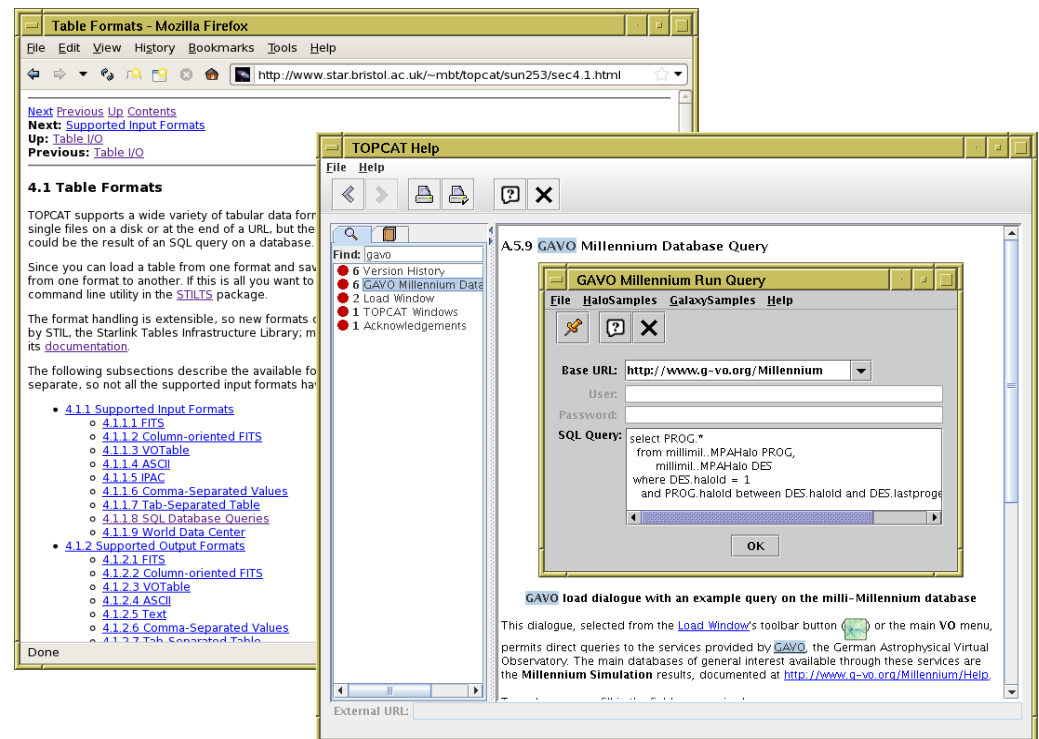
Demo

Expression Language Uses

- TOPCAT:

  - Synthetic columns

  - Algebraic subsets

  - Freeform column selectors (plots, crossmatches, ...)

  - Activation actions

- STILTS:

  - Synthetic columns

  - Row selections

  - Sort keys

  - Plotting values

  - Crossmatch values

  - . . .

Expression Language Syntax

Mostly like C/Python/Java/...

- Operators:
    - ▷ Arithmetic: + −, *, /
    - ▷ Logical: ==, !=, &&, ||, !
    - ▷ Conditional: *test* ? *value-if-true* : *value-if-false*
    - ▷ Array dereference: [*index*]
    - ▷ String concatenation: +

- Symbols (variable names) refer to constant or variable table values:
    - ▷ **Column names/$IDs:** take different value for each row (`RMAG`, `$6`)
    - ▷ **Subset names/_IDs:** take true/false value for each row (`galaxies`, `_3`)
    - ▷ **Parameters:** take same value for all rows (`param$ZP`, `ucd$pos_eq_ra_main`)

- Symbol syntax:
    - ▷ Starts with a letter (`[A-Za-z_$]`)
    - ▷ Continues with letters or numbers (`[A-Za-z_$0-9]`)
    - ▷ Sometimes column/parameter/subset names don't follow those rules:
        - ○ Name them sensibly in the first place
        - ○ Change them in the Columns Window (TOPCAT) or with `colmeta` (STILTS)
        - ○ Use alternative syntax (`$ID` for columns, `_ID` for subsets)

Expression Language Syntax (More)

- Literals

  - ▷ Enclose string values in double quotes (")
  - ▷ Use a decimal point for floating point (1.0/2, not 1/2)

- Special values

  - ▷ Blank value: NULL
  - ▷ Row index (first row is 1): index or $0
  - ▷ Random number $0 \rightarrow 1$: RANDOM

- Test for blank value:

  - ▷ To test if *name* is blank, use NULL_*name*

Expression Language Functions

- Usage:
  - Syntax `f(a, b, c)`
  - Use Function Browser for documentation (TOPCAT Functions window or STILTS `funcs`)
  - Arranged by category:
    - ▷ **Standard utilities:** Arithmetic, Arrays, Conversions, Formats, Maths, Strings
    - ▷ **Astro functions:** Coords, Distances, Fluxes, Tilings, Times
    - ▷ **Activation functions:** System, Image, Browsers, surveys, ... *(not STILTS)*

- Extensible
  - Can plug more in at runtime, but requires (simple) Java programming
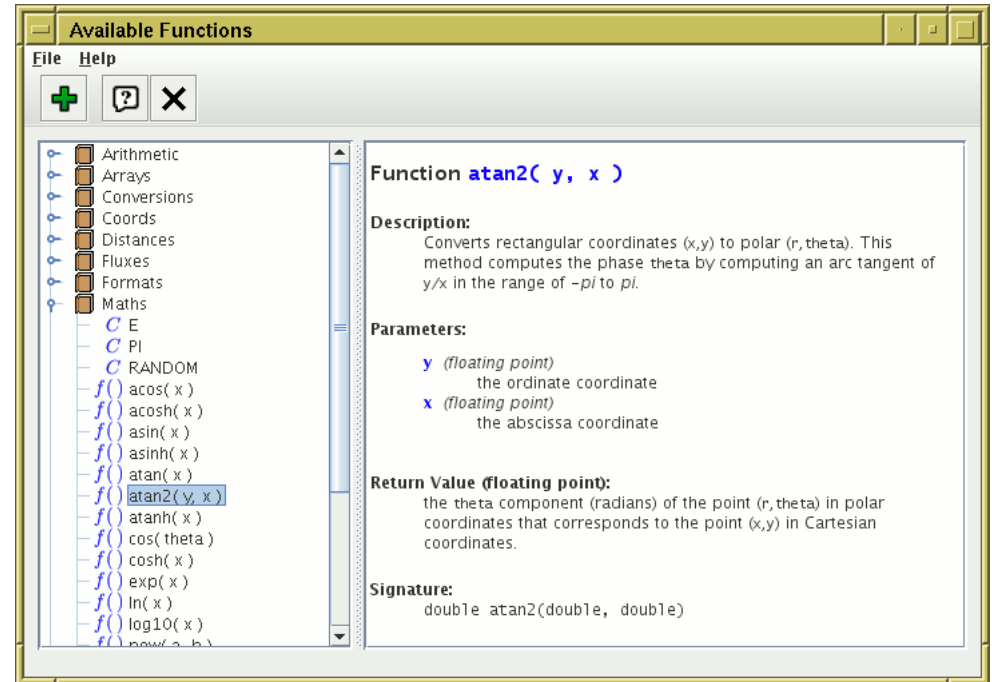
Expression Language Examples

- Numeric expressions (synthetic column)

  - Average: `(RMAG + BMAG) * 0.5`
  - Square root: `sqrt(variance)`
  - Sexagesimal to degrees: `radiansToDegrees(dmsToRadians(decDeg,decMin,decSec))`
  - Convert magic value to null: `jmag == 9999 ?  NULL : jmag`

- Logical expressions (subset)

  - In range: `RA > 100 && RA < 120 && Dec > 75 && Dec < 85`
  - Within circle (Euclidian): `$2*$2 + $3*$3 < 1`
  - Within circle (spherical): `skyDistanceDegrees(ra0,dec0,ra1,dec1)<1./3600.`
  - Fixed string matching: `startsWith(SECTOR, "ZZ")`
  - Regular expression: `matches(MORPH, "[Gg]al")`
  - Combine subsets: `(! _1) && (_2 || _3)`

- More in the Manual Chapter 9

STILTS

STIL Tool Set (STIL = Starlink Tables Infrastructure Library)

- Has pretty much the same capabilities as TOPCAT
- but works from the command line

TOPCAT	**STILTS**
GUI	Command line
Interactive	Scriptable
Easy to use	Reproducible
Good for data exploration	Good for batch/programmed use
Exploratory phase	Production phase
$\lesssim 10^6$ rows	Unlimited size (for most things)

Typical usage:

- start off with TOPCAT
- maybe move on to STILTS for more specialised requirements

STILTS Invocation

- Syntax:
  ```
  stilts <task-name> <param1>=<value1> <param2>=<value2> ...
  ```
- Simple example:
  ```
  stilts tcopy in=cat.ascii ifmt=ascii out=cat.fits ofmt=fits
  ```
- Complicated example:
  ```
  stilts plot2d
      in1=iras_psc.fits cmd1='addskycoords fk5 galactic RA DEC GLON GLAT'
      xdata1=GLON ydata1=GLAT auxdata1=FNU_100 auxlog=true auxflip=true size1=0 transparency1=3
      in2=messier.xml cmd2='addskycoords fk5 galactic RA DEC GLON GLAT'
      xdata2=GLON ydata2=GLAT txtlabel2=RADIUS>16?("M"+ID):"") cmd2='addcol SIZE sqrt(RADIUS/2)'
      xerror2=SIZE yerror2=SIZE subset2a=true hide2a=true colour2a=black errstyle2a=ellipse
      subset2b=true hide2b=true colour2b=black errstyle2b=filled_ellipse transparency2b=6
      xlabel='Galactic Longitude' ylabel='Galactic Latitude' title='The Sky'
      legend=false grid=false fontsize=16 fontstyle=bold-italic
      xlo=0 xhi=360 ylo=-90 yhi=+90 xpix=800 ypix=400
      out=skyplot.eps
  ```
- Quoting
  - ▷ Can get quite nasty — shell quoting and STILTS quoting interfere with each other
  - ▷ Rules of thumb:
    - ○ use single quotes outside parameters, double if needed inside
      ```
      cmd='keepcols "ID RA DEC"'
      ```
    - ○ avoid the need where possible by avoiding spaces
      ```
      cmd='addcol B_V BMAG-VMAG'
      ```
  - ▷ JyStilts can provide a better solution

STILTS: Help!

- The manual (HTML or PDF) contains:
  - Tutorial chapters on syntax and invocation
  - Expression language reference
  - Reference section for each command:
    - ▷ Description of what the command does
    - ▷ **Usage** section describing each parameter and the values it can take
    - ▷ **Examples** section with several examples of the command in use

- Help available on command line from STILTS itself:
  - List of available tasks:
    ```
    stilts -help
    ```
  - Usage (list of parameters) for each task:
    ```
    stilts tpipe -help
    ```
  - Detailed description of each parameter:
    ```
    stilts tpipe help=ofmt
    ```
  - Interactive mode:
    ```
    % stilts tmatch2
    in1 - Location of first input table:
    ```
  - Hopefully useful error messages

STILTS Pipelines

Manipulate tables by stacking up "filters" in a "pipeline"

- Like Unix pipelines (`grep | awk | sed | sort | ...`)
- "Filters" available for almost any manipulation:

  `addcol`: add column

  `select`: select only rows satisfying given condition

  `sort`: sort on values in one or more columns

  `head`: retain only first N rows

  `badval`: replace bad values, e.g. -9999→NULL

  . . . and many more

- Use them in STILTS commands as values of `cmd` (or `icmd`, `ocmd`, ...) parameters

  `stilts tpipe in=in.fits out=out.fits cmd=... cmd=...`

- Most make use of TOPCAT/STILTS algebraic expression language
- Examples:
  - ▷ Add a new calculated column:
    `cmd='addcol B_R BMAG-RMAG'`
  - ▷ Select only rows in a given region:
    `cmd='select skyDistanceDegrees(RA,DEC,78.63,-8.20)<0.001'`
  - ▷ Select only ten reddest objects:
    `cmd='sort RMAG-BMAG' cmd='head 10'`

STILTS Output Modes

Table output can be to a file, or some other destination

- Controlled by `omode` parameter on suitable commands
- Possible values:
  - ▷ `omode=out` (default): write to a file; associated `out` and `ofmt` parameters give filename and format
  - ▷ `omode=meta`: show metadata only
  - ▷ `omode=count`: count rows only
  - ▷ `omode=stats`: calculate column statistics only
  - ▷ `omode=topcat`: send table directly to running TOPCAT instance
  - ▷ ... and others

- Useful for experimentation, can avoid many run/save/load cycles; try a command, see (e.g.) what the resulting columns would be, tweak and try again
- Example:
  ```
  % tpipe in=x.fits cmd='select BMAG<15.1' omode=count
  columns: 28    rows: 0
     .  .  .    think again .  .  .
  ```

`tcopy`

Converts tables from one format to another

- Examples:
  - ▷ CSV to FITS:

    ```
    stilts tcopy in=x.csv ifmt=csv
            out=x.f ofmt=fits
    ```

  - ▷ FITS to VOTable:

    ```
    stilts tcopy in=x.fits out=x.vot
    ```

    (input format can be omitted for FITS or VOTable;
    output format can be omitted if a sensible extension is used)

- Intentionally simple

tpipe

Pipeline processor for tables

- Like `tcopy`, but
  - ▷ with filters (`cmd=...`)
  - ▷ with output mode (`omode=...`)

- Does anything that turns one input table into one output table

- Examples:
  - ▷ Restrict columns, keep only largest area rows:

    ```
    stilts tpipe 2dfgrs_ngp.fits \
            cmd='keepcols "SEQNUM AREA ECCENT"' \
            cmd='sort -down AREA' \
            cmd='head 20'
    ```

  - ▷ Cone search:

    ```
    stilts tpipe in=survey.fits
            cmd='select "skyDistance(hmsToRadians(RA),dmsToRadians(DEC), \
                                  hmsToRadians(2,28,11),dmsToRadians(-6,49,45) \
                    < 5 * ARC_MINUTE"' \
            omode=count
    ```

  - ▷ Sampled statistics:

    ```
    stilts tpipe USNOB.FITS cmd='every 1000000' omode=stats
    ```

$$\boxed{\texttt{calc}}$$

Simple calculator using TOPCAT/STILTS expression language

- Useful for getting expressions right before executing them in TOPCAT/STILTS
- Handy reckoner for astro or non-astro functions
- Examples:
  - ▷ Simple addition:
    ```
    % stilts calc "1 + 2"
       3
    ```

  - ▷ Time conversion:
    ```
    % stilts calc 'isoToMjd("2005-12-25T00:00:00")'
       53729.0
    ```

$$\boxed{\texttt{tmatch2}}$$

Cross-Matching between two tables

- All the same options available as TOPCAT's matcher
- Can also combine match coordinate spaces in more flexible ways
- Pipeline commands can be applied to input and/or output tables
- Examples:
  - ▷ Simple sky match:
    ```
    stilts tmatch2 in1=obs_v.xml in2=obs_i.xml out=obs_iv.xml \
                   matcher=sky values1="ra dec" values2="ra dec" params="2"
    ```

  - ▷ Sky match with preprocessing
    ```
    stilts tmatch2 survey.fits ifmt2=csv mycat.csv \
                   icmd1='addskycoords fk4 fk5 RA1950 DEC1950 RA2000 DEC2000' \
                   matcher=skyerr \
                   params=10 values1="RA2000 DEC2000 POS_ERR"  values2="RA DEC 0" \
                   join=2not1 omode=count
    ```

  - ▷ Sky match with postprocessing:
    ```
    stilts tmatch2 ifmt1=ascii ifmt2=ascii in1=cat-a.txt in2=cat-b.txt \
                   matcher=2d values1='X Y' values2='X Y' params=5 join=1and2 \
                   suffix1=_a suffix2=_b \
                   ocmd='addcol XDIFF X_a-X_b; addcol YDIFF Y_a-Y_b' \
                   ocmd'keepcols "XDIFF YDIFF"' omode=stats
    ```

`coneskymatch`

Multiple Cone Search, Simple Image, Simple Spectral Access query

- Performs one cone, SIA or SSA search for each row of a given table
- Pipeline commands can be applied to input and/or output tables
- Need to supply the URL of the VO service
    - ▷ Often the best way to obtain this is to use TOPCAT's VO windows

- Examples:
    - ▷ Multi Cone:

        ```
        stilts coneskymatch
                serviceurl='http://vizier.u-strasbg.fr/viz-bin/votable/-A?-source=II/246&'
                ra=RA2000 dec=DE2000 sr=0.05
                in=cat.fits out=cat_2mass.fits
        ```

    - ▷ Multi Simple Image Access:

        ```
        stilts coneskymatch
                serviceurl='http://isda.esac.esa.int:8080/aio/jsp/metadata.jsp?resource_class=ods'
                servicetype=sia dataformat=image/fits
                ocmd='keepcols ucd$VOX_Image_AccessReference'
                in=cat.fits out=fits_image_urls.txt
        ```

Plotting

Plot commands for histogram, 2d scatter plot, 3d Cartesian plot

- Output to screen or graphics file (GIF, PNG, EPS, PDF, ...)
- Examples:
  - ▷ Histogram:
    ```
    stilts plothist in=cat.xml xdata=Separation
    ```
  - ▷ 2d scatter plot:
    ```
    stilts plot2d in=cat.xml xdata=Vmag-Bmag ydata=Vmag yflip=true
    ```
  - ▷ 3d scatter plot:
    ```
    stilts plot3d in=millenn_g2.fits out=g2.pdf
                  xdata=x ydata=y zdata=z auxdata=snapnum
    ```
  - ▷ Multi-dataset plot:
    ```
    stilts plot2d in1=virgo.csv ifmt1=csv xdata1=u-r ydata1=u yflip1=true
                  in2=coma.csv  ifmt2=csv xdata2=u-r ydata2=u yflip2=true
                  out=color_mag.pdf
    ```

Other Commands

- The list of commands includes:

  - General purpose pipeline: `tpipe`

  - Format conversion: `tcopy` (or `tpipe`)

  - Simple calculations: `calc`

  - Crossmatching: `tskymatch2`, `tmatch2`, `tmatch1`, `tmatchn`

  - Multiple cone search: `coneskymatch`

  - TAP queries: `tapquery`, `tapresume`

  - Other joins: `tcat`, `tcatn`, `tjoin`

  - Calculate histogram: `tcube`

  - Multi-table file manipulation: `tmulti`, `tmultin`

  - Plotting: `plot2d`, `plot3d`, `plothist`

  - VOTable manipulation: `votlint`, `votcopy`

  - SQL specific: `sqlclient`, `sqlupdate`, `sqlskymatch`

JyStilts

STILTS can be run under Jython

- This may be more attractive to Python fans

- Jython *is* Python (written in Java), though no C-based packages (e.g. NumPy)

- Advantages:
  - ▷ String syntax is less horrible than from the shell
  - ▷ More efficient for complicated pipelines
  - ▷ Can write smaller/less unwieldy scripts

- Comparison:
  - ▷ From the shell:
    ```
    stilts tskymatch2 in1=survey.fits \
                      icmd1='addskycoords fk4 fk5 RA1950 DEC1950 RA2000 DEC2000' \
                      in2=mycat.csv ifmt2=csv \
                      icmd2='select VMAG>18' \
                      ra1=ALPHA dec1=DELTA ra2=RA2000 dec2=DEC2000 \
                      error=10 join=2not1 \
                      out=matched.fits
    ```
  - ▷ From JyStilts:
    ```
    >>> import stilts
    >>> t1 = stilts.tread('survey.fits')
    >>> t1 = t1.cmd_addskycoords(t1, 'fk4', 'fk5', 'RA1950', 'DEC1950', 'RA2000', 'DEC2000')
    >>> t2 = tread('mycat.csv', 'csv')
    >>> t2 = t2.cmd_select('VMAG>18')
    >>> tm = skymatch2(in1=t1, in2=t2, ra1='ALPHA', dec1='DELTA', error=10, join='2not1')
    >>> tm.write('matched.fits')
    ```

Summary

TOPCAT http://www.starlink.ac.uk/topcat/

STILTS http://www.starlink.ac.uk/stilts/

Feedback

- I'm happy to help with questions, usage issues, etc
- Suggestions always welcome*

*though I don't always guarantee to implement them