Yafit — Yet Another Fitter

Spectral Best Fitting Tool

http://www.star.bris.ac.uk/~mbt/yafit/

Mark Taylor, Bristol University
m.b.taylor@bristol.ac.uk

Input from: Paresh Prema (IoA)
Anita Richards (Jodrell)

Requirements Overview

Best Fitting Tool is a task required by DS4

What it has to do:

- Fit observed SEDs to library of calculated model spectra
- Observed SEDs are (so far) sets of photometry points
- Output is (at least) identification of the model which fits observations best

Calculations are quite straightforward

- χ^2 fitting

I/O and user interface are more involved

- Input observed and model data come in many formats

Input Data

1. An array of pre-calculated model spectra
   - Probably generated by some specialist code
   - Each model spectrum $\phi(\lambda)$ is flux as a function of spectral variable (various units on both axes)
   - Many variations of format (ASCII, VOTable, Galaxev, Starburst99, SVO, Pegase, . . .)
2. One or more observed set of photometric points
   - Each dataset contains a set of photometric observations
   - Each photometric observation (point) has:

     Spectral axis value X: may be wavelength, frequency, various units
     Spectral axis envelope ΔX: spectral range corresponding to filter
     — model as spike, box, Gaussian, bandpass function, . . .
     Flux axis value Y: may be flux in Jansky or other units, or on some magnitude scale
     Flux axis error ΔY: lower and upper errors may be same or different
   - Dataset may have an associated redshift value
   - Many variations of format (ASCII tables, VOTables, SED DM serializations, . . .)

(items greyed out are not currently implemented)

Processing Requirements

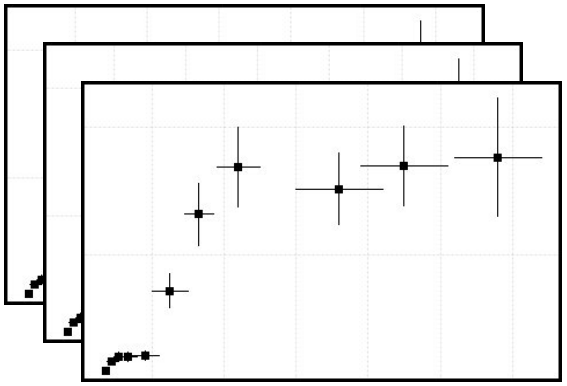
For each observed photometric dataset:

- Compare it against each known model spectrum (possibly rescaling and redshifting)
- Calculate goodness of fit measures (e.g. χ^2)
- Identify model spectrum which is the best fit to each observed photometric dataset

Output results in some way:

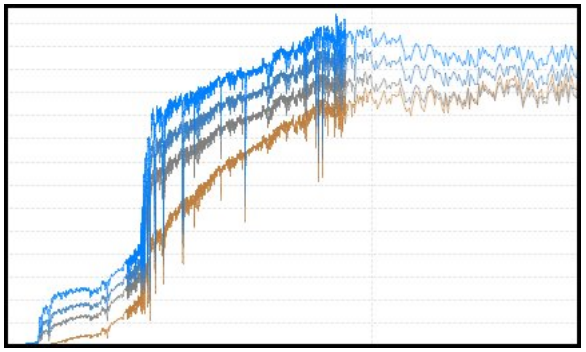
- Write summary table listing best fit for each photometric dataset
  (including model metadata, observation metadata and fitting statistics)
- Write table describing *all* model $\leftrightarrow$ observation fits
  (including model metadata, observation metadata and fitting statistics)
- Provide graphical presentation of summary/full fitting results

Processing Overview



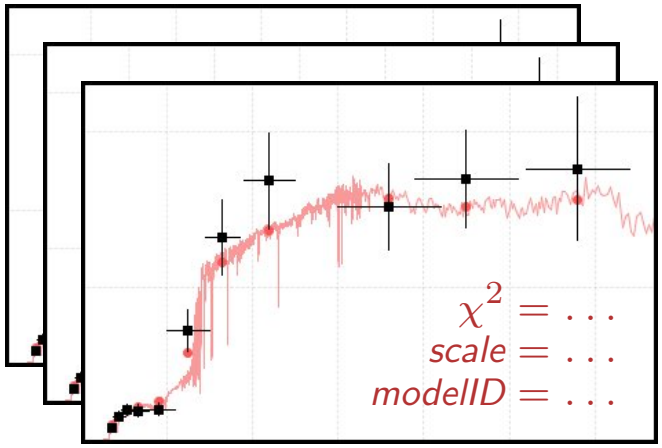
One or more photometric spectra

Fit photo spectrum
against each model



Array of model spectra

Select
lowest
χ^2



Best model fit to observation

Input Format Handling

Photometric spectra inputs

- Convert to specialised `.yobs` format before fitting
- Command `tableobs` converts table + key file → `.yobs` file

```
#         RA        DEC         J    J_ERR         H    H_ERR
      119.608     21.410     11.94    0.022     12.16    0.024
       53.094    -27.839     12.33    0.022     12.34    0.026
      213.870    -46.515     17.32    0.106     17.57    0.192
      204.143    -29.089     15.93    0.041     16.29    0.071
```

+

```
#    yColName    yerrColName         x    xWidth
           J          J_ERR     12400      5000
           H          H_ERR     16600      4000
```

- `tableobs` can perform other manipulations at the same time, e.g. unit conversion
- Converter for SED Data Model serialization(s) and others may be required in future

Model spectra inputs

- Convert to specialised `.ymod` format before fitting
- Command `copymodel` converts from various known formats to `.ymod` file
- Currently supported are Starburst99, Galaxev, SvoTar, SidewaysVOT
- `copymodel` can perform other manipulations at the same time, e.g. unit conversion
- Additional formats can be supported as requested

Minimal Example

1. Prepare observations file:

```
yafit tableobs in=photo_points.vot \
               key=sdss_bands.txt \
               out=photo_points.yobs
```

2. Prepare models file:

```
yafit copymodel ifmt=galaxev in=bc2003_salp.a1 in=bc2003_salp.a2 \
                out=gal_models.ymod
```

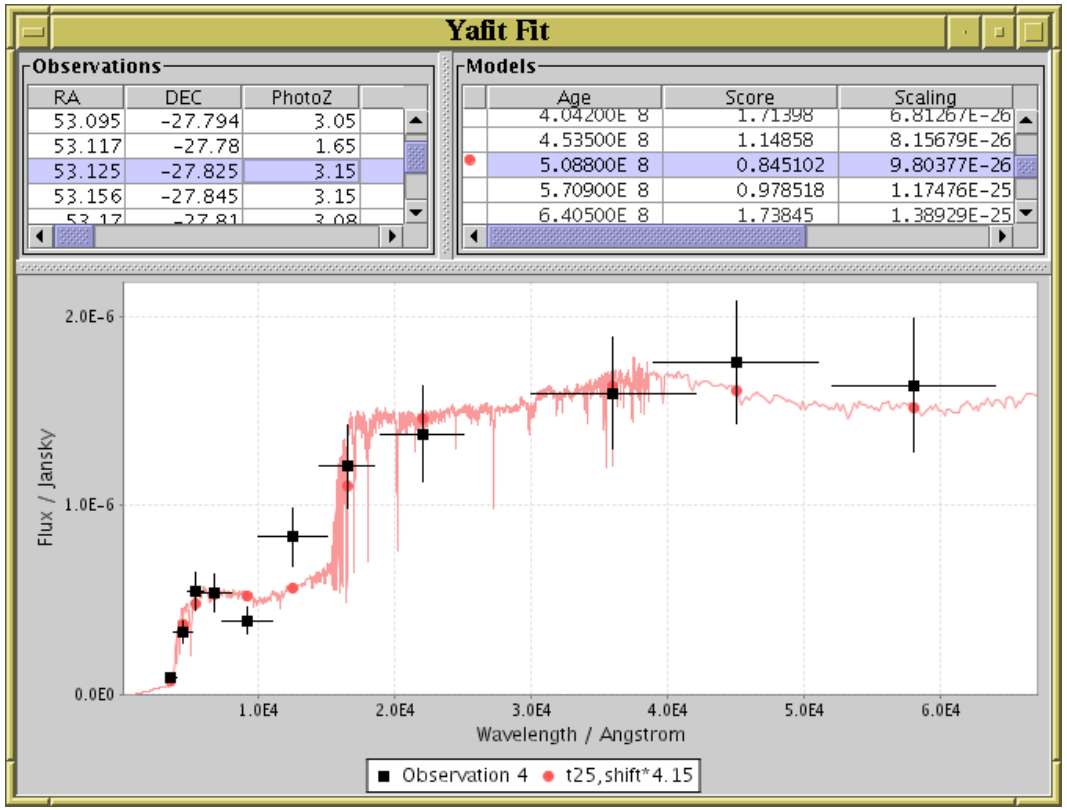
3. Perform fit:

```
yafit fit obs=photo_points.yobs model=gal_models.ymod \
          bestfits=out.vot \        — writes summary to VOTable
          gui=true                  — pops up window with results plotted
```

Output

Interactive display

- Graphics display
  - Observation points
  - Best fit model
  - Optionally other models
- Table display
  - Observation metadata
  - Model metadata
  - Fit statistics



Output table linking input observations with best fit models

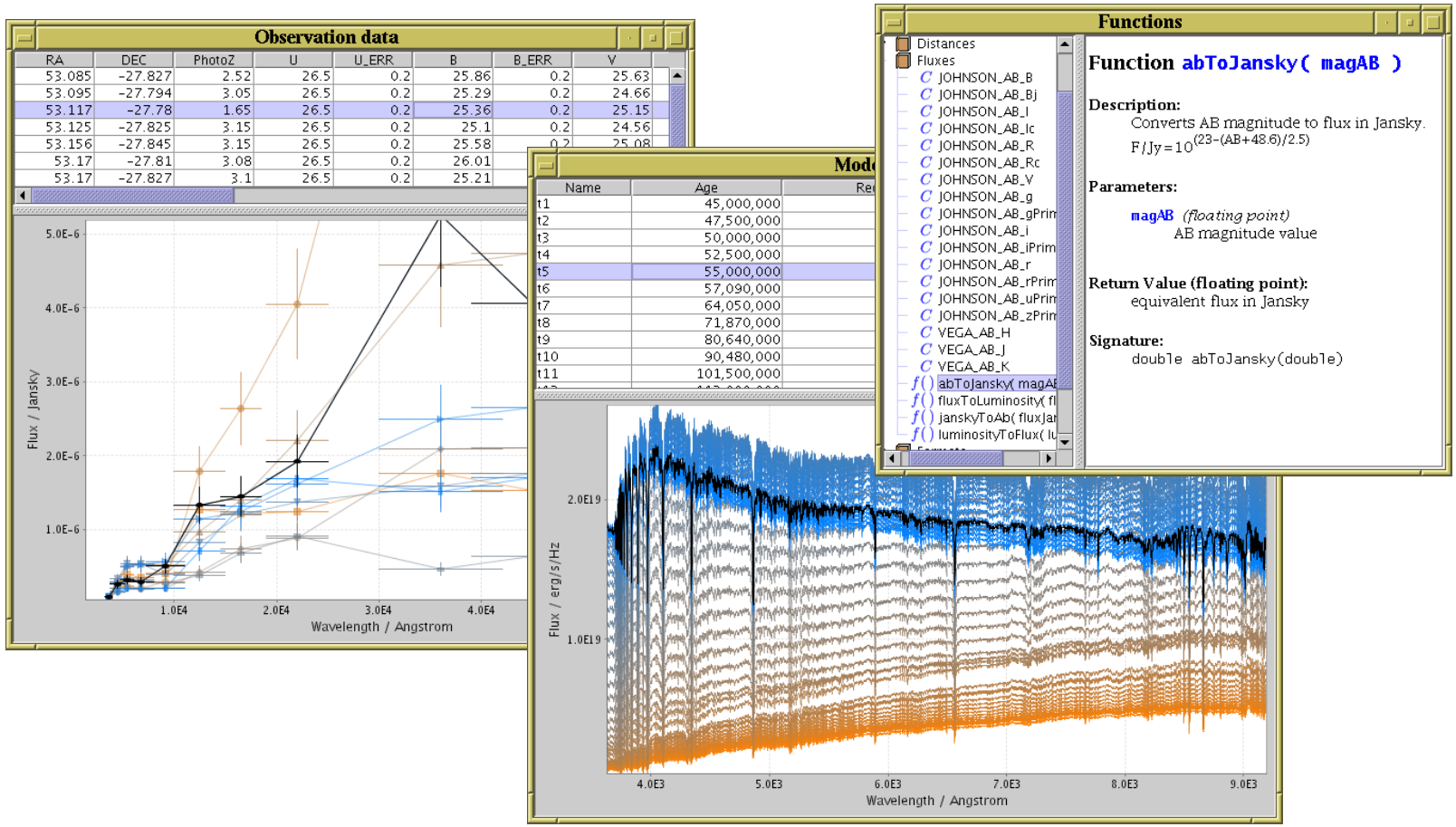
Input Metadata		Input Observations				Best Model		Fit Statistics	
RA	DEC	J	J_Err	H	H_ERR	MODEL_ID	AGE	SCORE	SCALE
119.608	21.410	11.94	0.022	12.16	0.024	m06	8.06E8	0.7430	1.29E-6
53.094	-27.839	12.33	0.022	12.34	0.026	m29	2.15E9	0.5998	4.60E-7
213.870	-46.515	17.32	0.106	17.57	0.192	m13	1.20E8	1.8392	5.19E-6
204.143	-29.089	15.93	0.041	16.29	0.071	m44	3.40E9	2.1345	8.11E-6

Other Capabilities

Standalone observation viewer (`plotobs` command)

Standalone model viewer (`plotmodels` command)

Powerful expression language for unit conversions etc



Current Status

Experimental release 0.1b

- Web page http://www.star.bris.ac.uk/~mbt/yafit/
- Full documentation (HTML/PDF reference and tutorial user guide)
- Working, but:
  - Subject to change (custom file formats, user interface, . . .)
  - Will probably require more observation & model input format handlers
  - Graphics are a bit clunky
  - Testing may throw up usability issues, feature requests