PhotRedCatalog

An implementation of the Munich Photometric Redshift code

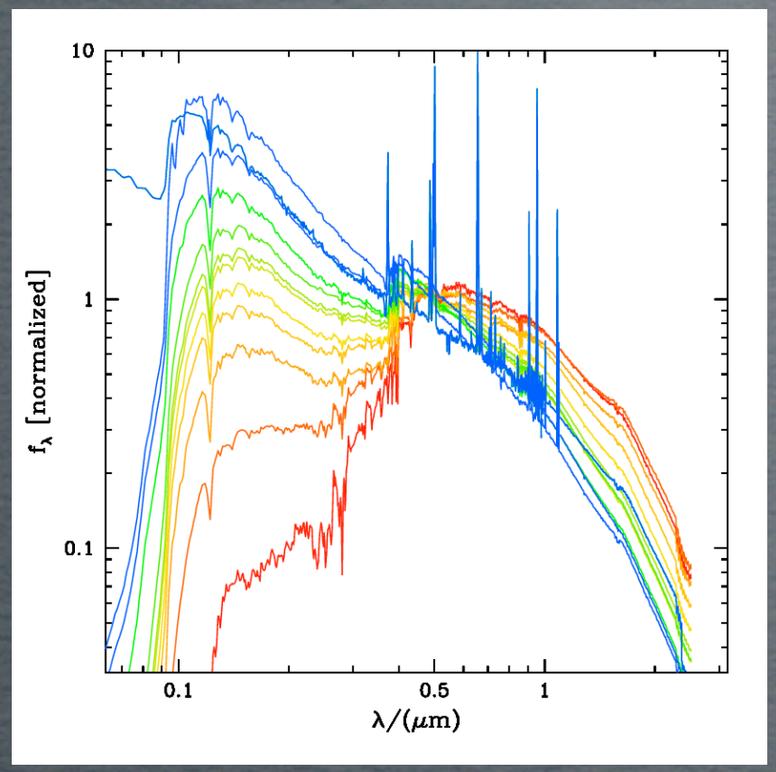
PhotRedCatalog

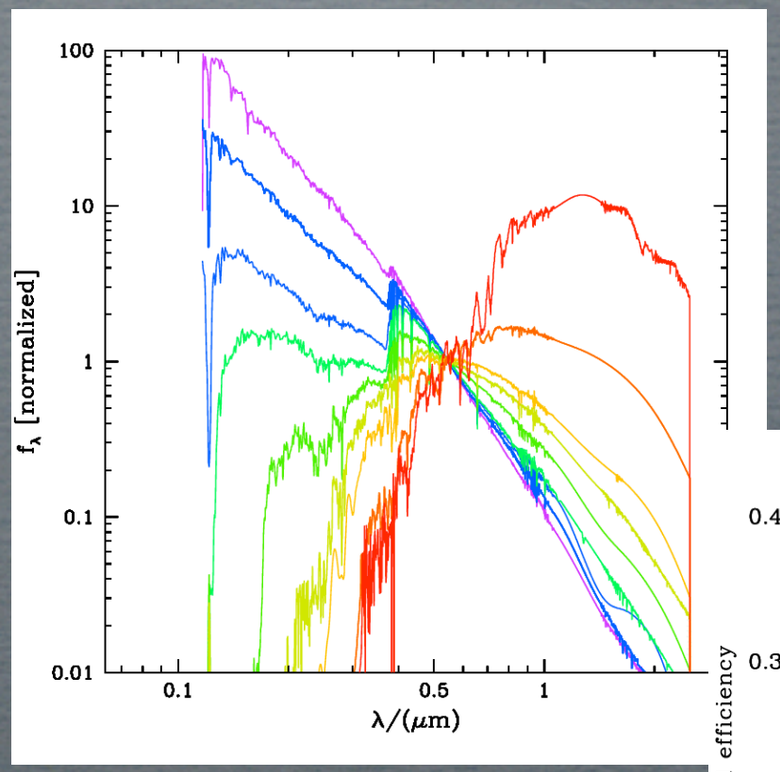
- a wrapper around SEDFILTER and PHOTOZ
- Compability with SourceLists and the Astro-Wise concepts
- All relevant information stored in the database
- Output as a new Sourcelist

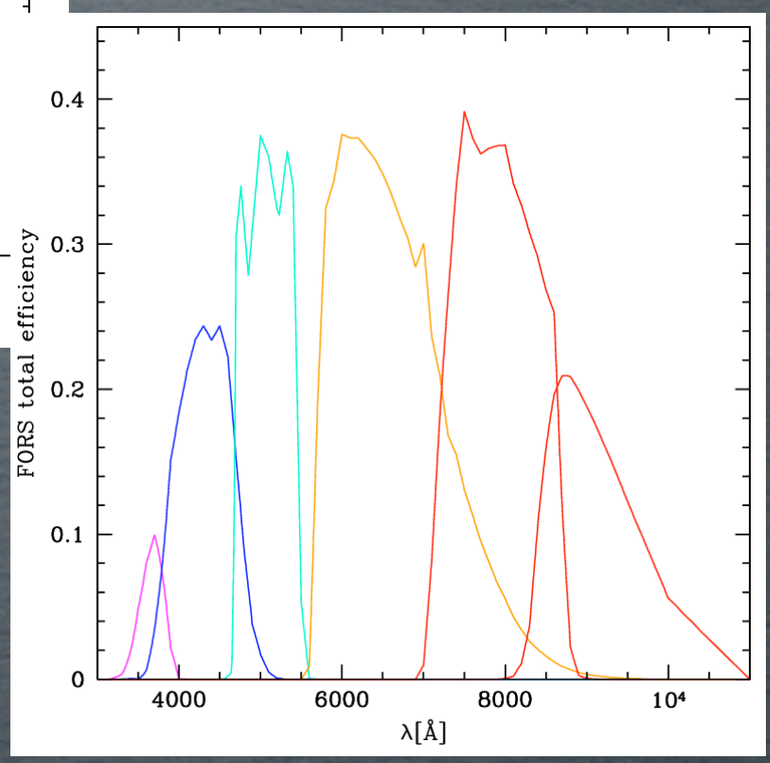
SEDFILTER & PHOTOZ

Input files

- Galactic and stellar SEDs





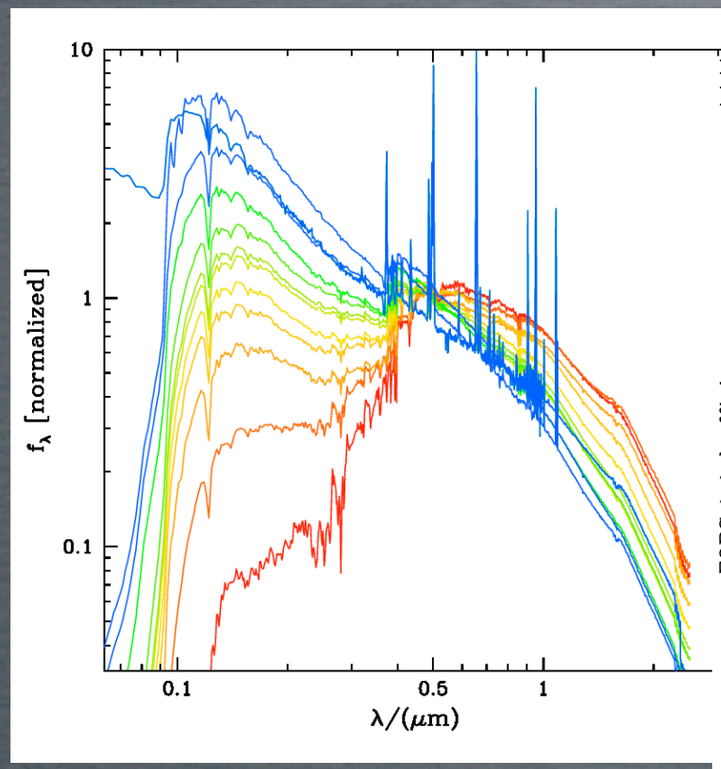


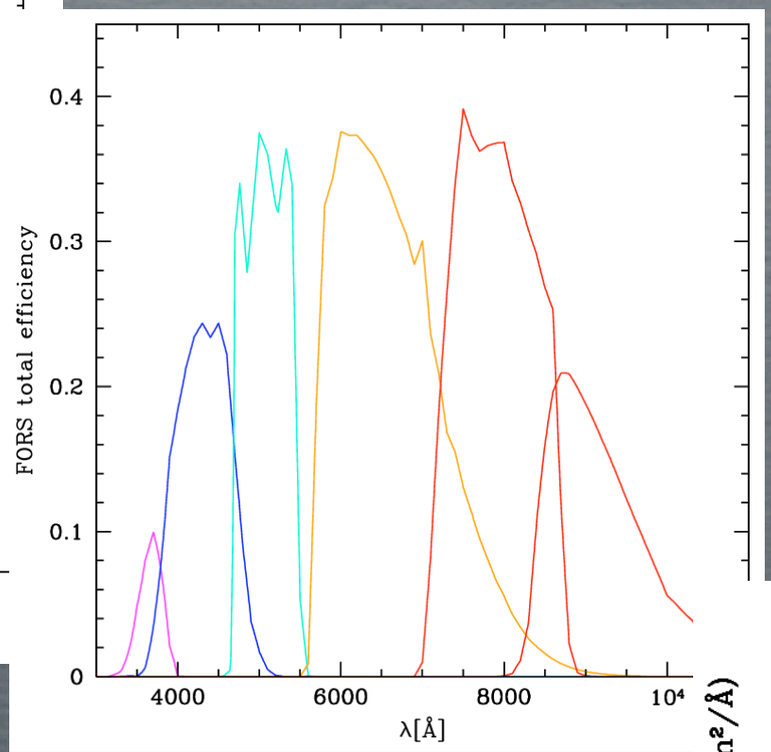
- Filter transmission curves

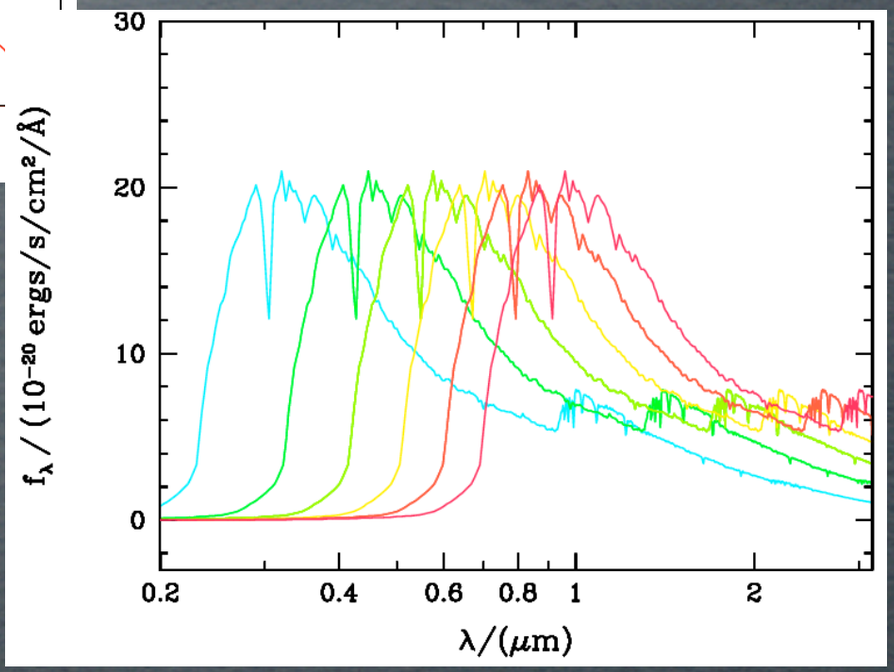
- Catalog with aperture photometry

SEDFILTER

- convolves the SEDs with the Filter curves

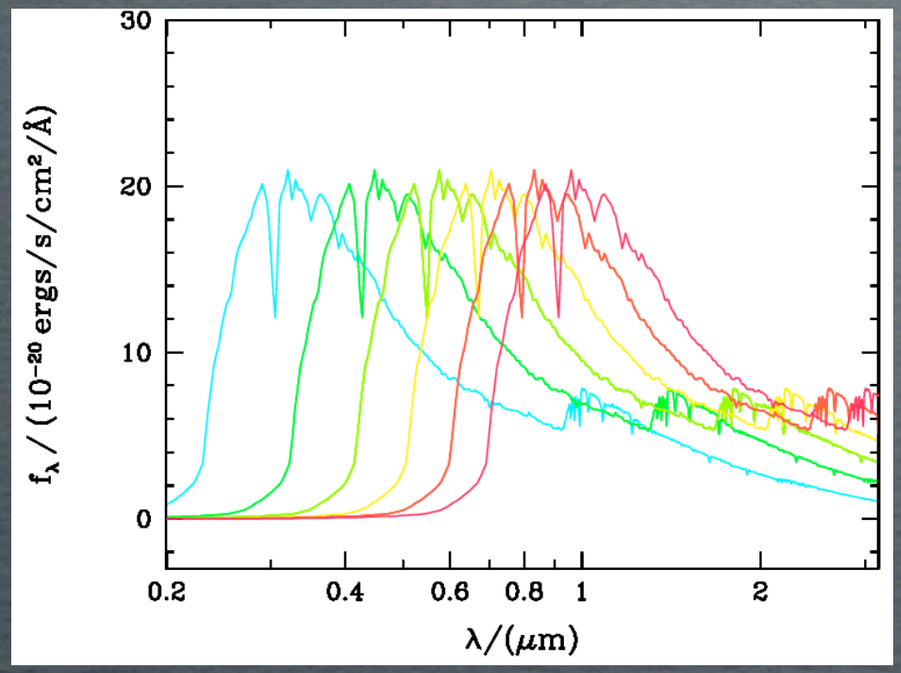


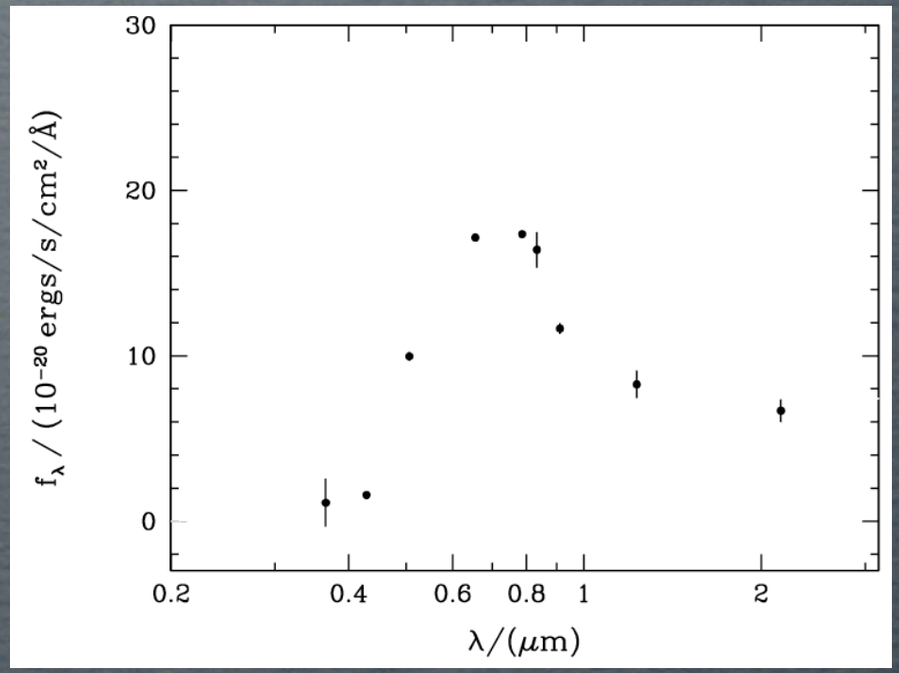




PHOTOZ

- Reads in convolved SEDs and the catalog
- Redshifts the SEDs
- Least-Squares fit of the magnitudes derived from the convolved SEDs against the observed data





PHOTOZ

Best fitting z and SED are determined by minimizing:

$$\chi^2(z, SED) = \frac{1}{N_{filt}} \sum_{i=1}^{N_{filt}} \frac{[f_i - \alpha f_i(z, SED)]^2}{\sigma_i^2 + [0.05\alpha f_i(z, SED)]^2}$$

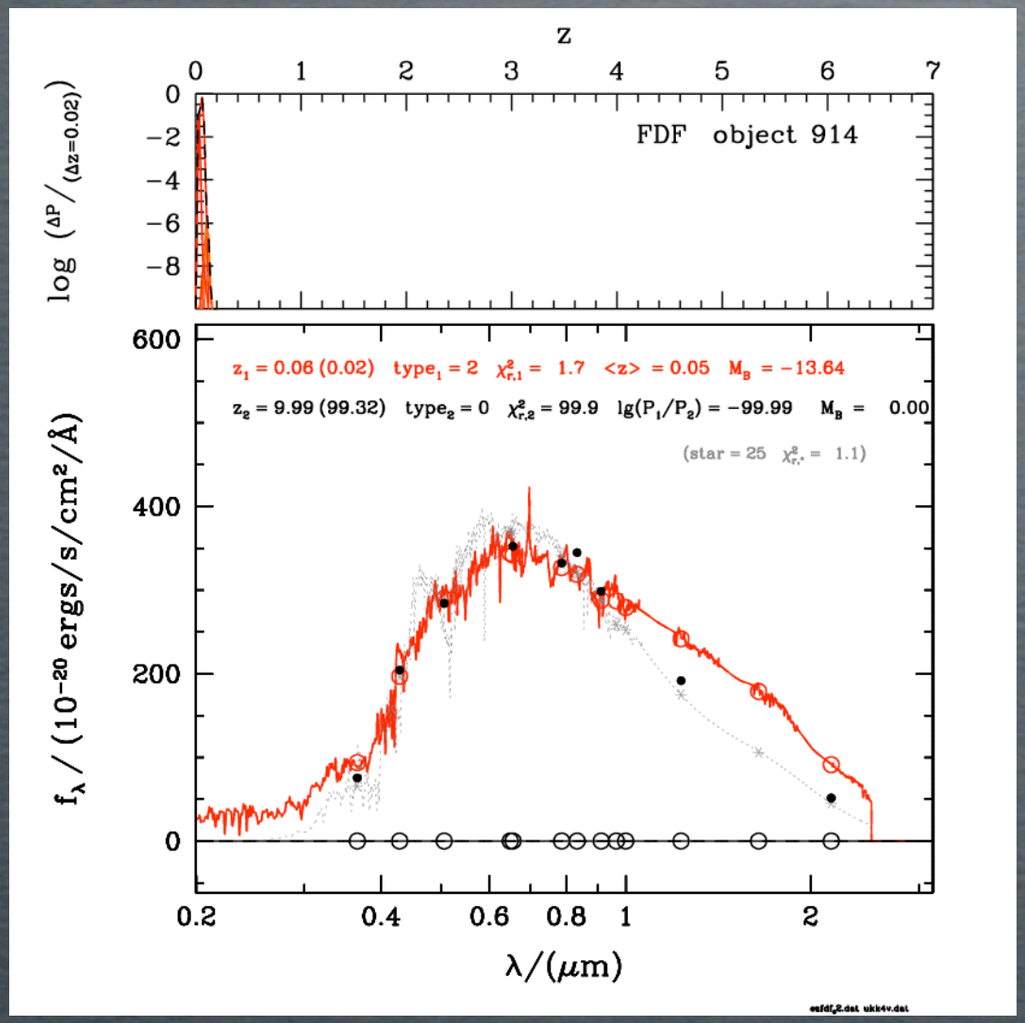
The probability of a source being a given redshift is:

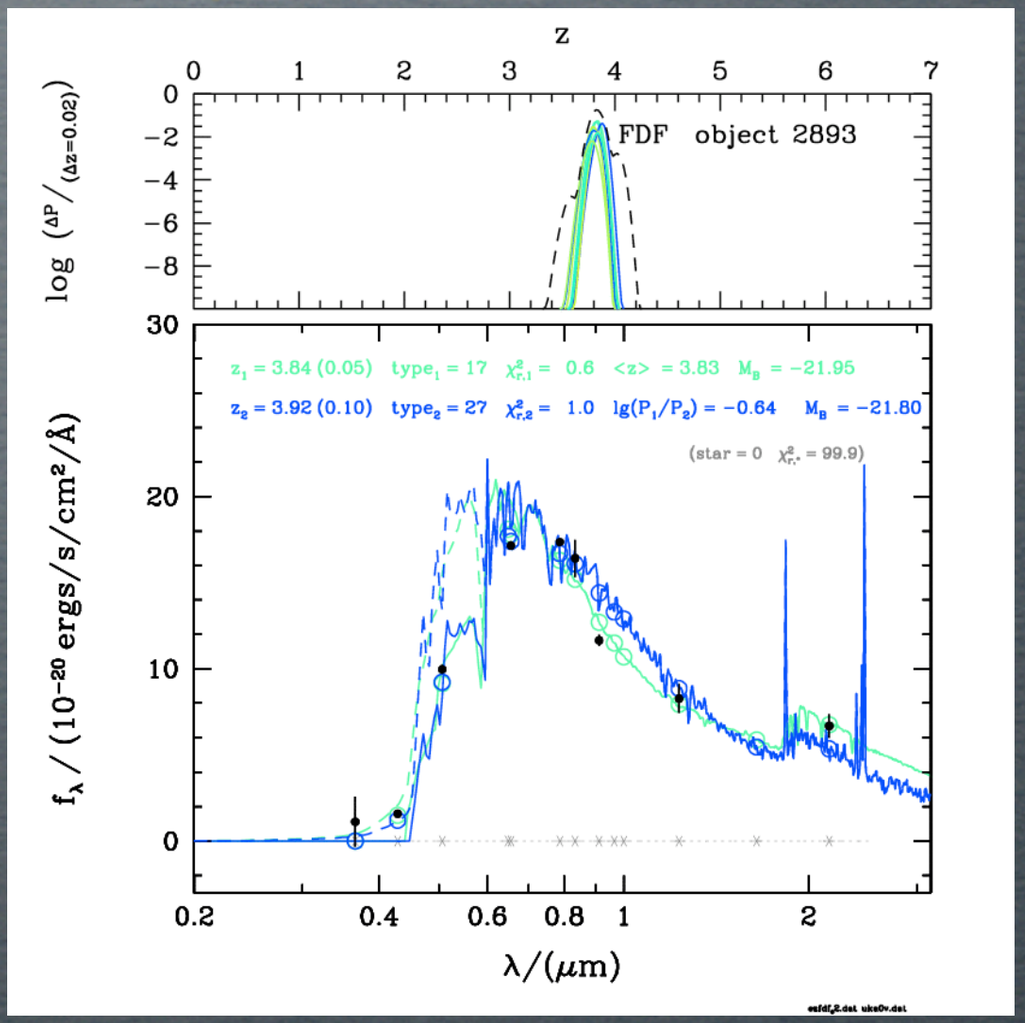
$$P_T = P_\chi \cdot P_L \cdot P_z = e^{-\frac{1}{2}\chi^2} \cdot e^{-k_\beta\left(\frac{M-M_\star}{\sigma}\right)^\beta} \cdot e^{-k_\gamma\left(\frac{z}{z_{lim}}\right)^\gamma}$$

SEDFILTER & PHOTOZ

Output

- Z of best fitting galactic SED, and bestfitting stellar SED





PhotRedCatalog

Input Classes

PhotRedFilter PhotRedSED

PhotRedStarlib

Internal Classes

CPhotRedSED CPhotRedStarlib

CPhotRedPlotSED PhotRedZeroPoints

PhotRedEigenspectra

User Interface

PhotRedConfig

PhotRedCatalog

PhotRedCatalog

PhotRedFilter

File with filter transmission curve

PhotRedSED

File with stellar/galactic SED

PhotRedStarlib

List of stellar SEDs

PhotRedCatalog

Convolved SEDs

- CPhotRedSED
  - CPhotRedStarlib
    - CPhotRedPlotSED

Photoz internal files

- PhotRedZeroPoints
  - PhotRedEigenspectra

PhotRedCatalog

```
class PhotRedConfig(DBObject)
```

```
SEDs = persistent( 'List of SEDs', PhotRedSED, [] )
filters = persistent( 'List of Filters', PhotRedFilter, [] )
starlib = persistent( 'Starlib', PhotRedStarlib, None )
name = persistent( 'Unique ID string', str, '' )

convolvedSEDs = persistent( 'List of convolved SEDs', CPhotRedSED, [] )
convolvedStarlib = persistent( 'Convolved Starlib', CPhotRedStarlib, None )
zeropoints = persistent( 'zeropoints file', PhotRedZeropoints, None )
eigenspectra = persistent( 'eigenspectra file', PhotRedEigenspectra, None )
```

```
def make(self):

    conffile = self.write_sedfilter_in()

    PhotRed.run_sedfilter(conffile)

    self.read_files()

    self.commit()
```

Write input file for sedfilter

Run ‘sedfilter < conffile’

Read and store the Convolved SEDs

PhotRedCatalog

```
class PhotRedCatalog(DBObject)
```

```
master = persistent( 'Master sourcelist', SourceList, None )
sourcelists = persistent( 'Associated input source lists', SourceList, [] )
associate_list = persistent( 'Associated catalog', AssociateList, None )

config = persistent( 'Photred Configuration', PhotRedConfig, None )
min_num_sources = persistent( 'Minimum number of detections', int, 3 )

extinc = persistent( 'Extinction values', float, [] )
model_error = persistent( 'Model errors', float, [] )

mag = persistent( 'Key for magnitude in Sourcelist', str, 'MAG_ISO' )
flux = persistent( 'Key for flux in Sourcelist', str, 'FLUX_ISO' )
fluxerr = persistent( 'Key for fluxerror in Sourcelist', str,'FLUXERR_ISO')

datpz1_name = persistent( 'datpz1 Sourcelist filename', str, '' )
datstar_name = persistent( 'datstar Sourcelist filename', str, '' )
datpz1 = persistent( 'Best fitting galactic results', SourceList, None )
datstar = persistent( 'Best fitting stellar results', SourceList, None )
```

PhotRedCatalog

class PhotRedCatalog(DBObject)

```
def make(self):

    <...some sanity checks...>

    self.make_association()

    self.config.write_files()

    self.write_files()

    PhotRed.run_photred('photoz.in')

    self.make_datpz1name()
    self.datpz1_to_ldac()

    self.make_datstarname()
    self.datstar_to_ldac()

    # self.associate_results()

    self.commit()
```

Master Association of input SourceLists

Write input file for photoz

Construct photometric catalog

Run photoz < photoz.in

Construct datpz1 SourceList

Construct datstar SourceList

Associate the SourceLists

PhotRedCatalog

datpz1 SourceList

```
RA,DEC,Xpos,Ypos,A,B,POSANG  - Standard photred parameters
obj                          - Object ID
best_z,err_z                 - redshift and error of best fit
mod                          - Model id
rchi2                        - reduced chi^2
z2                           - second best fitting z
lg_Pz2z1                     - Ratio of the probabilities of z2/z
<z>                          - weighted mean of the z distribution

fU,fB,fV,fR,fI,fJ,fH,fK,fF1,fF2,fF3,fF4,fF5,fF6,fF7
M_B,M_R,M_I,M_K              - fluxes of object and derived
absolute magnitudes

DMOD                         - derived distance modulus

f_dat/f_mod                  - Ratio between observed/model flux

best_model                   - Best fitting model
```

PhotRedCatalog

datstar SourceList

```
RA,DEC,Xpos,Ypos,A,B,POSANG  - Standard photred parameters
obj                          - Object ID
best_star                    - Model id of bestfitting star
rchi2                        - reduced chi²

fU,fB,fV,fR,fI,fJ,fH,fK,
fF1,fF2,fF3,fF4,fF5,fF6,fF7  - object fluxes

f_dat/f_mod                  - Ratio between observed/model flux
```

PhotRedConfig

An example from users view

```
awe> from Experimental.PhotRedCatalog import *
# We assume the SEDs and Filters were already ingested
# Create the PhotRedConfig object
# Select the filters
awe> pfu = (PhotRedFilter.filename == 'fors_bess_u_eso_tot.filter' )[0]
...
# Select the SEDs
awe> ps01 = (PhotRedSED.filename == 'manucci_soc.sed' )[0]
awe> ps02 = (PhotRedSED.filename == 'manucci_sac.sed' )[0]
...
# Create the PhotRedConfig
awe> pc = PhotRedConfig()
awe> pc.SEDs=[ps01,ps02,ps03,ps04,ps05,ps06,ps07,ps08,ps09,ps10,
     ps11,ps12,ps13,ps14,ps15,ps16,ps17,ps18,ps19,ps20,
     ps21,ps22,ps23,ps24,ps25,ps26,ps27,ps28,ps29]
awe> pc.filters=[pfu,pfb,pfv,pfr,pfi,pfj,pfk,pfz]
awe> pc.starlib=(PhotRedStarlib.filename=='starlib_pickles_v.lis')[0]
awe> pc.name='fors_demo'
awe> pc.make()
```

PhotRedCatalog

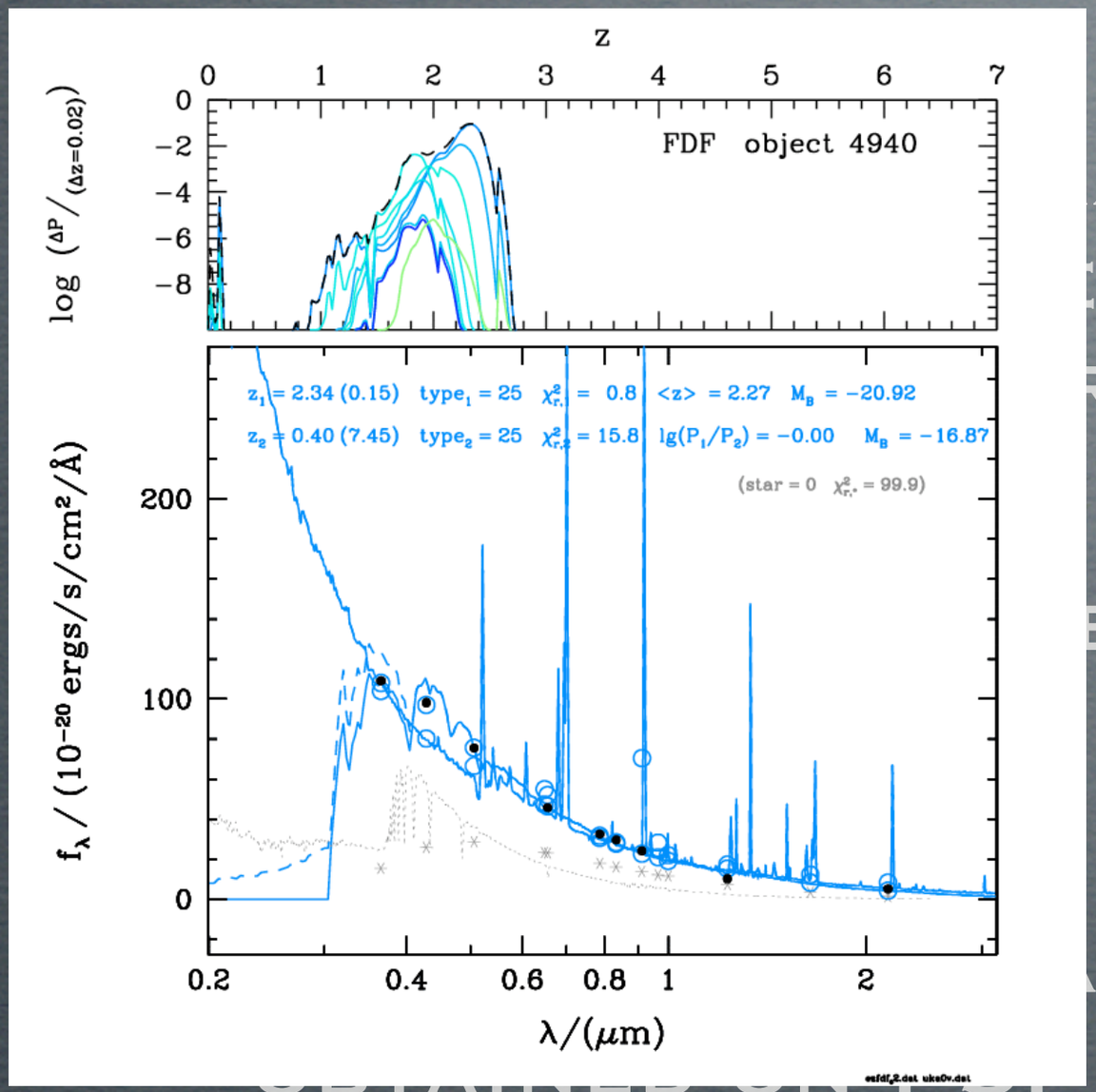
An example from users view

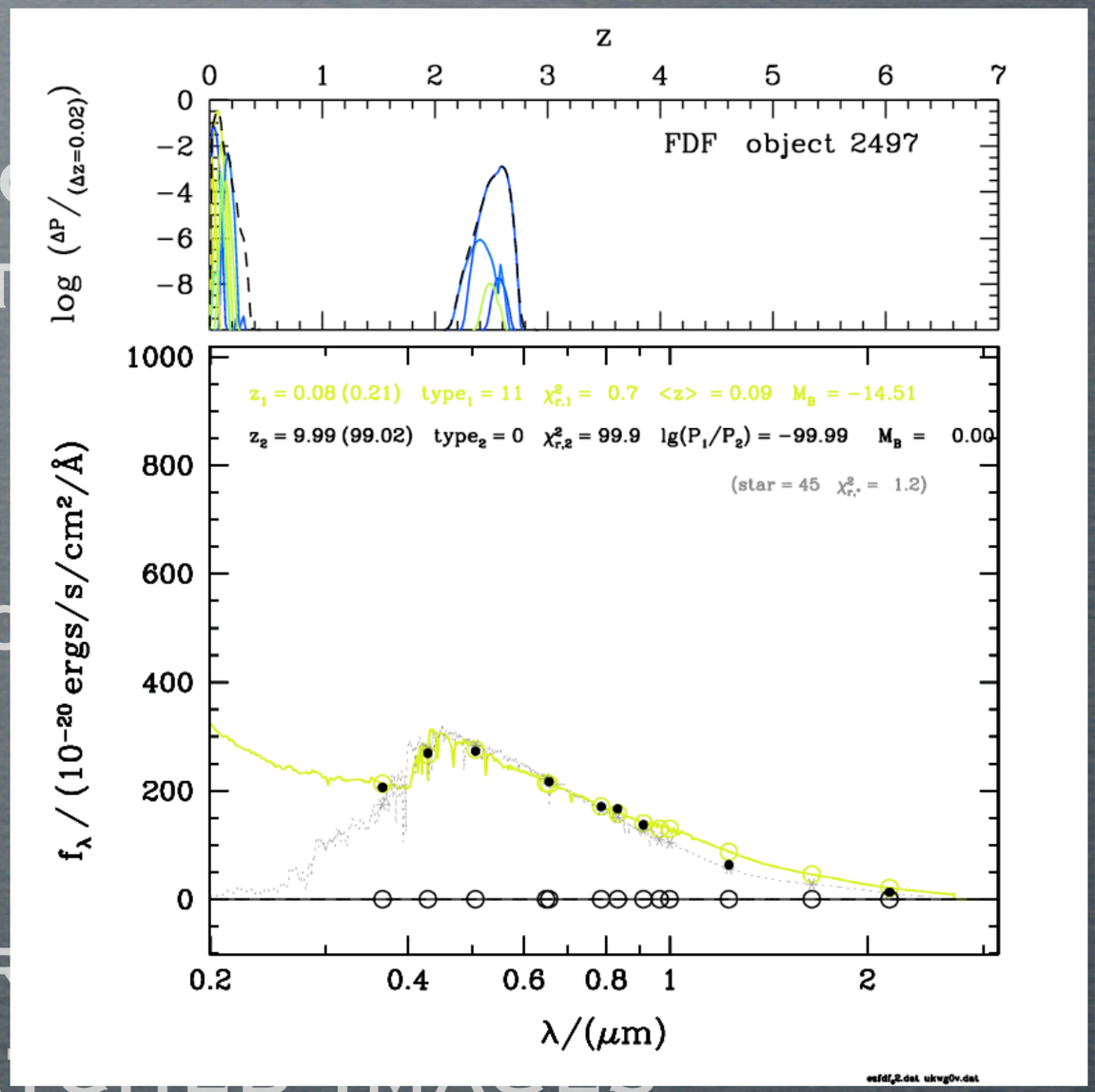
```
awe> from Experimental.PhotRedCatalog import *
# Select the config
awe> pc=(PhotRedConfig.name=='fors_demo')[0]
# Select the SourceLists
awe> sU=(SourceList.SLID==21)[0]
...
# Create the PhotRedCatalog
awe> pr = PhotRedCatalog()
pr.config=pc
pr.master=sI
pr.sourcelists=[sU,sB,sg,sR,sI,sJ,sK,sz]
pr.extinc=[-0.35,-0.15,-0.11,0.06,0.19,-0.09,-0.120,1.03]
pr.model_error=[0.07,0.03,0.03,0.03,0.05,0.15,0.3,0.1]
pr.mag='MAG_APER'
pr.flux='FLUX_APER'
pr.fluxerr='FLUXERR_APER'
pr.make()
```

Caveats

- Adequate spectral coverage is needed, i.e. at least 4-5 filters, including infrared
- SEDs degenerate for certain redshifts
- Only works on aperture photometry obtained on PSF matched images

Caveats


z
FDF object 4940
log (ΔP/(Δz=0.02))
z1 = 2.34 (0.15) type1 = 25 χ²r,1 = 0.8 <z> = 2.27 MB = −20.92
z2 = 0.40 (7.45) type2 = 25 χ²r,2 = 15.8 lg(P1/P2) = −0.00 MB = −16.87
(star = 0 χ²r,* = 99.9)
fλ / (10⁻²⁰ ergs/s/cm²/Å)
λ/(μm)
esfdf_2.dat uks0v.dat


z
FDF object 2497
log (ΔP/(Δz=0.02))
z1 = 0.08 (0.21) type1 = 11 χ²r,1 = 0.7 <z> = 0.09 MB = −14.51
z2 = 9.99 (99.02) type2 = 0 χ²r,2 = 99.9 lg(P1/P2) = −99.99 MB = 0.00
(star = 45 χ²r,* = 1.2)
fλ / (10⁻²⁰ ergs/s/cm²/Å)
λ/(μm)
esfdf_2.dat ukwg0v.dat

What's still missing

- Finish testing
- Add a name attribute
- Add a recipe
- Add plotting routine
- Import the datstar sourcelist

DEMO