BL_Lac_Two_Zone_EC

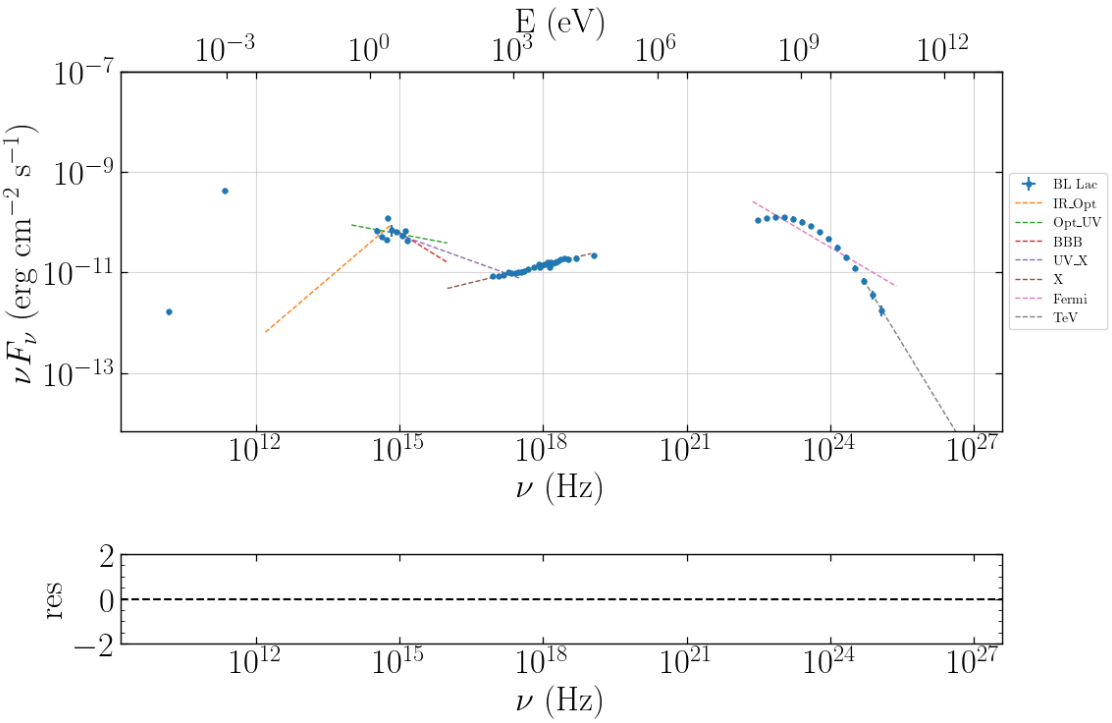
July 11, 2024

[ ]:
```
import matplotlib
matplotlib.rcParams['xtick.direction'] = 'in'
matplotlib.rcParams['ytick.direction'] = 'in'
matplotlib.rcParams['xtick.top'] = True
matplotlib.rcParams['ytick.right'] = True
matplotlib.rcParams['xtick.minor.visible'] = True
matplotlib.rcParams['ytick.minor.visible'] = True
matplotlib.rcParams['axes.linewidth'] = 1.0
matplotlib.rcParams['lines.linewidth'] = 1.5
matplotlib.rcParams['ytick.major.width'] = 0.9
matplotlib.rcParams['xtick.major.width'] = 0.9
matplotlib.rcParams['ytick.minor.width'] = 0.6
matplotlib.rcParams['xtick.minor.width'] = 0.6
matplotlib.rcParams['ytick.major.size'] = 5.0
matplotlib.rcParams['xtick.major.size'] = 5.0
matplotlib.rcParams['ytick.minor.size'] = 3.0
matplotlib.rcParams['xtick.minor.size'] = 3.0
matplotlib.rcParams["figure.dpi"] = 500
matplotlib.rcParams["figure.figsize"] = [5,4]
matplotlib.rcParams['axes.axisbelow'] = True

import numpy as np
import matplotlib
import pylab as plt
plt.rcParams.update({'text.usetex': True,'font.family': 'Proxima Nova','font.
 ↪size':25})
#plt.rcParams.update({'text.usetex': True,'font.family': 'sans-serif', 'font.
 ↪sans-serif': ['Latin Modern Sans'], 'font.size': 18})
plt.rcParams.update({'legend.fontsize':22})

import jetset
from jetset.data_loader import Data
from jetset.data_loader import ObsData, Data
from jetset.plot_sedfit import PlotSED
from jetset.test_data_helper import  test_SEDs
from jetset.sed_shaper import  SEDShape
import warnings
```

```
warnings.filterwarnings('ignore')
from jetset.minimizer import fit_SED, ModelMinimizer
from jetset.jet_model import Jet
from jetset.obs_constrain import ObsConstrain
from jetset.model_manager import  FitModel
from jetset.minimizer import ModelMinimizer
```

[ ]:
```
d=np.genfromtxt('/home/ayon-astro/BL Lac 2023/BL_Lac_2023_SED.txt')
data=Data(n_rows=d.shape[0])
data.set_field('x',d[:,0])
data.set_field('dx',value=d[:,1])
data.set_field('y',d[:,2])
data.set_field('dy',value=d[:,3])
data.set_meta_data('z',0.069)


#data.table
sed_data = ObsData(data_table= data, obj_name='BL Lac')
sed_data.group_data(bin_width= 0.01)

sed_data.add_systematics(0.1,[10.**6,10.**29])

my_shape=SEDShape(sed_data)

my_shape.eval_indices(minimizer='lsb',silent=True)

p=my_shape.plot_indices()
p.setlim(y_max=10E-8)
```

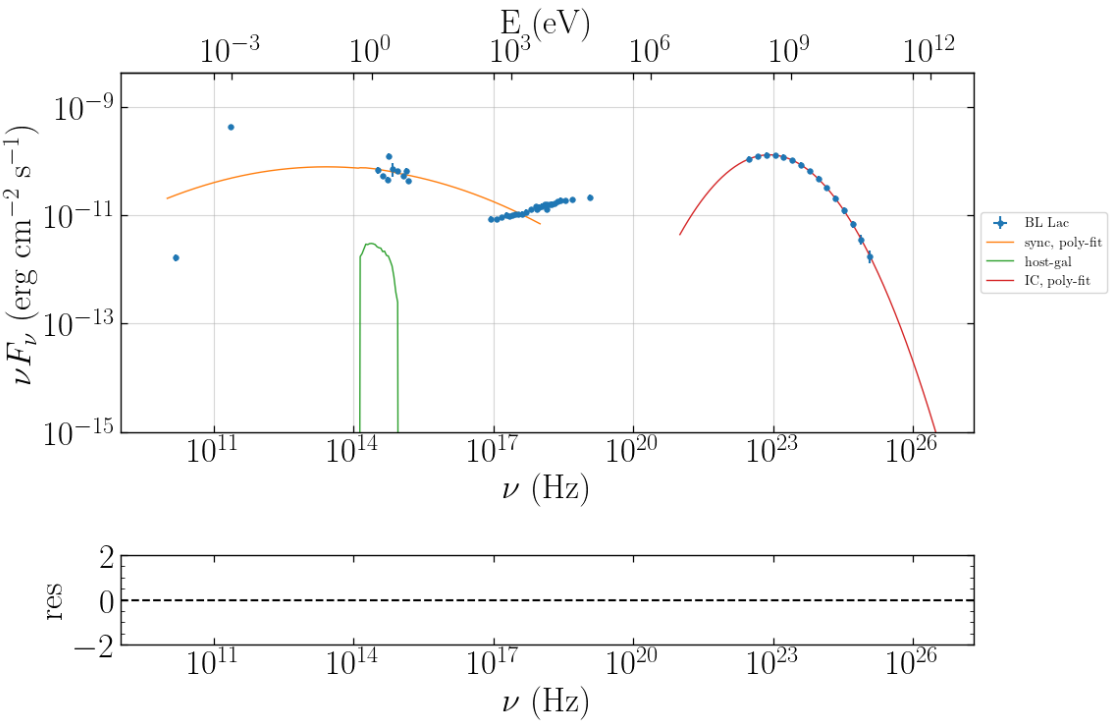
```
================================================================================

***  binning data  ***
---> N bins= 1525
---> bin_widht= 0.01
================================================================================

================================================================================

*** evaluating spectral indices for data ***
================================================================================
```



[ ]:
```
mm,best_fit=my_shape.sync_fit(check_host_gal_template=True,
                  use_log_par=True,
                  Ep_start=None,
                  minimizer='lsb',
                  silent=True,
                  fit_range=[10,18])

my_shape.IC_fit(fit_range=[21,27.],minimizer='minuit',silent=True)

p=my_shape.plot_shape_fit()
p.setlim(y_min=1E-15)
```

```
================================================================================

*** Log-Polynomial fitting of the synchrotron component ***
---> first blind fit run,  fit range: [10, 18]
--> class:  LSP

--> class:  LSP


<IPython.core.display.HTML object>

---> sync       nu_p=+1.341292e+01 (err=+4.235045e-01)  nuFnu_p=-1.010956e+01
```

```
(err=+1.258817e-01) curv.=-4.994971e-02 (err=+1.267424e-02)
================================================================================

================================================================================

*** Log-Polynomial fitting of the IC component ***
---> fit range: [21, 27.0]
---> LogCubic fit
====> simplex
====> migrad
====> simplex
====> migrad
====> simplex
====> migrad


<IPython.core.display.HTML object>

---> IC         nu_p=+2.291751e+01 (err=+6.394041e-02)  nuFnu_p=-9.888329e+00
(err=+2.103503e-02) curv.=-4.002335e-01 (err=+9.232541e-02)
================================================================================
```



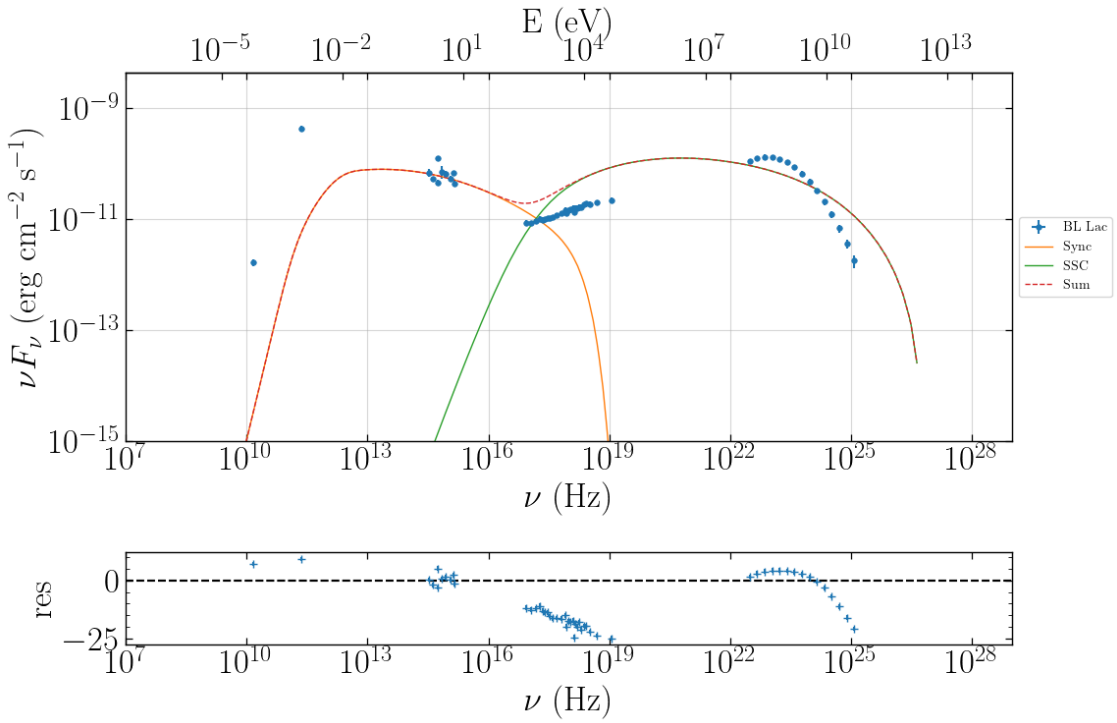
[ ]:
```
sed_obspar=ObsConstrain(beaming=25,
                        B_range=[0.1,1],
                        distr_e='lppl',
                        t_var_sec=3*86400,
                        nu_cut_IR=1E12,
                        SEDShape=my_shape)


prefit_jet=sed_obspar.
 ↪constrain_SSC_model(electron_distribution_log_values=False,silent=True)
prefit_jet.save_model('prefit_jet.pkl')
```

```
================================================================================

***  constrains parameters from observable ***

===> setting C threads to 4
```

<IPython.core.display.HTML object>

```
================================================================================
```

[ ]:
```
pl=prefit_jet.plot_model(sed_data=sed_data)
pl.add_model_residual_plot(prefit_jet,sed_data)
pl.setlim(y_min=1E-15,x_min=1E7,x_max=1E29)
```



[ ]:
```
compact_jet=Jet(name='compact_jet',beaming_expr='bulk_theta')
#compact_jet.add_EC_component(['EC_BLR'],disk_type='BB')
compact_jet.make_conical_jet()
compact_jet.set_EC_dependencies()
compact_jet.show_model()


extended_jet=compact_jet.clone()
extended_jet.name='extended_jet'
```

```
===> setting C threads to 4
adding par: R_H to  R
adding par: theta_open to  R
==> par R is depending on ['R_H', 'theta_open'] according to expr:   R =
np.tan(np.radians(theta_open))*R_H
setting R_H to 5.715026151380671e+16

----------------------------------------------------------------------------
model description:
----------------------------------------------------------------------------
type: Jet
name: compact_jet
geometry: spherical

electrons distribution:
 type: plc
 gamma energy grid size:  201
 gmin grid : 2.000000e+00
 gmax grid : 1.000000e+06
 normalization:  True
 log-values:  False
 ratio of cold protons to relativistic electrons: 1.000000e+00

radiative fields:
 seed photons grid size:  100
 IC emission grid size:  100
 source emissivity lower bound :  1.000000e-120
 spectral components:
   name:Sum, state: on
   name:Sum, hidden: False
   name:Sync, state: self-abs
   name:Sync, hidden: False
   name:SSC, state: on
   name:SSC, hidden: False
external fields transformation method: blob
```

```
SED info:
 nu grid size jetkernel: 1000
 nu size: 500
 nu mix (Hz): 1.000000e+06
 nu max (Hz): 1.000000e+30

flux plot lower bound   :  1.000000e-30

--------------------------------------------------------------------------------

<IPython.core.display.HTML object>

--------------------------------------------------------------------------------
===> setting C threads to 4
adding par: R_H to  R
adding par: theta_open to  R
==> par R is depending on ['R_H', 'theta_open'] according to expr:   R =
np.tan(np.radians(theta_open))*R_H
```

[ ]:
```
prefit_jet=Jet.load_model('prefit_jet.pkl')
```

```
===> setting C threads to 4
```

[ ]:
```
composite_jet=FitModel(nu_size=500, name='composite_jet')
composite_jet.add_component(compact_jet)
composite_jet.add_component(extended_jet)
#composite_jet.add_component(prefit_jet)
composite_jet.composite_expr='extended_jet+compact_jet'
composite_jet.show_pars()
```

```
<IPython.core.display.HTML object>
```

[ ]:
```
'''
composite_model=FitModel(nu_size=500,name='EBL corrected',template=my_shape.
 ↪host_gal)
composite_model.add_component(prefit_jet)
composite_model.add_component(ebl_franceschini)
composite_model.link_par(par_name='z_cosm', from_model='Franceschini_2008',␣
 ↪to_model='jet_leptonic')
composite_model.composite_expr='(jet_leptonic+host_galaxy)*Franceschini_2008'
composite_model.eval()
composite_model.plot_model()
'''
```

[ ]:
```
" \ncomposite_model=FitModel(nu_size=500,name='EBL corrected',template=my_shape.
host_gal)\ncomposite_model.add_component(prefit_jet)\ncomposite_model.add_compon
ent(ebl_franceschini)\ncomposite_model.link_par(par_name='z_cosm',
from_model='Franceschini_2008', to_model='jet_leptonic')\ncomposite_model.compos
```

```
ite_expr='(jet_leptonic+host_galaxy)*Franceschini_2008'\ncomposite_model.eval()\
ncomposite_model.plot_model()\n"
```

[ ]:
```
linked_pars=['z_cosm','theta_open','theta','T_Disk','L_Disk','T_DT']
for par in linked_pars:
    composite_jet.
 ↪link_par(par_name=par,from_model='extended_jet',to_model='compact_jet')
```

```
adding par: z_cosm to  z_cosm
adding par: theta_open to  theta_open
adding par: theta to  theta
```

[ ]:
```
#composite_jet.extended_jet.spectral_components.DT.hidden=True
#composite_jet.extended_jet.spectral_components.Disk.hidden=True
```

[ ]:
```
composite_jet.show_pars()
```

```
<IPython.core.display.HTML object>
```

[ ]:
```
'''
composite_jet.compact_jet.parameters.R_H.val=1E18
composite_jet.extended_jet.parameters.R_H.val=5E19
composite_jet.extended_jet.parameters.gamma_cut.val=1E3
composite_jet.extended_jet.parameters.B.val=0.1
composite_jet.extended_jet.parameters.N.val=1
#composite_jet.eval()
composite_jet.extended_jet.spectral_components.DT.hidden=True
composite_jet.extended_jet.spectral_components.Disk.hidden=True
composite_jet.eval()
p=composite_jet.plot_model(skip_components=False)
p.setlim(y_min=1E-16)
'''
```

[ ]:
```
' \ncomposite_jet.compact_jet.parameters.R_H.val=1E18\ncomposite_jet.extended_je
t.parameters.R_H.val=5E19\ncomposite_jet.extended_jet.parameters.gamma_cut.val=1
E3\ncomposite_jet.extended_jet.parameters.B.val=0.1\ncomposite_jet.extended_jet.
parameters.N.val=1\n#composite_jet.eval()\ncomposite_jet.extended_jet.spectral_c
omponents.DT.hidden=True\ncomposite_jet.extended_jet.spectral_components.Disk.hi
dden=True\ncomposite_jet.eval()\np=composite_jet.plot_model(skip_components=Fals
e)\np.setlim(y_min=1E-16)\n'
```

[ ]:
```
#composite_jet.freeze('jet_leptonic','z_cosm')
#composite_jet.free('jet_leptonic','R_H')
#composite_jet.jet_leptonic.parameters.beam_obj.fit_range=[5., 50.]
#composite_jet.jet_leptonic.parameters.R.fit_range=[10**15.5,10**17.5]
#composite_jet.jet_leptonic.parameters.gmax.fit_range=[1E4,1E6]
```

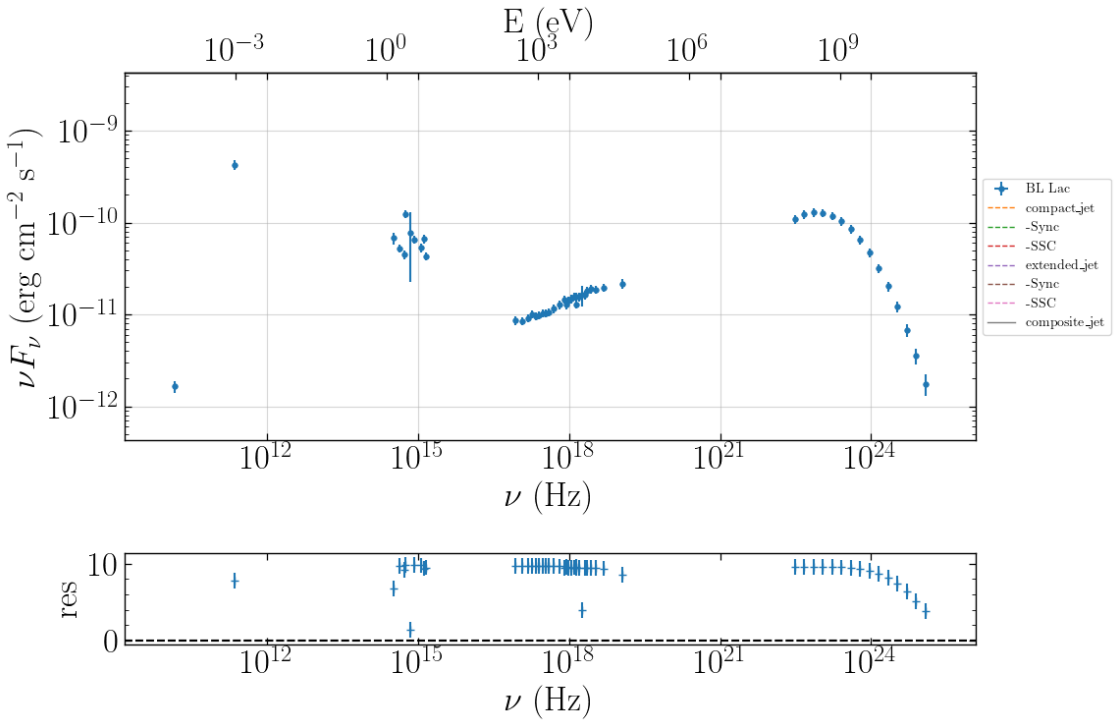
```
composite_jet.compact_jet.parameters.R_H.fit_range=[10**16, 10**18]
composite_jet.compact_jet.parameters.R.fit_range=[10**15, 10**17]

composite_jet.compact_jet.parameters.B.fit_range=[0.1,1]
composite_jet.compact_jet.parameters.gmax.fit_range=[1E5,1E6]

composite_jet.extended_jet.parameters.gmax.fit_range=[1E6,10**6.5]
composite_jet.extended_jet.parameters.R_H.fit_range_min=10**18.5
#composite_jet.extended_jet.parameters.R.fit_range=[10**16, 10**20]
composite_jet.extended_jet.parameters.gamma_cut.val=1E3
composite_jet.extended_jet.parameters.B.fit_range=[0.01,0.5]
#composite_jet.extended_jet.parameters.N.val=1


#composite_jet.host_galaxy.parameters.nuFnu_p_host.frozen=False
#composite_jet.host_galaxy.parameters.nu_scale.frozen=True
#composite_jet.jet_leptonic.parameters.z_cosm.frozen=True

model_minimizer_lsb=ModelMinimizer('minuit')
best_fit=model_minimizer_lsb.
 ↪fit(composite_jet,sed_data,1E11,1E29,fitname='SSC-best-fit-minuit',repeat=2)
```

```
filtering data in fit range = [1.000000e+11,1.000000e+29]
data length 51
================================================================================

*** start fit process ***
-----
fit run: 0

0it [00:00, ?it/s]

====> simplex
====> migrad
```

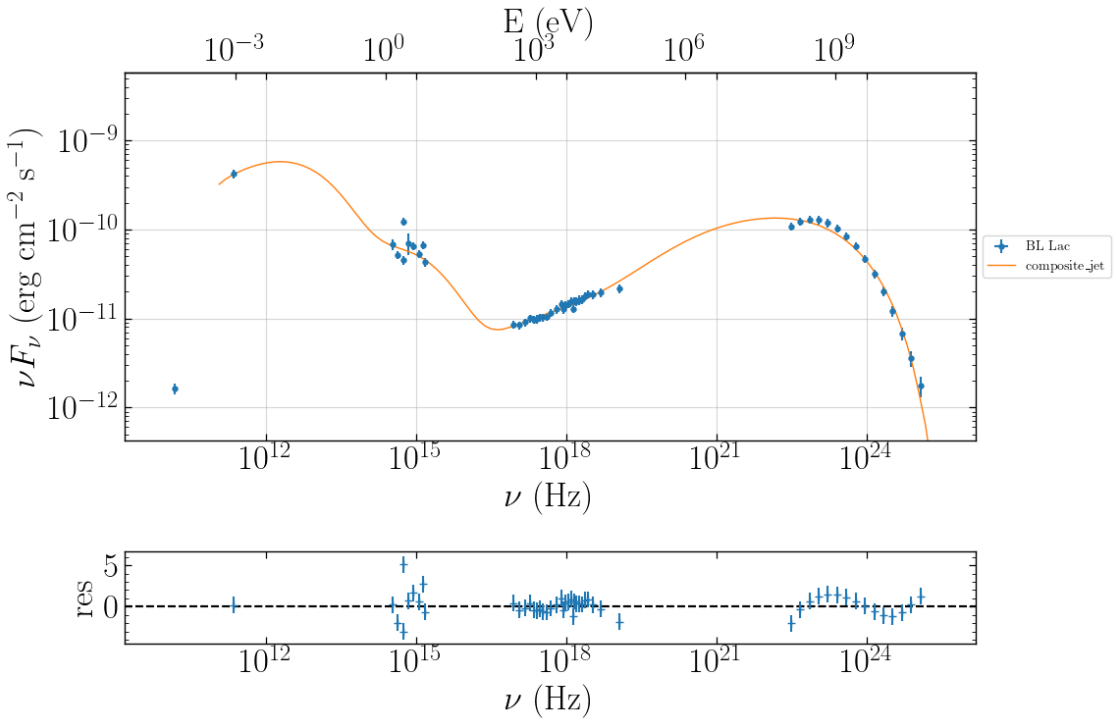
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[15], line 26
     18 #composite_jet.extended_jet.parameters.N.val=1
     19
     20
     21 #composite_jet.host_galaxy.parameters.nuFnu_p_host.frozen=False
     22 #composite_jet.host_galaxy.parameters.nu_scale.frozen=True
     23 #composite_jet.jet_leptonic.parameters.z_cosm.frozen=True
     25 model_minimizer_lsb=ModelMinimizer('minuit')
---> 26␣
 ↪best_fit=model_minimizer_lsb.fit(composite_jet,sed_data,1E11,1E29,fitname='SS-best-fit-mi
```

```
File ~/mambaforge/envs/jetset_latest/lib/python3.10/site-packages/jetset/
 ↪minimizer.py:465, in ModelMinimizer.fit(self, fit_model, sed_data,␣
 ↪nu_fit_start, nu_fit_stop, fitname, fit_workplace, loglog, silent,␣
 ↪get_conf_int, max_ev, use_fake_err, use_UL, skip_minimizer, repeat)
    463     self.asymm_errors = self.minimizer.asymm_errors
    464 self.covar=self.minimizer.covar
--> 465 self.corr=self.minimizer.corr
    467 self.reset_to_best_fit()
    468 self.minimizer._fit_stats()

File ~/mambaforge/envs/jetset_latest/lib/python3.10/site-packages/jetset/
 ↪minimizer.py:699, in Minimizer.corr(self)
    696 @property
    697 def corr(self):
    698     if self.covar is not None:
--> 699         v = np.sqrt(np.diag(self.covar))
    700         outer_v = np.outer(v, v)
    701         correlation = self.covar / outer_v

File ~/mambaforge/envs/jetset_latest/lib/python3.10/site-packages/numpy/lib/
 ↪twodim_base.py:303, in diag(v, k)
    301     return diagonal(v, k)
    302 else:
--> 303     raise ValueError("Input must be 1- or 2-d.")

ValueError: Input must be 1- or 2-d.
```

[ ]:
```
composite_jet.set_nu_grid(1E6,1E30,200)
composite_jet.eval()
p=composite_jet.plot_model(sed_data=sed_data, skip_components = False)
p.save('BL_Lac_Two_Zone_Model_ayon.png')
```



[ ]:
```
composite_jet.set_nu_grid(1E6,1E30,200)
composite_jet.eval()
p=composite_jet.plot_model(sed_data=sed_data, skip_components = compact_jet)
```



[ ]: