

## Using\_pyCloudy\_3

June 2, 2020

```
[1]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import os
home_dir = os.environ['HOME'] + '/'
import pyCloudy as pc
print(pc.__version__)
```

0.9.10b1

```
[2]: pc.config.cloudy_exe = '/usr/local/Cloudy/c17.02/source/cloudy.exe'
```

```
[3]: dir_ = '/tmp/models/'
pc.print_make_file(dir_)
```

```
[4]: def set_models(dir_, model_name):
    emis_tab = ['H 1 4861.33A',
                'H 1 6562.81A',
                'Ca B 8500.35A',
                'N 2 6583.45A',
                'O 1 6300.30A',
                'O 2 3726.03A',
                'O 2 3728.81A',
                'O 3 5006.84A',
                'BLND 4363.00A'
    ]
    emis_tab_c13 = ['H 1 4861',
                    'H 1 6563',
                    'He 1 5876',
                    'N 2 6584',
                    'O 1 6300',
                    'O II 3726',
                    'O II 3729',
                    'O 3 5007',
                    'TOTL 4363',
                    'O 1 63.17m',
                    'O 1 145.5m',
                    'C 2 157.6m',
```

```

                'H 1 4.051m']

a = 2.
b = 1.0
thetas = np.linspace(0., 90., 6)
thetas_rad = np.pi / 180. * thetas
fact_elli = a * b / np.sqrt((b * np.sin(thetas_rad))**2 + (a * np.
↪cos(thetas_rad))**2)
rs_in = 16.5 + np.log10(fact_elli)
densities = 4 - np.log10(fact_elli) * 2

model = pc.CloudyInput()
model.set_BB(80000., 'q(H)', 47.3)
model.set_grains()
model.set_emis_tab(emis_tab)

for theta, r_in, density in zip(thetas, rs_in, densities):
    model.model_name = '{0}/{1}_{2:.0f}'.format(dir_, model_name, theta)
    model.set_cste_density(density)
    model.set_radius(r_in)
    model.set_theta_phi(theta)
    model.print_input(to_file = True, verbose = False)

```

```

[5]: def def_profiles(m3d):
    """
    This uses the default velocity law (polynome) and default profile (gaussian)
    """
    m3d.set_velocity(params = [20.,60.])
    m3d.config_profile(size_spectrum = 51, vel_max = 50, v_turb = 0.01)

```

```

[6]: def def_profiles_user(m3d):
    """
    Use this to define your own expansion velocity
    """
    def velo_polynome(params):
        """
        User defined expansion velocity
        """
        # params is a 2 elements table, the first element is a table of
        ↪parameters, the second one the cob_coord
        # which is needed to know r, x, y and z to define the velocity.
        coeffs = params[0]
        cub_coord = params[1]
        tmp = 0.
        for i, coeff in enumerate(coeffs):
            # for each parameter we add the corresponding coeff * R**power
            tmp = tmp + coeff * cub_coord.r**i
        tmp = tmp / cub_coord.r

```

```

    # to avoid the singularity:
    tt = (cub_coord.r == 0.)
    tmp[tt] = 0
    # Projecting on each one of the 3 axes to obtain the velocity components
    vel_x = tmp * cub_coord.x / np.max(cub_coord.x)
    vel_y = tmp * cub_coord.y / np.max(cub_coord.y)
    vel_z = tmp * cub_coord.z / np.max(cub_coord.z)
    return vel_x, vel_y, vel_z

def Hb_prof(x, zeta_0):
    """
    The Hbeta profile is sum of 2 blocks of lines (actually 3 + 4 lines)
    """
    res1 = .41 / zeta_0 / np.sqrt(np.pi) * np.exp(-(((x-2.7)/zeta_0)**2))
    res2 = .59 / zeta_0 / np.sqrt(np.pi) * np.exp(-(((x+2.0)/zeta_0)**2))
    return res1 + res2

m3d.set_velocity(velocity_law='user', params = [[20.,60.], m3d.cub_coord],
↪user_function = velo_polynome)
m3d.config_profile(size_spectrum = 41, vel_max = 25, profile_function =
↪Hb_prof, v_turb = 0.01)

```

```

[7]: def plot_profiles(m3d, x_pos, y_pos):
    plt.plot(m3d.vel_tab,m3d.get_profile('H__1_486133A', axis='x')[:
↪,x_pos,y_pos] * 5, label = r'H$\beta$')
    plt.plot(m3d.vel_tab,m3d.get_profile('N__2_658345A', axis='x')[:
↪,x_pos,y_pos] * 5, label = r'[NII]$\\lambda$6584')
    plt.plot(m3d.vel_tab,m3d.get_profile('O__3_500684A', axis='x')[:
↪,x_pos,y_pos], label = r'[OIII]$\\lambda$5007')
    plt.legend()

```

```

[8]: def other_plots(m3d, proj_axis):
    plt.subplot(331)
    plt.imshow(m3d.get_emis('H__1_486133A').sum(axis = proj_axis)*m3d.cub_coord.
↪cell_size)
    plt.title('Hb')
    plt.colorbar()

    plt.subplot(332)
    plt.imshow(m3d.get_emis('N__2_658345A').sum(axis = proj_axis)*m3d.cub_coord.
↪cell_size)
    plt.title('[NII]')
    plt.colorbar()

    plt.subplot(333)

```

```

plt.imshow(m3d.get_emis('O__3_500684A').sum(axis = proj_axis)*m3d.cub_coord.
→cell_size)
plt.title(' [OIII] ')
plt.colorbar()

plt.subplot(334)
plt.imshow(m3d.get_emis('N__2_658345A').sum(axis = proj_axis)/m3d.
→get_emis('H__1_486133A').sum(axis = proj_axis))
plt.title(' [NII]/Hb ')
plt.colorbar()

plt.subplot(335)
plt.imshow(m3d.get_emis('O__3_500684A').sum(axis = proj_axis)/m3d.
→get_emis('H__1_486133A').sum(axis = proj_axis))
plt.title(' [OIII]/Hb ')
plt.colorbar()

plt.subplot(336)
plt.imshow(m3d.get_ionic('O',1)[n_cut,:,:])
plt.title('O+ cut')
plt.colorbar()

plt.subplot(337)
plt.scatter(m3d.get_ionic('O',1).ravel(),m3d.get_ionic('N',1).ravel()/m3d.
→get_ionic('O',1).ravel(),
               c=np.abs(m3d.cub_coord.theta.ravel()), edgecolors = 'none')
plt.title('Colored by |Theta|')
plt.xlabel('O+ / O')
plt.ylabel('N+/O+ / N/O')
plt.colorbar()

plt.subplot(338)
plt.scatter(m3d.get_ionic('O',1).ravel(),m3d.get_ionic('N',1).ravel()/m3d.
→get_ionic('O',1).ravel(),
               c=m3d.relative_depth.ravel(),vmin = 0, vmax = 1, edgecolors =_
→'none')
plt.title('Colored by position in the nebula')
plt.xlabel('O+ / O')
plt.ylabel('N+/O+ / N/O')
plt.colorbar()

plt.subplot(339)
C1 = (m3d.get_ionic('N',1)/m3d.get_ionic('O',1)*m3d.get_ionic('N',2))
C2 = (m3d.get_ionic('N',2))
tt = (m3d.get_ionic('O',1) == 0)
C1[tt] = 0

```

```

C2[tt] = 0
V = C1.sum(axis = proj_axis) / C2.sum(axis = proj_axis)
plt.imshow(V)
plt.colorbar()
plt.title('N+/O+ / N/O weighted by NII')
plt.contour(V, levels=[1.0])

```

```

[9]: model_name = "M3D_1"
pc.log_.calling = 'Model3D : ' + model_name
pc.log_.level = 3

```

```

[10]: dim = 101
n_cut = int((dim-1) / 2)
proj_axis = 0

```

```

[13]: set_models(dir_, model_name)

```

```

CloudyInput: Input written in /tmp/models//M3D_1_0.in
CloudyInput: Input written in /tmp/models//M3D_1_18.in
CloudyInput: Input written in /tmp/models//M3D_1_36.in
CloudyInput: Input written in /tmp/models//M3D_1_54.in
CloudyInput: Input written in /tmp/models//M3D_1_72.in
CloudyInput: Input written in /tmp/models//M3D_1_90.in

```

```

[14]: pc.print_make_file(dir_ = dir_)
pc.run_cloudy(dir_ = dir_, n_proc = 6, model_name = model_name, use_make = True)

```

```

run_cloudy: running: cd /tmp/models/ ; make -j 6 name="M3D_1"
run_cloudy: ending: cd /tmp/models/ ; make -j 6 name="M3D_1"

```

```

[11]: liste_of_models = pc.load_models('{0}/{1}'.format(dir_, model_name),
↳ list_elem=['H', 'He', 'C', 'N', 'O', 'Ar', 'Ne'],
read_cont = False, read_grains =
↳ False)

```

```

CloudyModel /tmp/models/M3D_1_18: Creating CloudyModel for
/tmp/models/M3D_1_18
CloudyModel /tmp/models/M3D_1_18: Be abundance not defined
CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.rad read
CloudyModel /tmp/models/M3D_1_18: Number of zones: 180
CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.phy read
CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_H read
CloudyModel /tmp/models/M3D_1_18: filling H with 3 columns
CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_He read
CloudyModel /tmp/models/M3D_1_18: filling He with 3 columns
CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_C read
CloudyModel /tmp/models/M3D_1_18: filling C with 13 columns
CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_N read

```

```

CloudyModel /tmp/models/M3D_1_18: filling N with 8 columns
CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_0 read
CloudyModel /tmp/models/M3D_1_18: filling 0 with 12 columns
CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_Ar read
CloudyModel /tmp/models/M3D_1_18: filling Ar with 19 columns
CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_Ne read
CloudyModel /tmp/models/M3D_1_18: filling Ne with 11 columns
CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.emis read
CloudyModel /tmp/models/M3D_1_18: Number of emissivities: 9
CloudyModel /tmp/models/M3D_1_36: Creating CloudyModel for
/tmp/models/M3D_1_36
CloudyModel /tmp/models/M3D_1_36: Be abundance not defined
CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.rad read
CloudyModel /tmp/models/M3D_1_36: Number of zones: 180
CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.phy read
CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_H read
CloudyModel /tmp/models/M3D_1_36: filling H with 3 columns
CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_He read
CloudyModel /tmp/models/M3D_1_36: filling He with 3 columns
CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_C read
CloudyModel /tmp/models/M3D_1_36: filling C with 13 columns
CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_N read
CloudyModel /tmp/models/M3D_1_36: filling N with 8 columns
CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_0 read
CloudyModel /tmp/models/M3D_1_36: filling 0 with 12 columns
CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_Ar read
CloudyModel /tmp/models/M3D_1_36: filling Ar with 19 columns
CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_Ne read
CloudyModel /tmp/models/M3D_1_36: filling Ne with 11 columns
CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.emis read
CloudyModel /tmp/models/M3D_1_36: Number of emissivities: 9
CloudyModel /tmp/models/M3D_1_90: Creating CloudyModel for
/tmp/models/M3D_1_90
CloudyModel /tmp/models/M3D_1_90: Be abundance not defined
CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.rad read
CloudyModel /tmp/models/M3D_1_90: Number of zones: 176
CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.phy read
CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_H read
CloudyModel /tmp/models/M3D_1_90: filling H with 3 columns
CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_He read
CloudyModel /tmp/models/M3D_1_90: filling He with 3 columns
CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_C read
CloudyModel /tmp/models/M3D_1_90: filling C with 13 columns
CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_N read
CloudyModel /tmp/models/M3D_1_90: filling N with 8 columns
CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_0 read
CloudyModel /tmp/models/M3D_1_90: filling 0 with 12 columns
CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_Ar read

```

```

CloudyModel /tmp/models/M3D_1_90: filling Ar with 19 columns
CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_Ne read
CloudyModel /tmp/models/M3D_1_90: filling Ne with 11 columns
CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.emis read
CloudyModel /tmp/models/M3D_1_90: Number of emissivities: 9
CloudyModel /tmp/models/M3D_1_0: Creating CloudyModel for
/tmp/models/M3D_1_0
CloudyModel /tmp/models/M3D_1_0: Be abundance not defined
CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.rad read
CloudyModel /tmp/models/M3D_1_0: Number of zones: 182
CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.phy read
CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_H read
CloudyModel /tmp/models/M3D_1_0: filling H with 3 columns
CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_He read
CloudyModel /tmp/models/M3D_1_0: filling He with 3 columns
CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_C read
CloudyModel /tmp/models/M3D_1_0: filling C with 13 columns
CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_N read
CloudyModel /tmp/models/M3D_1_0: filling N with 8 columns
CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_O read
CloudyModel /tmp/models/M3D_1_0: filling O with 12 columns
CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_Ar read
CloudyModel /tmp/models/M3D_1_0: filling Ar with 19 columns
CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_Ne read
CloudyModel /tmp/models/M3D_1_0: filling Ne with 11 columns
CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.emis read
CloudyModel /tmp/models/M3D_1_0: Number of emissivities: 9
CloudyModel /tmp/models/M3D_1_54: Creating CloudyModel for
/tmp/models/M3D_1_54
CloudyModel /tmp/models/M3D_1_54: Be abundance not defined
CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.rad read
CloudyModel /tmp/models/M3D_1_54: Number of zones: 179
CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.phy read
CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_H read
CloudyModel /tmp/models/M3D_1_54: filling H with 3 columns
CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_He read
CloudyModel /tmp/models/M3D_1_54: filling He with 3 columns
CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_C read
CloudyModel /tmp/models/M3D_1_54: filling C with 13 columns
CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_N read
CloudyModel /tmp/models/M3D_1_54: filling N with 8 columns
CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_O read
CloudyModel /tmp/models/M3D_1_54: filling O with 12 columns
CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_Ar read
CloudyModel /tmp/models/M3D_1_54: filling Ar with 19 columns
CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_Ne read
CloudyModel /tmp/models/M3D_1_54: filling Ne with 11 columns
CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.emis read

```

```

CloudyModel /tmp/models/M3D_1_54: Number of emissivities: 9
CloudyModel /tmp/models/M3D_1_72: Creating CloudyModel for
/tmp/models/M3D_1_72
CloudyModel /tmp/models/M3D_1_72: Be abundance not defined
CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.rad read
CloudyModel /tmp/models/M3D_1_72: Number of zones: 177
CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.phy read
CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_H read
CloudyModel /tmp/models/M3D_1_72: filling H with 3 columns
CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_He read
CloudyModel /tmp/models/M3D_1_72: filling He with 3 columns
CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_C read
CloudyModel /tmp/models/M3D_1_72: filling C with 13 columns
CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_N read
CloudyModel /tmp/models/M3D_1_72: filling N with 8 columns
CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_O read
CloudyModel /tmp/models/M3D_1_72: filling O with 12 columns
CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_Ar read
CloudyModel /tmp/models/M3D_1_72: filling Ar with 19 columns
CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_Ne read
CloudyModel /tmp/models/M3D_1_72: filling Ne with 11 columns
CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.emis read
CloudyModel /tmp/models/M3D_1_72: Number of emissivities: 9
load_models: 6 models read

```

```

[12]: M=liste_of_models[0]
      M.emis_labels

```

```

[12]: array(['H__1_486133A', 'H__1_656281A', 'CA_B_587564A', 'N__2_658345A',
            'O__1_630030A', 'O__2_372603A', 'O__2_372881A', 'O__3_500684A',
            'BLND_436300A'], dtype='<U12')

```

```

[16]: m3d = pc.C3D(liste_of_models, dims = [dim, dim, dim], angles = [45,45,0],
      ↪ plan_sym = True)

```

```

C3D: Entering C3D
CubCoord: building a cube of 101x101x101
CubCoord: Rotation matrix by 45.0, 45.0, 0.0 degrees.
C3D: CubCoord done.
C3D: interp_bi done.
C3D: Interpolation mesh done
C3D: All 3D values reset

```

```

[17]: def_profiles(m3d)

```

```

[18]: plt.figure(figsize=(10,10))
      plot_profiles(m3d, 55, 55)

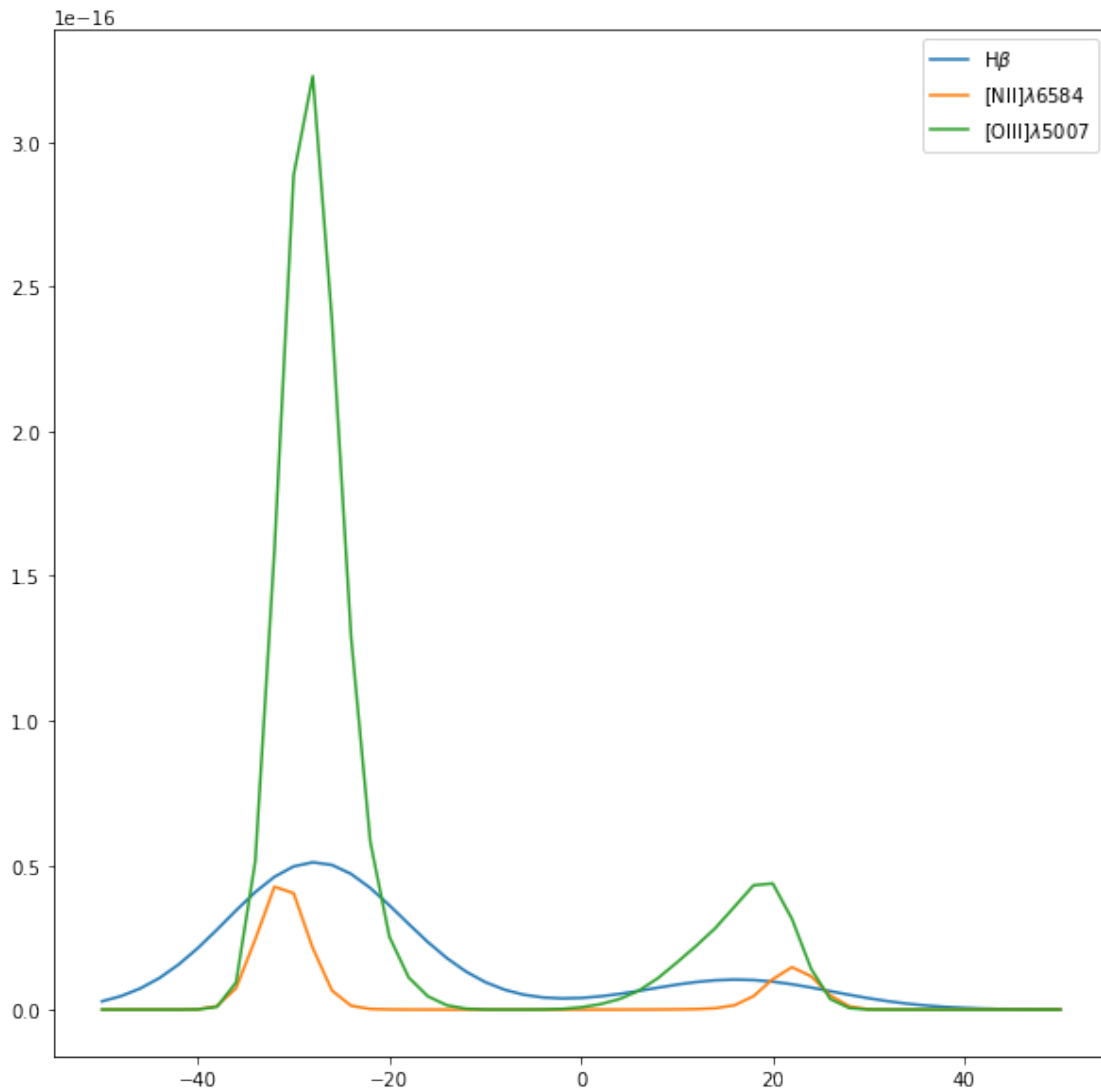
```



```

C3D: get_emis(0) interpolated using numpy-method
C3D: te interpolated using numpy-method
C3D: line H__1_486133A : profile computed on axis x
C3D: get_emis(3) interpolated using numpy-method
C3D: line N__2_658345A : profile computed on axis x
C3D: get_emis(7) interpolated using numpy-method
C3D: line O__3_500684A : profile computed on axis x

```



```

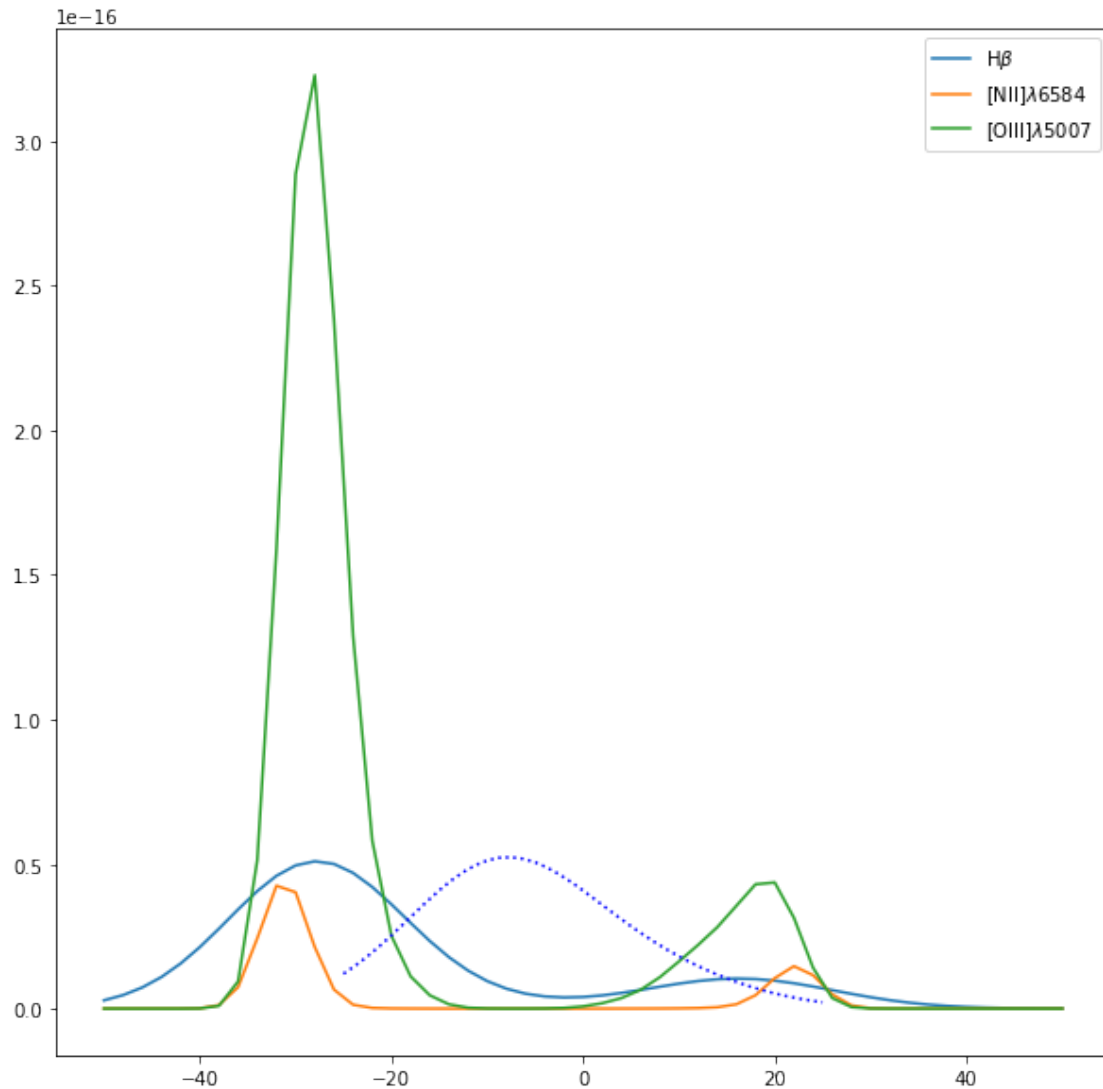
[19]: plt.figure(figsize=(10,10))
      plot_profiles(m3d, 55, 55)
      def_profiles_user(m3d)
      plt.plot(m3d.vel_tab,m3d.get_profile('H__1_486133A', axis='x')[:,55,55] * 5, ':
      ↪b', label = r'H$\beta$')

```

```
/Users/christophemorriset/anaconda3/lib/python3.7/site-  
packages/ipykernel_launcher.py:17: RuntimeWarning: divide by zero encountered in  
true_divide
```

```
C3D: line H__1_486133A : profile computed on axis x
```

```
[19]: [<matplotlib.lines.Line2D at 0x1a30e5ec50>]
```



```
[20]: plt.figure(figsize=(15,15))  
other_plots(m3d, proj_axis)
```

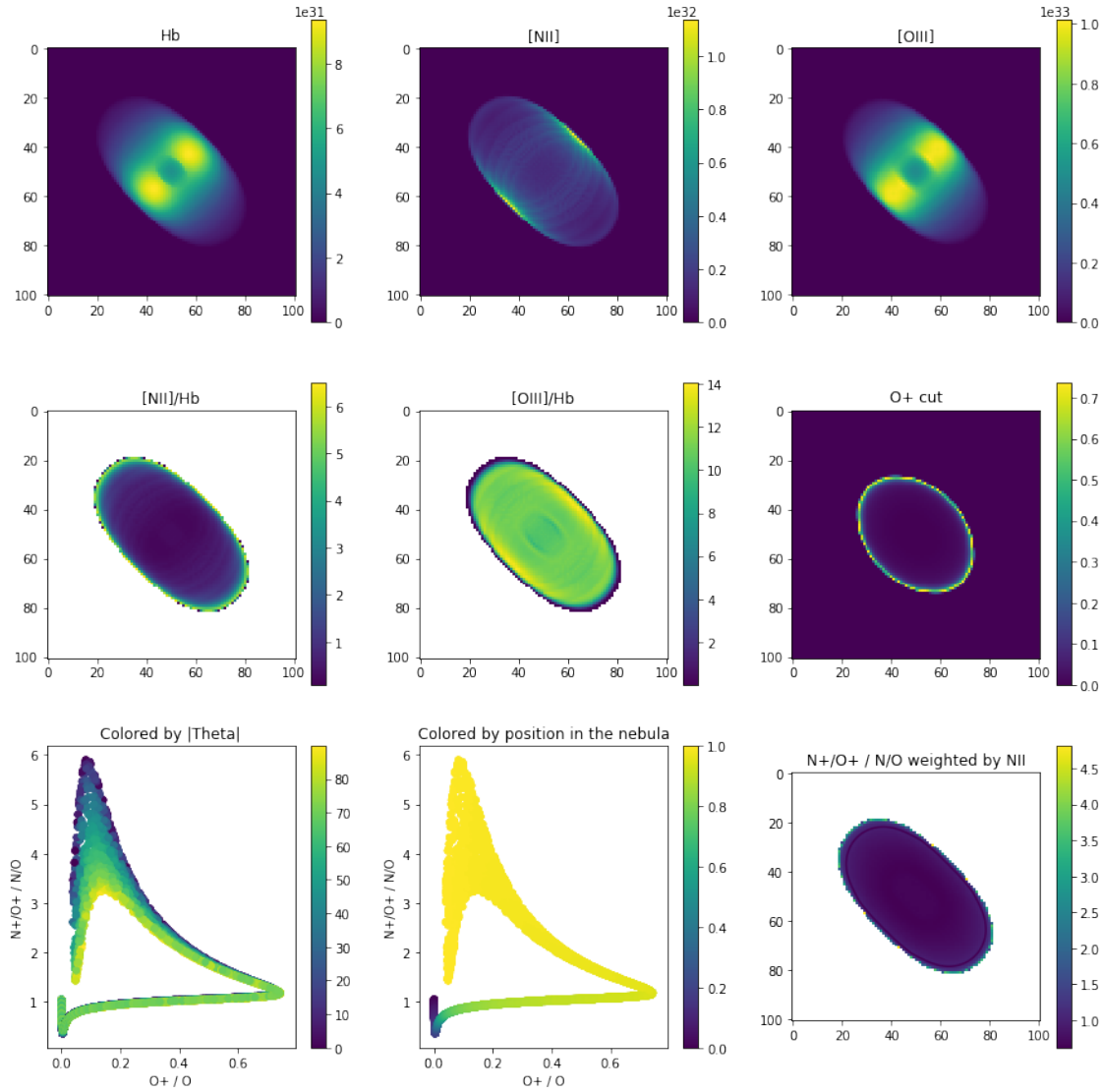
```
/Users/christophemorriset/anaconda3/lib/python3.7/site-  
packages/ipykernel_launcher.py:18: RuntimeWarning: invalid value encountered in  
true_divide  
/Users/christophemorriset/anaconda3/lib/python3.7/site-
```

```
packages/ipykernel_launcher.py:23: RuntimeWarning: invalid value encountered in
true_divide
/Users/christophemorrisset/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:33: RuntimeWarning: invalid value encountered in
true_divide

    C3D: get_ionic('O', 1) interpolated using numpy-method
    C3D: get_ionic('N', 1) interpolated using numpy-method

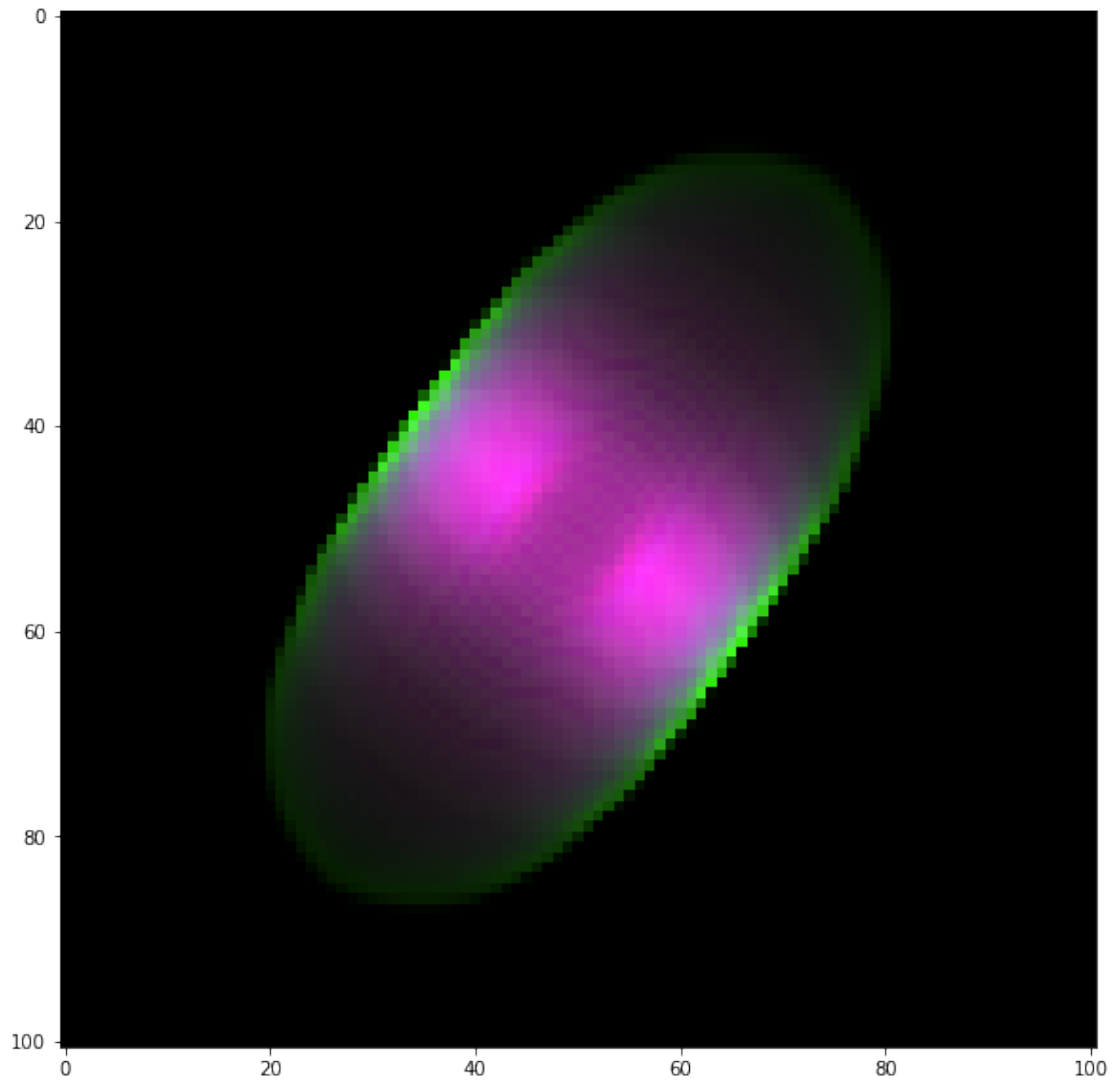
/Users/christophemorrisset/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:41: RuntimeWarning: invalid value encountered in
true_divide
/Users/christophemorrisset/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:49: RuntimeWarning: invalid value encountered in
true_divide
/Users/christophemorrisset/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:54: RuntimeWarning: invalid value encountered in
true_divide

    C3D: get_ionic('N', 2) interpolated using numpy-method
```



```
[21]: im = m3d.get_RGB(list_emis = [0, 3, 7])
plt.figure(1, figsize=(10,10))
plt.imshow(im)
```

```
[21]: <matplotlib.image.AxesImage at 0x1a281906d0>
```



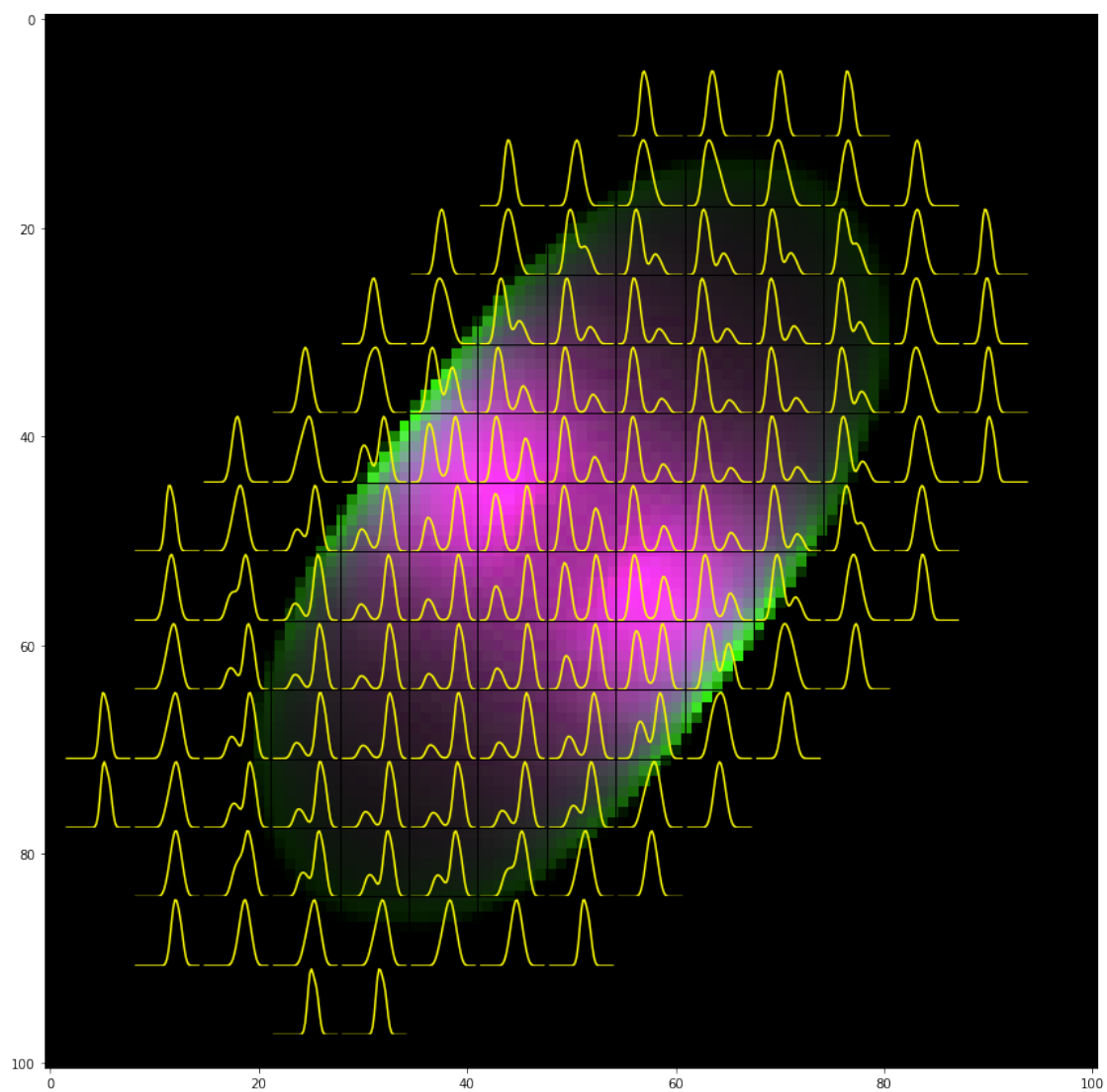
```
[22]: im = m3d.get_RGB(list_emis = [0, 3, 7])
plt.figure(1, figsize=(15,15))
plt.imshow(im)
m3d.plot_profiles(ref = 3, i_fig = 1, Nx=20, Ny=20)
```

```
C3D: line N__2_658345A : profile computed on axis x
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
  prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
  prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
```

```

RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)
/Users/christophemorisset/Dropbox/Python/pyCloudy/pyCloudy/c3d/model_3d.py:943:
RuntimeWarning: invalid value encountered in true_divide
    prof /= np.max(prof)

```



[ ]: