In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import matplotlib.patches as patches
import astropy.io.fits as fits
from astropy import units as u
from collections import defaultdict
from matplotlib.patches import  ConnectionPatch
```

In [ ]:
```
```

In [2]:
```
import DPConCFil
from DPConCFil.Clump_Class import *
from DPConCFil.Filament_Class import *
import DPConCFil.Plot_and_Save_Funs as Plot_and_Save_Funs
import DPConCFil.Profile_Funs as Profile_Funs
```

In [ ]:
```
```

The reference of the Example data

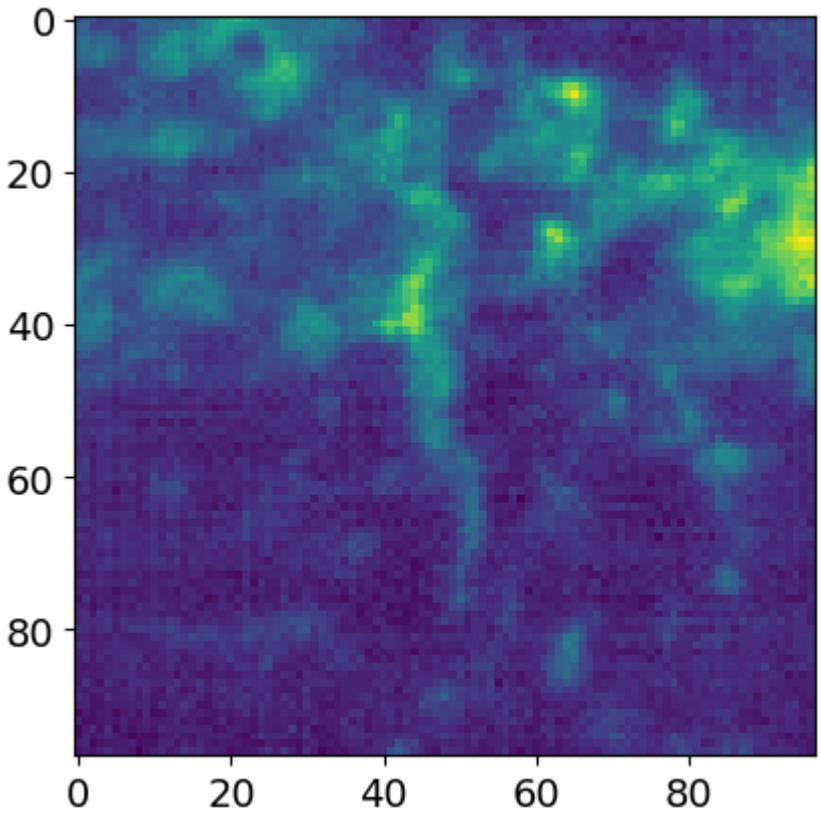
'Example_Filaments_13CO_1.fits' is the ^{13}CO ($J = 1 - 0$) emission line of the Milky Way Imaging Scroll Painting (MWISP) within $17.7^{\circ} \leq l \leq 18.5^{\circ}$, $0^{\circ} \leq b \leq 0.8^{\circ}$ and 5 km s^{-1} $\leq v \leq$ 30 km s^{-1}.

MWISP project is a multi-line survey in $^{12}CO/^{13}CO/C^{18}O$ along the northern galactic plane with PMO-13.7m telescope.

In [ ]:
```
```

In [3]:
```
file_example = 'Example_Filaments_13CO_1'
file_name = "../Example_Files/Data/{}.fits".format(file_example)
```

In [4]:
```
real_data = fits.getdata(file_name)
plt.imshow(real_data.sum(0))
plt.show()
```



In [ ]:

Calculate the clump information

The parameters of FacetClumps. Please see the introduction of FacetClumps for more details.

In [5]:
```
SWindow = 3 # [3,5,7]
KBins = 35 # [10,...,60]
FwhmBeam = 2
VeloRes = 2
SRecursionLBV = [9, 4] # [(2+FwhmBeam)**2,3+VeloRes]

header = fits.getheader(file_name)
RMS = header['RMS']
Threshold = 5 * RMS

parameters_FacetClumps = [RMS, Threshold, SWindow, KBins, FwhmBeam, VeloRes, SRe
```

In [ ]:

Construct clump objects. These file names are necessary parameters.

file_name: File name.

mask_name: Mask name, the file use to store the region information or store the region information.

outcat_name: The file used to store clump table in pixel coordinate system.

outcat_wcs_name: The file used to store clump table in WCS coordinate system.

In [ ]:

In [6]:
```
mask_name = 'Example_Files/Clump/mask_{}.fits'.format(file_example)
outcat_name = 'Example_Files/Clump/outcat_{}.csv'.format(file_example)
outcat_wcs_name = 'Example_Files/Clump/outcat_wcs_{}.csv'.format(file_example)
```

In [7]:
```
clumpsObj = ClumpInfor(file_name,mask_name,outcat_name,outcat_wcs_name)
```

In [ ]:

Calculate the clump information from FacetClumps.

In this case, the parameters of FacetClumps is essential. More clump detection algorithms can also be added to this process.

The angle of the clumps detected by FacetClumps is obtained by diagonalizing the moment of inertia matrix, please the article of FacetClumps for more details. Performing a two-dimensional single Gaussian fitting on the velocity integrated map of a clump can provide more accurate position and direction information of the clump in spatial direction.

When 'fit_flag=True', it indicates that the fitting will be used. This will benefit the performance of DPConFil.

In [ ]:

In [8]:
```
clumpsObj.Cal_Infor_From_Mask_Or_Algorithm(mask_or_algorithm='FacetClumps',param
clumpsObj.Get_Clumps_Infor(fit_flag = True)
```

```
100%|██████████| 63/63 [00:02<00:00, 26.35it/s]
100%|██████████| 39/39 [00:07<00:00,  5.26it/s]
100%|██████████| 39/39 [00:00<00:00, 177.40it/s]
Number: 126
Time: 10.97
100%|██████████| 126/126 [00:01<00:00, 71.80it/s]
Fitting Clumps Time: 1.77
```

In [ ]:

Calculate the clump information from the mask file 'mask_name'.

The mask is the region information of clumps, which can be obtained by any clump detection algorithm.

In [9]:
```
clumpsObj.Cal_Infor_From_Mask_Or_Algorithm(mask_or_algorithm='mask')
clumpsObj.Get_Clumps_Infor(fit_flag = True)
```

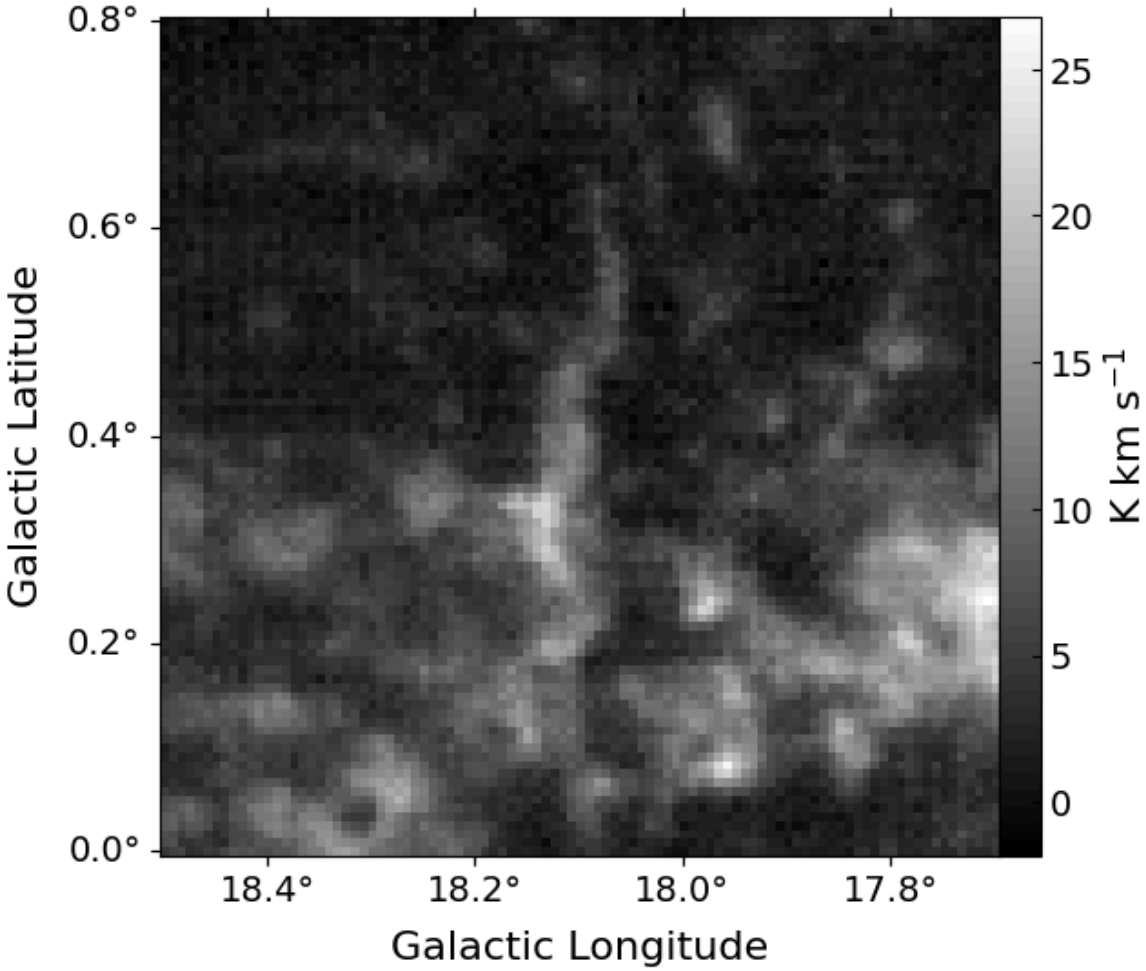
```
Number: 126
Time: 0.08
```

```
100%|██████████| 126/126 [00:01<00:00, 72.79it/s]
Fitting Clumps Time: 1.75
```

In [ ]:

Plot the original image. If save_path=None, the image will not be saved.

In [10]:
```
save_path = 'Images/Example_Data.pdf'
Plot_and_Save_Funs.Plot_Origin_Data(clumpsObj,figsize=(8,6),fontsize=16,spacing=
```



In [ ]:

Plot the detection results and save the image.

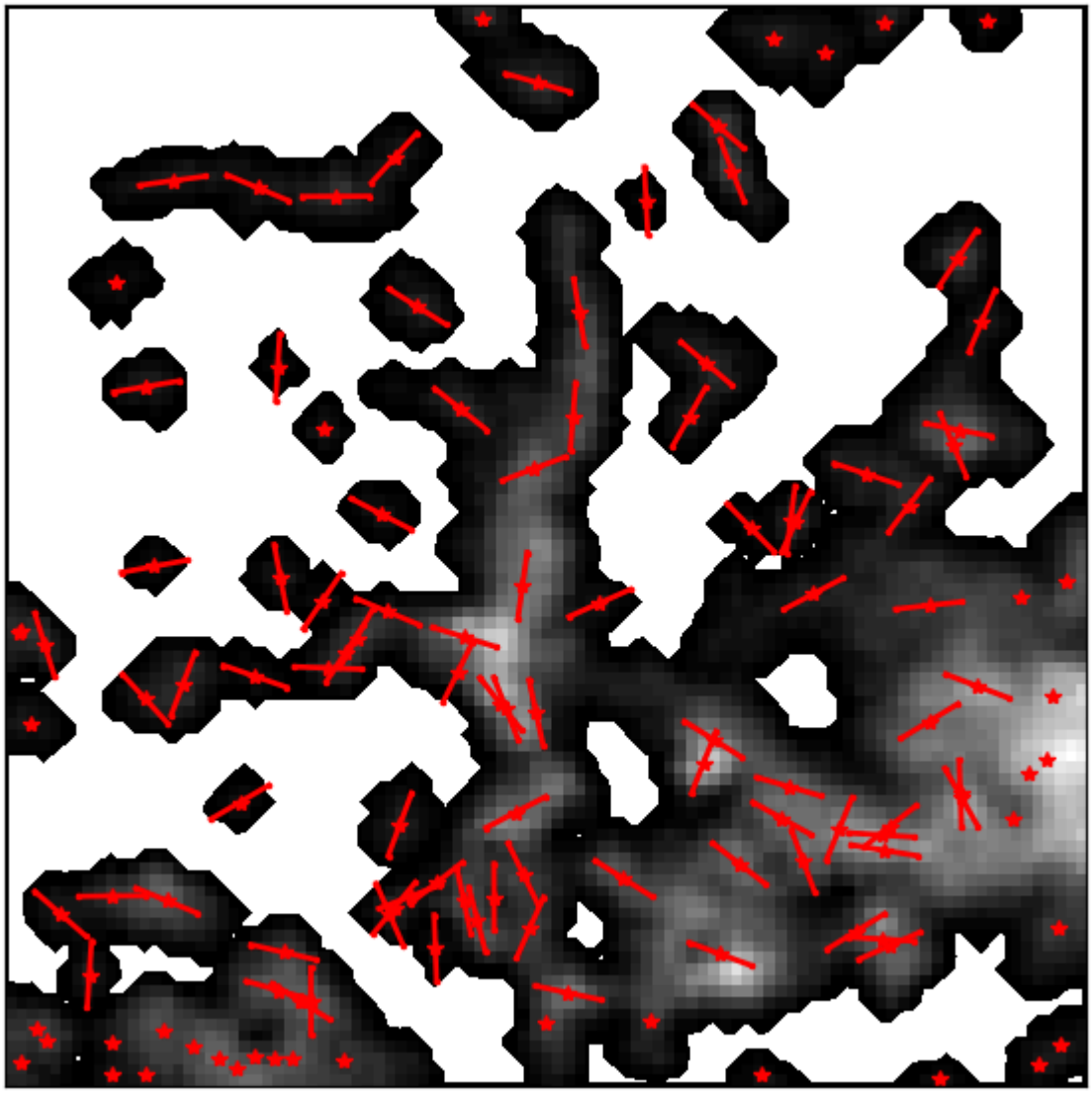
The total number of clumps is 127, with 89 of them not touching the edge. The red asterisks denote the central position of the clumps, and the red lines denote the direction of the principal axis of the clumps.

In [11]:
```
edges = clumpsObj.edges
print('Total number:',len(edges))
print('NO edges number:',np.where(edges==0)[0].shape[0])
```

```
Total number: 126
NO edges number: 88
```

In [ ]:

In [12]:
```
save_path = 'Images/Clumps_Infor.pdf'
Plot_and_Save_Funs.Plot_Clumps_Infor(clumpsObj,figsize=(8,6),line_scale=3,save_p
```



In [ ]:
```

```

Get the clumps information from the clumpsObj.

In [13]:
```
clump_angles = clumpsObj.angles
clump_edges = clumpsObj.edges
clump_centers = clumpsObj.centers
clump_centers_wcs = clumpsObj.centers_wcs
origin_data = clumpsObj.origin_data
regions_data = clumpsObj.regions_data
data_wcs = clumpsObj.data_wcs
connected_ids_dict = clumpsObj.connected_ids_dict
clump_coords_dict = clumpsObj.clump_coords_dict

clumps_data = np.zeros_like(origin_data)
clumps_data[regions_data>0] = origin_data[regions_data>0]
```

In [ ]:
```

```

The Algorithm in the PPP Space: DisPerSE

P5RMS_R5RMS:

RMS ~ 0.22

persistence threshold = 5*RMS

robustness threshold = 5*RMS

In [14]:
```
# sk_file_name = '../Filaments/Data/DisPerSE/P4RMS_R4RMS.fits'
sk_file_name = '../Filaments/Data/DisPerSE/P5RMS_R5RMS.fits'
# sk_file_name = '../Filaments/Data/DisPerSE/P6RMS_R6RMS.fits'
# sk_file_name = '../Filaments/Data/DisPerSE/P8RMS_R8RMS.fits'

# print('file_name:',sk_file_name)
skeleton_data_T = fits.getdata(sk_file_name)

skeleton_data = np.zeros_like(skeleton_data_T)
skeleton_data[skeleton_data_T>0]=1

skeleton_label = measure.label(skeleton_data,connectivity=3)
skeletons_list = measure.regionprops(skeleton_label)
```

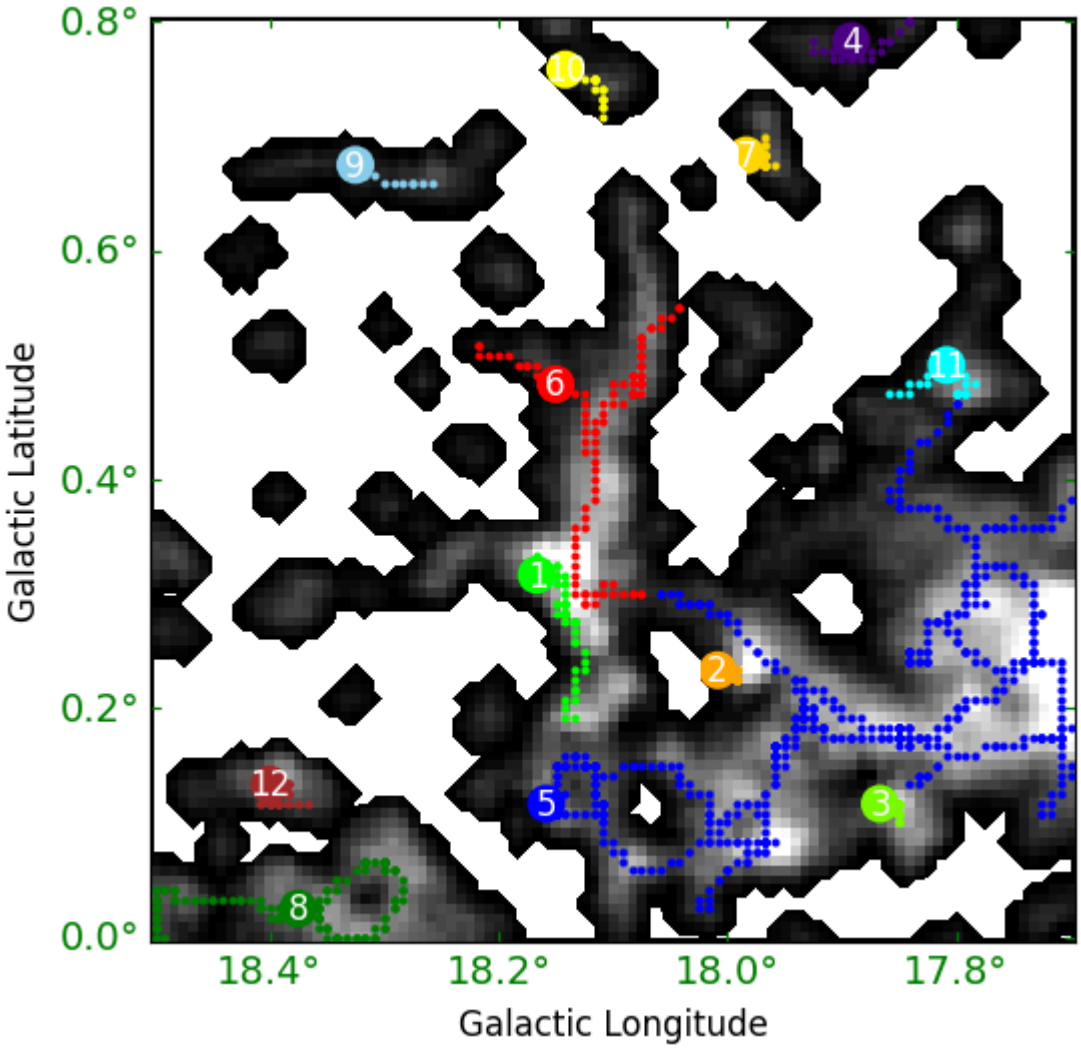
In [ ]:
```
```

In [15]:
```
figsize=(8,6)
fontsize=12
spacing=12*u.arcmin
```

In [ ]:
```
```

In [16]:
```
fig = plt.figure(figsize=(8,6))
ax0 = fig.add_subplot(111,projection=data_wcs.celestial)


colors_T = ['lime','orange','lawngreen','indigo','blue','red','gold','green','sk
colors_id = 0
circle_radius = 2
for index in range(len(skeletons_list)):
    skeleton_coords = skeletons_list[index].coords[:,1:]
    if len(skeleton_coords) > 10:
        center_x = skeleton_coords[len(skeleton_coords)//2][1]-2
        center_y = skeleton_coords[len(skeleton_coords)//2][0]+1
        circle = patches.Circle((center_x, center_y), circle_radius, facecolor=c
        ax0.add_patch(circle)
        ax0.text(center_x, center_y, "{}".format(colors_id+1), fontsize=12, colo

        for i in range(len(skeleton_coords)):
            ax0.plot(skeleton_coords[i][1],skeleton_coords[i][0],color=colors_T[
        colors_id += 1


img = clumps_data.sum(0)
vmin = np.min(img[np.where(img!=0)])
vmax = np.nanpercentile(img[np.where(img!=0)], 98.)
ax0.imshow(img,
           origin='lower',
           cmap='gray',
```

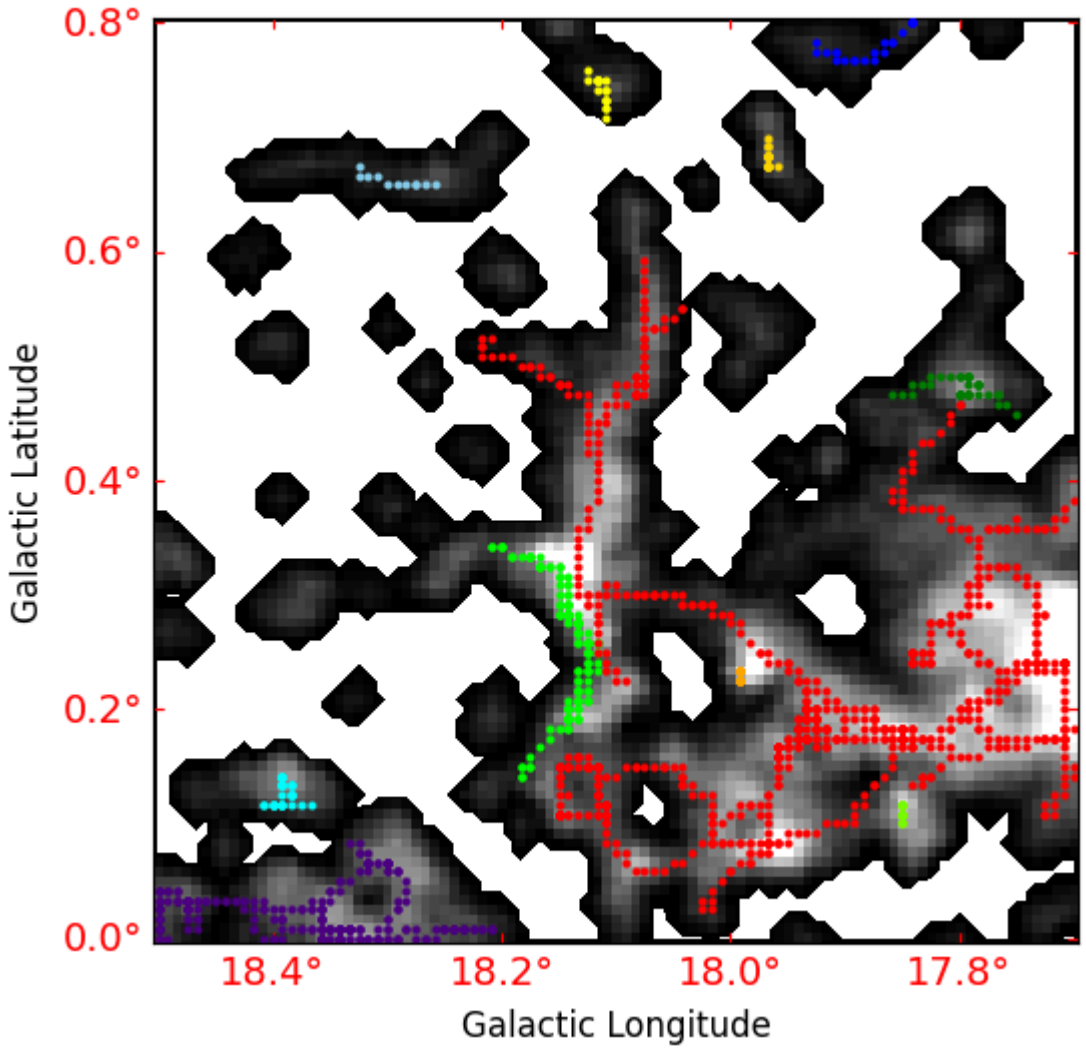
```
            interpolation='none',
            norm = colors.Normalize(vmin = vmin, vmax =  vmax))
ax0.contourf(img,
             levels = [0., .01],
             colors = 'w')
plt.rcParams['xtick.direction'] = 'in'
plt.rcParams['ytick.direction'] = 'in'
plt.rcParams['xtick.color'] = 'red'
plt.rcParams['ytick.color'] = 'red'
plt.xlabel("Galactic Longitude",fontsize=fontsize)
plt.ylabel("Galactic Latitude",fontsize=fontsize)
lon = ax0.coords[0]
lat = ax0.coords[1]
lon.set_major_formatter("d.d")
lat.set_major_formatter("d.d")
lon.set_ticks(spacing=spacing)
lat.set_ticks(spacing=spacing)

# plt.savefig('../Images/DisPerSe_LB.pdf', format='pdf', dpi=1000)
plt.show()
```



In [ ]:

In [17]:
```
sk_file_name = '../Filaments/Data/DisPerSE/P4RMS_R4RMS.fits'
# sk_file_name = '../Filaments/Data/DisPerSE/P5RMS_R5RMS.fits'
# sk_file_name = '../Filaments/Data/DisPerSE/P6RMS_R6RMS.fits'
# sk_file_name = '../Filaments/Data/DisPerSE/P8RMS_R8RMS.fits'
```

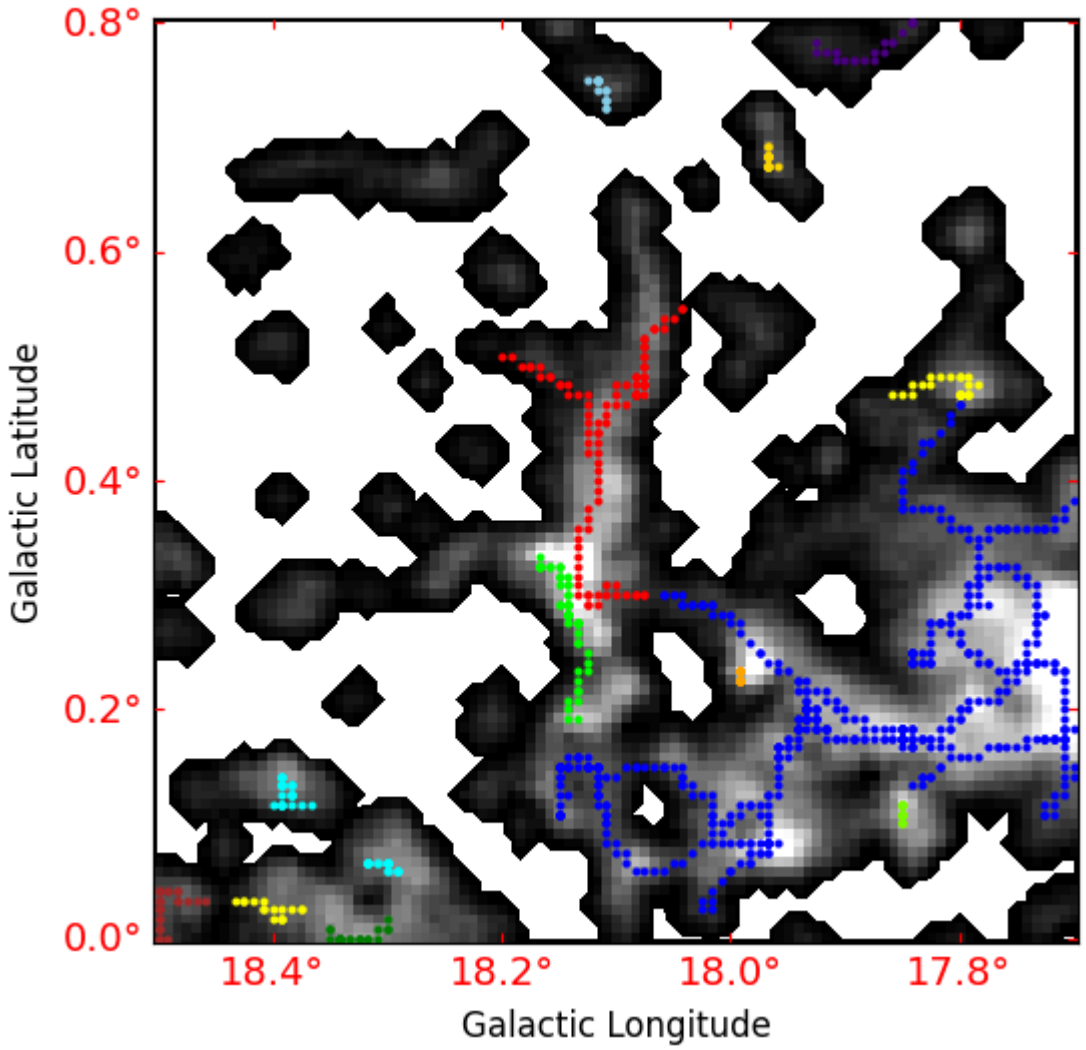
```
# print('file_name:',sk_file_name)
skeleton_data_T = fits.getdata(sk_file_name)

skeleton_data = np.zeros_like(skeleton_data_T)
skeleton_data[skeleton_data_T>0]=1

skeleton_label = measure.label(skeleton_data,connectivity=3)
skeletons_list = measure.regionprops(skeleton_label)


fig = plt.figure(figsize=(8,6))
ax0 = fig.add_subplot(111,projection=data_wcs.celestial)


colors_T = ['lime','orange','lawngreen','indigo','blue','red','gold','green','sk
            'yellow','cyan','brown']
colors_id = 0
circle_radius = 2
for index in range(len(skeletons_list)):
    skeleton_coords = skeletons_list[index].coords[:,1:]
    if len(skeleton_coords) > 10:
        center_x = skeleton_coords[len(skeleton_coords)//2][1]-2
        center_y = skeleton_coords[len(skeleton_coords)//2][0]+1
        circle = patches.Circle((center_x, center_y), circle_radius, facecolor=c
#         ax0.add_patch(circle)
#         ax0.text(center_x, center_y, "{}".format(colors_id+1), fontsize=12, co

        for i in range(len(skeleton_coords)):
            ax0.plot(skeleton_coords[i][1],skeleton_coords[i][0],color=colors_T[
        colors_id += 1


img = clumps_data.sum(0)
vmin = np.min(img[np.where(img!=0)])
vmax = np.nanpercentile(img[np.where(img!=0)], 98.)
ax0.imshow(img,
           origin='lower',
           cmap='gray',
           interpolation='none',
           norm = colors.Normalize(vmin = vmin, vmax =  vmax))
ax0.contourf(img,
             levels = [0., .01],
             colors = 'w')
plt.rcParams['xtick.direction'] = 'in'
plt.rcParams['ytick.direction'] = 'in'
plt.rcParams['xtick.color'] = 'red'
plt.rcParams['ytick.color'] = 'red'
plt.xlabel("Galactic Longitude",fontsize=fontsize)
plt.ylabel("Galactic Latitude",fontsize=fontsize)
lon = ax0.coords[0]
lat = ax0.coords[1]
lon.set_major_formatter("d.d")
lat.set_major_formatter("d.d")
lon.set_ticks(spacing=spacing)
lat.set_ticks(spacing=spacing)

# plt.savefig('../Images/DisPerSe_LB_4RMS.pdf', format='pdf', dpi=1000)
plt.show()
```



In [ ]:

In [18]:
```
# sk_file_name = '../Filaments/Data/DisPerSE/P4RMS_R4RMS.fits'
# sk_file_name = '../Filaments/Data/DisPerSE/P5RMS_R5RMS.fits'
sk_file_name = '../Filaments/Data/DisPerSE/P6RMS_R6RMS.fits'
# sk_file_name = '../Filaments/Data/DisPerSE/P8RMS_R8RMS.fits'

# print('file_name:',sk_file_name)
skeleton_data_T = fits.getdata(sk_file_name)

skeleton_data = np.zeros_like(skeleton_data_T)
skeleton_data[skeleton_data_T>0]=1

skeleton_label = measure.label(skeleton_data,connectivity=3)
skeletons_list = measure.regionprops(skeleton_label)


fig = plt.figure(figsize=(8,6))
ax0 = fig.add_subplot(111,projection=data_wcs.celestial)


colors_T = ['lime','orange','lawngreen','indigo','blue','red','gold','green','sk
            'yellow','cyan','brown']
colors_id = 0
circle_radius = 2
for index in range(len(skeletons_list)):
    skeleton_coords = skeletons_list[index].coords[:,1:]
```

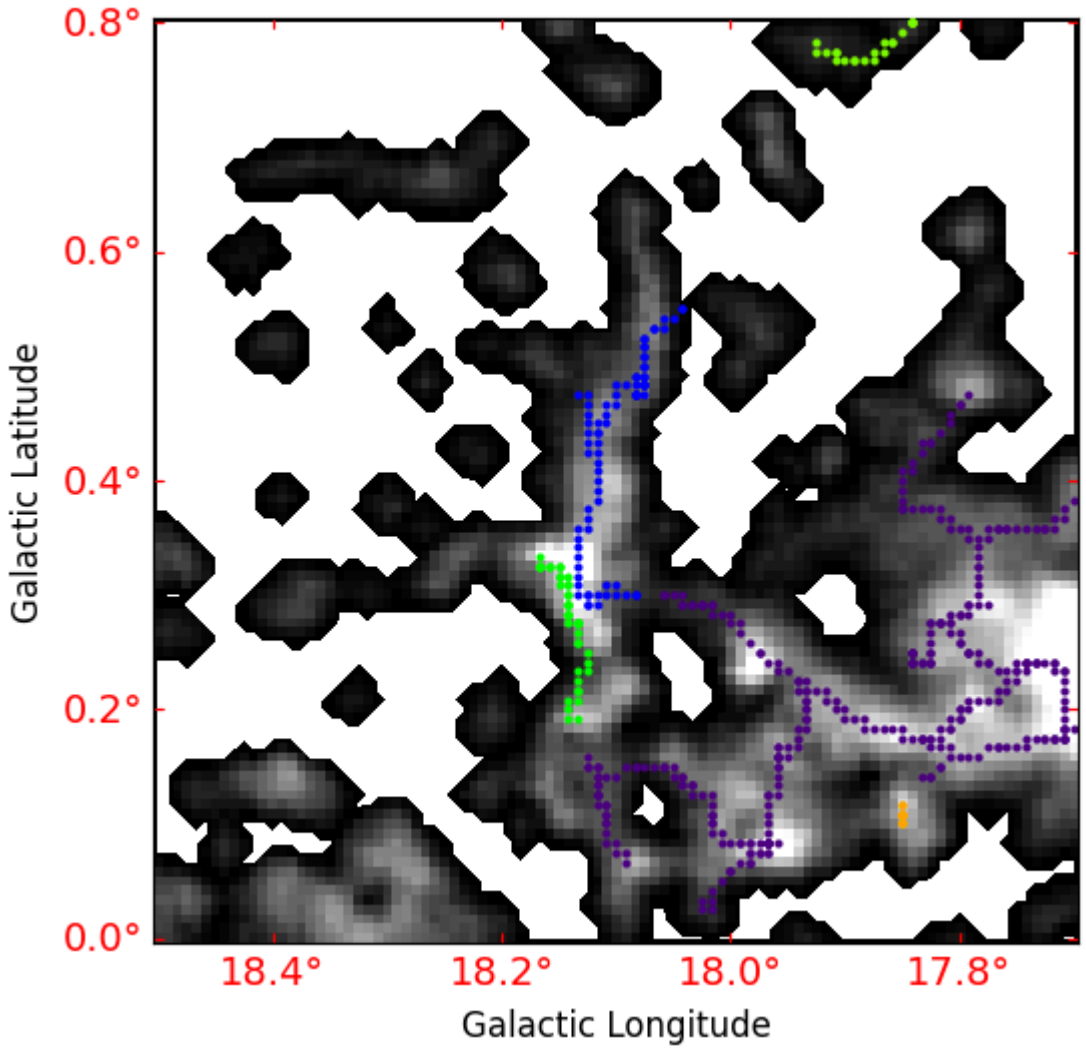
```
    if len(skeleton_coords) > 10:
        center_x = skeleton_coords[len(skeleton_coords)//2][1]-2
        center_y = skeleton_coords[len(skeleton_coords)//2][0]+1
        circle = patches.Circle((center_x, center_y), circle_radius, facecolor=c
#         ax0.add_patch(circle)
#         ax0.text(center_x, center_y, "{}".format(colors_id+1), fontsize=12, co

        for i in range(len(skeleton_coords)):
            ax0.plot(skeleton_coords[i][1],skeleton_coords[i][0],color=colors_T[
        colors_id += 1


img = clumps_data.sum(0)
vmin = np.min(img[np.where(img!=0)])
vmax = np.nanpercentile(img[np.where(img!=0)], 98.)
ax0.imshow(img,
           origin='lower',
           cmap='gray',
           interpolation='none',
           norm = colors.Normalize(vmin = vmin, vmax =  vmax))
ax0.contourf(img,
             levels = [0., .01],
             colors = 'w')
plt.rcParams['xtick.direction'] = 'in'
plt.rcParams['ytick.direction'] = 'in'
plt.rcParams['xtick.color'] = 'red'
plt.rcParams['ytick.color'] = 'red'
plt.xlabel("Galactic Longitude",fontsize=fontsize)
plt.ylabel("Galactic Latitude",fontsize=fontsize)
lon = ax0.coords[0]
lat = ax0.coords[1]
lon.set_major_formatter("d.d")
lat.set_major_formatter("d.d")
lon.set_ticks(spacing=spacing)
lat.set_ticks(spacing=spacing)

# plt.savefig('../Images/DisPerSe_LB_6RMS.pdf', format='pdf', dpi=1000)
plt.show()
```



In [ ]:

In [20]:
```
# sk_file_name = '../Filaments/Data/DisPerSE/P4RMS_R4RMS.fits'
# sk_file_name = '../Filaments/Data/DisPerSE/P5RMS_R5RMS.fits'
# sk_file_name = '../Filaments/Data/DisPerSE/P6RMS_R6RMS.fits'
sk_file_name = '../Filaments/Data/DisPerSE/P8RMS_R8RMS.fits'

# print('file_name:',sk_file_name)
skeleton_data_T = fits.getdata(sk_file_name)

skeleton_data = np.zeros_like(skeleton_data_T)
skeleton_data[skeleton_data_T>0]=1

skeleton_label = measure.label(skeleton_data,connectivity=3)
skeletons_list = measure.regionprops(skeleton_label)


fig = plt.figure(figsize=(8,6))
ax0 = fig.add_subplot(111,projection=data_wcs.celestial)


colors_T = ['lime','orange','lawngreen','indigo','blue','red','gold','green','sk
            'yellow','cyan','brown']
colors_id = 0
circle_radius = 2
for index in range(len(skeletons_list)):
    skeleton_coords = skeletons_list[index].coords[:,1:]
```

```
    if len(skeleton_coords) > 10:
        center_x = skeleton_coords[len(skeleton_coords)//2][1]-2
        center_y = skeleton_coords[len(skeleton_coords)//2][0]+1
        circle = patches.Circle((center_x, center_y), circle_radius, facecolor=c
#         ax0.add_patch(circle)
#         ax0.text(center_x, center_y, "{}".format(colors_id+1), fontsize=12, co

        for i in range(len(skeleton_coords)):
            ax0.plot(skeleton_coords[i][1],skeleton_coords[i][0],color=colors_T[
        colors_id += 1


img = clumps_data.sum(0)
vmin = np.min(img[np.where(img!=0)])
vmax = np.nanpercentile(img[np.where(img!=0)], 98.)
ax0.imshow(img,
           origin='lower',
           cmap='gray',
           interpolation='none',
           norm = colors.Normalize(vmin = vmin, vmax =  vmax))
ax0.contourf(img,
             levels = [0., .01],
             colors = 'w')
plt.rcParams['xtick.direction'] = 'in'
plt.rcParams['ytick.direction'] = 'in'
plt.rcParams['xtick.color'] = 'red'
plt.rcParams['ytick.color'] = 'red'
plt.xlabel("Galactic Longitude",fontsize=fontsize)
plt.ylabel("Galactic Latitude",fontsize=fontsize)
lon = ax0.coords[0]
lat = ax0.coords[1]
lon.set_major_formatter("d.d")
lat.set_major_formatter("d.d")
lon.set_ticks(spacing=spacing)
lat.set_ticks(spacing=spacing)

# plt.savefig('../Images/DisPerSe_LB_8RMS.pdf', format='pdf', dpi=1000)
plt.show()
```



In [ ]:

DisPerSE in Simulation

In [19]:
```
RMS = 0.1
Threshold = 5 * RMS

parameters_FacetClumps = [RMS, Threshold, SWindow, KBins, FwhmBeam, VeloRes, SRe
```

In [ ]:

In [20]:
```
file_name = 'RandA_260_13CO.fits'

file_example = 'Simulation_1'
mask_name = '../Example_Files/Clump/mask_{}.fits'.format(file_example)
outcat_name = '../Example_Files/Clump/outcat_{}.csv'.format(file_example)
outcat_wcs_name = '../Example_Files/Clump/outcat_wcs_{}.csv'.format(file_example
```

In [ ]:

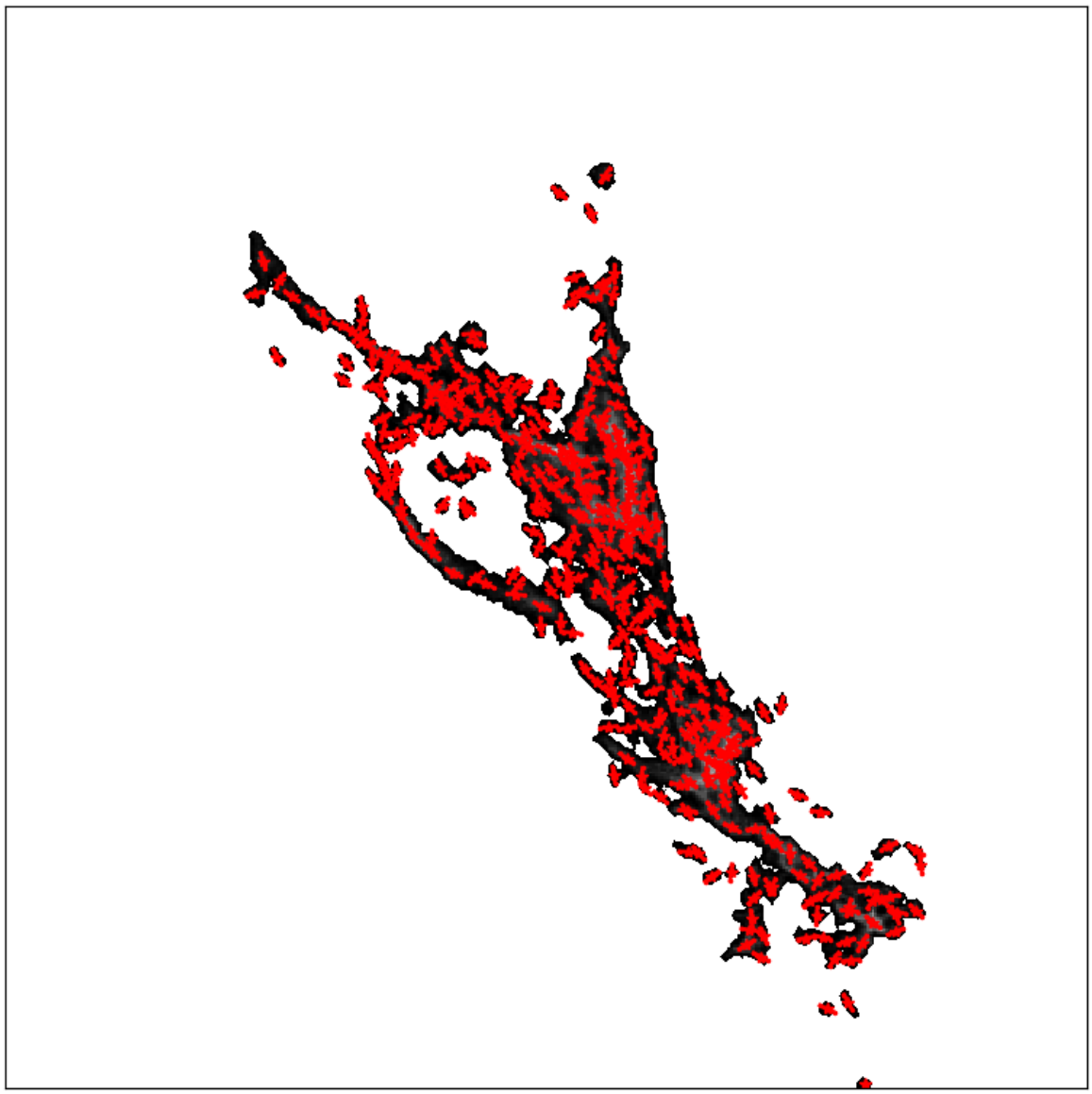
In [21]:
```
clumpsObj = ClumpInfor(file_name,mask_name,outcat_name,outcat_wcs_name)

# clumpsObj.Cal_Infor_From_Mask_Or_Algorithm(mask_or_algorithm='FacetClumps',par
# clumpsObj.Get_Clumps_Infor(fit_flag = True)
```

```
clumpsObj.Cal_Infor_From_Mask_Or_Algorithm(mask_or_algorithm='mask')
clumpsObj.Get_Clumps_Infor(fit_flag = True)
```

```
Number: 373
Time: 0.52
100%|██████████| 373/373 [00:07<00:00, 47.57it/s]
Fitting Clumps Time: 8.15
```

In [ ]:

In [22]:
```
save_path = 'Images/Clumps_Infor.pdf'
Plot_and_Save_Funs.Plot_Clumps_Infor(clumpsObj,figsize=(12,8),line_scale=3,save_
```



In [ ]:

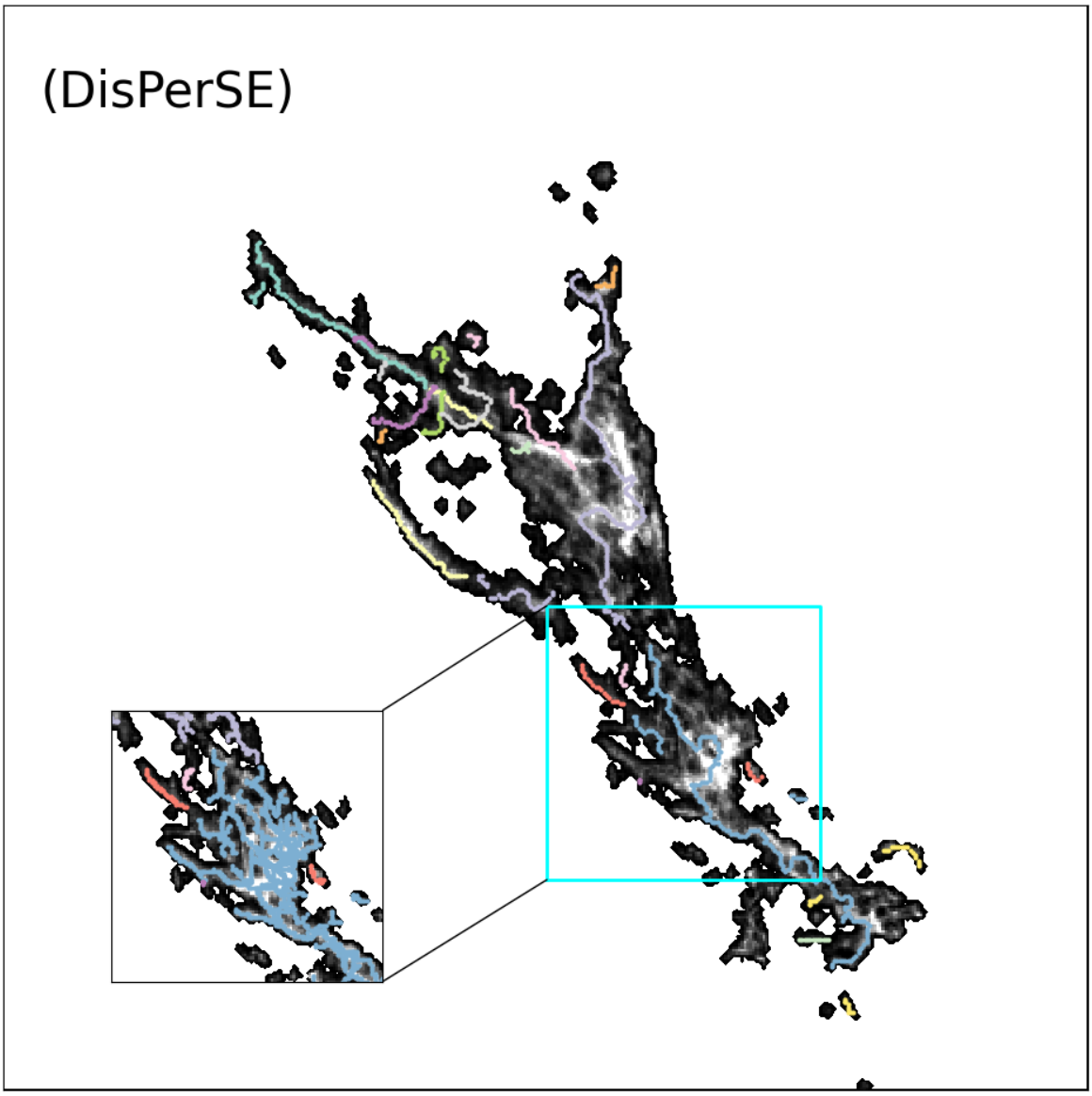
In [23]:
```
clump_angles = clumpsObj.angles
clump_edges = clumpsObj.edges
clump_centers = clumpsObj.centers
clump_centers_wcs = clumpsObj.centers_wcs
origin_data = clumpsObj.origin_data
regions_data = clumpsObj.regions_data
data_wcs = clumpsObj.data_wcs
connected_ids_dict = clumpsObj.connected_ids_dict
clump_coords_dict = clumpsObj.clump_coords_dict
```

```
clumps_data = np.zeros_like(origin_data)
clumps_data[regions_data>0] = origin_data[regions_data>0]
```

In [ ]:

In [24]:
```
sk_file_name = 'P10RMS_R10RMS.fits'

# print('file_name:',sk_file_name)
skeleton_data_T = fits.getdata(sk_file_name)

skeleton_data = np.zeros_like(skeleton_data_T)
skeleton_data[skeleton_data_T>0]=1

skeleton_label = measure.label(skeleton_data,connectivity=3)
skeletons_list = measure.regionprops(skeleton_label)

fontsize = 30
fig = plt.figure(figsize=(14,12))
ax0 = fig.add_subplot(111)#,projection=data_wcs.celestial)

ax0.text(13,362,r'({})'.format('DisPerSE'),color='black',fontsize=fontsize)

colors_T = plt.cm.Set3(np.linspace(0, 1, 30))
# colors_T = colors_T[::-1]
colors_id = 0
for index in range(len(skeletons_list)):
    skeleton_coords = skeletons_list[index].coords[:,1:]

    if len(skeleton_coords) > 10:
        G_longest_skeleton, T_longest_skeleton = FCFA.Graph_Infor_Connected(skel
        max_path, max_edges = FCFA.Get_Max_Path_Weight(T_longest_skeleton)
        skeleton_coords = skeleton_coords[max_path]
        for i in range(len(skeleton_coords)):
            ax0.plot(skeleton_coords[i][1],skeleton_coords[i][0],color=colors_T[
        colors_id += 1


img = clumps_data.sum(0)
vmin = np.min(img[np.where(img!=0)])
vmax = np.nanpercentile(img[np.where(img!=0)], 98.)
ax0.imshow(img,
           origin='lower',
           cmap='gray',
           interpolation='none',
           norm = colors.Normalize(vmin = vmin, vmax =  vmax))
ax0.contourf(img,
             levels = [0., .01],
             colors = 'w')

linked = 'left'
xlim_left, xlim_right = 180+20,281+20
ylim_bottom, ylim_top = 97-20,198-20
axins = ax0.inset_axes((0.1, 0.1, 0.25, 0.25))
axins.set_xlim(xlim_left, xlim_right)
axins.set_ylim(ylim_bottom, ylim_top)

ax0.plot([xlim_left,xlim_right,xlim_right,xlim_left,xlim_left],
           [ylim_bottom,ylim_bottom,ylim_top,ylim_top,ylim_bottom],linewidth=2,
```

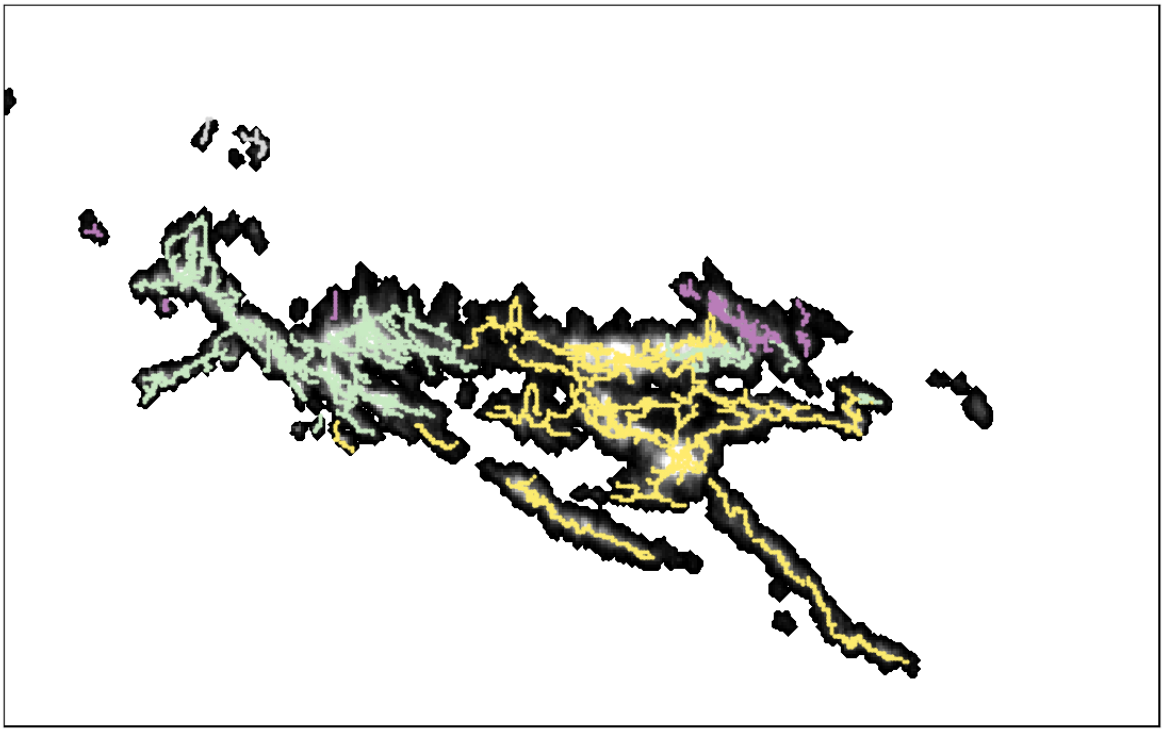
```
if linked == 'bottom':
    xyA_1, xyB_1 = (xlim_left,ylim_top), (xlim_left,ylim_bottom)
    xyA_2, xyB_2 = (xlim_right,ylim_top), (xlim_right,ylim_bottom)
elif  linked == 'top':
    xyA_1, xyB_1 = (xlim_left,ylim_bottom), (xlim_left,ylim_top)
    xyA_2, xyB_2 = (xlim_right,ylim_bottom), (xlim_right,ylim_top)
elif  linked == 'left':
    xyA_1, xyB_1 = (xlim_right,ylim_top), (xlim_left,ylim_top)
    xyA_2, xyB_2 = (xlim_right,ylim_bottom), (xlim_left,ylim_bottom)
elif  linked == 'right':
    xyA_1, xyB_1 = (xlim_left,ylim_top), (xlim_right,ylim_top)
    xyA_2, xyB_2 = (xlim_left,ylim_bottom), (xlim_right,ylim_bottom)

con = ConnectionPatch(xyA=xyA_1,xyB=xyB_1,coordsA="data",
                      coordsB="data",axesA=axins,axesB=ax0)
axins.add_artist(con)
con = ConnectionPatch(xyA=xyA_2,xyB=xyB_2,coordsA="data",
                      coordsB="data",axesA=axins,axesB=ax0)
axins.add_artist(con)
axins.set_xticks([])
axins.set_yticks([])

colors_id = 0
for index in range(len(skeletons_list)):
    skeleton_coords = skeletons_list[index].coords[:,1:]

    if len(skeleton_coords) > 10:
#         G_longest_skeleton, T_longest_skeleton = FCFA.Graph_Infor_Connected(sk
#         max_path, max_edges = FCFA.Get_Max_Path_Weight(T_longest_skeleton)
#         skeleton_coords = skeleton_coords[max_path]
        for i in range(len(skeleton_coords)):
            axins.plot(skeleton_coords[i][1],skeleton_coords[i][0],color=colors_
        colors_id += 1


img = clumps_data.sum(0)
img_color = img[xlim_left:xlim_right,ylim_bottom:ylim_top]
vmin = np.min(img_color[np.where(img_color!=0)])
vmax = np.nanpercentile(img_color[np.where(img_color!=0)], 98.)
axins.imshow(img,
           origin='lower',
           cmap='gray',
           interpolation='none',
           norm = colors.Normalize(vmin = vmin, vmax =  vmax))
axins.contourf(img,
             levels = [0., .01],
             colors = 'w')

plt.rcParams['xtick.direction'] = 'in'
plt.rcParams['ytick.direction'] = 'in'
plt.rcParams['xtick.color'] = 'red'
plt.rcParams['ytick.color'] = 'red'
# plt.xlabel("Galactic Longitude",fontsize=fontsize)
# plt.ylabel("Galactic Latitude",fontsize=fontsize)
plt.xticks([]),plt.yticks([])

# plt.savefig('../Images/HD_DisPerSE.pdf', format='pdf', dpi=1000)
plt.show()
```



In [ ]:

In [25]:
```
fig = plt.figure(figsize=(14,12))
ax0 = fig.add_subplot(111)#,projection=data_wcs.celestial)

# ax0.text(13,362,r'({})'.format('DisPerSE'),color='black',fontsize=fontsize)

colors_T = plt.cm.Set3(np.linspace(0, 1, len(skeletons_list)))
colors_T = colors_T[::-1]

colors_id = 0
circle_radius = 2
for index in range(len(skeletons_list)):
    skeleton_coords = skeletons_list[index].coords[:,:2]
    if len(skeleton_coords) > 10:
        for i in range(len(skeleton_coords)):
            ax0.plot(skeleton_coords[i][1],skeleton_coords[i][0],color=colors_T[
        colors_id += 1


img = clumps_data.sum(2)
vmin = np.min(img[np.where(img!=0)])
vmax = np.nanpercentile(img[np.where(img!=0)], 98.)
ax0.imshow(img,
```

```
            origin='lower',
            cmap='gray',
            interpolation='none',
            norm = colors.Normalize(vmin = vmin, vmax =  vmax))
ax0.contourf(img,
             levels = [0., .01],
             colors = 'w')
plt.rcParams['xtick.direction'] = 'in'
plt.rcParams['ytick.direction'] = 'in'
plt.rcParams['xtick.color'] = 'red'
plt.rcParams['ytick.color'] = 'red'
# plt.xlabel("Galactic Longitude",fontsize=fontsize)
# plt.ylabel("Galactic Latitude",fontsize=fontsize)
plt.xticks([]),plt.yticks([])

# plt.savefig('../Images/HD_DisPerSE_LV.pdf', format='pdf', dpi=1000)
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: