Felzenszwalb Clustering

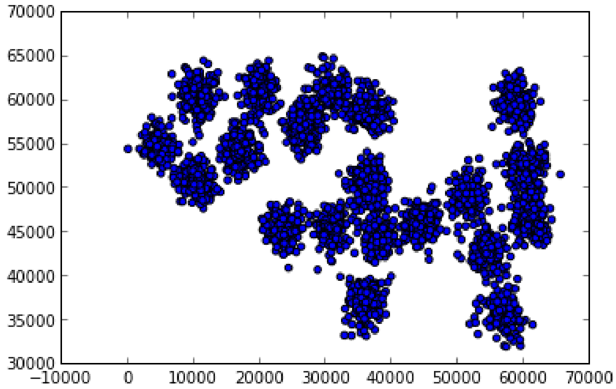
Load up data

```
import pandas as pd
syn = pd.read_csv("synthetic.txt",names=['a','b'],sep="    ")
print syn.shape
data = np.array(syn)
```

```
(3000, 2)
```

It looks like this

```
plt.scatter(data[:,0],data[:,1])
plt.savefig('test_02.png')
```



```
(3000, 2)
```

Create pairwise Euclidian distances

```
from sklearn.metrics.pairwise import euclidean_distances
X = euclidean_distances(data, data)
print X.shape
print X[1,2]
```

```
(3000, 3000)
622.540761718
```

Remove distances that are too big, so we can create a sparse array

```
import scipy.sparse as sps
from scipy.io import mmwrite, mmread
print X.shape
X2 = X.copy()
print X2.mean(), X2.std()
X2[X > 2000] = 0.0
print X2.shape
X3 = sps.lil_matrix(X2)
print X3.shape
print 'before triu', len(X3.nonzero()[0])
X4 = sps.triu(X3)
print  'after triu', len(X4.nonzero()[0])
```

```
(3000, 3000)
23275.2891376 13430.4720182
(3000, 3000)
(3000, 3000)
before triu 174020
after triu 87010
```

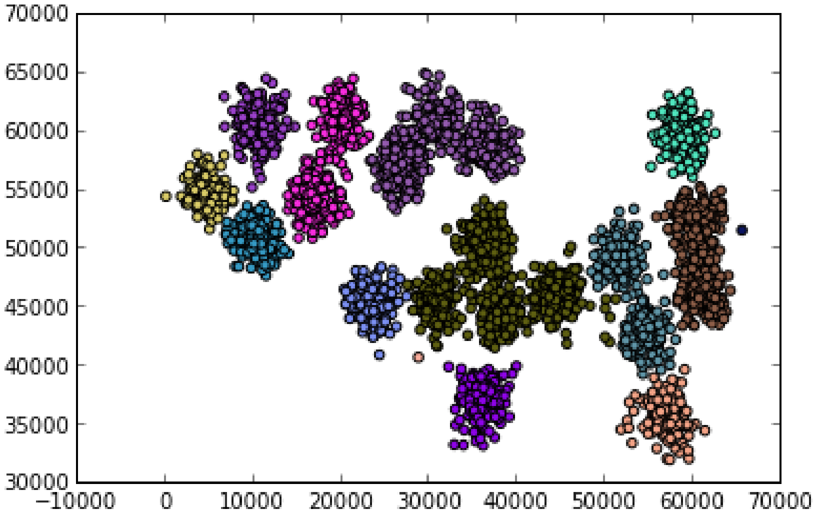
Take only upper triangular part of the matrix because it is symmetric, cuts the number of non-zero cells by half,

```
import scipy.sparse as sps
from scipy.io import mmwrite, mmread
mmwrite('/tmp/syndist', X4)
```

```
!../felzclust/felzclust /tmp/syndist.mtx 25000 20  > /tmp/out
df = pd.read_csv('/tmp/out',sep=';')
syn['cluster'] = df['cluster']
print len(syn['cluster'].unique()), 'clusters found'
print syn[:5]
```

```
14 clusters found
       a      b  cluster
0  54620  43523      238
1  52694  42750      238
2  53253  43024      238
3  54925  42624      238
4  54973  43980      238
```

```
import random
for clust in syn['cluster'].unique():
    tmp = np.array(syn[syn['cluster'] == clust][['a','b']])
    plt.scatter(tmp[:,0], tmp[:,1], c=np.random.rand(3,1))
plt.savefig('test_01.png')
```


70000
65000
60000
55000
50000
45000
40000
35000
30000
−10000
0
10000
20000
30000
40000
50000
60000
70000