In [3]:
```
import scipy.io as sio
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt #for ploting
import math #logarithmic  computation
matFile = sio.loadmat("Homo_sapiens.mat") # read mat file
```

In [4]:
```
matFile
```

Out[4]:
```
{'__header__': b'MATLAB 5.0 MAT-file Platform: posix, Created on: Thu Nov 12 18:54:12 2015',
 '__version__': '1.0',
 '__globals__': [],
 'group': <3890x50 sparse matrix of type '<class 'numpy.float64'>'
        with 6640 stored elements in Compressed Sparse Column format>,
 'network': <3890x3890 sparse matrix of type '<class 'numpy.float64'>'
        with 76584 stored elements in Compressed Sparse Column format>}
```

In [5]:
```
network=matFile["network"].todense()
```

In [46]:
```
with open("directed_network.txt","w") as wf:
    for i in range(0,3890):
        for j in range(0,3890):
            wf.write("\t".join([str(i),str(j),str(network[i,j])]))
            wf.write("\n")
```

In [6]:
```
with open("directed_network_node2vec.txt","w") as wf:
    for i in range(0,3890):
        for j in range(0,3890):
            if network[i,j] > 0.00001:
                wf.write("\t".join([str(i),str(j)]))
                wf.write("\n")
```

In [55]:
```
!python2 ~/Desktop/externalDisk/node2vec/src/main.py --input ./directed_network_node2vec.txt --output ./node2vec.emd --dimensions 2 --walk-length 40  --workers 12 --directed
```

```
Walk iteration:
1 / 10
2 / 10
3 / 10
4 / 10
5 / 10
6 / 10
7 / 10
8 / 10
9 / 10
10 / 10
```

In [57]:
```
!deepwalk --format mat --input Homo_sapiens.mat --max-memory-data-size 1500000
0 --number-walks 80 --representation-size 2 --walk-length 40 --window-size 10
 --workers 12 --output ./deepwalk.embeddings
```

```
Number of nodes: 3890
Number of walks: 311200
Data size (walks*length): 12448000
Walking...
Training...
2019-04-03 00:41:28 WARNING word2vec.py: 453 consider setting layer size to a
multiple of 4 for greater performance
```

In [50]:
```
!./line -train directed_network.txt -output line.emb -binary 0 -size 2 -order
2 -negative 5 -sample 1
```

```
--------------------------------
Order: 2
Samples: 1M
Negative: 5
Dimension: 2
Initial rho: 0.025000
--------------------------------
Number of edges: 15132100
Number of vertices: 3890
--------------------------------
Rho: 0.000498  Progress: 99.010%
Total time: 2.484214
```

In [7]:
```
!head deepwalk.embeddings
```

```
3890 2
3630 0.6900835 -0.1099861
824 0.79812306 -0.32175177
1219 0.69618225 -0.06892041
291 0.75003415 -0.23189463
1042 0.7789927 -0.27625772
3361 0.7583472 -0.24201731
1556 0.6759308 0.027903436
1324 0.7641915 -0.28979298
1027 0.75827074 -0.20404878
```

In [12]:
```
!head line.emb
```

```
3890 2
0 -0.613145 1.182144
1 -0.789033 0.956220
2 -0.057798 0.474585
3 -0.084796 0.652828
4 -0.434152 1.473698
5 -0.655262 0.793805
6 -0.163866 0.101380
7 -0.244334 0.513086
8 -0.177173 0.236363
```

In [13]:

```
!head node2vec.emd
```

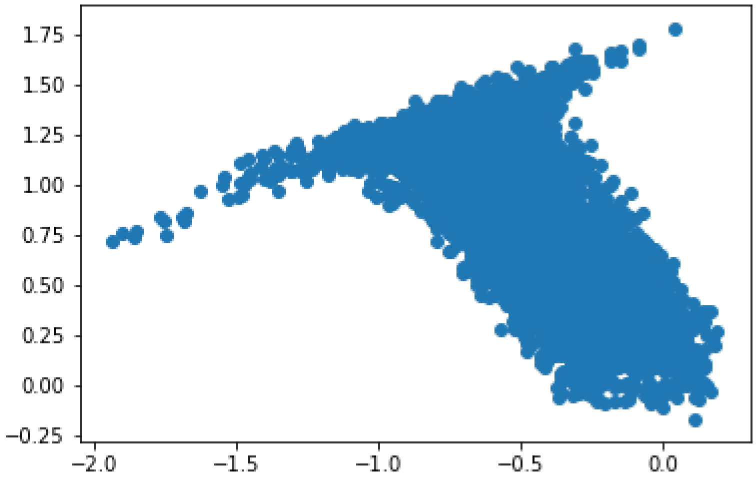
```
3890 2
3630 0.907081 -0.720103
824 1.291277 -0.426085
1219 0.983246 -0.661871
291 1.102590 -0.567622
3361 1.162797 -0.529489
1556 0.850200 -0.850889
1042 1.308276 -0.424843
1324 1.184480 -0.508187
1027 1.206057 -0.496429
```

In [43]:

```
nodeID = []
coordinate1 = []
coordinate2 = []
with open("line.emb","r") as rf:
    rf.readline()
    for line in rf:
        x, y, z = line.strip().split(" ")
        nodeID.append(x)
        coordinate1.append(y)
        coordinate2.append(z)

coordinate1 = list(map(float, coordinate1))
coordinate2 = list(map(float, coordinate2))

plt.scatter(coordinate1[:],coordinate2[:])
plt.show()
```

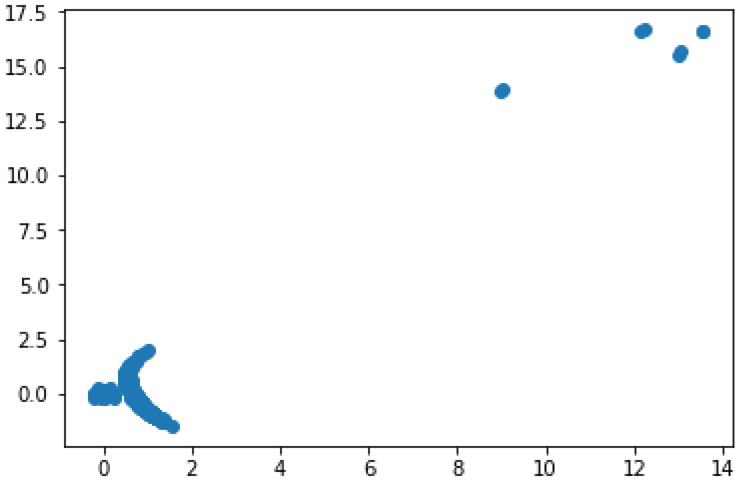


In [44]:

```
nodeID = []
coordinate1 = []
coordinate2 = []
with open("deepwalk.embeddings","r") as rf:
    rf.readline()
    for line in rf:
        x, y, z = line.strip().split(" ")
        nodeID.append(x)
        coordinate1.append(y)
        coordinate2.append(z)

coordinate1 = list(map(float, coordinate1))
coordinate2 = list(map(float, coordinate2))

plt.scatter(coordinate1[:],coordinate2[:])
plt.show()
```

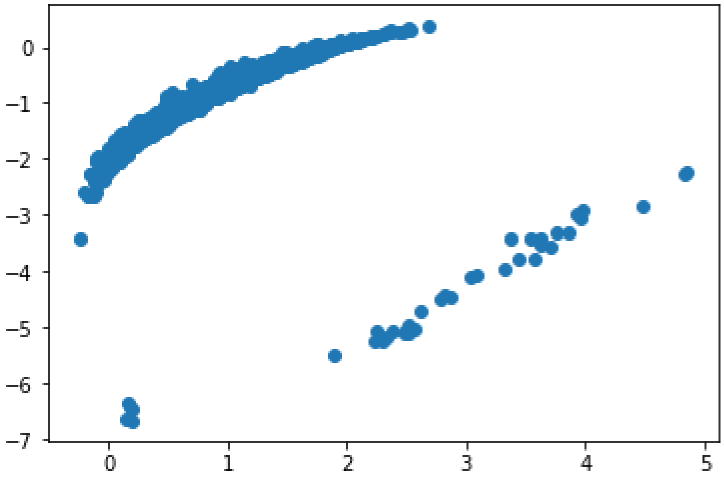


In [45]:

```
nodeID = []
coordinate1 = []
coordinate2 = []
with open("node2vec.emd","r") as rf:
    rf.readline()
    for line in rf:
        x, y, z = line.strip().split(" ")
        nodeID.append(x)
        coordinate1.append(y)
        coordinate2.append(z)

coordinate1 = list(map(float, coordinate1))
coordinate2 = list(map(float, coordinate2))

plt.scatter(coordinate1[:],coordinate2[:])
plt.show()
```



During the installation of LINE, gsl library was required, and I tried to install gsl-2.5. However, compiled program was still not excutable. Finally, I found 'gsl-bin' in apt-get, and LINE was runable after installation of 'gsl-bin'.

Conda create was used for create isolated environment for different Python version.

deepwalk and node2vec require two different versions of gensim

Node2vec and LINE finished job quickly, and Deepwalk took much long time.

In [ ]: