Download data

Data available via scikit-learn

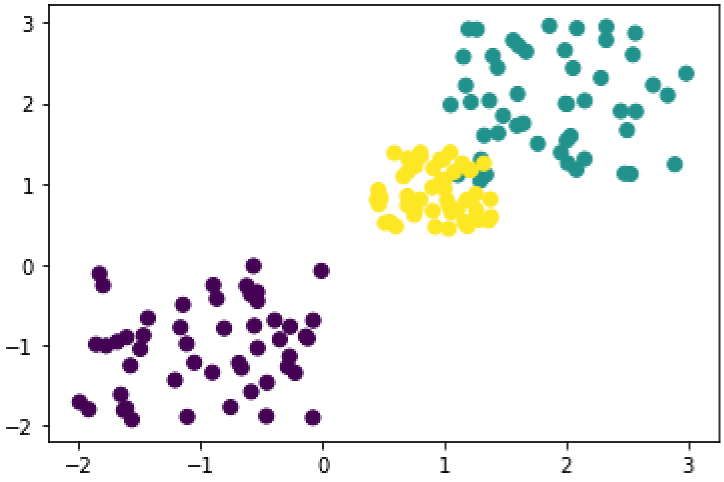
https://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets (https://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets)

https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1 (https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1)

In [1]:

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
##### clustering (unsupervised/descrete) ####
# random generated data
# Generating 3 destinct cluster, 2 of them slightly overlapping
X = -2 * np.random.rand(50,2)
X1 = 1 + 2 * np.random.rand(50,2)
X2 = 0.4 + np.random.rand(50,2)

X = np.append(X,X1,axis=0)
X = np.append(X,X2,axis=0)
colors = [0]*50+[1]*50+[2]*50
# === PLOTTING DATA
print(X.shape)
plt.scatter(X[ : , 0], X[ :, 1], s = 50, c = colors)
plt.show()
# ===

# gene expression
nci60 = pd.read_csv('data/NCI60.txt',sep='\t', index_col=False)
# print(nci60.shape)
# display(nci60)
### pre-process
print(nci60.shape)
# expression data starts at column 5
expr = nci60.iloc[:,5:]
print(expr.shape)


##### dimensionality reduction (unuspervised/continuous) #####
#https://idyll.pub/post/dimensionality-reduction-293e465c2a3443e8941b016d/
#gene expression
```

```
(150, 2)
```



```
(198, 389)
(198, 384)
```

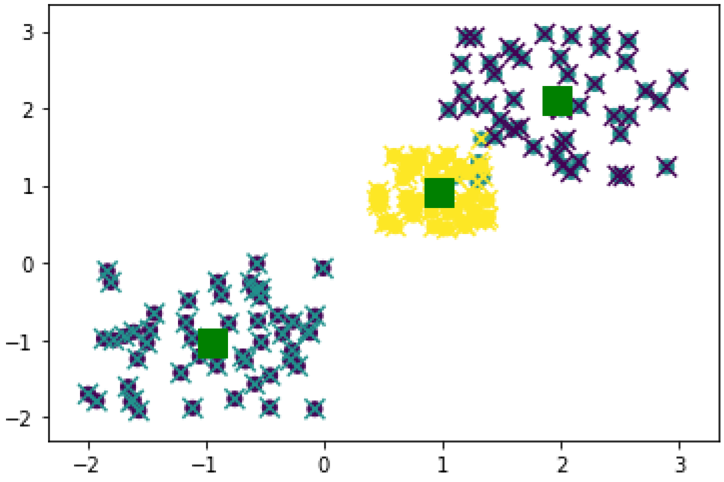
K-Means

K-means synthenic data

In [2]:

```
#Synthetic data
from sklearn.cluster import KMeans
Kmean = KMeans(n_clusters=3)
Kmean.fit(X)

print(Kmean.cluster_centers_)

# Plots original points
plt.scatter(X[ : , 0], X[ : , 1], s = 50, c=colors)
# Plots X labels after k-mens
plt.scatter(X[ : , 0], X[ : , 1], s = 100, marker='x', c=Kmean.labels_)

# Plots centroids
for cl in Kmean.cluster_centers_:
    plt.scatter(cl[0], cl[1], s=200, c='g', marker='s')
plt.show()
```

```
[[ 1.95967537  2.10576763]
 [-0.93685732 -1.02268501]
 [ 0.97224279  0.90732231]]
```

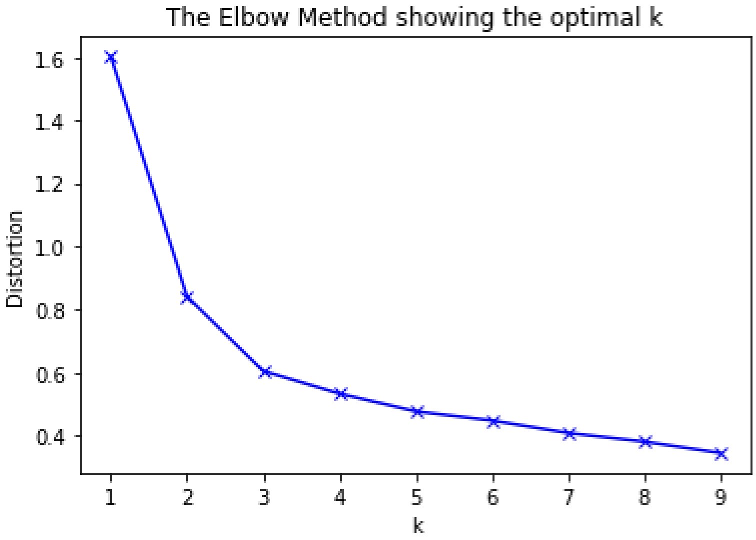


Elbow method for determing K

https://pythonprogramminglanguage.com/kmeans-elbow-method/
(https://pythonprogramminglanguage.com/kmeans-elbow-method/)

In [3]:

```
from scipy.spatial.distance import cdist

# k means determine k
distortions = []
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k).fit(X)
    kmeanModel.fit(X)
    # calcualtes the distance between each point and each cluster center
    # takes the minimum of each calcaulation (hence the difference between point and ITS cluster center)
    # takes average of all distances
    distortions.append(sum(np.min(cdist(X, kmeanModel.cluster_centers_, 'euclidean'), axis=1)) / X.shape[0])

# Plot the elbow
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```



K-means on Gene Expression data

In [4]:

```
display(nci60)
display(expr)
# fit the data, there is 9 tissue types
Kmean = KMeans(n_clusters=16)
Kmean.fit(expr)
# see how many rows ended up with each labels
print(pd.Series(Kmean.labels_).value_counts())
print(nci60.batch.value_counts())

# # how we do visulaize this?

plt.scatter(nci60.PC2, nci60.PC3, s = 10, c='black')
plt.scatter(nci60.PC2, nci60.PC3, s = 100, marker='x', c=Kmean.labels_)
plt.show()

plt.scatter(nci60.PC2, nci60.PC3, s = 10, c='black')
plt.scatter(nci60.PC2, nci60.PC3, s = 100, marker='x', c=nci60.batch.astype('category').cat.codes)
plt.show()
```

	tissue	batch	PC1	PC2	PC3	ACAD10	ACOT9	ACP5	
0	GSM803615	cancer_leukemia	1	-0.056842	-0.044581	-0.113268	-0.225523	-0.657666	-
1	GSM803674	cancer_leukemia	1	-0.056585	-0.042994	-0.114095	-0.115336	-0.745160	-
2	GSM803733	cancer_leukemia	1	-0.055702	-0.045556	-0.112062	0.061519	0.074016	-
3	GSM803616	cancer_leukemia	1	-0.061005	-0.036062	-0.123295	-0.148417	0.195879	-
4	GSM803675	cancer_leukemia	1	-0.060836	-0.032265	-0.127681	-0.198627	0.117102	-
5	GSM803734	cancer_leukemia	1	-0.056634	-0.049184	-0.128937	-0.083067	0.694710	-
6	GSM803617	cancer_leukemia	1	-0.058637	-0.027791	-0.107388	-0.154221	0.820919	-
7	GSM803676	cancer_leukemia	1	-0.058000	-0.029718	-0.105948	-0.213438	0.862094	-
8	GSM803735	cancer_leukemia	1	-0.053656	-0.054039	-0.126916	0.104375	0.651555	-
9	GSM803618	cancer_leukemia	1	-0.061899	-0.064274	-0.104224	-0.081437	0.827006	-
10	GSM803677	cancer_leukemia	1	-0.060902	-0.064753	-0.107330	-0.183441	0.852921	-
11	GSM803736	cancer_leukemia	1	-0.062722	-0.062989	-0.100799	-0.132196	1.089384	-
12	GSM803619	cancer_leukemia	1	-0.065582	-0.017462	-0.055630	-0.237170	0.864546	-
13	GSM803678	cancer_leukemia	1	-0.064839	-0.021478	-0.060321	-0.293436	0.878104	-
14	GSM803737	cancer_leukemia	1	-0.064517	-0.018884	-0.056781	-0.224833	0.954050	-
15	GSM803620	cancer_leukemia	1	-0.067114	-0.034380	-0.089720	-0.135940	0.892611	-
16	GSM803679	cancer_leukemia	1	-0.067142	-0.034965	-0.086943	-0.139152	0.960512	-
17	GSM803738	cancer_leukemia	1	-0.066316	-0.043587	-0.095912	-0.101752	1.026879	-
18	GSM803621	cancer_breast	1	-0.078022	0.047675	0.067187	-0.451185	0.940607	-
19	GSM803680	cancer_breast	1	-0.077909	0.048376	0.065161	-0.488317	0.976915	-
20	GSM803739	cancer_breast	1	-0.078001	0.044726	0.064545	-0.341475	1.100175	-
21	GSM803622	cancer_breast	1	-0.076179	0.063242	0.091899	-0.652644	0.618990	-
22	GSM803681	cancer_breast	1	-0.076027	0.064693	0.091411	-0.629674	0.598216	-
23	GSM803740	cancer_breast	1	-0.075180	0.063560	0.094382	-0.569921	0.655253	-
24	GSM803623	cancer_breast	1	-0.072455	-0.068390	-0.088982	-0.345481	0.251308	-
25	GSM803682	cancer_breast	1	-0.072869	-0.071157	-0.084977	-0.186985	0.359076	-
26	GSM803741	cancer_breast	1	-0.072874	-0.069472	-0.087196	-0.193552	0.421299	-
27	GSM803624	cancer_ovarian	1	-0.077715	0.004067	0.010490	-0.568091	0.616888	-
28	GSM803683	cancer_ovarian	1	-0.080732	0.006610	0.028655	-0.655521	0.658422	-
29	GSM803742	cancer_ovarian	1	-0.080749	0.006198	0.029227	-0.454352	0.804546	-
...	...	...	...	...	...	...	...	...	
168	GSM803672	cancer_melanoma	1	-0.072058	0.019348	0.045163	-0.340114	0.656020	
169	GSM803730	cancer_melanoma	1	-0.073148	0.018081	0.040628	-0.489069	0.747809	
170	GSM803787	cancer_melanoma	1	-0.074256	0.023280	0.047394	-0.375733	0.791462	
171	GSM803673	cancer_breast	1	-0.072343	-0.060815	-0.056714	-0.212072	0.612341	-

	tissue	batch	PC1	PC2	PC3	ACAD10	ACOT9	ACP5	
172	GSM803731	cancer_breast	1	-0.072348	-0.058145	-0.057504	-0.220204	0.639464	-
173	GSM803788	cancer_breast	1	-0.072220	-0.060890	-0.058461	-0.174758	0.691322	-
174	GSM18933	normal_brain	2	-0.021123	0.201263	-0.082889	-0.333775	-0.469165	-
175	GSM18934	normal_brain	2	-0.020917	0.203883	-0.088022	-0.298711	-0.389676	-
176	GSM18923	normal_brain	2	-0.017381	0.198927	-0.075512	-0.174182	-0.412700	-
177	GSM18924	normal_brain	2	-0.017633	0.198388	-0.081087	-0.121340	-0.431464	-
178	GSM18929	normal_brain	2	-0.018627	0.211270	-0.084287	-0.388273	-0.395204	-
179	GSM18930	normal_brain	2	-0.022591	0.188253	-0.072763	-0.278641	-0.470682	-
180	GSM18917	normal_brain	2	-0.023064	0.193976	-0.104031	-0.397680	-0.465619	-
181	GSM18918	normal_brain	2	-0.022722	0.198841	-0.098570	-0.125040	-0.515103	-
182	GSM18997	normal_ovary	2	-0.013734	0.170145	-0.110469	0.037138	-0.304679	-
183	GSM18998	normal_ovary	2	-0.013945	0.167311	-0.109352	-0.093727	-0.316046	-
184	GSM18957	normal_prostate	2	-0.034817	0.150056	-0.107471	-0.517187	-0.473539	-
185	GSM18958	normal_prostate	2	-0.033561	0.140488	-0.114631	-0.350905	-0.402720	-
186	GSM18889	normal_blood	2	-0.027525	0.126379	-0.165823	-0.263033	-0.073894	-
187	GSM18890	normal_blood	2	-0.028814	0.121179	-0.166312	-0.217535	-0.127278	-
188	GSM18877	normal_blood	2	-0.016501	0.113063	-0.174482	-0.011065	-0.153146	-
189	GSM18878	normal_blood	2	-0.016110	0.113215	-0.176995	-0.021145	-0.136286	-
190	GSM18875	normal_blood	2	-0.022206	0.108500	-0.164317	-0.182053	-0.337838	-
191	GSM18876	normal_blood	2	-0.019868	0.119745	-0.167830	-0.325988	-0.284017	-
192	GSM18949	normal_lung	2	-0.026617	0.161477	-0.096402	-0.422320	-0.494245	
193	GSM18950	normal_lung	2	-0.022855	0.170397	-0.100391	-0.378157	-0.352568	
194	GSM19001	normal_skin	2	-0.014454	0.172637	-0.113487	0.444467	-0.549839	-
195	GSM19002	normal_skin	2	-0.014447	0.174409	-0.117696	0.721574	-0.511333	-
196	GSM18955	normal_kidney	2	-0.030083	0.121079	-0.114942	-0.160365	-0.533933	
197	GSM18956	normal_kidney	2	-0.029734	0.126442	-0.114295	-0.113225	-0.417798	

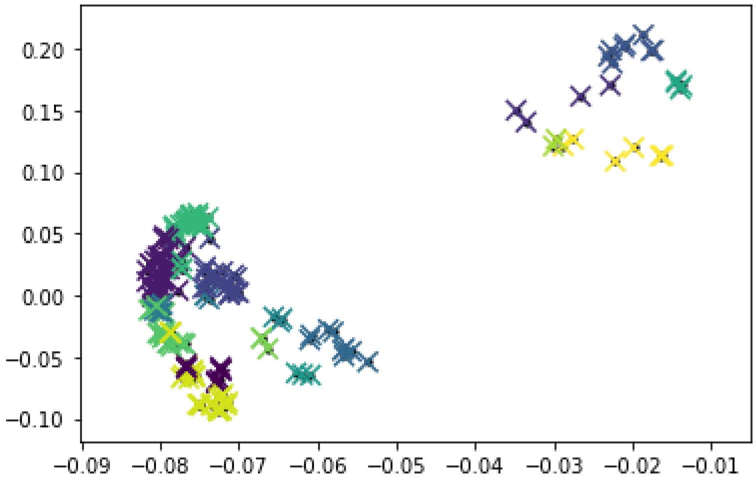
198 rows × 389 columns

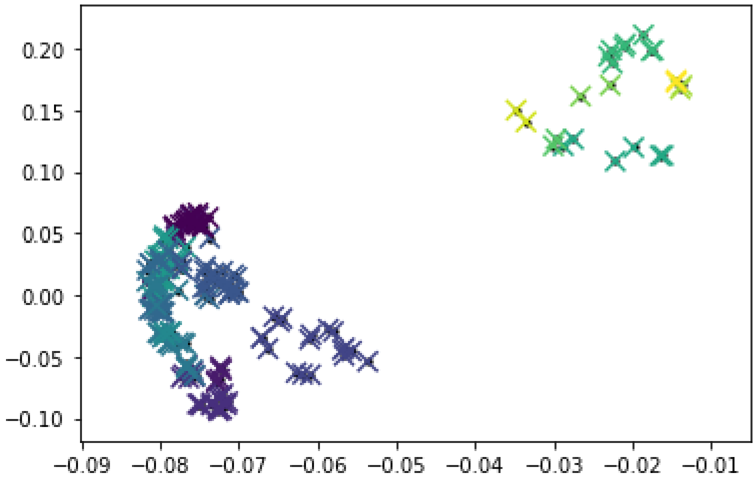
	ACAD10	ACOT9	ACP5	ACSL3	ACTN1	ACTN4	ACVR2A	ADCK4	
0	-0.113268	-0.225523	-0.657666	-0.719608	0.976008	1.111694	0.503365	-0.544117	-1.17
1	-0.114095	-0.115336	-0.745160	-0.694094	0.943915	1.002691	0.541647	-0.107079	-1.15
2	-0.112062	0.061519	0.074016	-0.465058	1.225343	0.800822	0.751530	-0.304991	-1.10
3	-0.123295	-0.148417	0.195879	-0.766730	1.087944	1.520482	1.073094	-0.145891	-1.02
4	-0.127681	-0.198627	0.117102	-0.464126	1.095090	1.549793	1.069821	0.029051	-1.16
5	-0.128937	-0.083067	0.694710	-0.342008	0.865687	1.393348	1.056934	-0.321569	-1.23
6	-0.107388	-0.154221	0.820919	-0.552283	1.209606	1.006519	0.385098	-0.079533	-1.05
7	-0.105948	-0.213438	0.862094	-0.628550	1.167257	1.036698	0.333438	-0.449753	-1.08
8	-0.126916	0.104375	0.651555	-0.419143	0.803217	1.201640	1.214298	0.505413	-0.92
9	-0.104224	-0.081437	0.827006	-0.493754	0.311060	0.098200	0.192633	-0.071105	-1.10
10	-0.107330	-0.183441	0.852921	-0.470541	0.290231	-0.026508	0.122113	0.041792	-1.15
11	-0.100799	-0.132196	1.089384	-0.307204	0.011621	-0.278588	0.241486	-0.426995	-1.20
12	-0.055630	-0.237170	0.864546	-0.840067	0.900479	1.689163	1.351323	-0.071631	-1.15
13	-0.060321	-0.293436	0.878104	-0.863561	0.920503	1.672084	1.384987	-0.059195	-1.18
14	-0.056781	-0.224833	0.954050	-0.845293	0.876780	1.488628	1.321521	-0.125633	-1.17
15	-0.089720	-0.135940	0.892611	-0.600149	1.010743	1.150900	0.959611	0.286767	-1.19
16	-0.086943	-0.139152	0.960512	-0.392410	0.965920	1.242161	1.051196	0.258448	-1.17
17	-0.095912	-0.101752	1.026879	-0.437341	0.814904	0.986422	0.879281	-0.173074	-1.14
18	0.067187	-0.451185	0.940607	-0.984481	0.755387	1.609322	0.873873	-0.225148	-1.42
19	0.065161	-0.488317	0.976915	-0.925485	0.747708	1.592533	0.958221	-0.238177	-1.53
20	0.064545	-0.341475	1.100175	-0.898786	0.689580	1.450014	0.890532	-0.462409	-1.49
21	0.091899	-0.652644	0.618990	-0.868125	0.472090	1.679343	0.742354	0.162686	-1.55
22	0.091411	-0.629674	0.598216	-0.941766	0.489222	1.749456	0.764734	0.128605	-1.54
23	0.094382	-0.569921	0.655253	-0.793412	0.447984	1.574006	0.702518	0.077412	-1.53
24	-0.088982	-0.345481	0.251308	-0.751214	0.637299	1.005870	0.604459	0.118928	-1.22
25	-0.084977	-0.186985	0.359076	-0.810905	0.574677	1.093582	0.846591	0.107682	-1.33
26	-0.087196	-0.193552	0.421299	-0.793472	0.852449	0.861591	0.725070	0.059282	-1.33
27	0.010490	-0.568091	0.616888	-1.029302	0.960891	1.780489	0.482121	0.290475	-1.53
28	0.028655	-0.655521	0.658422	-0.864975	0.858157	1.999481	0.733460	0.219468	-1.54
29	0.029227	-0.454352	0.804546	-0.732586	0.702438	1.773233	0.784342	0.139265	-1.52
...	...	...	...	...	...	...	...	...	
168	0.045163	-0.340114	0.656020	0.284666	1.160679	1.300248	0.765416	-0.135345	-1.55
169	0.040628	-0.489069	0.747809	0.416655	1.144911	1.324996	0.843852	-0.102116	-1.49
170	0.047394	-0.375733	0.791462	0.308832	1.202791	1.445058	1.075510	-0.245130	-1.52
171	-0.056714	-0.212072	0.612341	-0.869982	0.143276	1.074310	0.387683	-0.259083	-1.29

	ACAD10	ACOT9	ACP5	ACSL3	ACTN1	ACTN4	ACVR2A	ADCK4	
172	-0.057504	-0.220204	0.639464	-0.898149	0.210101	1.088200	0.397209	-0.171641	-1.309
173	-0.058461	-0.174758	0.691322	-0.808267	0.128221	0.929756	0.411656	-0.151958	-1.407
174	-0.082889	-0.333775	-0.469165	-0.496119	0.318325	-0.132482	-0.085508	-0.144946	-0.494
175	-0.088022	-0.298711	-0.389676	-0.470748	0.364341	-0.241940	-0.065919	-0.263852	-0.495
176	-0.075512	-0.174182	-0.412700	-0.415713	0.222922	-0.035956	-0.012674	-0.008977	-0.403
177	-0.081087	-0.121340	-0.431464	-0.451556	0.295517	-0.056207	0.129624	-0.205905	-0.415
178	-0.084287	-0.388273	-0.395204	-0.434761	0.571569	0.309640	0.185475	-0.374542	-0.497
179	-0.072763	-0.278641	-0.470682	-0.414257	0.503134	0.759271	0.424932	-0.360431	-0.480
180	-0.104031	-0.397680	-0.465619	-0.379339	0.083240	-0.199214	-0.274718	-0.059897	-0.464
181	-0.098570	-0.125040	-0.515103	-0.340250	0.227368	-0.239526	-0.158643	-0.011989	-0.523
182	-0.110469	0.037138	-0.304679	-0.121934	-0.175915	0.297356	0.196031	-0.190083	-0.354
183	-0.109352	-0.093727	-0.316046	-0.125807	-0.098384	0.229316	0.235352	-0.113389	-0.332
184	-0.107471	-0.517187	-0.473539	-0.133990	-0.148965	2.621749	0.281129	-0.237538	-0.570
185	-0.114631	-0.350905	-0.402720	-0.254863	-0.187989	1.563886	1.812410	-0.388373	-0.58
186	-0.165823	-0.263033	-0.073894	-0.333808	-0.170585	-0.402311	0.006430	-0.347448	-0.437
187	-0.166312	-0.217535	-0.127278	-0.354470	-0.188380	-0.419604	0.021133	-0.344235	-0.420
188	-0.174482	-0.011065	-0.153146	-0.158601	-0.322998	0.202676	0.079936	-0.312336	-0.425
189	-0.176995	-0.021145	-0.136286	-0.047678	-0.256053	0.260258	0.115175	-0.107318	-0.420
190	-0.164317	-0.182053	-0.337838	-0.265893	-0.267236	-0.440287	0.671119	-0.286805	-0.428
191	-0.167830	-0.325988	-0.284017	-0.260497	-0.382963	-0.311795	0.670770	-0.182638	-0.465
192	-0.096402	-0.422320	-0.494245	2.889138	-0.477018	0.377125	0.522240	-0.429912	-0.573
193	-0.100391	-0.378157	-0.352568	2.035244	-0.439438	0.411293	0.383717	-0.274559	-0.507
194	-0.113487	0.444467	-0.549839	-0.468444	0.120016	0.582435	0.755217	0.606611	-0.659
195	-0.117696	0.721574	-0.511333	-0.326439	-0.250943	0.352607	0.454183	0.761518	-0.52
196	-0.114942	-0.160365	-0.533933	0.477747	-0.362841	-0.178120	1.122167	-0.011467	-0.499
197	-0.114295	-0.113225	-0.417798	0.289113	-0.359780	-0.035457	1.201101	-0.023936	-0.523

198 rows × 384 columns

```
1     42
14    27
10    27
3     23
11    19
6      9
5      9
0      9
4      8
15     6
9      4
2      4
12     3
8      3
7      3
13     2
dtype: int64
cancer_non-small cell lung    26
cancer_melanoma               26
cancer_renal                  23
cancer_colon                  21
cancer_ovarian                21
cancer_leukemia               18
cancer_CNS                    18
cancer_breast                 15
normal_brain                   8
cancer_prostate                6
normal_blood                   6
normal_ovary                   2
normal_kidney                  2
normal_lung                    2
normal_prostate                2
normal_skin                    2
Name: batch, dtype: int64
```



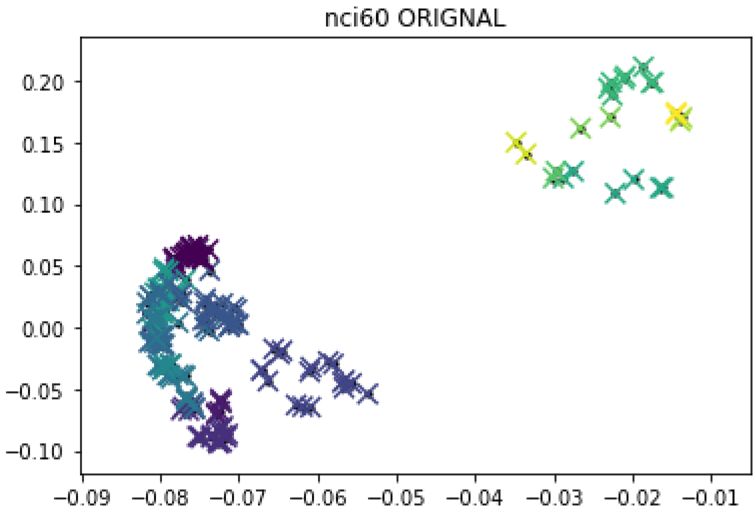


Hierarchical Clustering

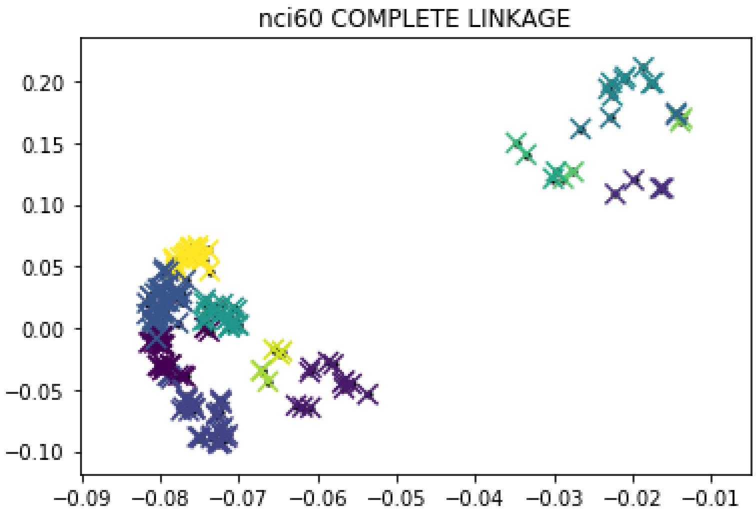
In [5]:

```
from sklearn.cluster import AgglomerativeClustering
single = AgglomerativeClustering(n_clusters=16, linkage='single').fit(expr)
complete = AgglomerativeClustering(n_clusters=16, linkage='complete').fit(expr
)
average = AgglomerativeClustering(n_clusters=16, linkage='average').fit(expr)


plt.scatter(nci60.PC2, nci60.PC3, s = 10, c='black')
plt.scatter(nci60.PC2, nci60.PC3, s = 100, marker='x', c=nci60.batch.astype('c
ategory').cat.codes)
plt.title("nci60 ORIGNAL")
plt.show()

plt.scatter(nci60.PC2, nci60.PC3, s = 10, c='black')
plt.scatter(nci60.PC2, nci60.PC3, s = 100, marker='x', c=complete.labels_)
plt.title("nci60 COMPLETE LINKAGE")
plt.show()

plt.scatter(nci60.PC2, nci60.PC3, s = 10, c='black')
plt.scatter(nci60.PC2, nci60.PC3, s = 100, marker='x', c=single.labels_)
plt.title("nci60 SINGLE LINKAGE")
plt.show()

plt.scatter(nci60.PC2, nci60.PC3, s = 10, c='black')
plt.scatter(nci60.PC2, nci60.PC3, s = 100, marker='x', c=average.labels_)
plt.title("nci60 AVERAGE LINKAGE")
plt.show()


# ON YOU OWN - Explore for synthetic data set (comment nci60 plots and uncomme
nt the ones below)
display(X)
single = AgglomerativeClustering(n_clusters=16, linkage='single').fit(X)
complete = AgglomerativeClustering(n_clusters=16, linkage='complete').fit(X)
average = AgglomerativeClustering(n_clusters=16, linkage='average').fit(X)

#### SYNTHETIC DATA ####
plt.scatter(X[ : , 0], X[ :, 1], s = 50, c = colors)
plt.title("SYNTHETIC DATA ORIGNAL")
plt.show()

plt.scatter(X[ : , 0], X[ :, 1], s = 50, c = complete.labels_)
plt.title("SYNTHETIC DATA COMPLETE LINKAGE")
plt.show()

plt.scatter(X[ : , 0], X[ :, 1], s = 50, c = single.labels_)
plt.title("SYNTHETIC DATA SINGLE LINKAGE")
plt.show()

plt.scatter(X[ : , 0], X[ :, 1], s = 50, c = average.labels_)
plt.title("SYNTHETIC DATA AVERAGE LINKAGE")
plt.show()

# reset assigments back to nci60
from sklearn.cluster import AgglomerativeClustering
single = AgglomerativeClustering(n_clusters=16, linkage='single').fit(expr)
```

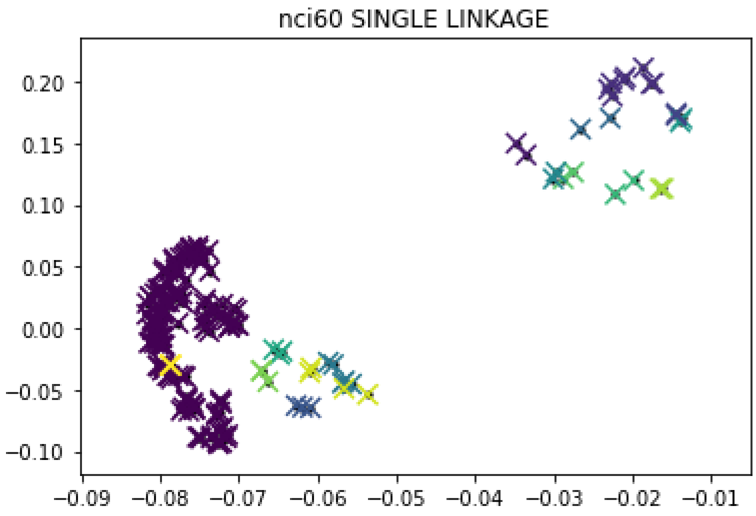
```
complete = AgglomerativeClustering(n_clusters=16, linkage='complete').fit(expr
)
average = AgglomerativeClustering(n_clusters=16, linkage='average').fit(expr)
```



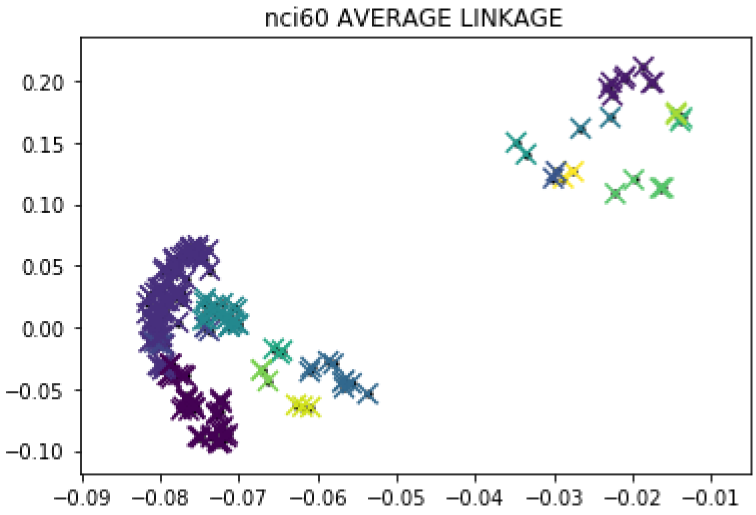
nci60 ORIGNAL



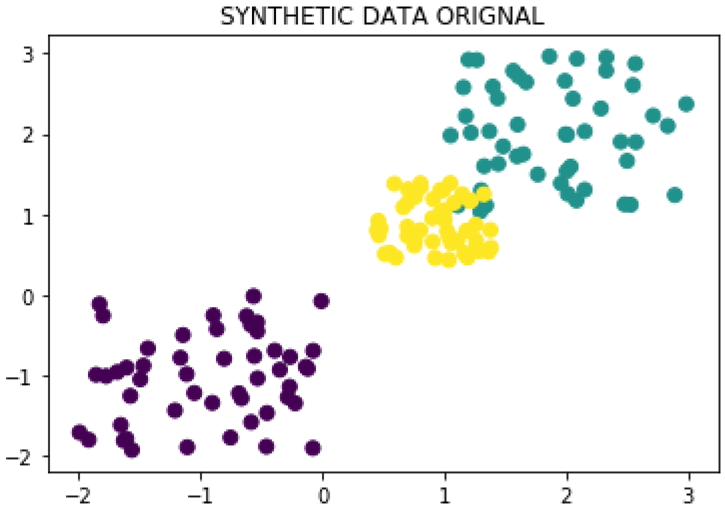
nci60 COMPLETE LINKAGE



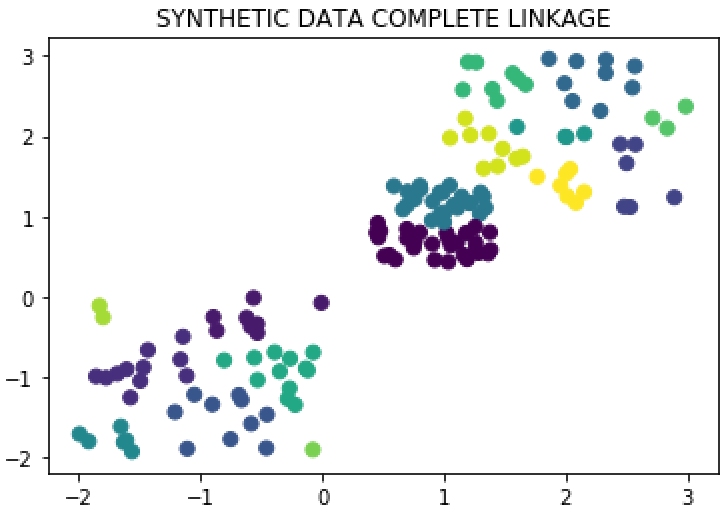
nci60 SINGLE LINKAGE

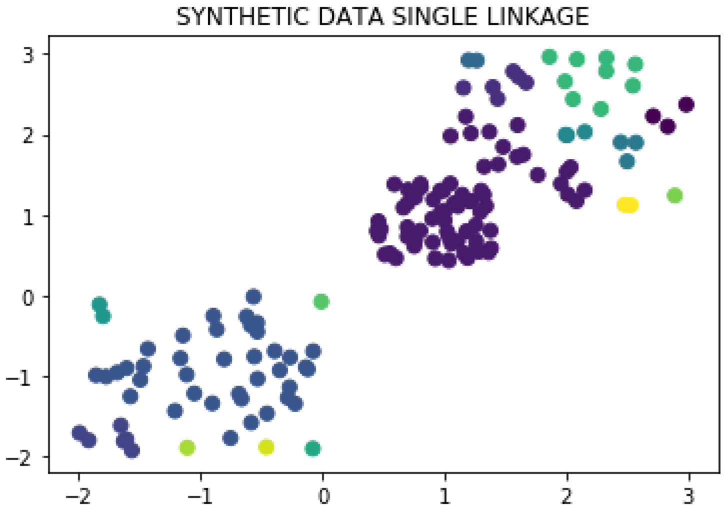

nci60 AVERAGE LINKAGE
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.09
−0.08
−0.07
−0.06
−0.05
−0.04
−0.03
−0.02
−0.01

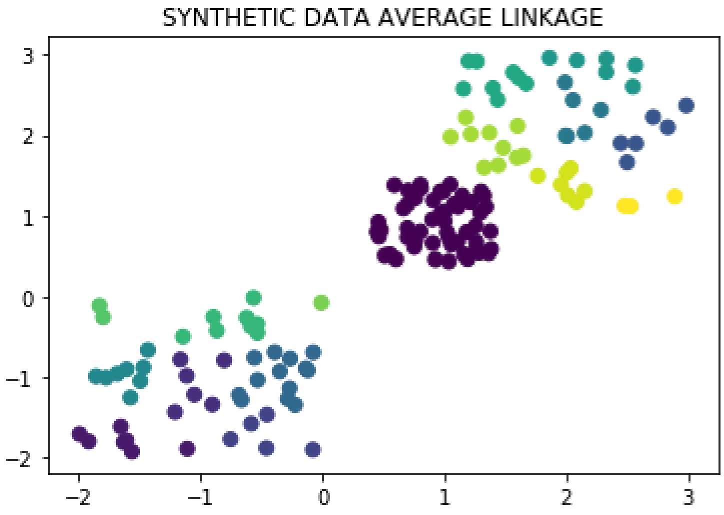
```
array([[-0.58614565, -0.36467805],
       [-0.90224172, -1.33496352],
       [-0.56364119, -0.01041619],
       [-1.16423667, -0.77561396],
       [-1.68449171, -0.95266979],
       [-1.77542656, -1.00239811],
       [-1.60880872, -1.78030268],
       [-1.49470448, -1.04238976],
       [-0.26301732, -0.76500631],
       [-0.07305393, -0.68794648],
       [-1.10846992, -1.8851535 ],
       [-0.55698241, -0.75416486],
       [-0.00628236, -0.07167975],
       [-0.80483157, -0.78645138],
       [-0.68348971, -1.21672748],
       [-0.53034776, -1.03105704],
       [-0.75248915, -1.76616106],
       [-0.38960591, -0.68625127],
       [-0.34755626, -0.92428656],
       [-1.85762719, -0.9855118 ],
       [-0.07566832, -1.89662912],
       [-0.45285802, -1.4607254 ],
       [-0.2824667 , -1.26440686],
       [-1.82988391, -0.10859332],
       [-0.53338648, -0.3349189 ],
       [-0.66214102, -1.27882708],
       [-0.13333926, -0.88581654],
       [-1.65465471, -1.60938094],
       [-1.04998091, -1.21240957],
       [-1.91863054, -1.79395942],
       [-1.60720815, -0.89802712],
       [-1.80038576, -0.25211163],
       [-1.56347422, -1.91999473],
       [-0.53235499, -0.44563425],
       [-0.8657694 , -0.41761328],
       [-1.99339366, -1.70266561],
       [-0.22149967, -1.33982027],
       [-1.20923908, -1.42932585],
       [-1.11303828, -0.97850941],
       [-0.11786963, -0.91070268],
       [-1.63043748, -1.80375125],
       [-1.46858995, -0.87393386],
       [-1.43207982, -0.65864118],
       [-1.57573189, -1.24862358],
       [-0.62018282, -0.25776806],
       [-0.58299694, -1.57422263],
       [-0.45764442, -1.87743781],
       [-1.14352222, -0.49266627],
       [-0.89383565, -0.24881856],
       [-0.26715187, -1.13448574],
       [ 1.76898703,  1.49561098],
       [ 1.4387817 ,  2.44038041],
       [ 2.09132662,  2.93061421],
       [ 2.98981008,  2.36895858],
       [ 1.56717348,  2.78106887],
       [ 1.61127604,  2.71401924],
       [ 2.71744288,  2.22513781],
       [ 1.959899  ,  1.38546294],
       [ 2.33119784,  2.78390369],
       [ 1.86524473,  2.96141989],
       [ 1.99299893,  2.65618008],
       [ 2.89608379,  1.23911675],
       [ 2.45132214,  1.90051173],
       [ 1.05247248,  1.97875835],
       [ 2.48292014,  1.1241694 ],
       [ 2.55147741,  2.60307174],
       [ 2.03962428,  1.59051618],
       [ 1.39993761,  2.58550107],
       [ 1.2680024 ,  2.91656323],
       [ 2.05881088,  2.43574527],
       [ 1.29636406,  1.03593222],
       [ 2.28846681,  2.3138754 ],
       [ 2.00731405,  1.53561653],
       [ 2.15568529,  2.0297436 ],
       [ 1.59862766,  1.72101198],
       [ 1.32805481,  1.59852493],
       [ 1.1771605 ,  2.21976727],
       [ 2.15437945,  1.30521939],
       [ 1.30310814,  1.2999548 ],
       [ 1.48448298,  1.84354467],
       [ 1.44391343,  1.62668923],
       [ 1.11186395,  1.10716078],
       [ 1.21907568,  2.01248714],
       [ 1.60315398,  2.11461661],
       [ 2.01002779,  1.98839608],
       [ 2.53332976,  1.11888029],
       [ 2.08694287,  1.17245155],
       [ 1.99812262,  1.99442514],
       [ 2.33307681,  2.94834117],
       [ 1.65077505,  1.74746583],
       [ 1.19969185,  2.92042695],
       [ 2.57607676,  1.89648143],
       [ 1.67354851,  2.64123566],
       [ 2.01586583,  1.25368482],
       [ 2.83733092,  2.09792749],
       [ 1.3437576 ,  1.110126  ],
       [ 1.15778014,  2.57640731],
       [ 2.50346954,  1.66319848],
       [ 2.57104978,  2.86975664],
       [ 1.37125398,  2.03118208],
       [ 1.28620449,  0.54859098],
       [ 0.80459241,  1.38497415],
       [ 1.21288403,  1.16625276],
       [ 0.69716284,  0.84597589],
       [ 1.32613809,  1.24920548],
       [ 0.96805901,  1.30053084],
       [ 1.02371011,  0.79221919],
       [ 1.19104751,  0.48697259],
       [ 1.19155568,  0.47054659],
       [ 0.90220226,  0.95225181],
       [ 0.75326899,  0.61207495],
       [ 0.45820174,  0.92010387],
       [ 0.5537928 ,  0.52225853],
       [ 0.91012194,  1.18967714],
       [ 0.47844702,  0.82180734],
       [ 0.51246173,  0.51084622],
       [ 1.0389799 ,  0.72302414],
       [ 1.38513573,  0.58575292],
       [ 1.21817817,  1.17179802],
       [ 0.80362472,  0.80367465],
       [ 0.76322647,  0.68805246],
       [ 1.25939339,  0.87179997],
       [ 0.92613987,  0.46175271],
       [ 1.26631977,  0.58755   ],
       [ 0.69747711,  0.73022857],
       [ 0.60272985,  0.46671802],
       [ 0.7012034 ,  1.14674684],
       [ 0.44734701,  0.79998993],
       [ 0.972434  ,  1.02857149],
       [ 1.18408048,  0.79749879],
       [ 0.46274471,  0.74044433],
       [ 1.03737159,  0.43769048],
       [ 0.99249206,  1.30034505],
       [ 0.59008587,  1.37932566],
       [ 1.146961  ,  1.25012106],
       [ 1.00294131,  0.93070625],
       [ 1.08194448,  1.14427543],
       [ 0.80895044,  1.34209175],
       [ 0.70274191,  1.31180315],
       [ 1.36554818,  0.54074522],
       [ 1.37986915,  0.80511684],
       [ 0.9055497 ,  0.65983965],
       [ 0.66417412,  1.08879116],
       [ 1.26073235,  0.67629915],
       [ 1.05027225,  1.38771284],
       [ 1.00743184,  1.05344743],
       [ 1.1110193 ,  0.67909654],
       [ 1.05847553,  0.64263971],
       [ 1.16219881,  0.52372496],
       [ 0.76257976,  1.21936463]])
```



SYNTHETIC DATA ORIGNAL

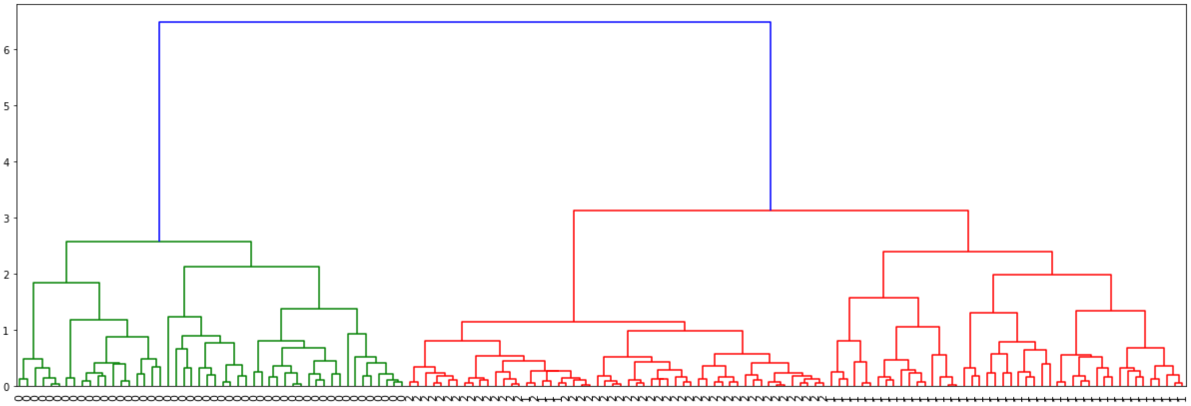






Hierarchical Clustering with Scipy

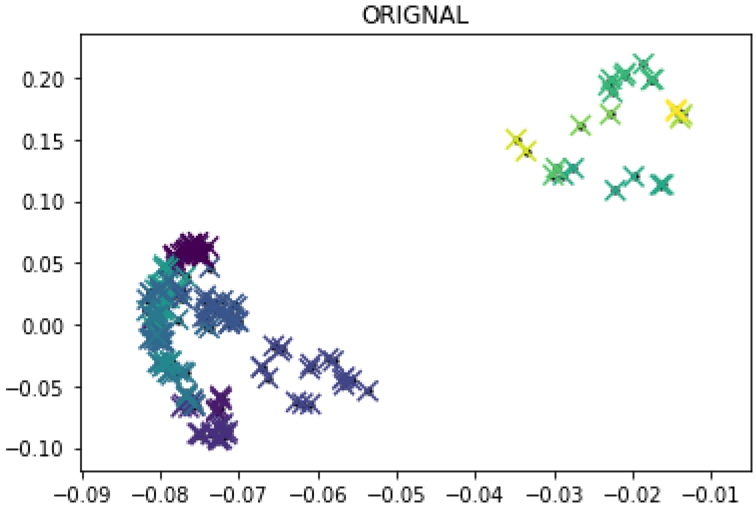
In [6]:
```
from scipy.cluster.hierarchy import dendrogram, linkage
# cannot choose number of clusters
Z = linkage(expr,'complete')
# display(Z)
# plt.figure(figsize=(21, 7))
# dendrogram(Z,orientation='top',
#              labels=nci60.tissue,
#              distance_sort='descending',
#              leaf_font_size='11',
#              show_leaf_counts=True)
# plt.show()

# synthetic data
Z = linkage(X,'complete')
plt.figure(figsize=(21, 7))
dendrogram(Z,orientation='top',
            labels=colors,
            distance_sort='descending',
            leaf_font_size='11',
            show_leaf_counts=True)
plt.show()

# reset back to nci60
Z = linkage(expr,'complete')
```

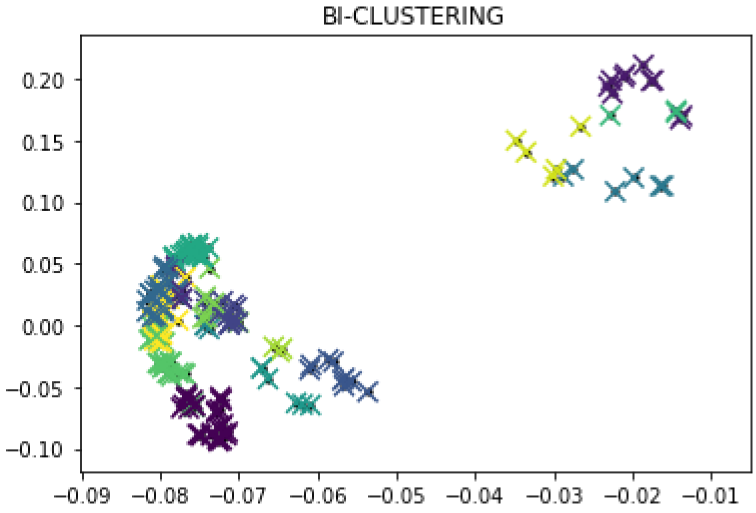


Bi-Clustering for Omics data

In [7]:

```
from sklearn.cluster import SpectralBiclustering
bicluster = SpectralBiclustering(n_clusters=16, random_state=0).fit(expr)

plt.scatter(nci60.PC2, nci60.PC3, s = 10, c='black')
plt.scatter(nci60.PC2, nci60.PC3, s = 100, marker='x', c=nci60.batch.astype('c
ategory').cat.codes)
plt.title("ORIGNAL")
plt.show()

plt.scatter(nci60.PC2, nci60.PC3, s = 10, c='black')
plt.scatter(nci60.PC2, nci60.PC3, s = 100, marker='x', c=bicluster.row_labels_
)
plt.title("BI-CLUSTERING")
plt.show()
```





Lab

Try unsupervised learning on the synthetic dataset in Orange

Use code below for the python data widget

alt text

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from Orange.data import Domain, Table
##### clustering (unsupervised/descrete) ####
# radmon generated data
# Generating 3 destinct cluster, 2 of them slightly overlapping
X = -2 * np.random.rand(50,2)
X1 = 1 + 2 * np.random.rand(50,2)
X2 = 0.4 + np.random.rand(50,2)

X = np.append(X,X1,axis=0)
X = np.append(X,X2,axis=0)
colors = [0]*50+[1]*50+[2]*50
dfX = pd.DataFrame(X)
dfX['color'] = colors

out_data = Table(dfX)
```