

Visualizing LUAD Gene Expression

October 5, 2019

1 Visualizing LUAD Gene Expression

1.0.1 importing libraries

```
In [24]: import pandas as pd
import numpy as np
```

1.0.2 reading data

```
In [25]: df = pd.read_csv('LUAD_rsem_normal_tumor.sample.txt', sep='\t')
```

1.0.3 exploring the dataframe

```
In [26]: df.head()
```

```
Out[26]:
```

	Gene	Normal	Normal.1	Normal.2	Normal.3	Normal.4	\
0	A1BG	1	86.1017	51.2577	103.7700	99.6544	
1	A1CF	29974	0.0000	0.0000	0.0000	0.0000	
2	A2BP1	54715	0.3082	0.0000	1.3937	0.0000	
3	A2LD1	87769	57.1125	60.2753	81.6237	78.6636	
4	A2M	2	133594.6379	133695.1685	113644.1812	73240.7488	
		Normal.5	Normal.6	Normal.7	Normal.8	...	Tumor.5 \
0		138.7365	74.2530	237.2049	243.9686	...	359.4863
1		0.5692	0.0000	0.3747	0.0000	...	0.0000
2		0.0000	0.4745	1.1240	0.0000	...	0.0000
3		68.1047	70.1963	91.5024	59.8984	...	107.2146
4		68335.7314	135432.6125	61787.8157	33512.8328	...	4884.4635
		Tumor.6	Tumor.7	Tumor.8	Tumor.9	Tumor.10	Tumor.11 \
0		79.6312	54.5498	73.1611	158.8152	41.9737	163.1712
1		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2		0.0000	0.3603	2.8188	1.7683	0.0000	0.0000
3		67.9799	68.1639	161.5436	83.8904	43.9649	101.5073
4		47086.2951	36847.5940	28800.7202	44583.4660	15691.2018	19518.7370
		Tumor.12	Tumor.13	Tumor.14			
0		64.8333	29.2505	242.1940			

```

1      0.0000      0.0000      0.0000
2    264.5390      0.0000      8.2090
3     81.1028    168.4186     79.9254
4  21294.6135  11774.1682  17508.8284

```

```
[5 rows x 31 columns]
```

```
In [27]: display(df.head())
display(df.shape)
```

	Gene	Normal	Normal.1	Normal.2	Normal.3	Normal.4	\
0	A1BG	1	86.1017	51.2577	103.7700	99.6544	
1	A1CF	29974	0.0000	0.0000	0.0000	0.0000	
2	A2BP1	54715	0.3082	0.0000	1.3937	0.0000	
3	A2LD1	87769	57.1125	60.2753	81.6237	78.6636	
4	A2M	2	133594.6379	133695.1685	113644.1812	73240.7488	

	Normal.5	Normal.6	Normal.7	Normal.8	...	Tumor.5	\
0	138.7365	74.2530	237.2049	243.9686	...	359.4863	
1	0.5692	0.0000	0.3747	0.0000	...	0.0000	
2	0.0000	0.4745	1.1240	0.0000	...	0.0000	
3	68.1047	70.1963	91.5024	59.8984	...	107.2146	
4	68335.7314	135432.6125	61787.8157	33512.8328	...	4884.4635	

	Tumor.6	Tumor.7	Tumor.8	Tumor.9	Tumor.10	Tumor.11	\
0	79.6312	54.5498	73.1611	158.8152	41.9737	163.1712	
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
2	0.0000	0.3603	2.8188	1.7683	0.0000	0.0000	
3	67.9799	68.1639	161.5436	83.8904	43.9649	101.5073	
4	47086.2951	36847.5940	28800.7202	44583.4660	15691.2018	19518.7370	

	Tumor.12	Tumor.13	Tumor.14
0	64.8333	29.2505	242.1940
1	0.0000	0.0000	0.0000
2	264.5390	0.0000	8.2090
3	81.1028	168.4186	79.9254
4	21294.6135	11774.1682	17508.8284

```
[5 rows x 31 columns]
```

```
(20502, 31)
```

1.0.4 problems w the data

1. index is random
2. normal and tumor separated by trial readings
3. gene expression is not tidy

1.0.5 we want something like || gene || normal or tumor || expression level ||

1.0.6 get columns

```
In [28]: df.columns.tolist()
```

```
Out[28]: ['Gene',  
          'Normal',  
          'Normal.1',  
          'Normal.2',  
          'Normal.3',  
          'Normal.4',  
          'Normal.5',  
          'Normal.6',  
          'Normal.7',  
          'Normal.8',  
          'Normal.9',  
          'Normal.10',  
          'Normal.11',  
          'Normal.12',  
          'Normal.13',  
          'Normal.14',  
          'Tumor',  
          'Tumor.1',  
          'Tumor.2',  
          'Tumor.3',  
          'Tumor.4',  
          'Tumor.5',  
          'Tumor.6',  
          'Tumor.7',  
          'Tumor.8',  
          'Tumor.9',  
          'Tumor.10',  
          'Tumor.11',  
          'Tumor.12',  
          'Tumor.13',  
          'Tumor.14']
```

1.0.7 reshape or “melt” data to make it “stacked” (long format instead of wide format)

```
In [29]: ?pd.DataFrame.melt
```

```
In [30]: df_stack = pd.melt(df, id_vars=['Gene'], value_vars = ['Normal',  
          'Normal.1',  
          'Normal.2',  
          'Normal.3',  
          'Normal.4',  
          'Normal.5',  
          'Normal.6',
```

```

'Normal.7',
'Normal.8',
'Normal.9',
'Normal.10',
'Normal.11',
'Normal.12',
'Normal.13',
'Normal.14',
'Tumor',
'Tumor.1',
'Tumor.2',
'Tumor.3',
'Tumor.4',
'Tumor.5',
'Tumor.6',
'Tumor.7',
'Tumor.8',
'Tumor.9',
'Tumor.10',
'Tumor.11',
'Tumor.12',
'Tumor.13',
'Tumor.14'], var_name='Cancerous', value_name='Expression')

```

In [31]: df_stack.shape

Out[31]: (615060, 3)

In [32]: df_stack.tail()

```

Out[32]:
      Gene Cancerous  Expression
615055  ZYG11A  Tumor.14      61.9403
615056  ZYG11B  Tumor.14     807.4627
615057    ZYX  Tumor.14    4368.6567
615058  ZZEF1  Tumor.14    1033.5821
615059   ZZZ3  Tumor.14     641.0448

```

1.0.8 remove the numbers in 'Cancerous'

```

In [33]: for index, data in df_stack.iterrows():
          if (data.Cancerous.startswith('Normal')):
              df_stack.at[index, 'Cancerous'] = 'Normal'
          else:
              df_stack.at[index, 'Cancerous'] = 'Tumor'

```

In [34]: df_stack

```

Out[34]:
      Gene Cancerous  Expression
0      A1BG   Normal      1.0000

```

1	A1CF	Normal	29974.0000
2	A2BP1	Normal	54715.0000
3	A2LD1	Normal	87769.0000
4	A2M	Normal	2.0000
...
615055	ZYG11A	Tumor	61.9403
615056	ZYG11B	Tumor	807.4627
615057	ZYX	Tumor	4368.6567
615058	ZZEF1	Tumor	1033.5821
615059	ZZZ3	Tumor	641.0448

[615060 rows x 3 columns]

1.0.9 using groupby to calculate mean for normal and tumor

```
In [35]: df_stack.groupby(['Cancerous'])['Expression'].mean()
```

```
Out [35]: Cancerous
Normal    123734.934183
Tumor      963.243134
Name: Expression, dtype: float64
```

1.0.10 log transform gene expression to normalize, ie. put normal and tumor values on the same scale

```
In [37]: expr = df_stack['Expression']
df_stack['Expression_Norm'] = np.log2(expr+1)
df_stack.head()
```

```
Out [37]:
```

	Gene	Cancerous	Expression	Expression_Norm
0	A1BG	Normal	1.0	1.000000
1	A1CF	Normal	29974.0	14.871472
2	A2BP1	Normal	54715.0	15.739675
3	A2LD1	Normal	87769.0	16.421440
4	A2M	Normal	2.0	1.584963

1.0.11 query

```
In [38]: df_stack.query('Expression_Norm > 1.47 & Expression_Norm <1.5').get('Gene').unique()
```

```
Out [38]: array(['KCNK4', 'HBG2', 'KRTAP5-2', 'SLC25A21', 'SLC5A10', 'CDRT1',
                'AACSL', 'ABCG5', 'ACSBG2', 'ADCYAP1', 'AKNAD1', 'ANKRD55',
                'AQP7P1', 'ASAH2', 'ASB12', 'ASTN1', 'ATP4B', 'BCL8', 'BRSK2',
                'BTBD16', 'C12orf36', 'C12orf71', 'C13orf16', 'C14orf19',
                'C15orf2', 'C16orf3', 'C18orf56', 'C19orf71', 'C21orf81',
                'C7orf16', 'C7orf61', 'C9orf128', 'CA10', 'CAMK2N2', 'CAPN14',
                'CCDC144B', 'CDH22', 'CLEC17A', 'CNGA3', 'COL25A1', 'CPN2', 'CR1L',
                'CRYBA4', 'CSNK1A1L', 'CSRNP3', 'CYP2A7', 'DDN', 'DGCR10',
                'DMRTA2', 'DPEP3', 'DPY19L2P4', 'EPX', 'F7', 'FAM133A', 'FAM187B',
```

'FAM194A', 'FAM74A1', 'FAM90A1', 'FBX043', 'FOXD4L2', 'FOXI1',
 'GABRD', 'GPC2', 'GPR150', 'GPR21', 'GPR25', 'HAND2', 'HAP1',
 'HBE1', 'HIST1H1E', 'HIST1H4E', 'HIST1H4I', 'HOXC6', 'HPDL', 'HPR',
 'HRG', 'HYAL4', 'IGSF1', 'IL17C', 'IMPG2', 'INSM1', 'IRGM',
 'KCNB2', 'KCTD19', 'KERA', 'KIR3DX1', 'KRT16', 'LOC100128573',
 'LOC100133545', 'LOC121952', 'LOC168474', 'LOC283404', 'LOC283856',
 'LPA', 'LPAL2', 'LY6G6E', 'MAB21L2', 'MGAT4C', 'MGC45800', 'MNX1',
 'MOV10L1', 'NCRNA00169', 'NCRNA00235', 'NPFFR1', 'NPW', 'NTN5',
 'OCLM', 'OR13A1', 'PADI3', 'PCDHB1', 'PCDHGC4', 'PFKFB1', 'PLIN1',
 'POTEE', 'PRO0628', 'PRSSL1', 'RAB3C', 'RNASE3', 'ROPN1',
 'RPL23AP32', 'RPS15AP10', 'RPS4Y1', 'SCN10A', 'SLC22A1', 'SLC24A5',
 'SLC28A2', 'SLC36A3', 'SLC38A3', 'SLC6A15', 'SLC6A17', 'SLC9A11',
 'SLFN14', 'SMPX', 'SPINK2', 'SRRM4', 'SSTR4', 'TDRD5', 'TEX12',
 'TMEM211', 'TUSC5', 'TXNDC3', 'VWA5B1', 'VWA5B2', 'WDR88', 'WNT16',
 'ZNF878', 'ZSCAN5B', 'A2ML1', 'ABCB11', 'ABRA', 'AMH', 'ANKRD34B',
 'ATP2B3', 'AURKAPS1', 'B4GALNT4', 'BCAN', 'BMP15', 'C10orf53',
 'C10orf91', 'C11orf20', 'C1orf111', 'C1QTNF4', 'C3orf35',
 'C7orf54', 'C9', 'CEND1', 'CHRNA2', 'CLCN1', 'CLDN20', 'CPA5',
 'CXorf58', 'DCST2', 'DPRXP4', 'DRD4', 'DRD5', 'DSG3', 'DUSP9',
 'ELAVL4', 'ENTHD1', 'ENTPD8', 'FAM22G', 'FGF17', 'FLG2', 'FSD1',
 'FUNDC2P2', 'GAP43', 'GDPD4', 'GFAP', 'GRM6', 'HCG22', 'HOXA11',
 'HOXB9', 'HSF5', 'INHA', 'KCNQ2', 'KLF14', 'KRTAP5-1', 'LDHAL6B',
 'LOC148709', 'LOC285796', 'LOC442459', 'LY6D', 'LY6G6C', 'MAS1L',
 'MSLNL', 'MYH8', 'NBLA00301', 'NDST3', 'NELL1', 'NPFFR2', 'NPPA',
 'NPY6R', 'ODF1', 'PDYN', 'PHF21B', 'PNMA5', 'PRH2', 'PTPRN',
 'RFPL1', 'RFX4', 'RNF148', 'RP1-177G6.2', 'RXFP2', 'SALL1',
 'SALL4', 'SCARNA12', 'SCXB', 'SERPINC1', 'SLC38A11', 'SLC6A2',
 'SLC7A14', 'SLC9A10', 'SPATA1', 'SPC24', 'STRC', 'TGM5', 'TMEM74',
 'TOMM20L', 'TRIM50', 'TTR', 'TUBB4Q', 'UPK3A', 'ZNF157', 'ZP1',
 'VSTM2A', 'SLC7A5P1', 'C1QTNF9', 'CASKIN1', 'ABHD12B', 'ABHD1',
 'ACTC1', 'AGRP', 'ALS2CR12', 'ANKRD20A4', 'AP3B2', 'AQP10', 'AQP6',
 'ATP1A4', 'BFSP2', 'BLID', 'BREA2', 'C13orf26', 'C15orf54',
 'C20orf197', 'C21orf99', 'C2orf70', 'C6orf227', 'C7orf52',
 'C7orf69', 'C8orf85', 'CACNA1G', 'CEACAM3', 'CEACAM7', 'CEBPE',
 'CLDN6', 'CLEC18B', 'CLPS', 'COLEC10', 'CRIP3', 'CRYGN', 'CYP2C8',
 'CYP8B1', 'DGKI', 'DIRC1', 'DISC2', 'DLX3', 'DUSP26', 'ERAS',
 'ERVFRDE1', 'FAM135B', 'FLJ40504', 'FOXH1', 'GATA1', 'GCK', 'GCKR',
 'GDPD2', 'GGT8P', 'GNMT', 'GRM1', 'HBD', 'HIST2H3C', 'HOXA13',
 'IGSF5', 'IL1F5', 'KCNA2', 'KCNE2', 'KRT13', 'LEMD1', 'LGALS4',
 'LHFPL1', 'LOC100132831', 'LOC145837', 'LOC440461', 'MEGF10',
 'MMP8', 'MUC6', 'MYBPH', 'MYH6', 'MYO1A', 'NCAN', 'NEU4',
 'NEUROG2', 'NKX6-1', 'ODF3L2', 'OTUD7A', 'PLA2G3', 'POU4F2',
 'POU5F1B', 'PRAME', 'PRG2', 'PRND', 'PRSS45', 'PTTG2', 'RAPSN',
 'RERGL', 'RIMS4', 'RPA4', 'RXFP3', 'S100A5', 'SBSN', 'SCARNA2',
 'SH3GL2', 'SHISA7', 'SLC10A5', 'SPATA4', 'SPATA8', 'ST8SIA6',
 'TAF7L', 'TAS1R1', 'TDRD12', 'TMEM196', 'TMEM88B', 'TMPRSS9',
 'TNNT3', 'TP53AIP1', 'TTY14', 'TUBBP5', 'TULP2', 'VIT', 'ZNF389',
 'ZNF492', 'ST7OT3', 'HEPHL1', 'PKD2L2', 'STL', 'C20orf106', 'AAA1',

'AK5', 'ANGPT4', 'ARSH', 'ASCL3', 'AZU1', 'C14orf162', 'C15orf5',
 'C1orf182', 'C1orf227', 'C20orf200', 'C21orf121', 'C21orf62',
 'C22orf43', 'C6orf142', 'C8A', 'CASP12', 'CASR', 'CDH16', 'CDHR5',
 'CDK15', 'CLEC12B', 'CLEC2L', 'CLEC4D', 'CPA4', 'CRB1', 'CTXN2',
 'CYP4A11', 'DNASE1L2', 'EPA6', 'EPA8', 'FABP7', 'FAM106A',
 'FAM106C', 'FAM132A', 'FAM183B', 'FAM19A1', 'FER1L5', 'FLJ37201',
 'FLJ39739', 'FLJ45079', 'FOX4', 'GALR2', 'GBA3', 'GDAP1L1',
 'GDNF', 'GFI1B', 'GPR27', 'GPR52', 'GSTM2P1', 'GTSF1L', 'HAPLN4',
 'HEMGN', 'HERC2P4', 'HTR1B', 'IGSF11', 'IL1F9', 'INSRR', 'IVL',
 'KCNA4', 'KRT4', 'LEFTY1', 'LOC100128542', 'LOC100189589',
 'LOC100216001', 'LOC285954', 'LOC342346', 'LOC389791', 'LOC440905',
 'LRRC3B', 'LYZL4', 'MATN4', 'MC5R', 'MIXL1', 'MT1H', 'MYCNOS',
 'NEFM', 'NKPD1', 'NT5C1B', 'OMP', 'OR10AD1', 'OVCH1', 'PAK7',
 'PCK1', 'PDZK1', 'PLA2G12B', 'PTPRVP', 'PYY2', 'PYY', 'RLN1',
 'RPL29P2', 'SCARNA7', 'SCARNA9', 'SLC10A6', 'SLC18A3', 'SLC22A16',
 'SLC25A18', 'SLC28A1', 'SLC45A2', 'SLC5A12', 'SLC5A7', 'SLC7A3',
 'SLN', 'SNCB', 'SPIC', 'SPRR2A', 'SULT4A1', 'SYT14', 'TCAM1P',
 'TCL1B', 'TDH', 'TERT', 'TMPRSS7', 'TNFSF18', 'UBE2U', 'USH2A',
 'UTF1', 'VCX', 'VSX2', 'WFIKK2', 'WNT1', 'YBX2', 'ZBP2',
 'ZYG11A', 'ABCC12', 'ABCC8', 'ACCN4', 'ACTN2', 'ADAM21P1',
 'ADAMTS18', 'ADARB2', 'ADRB3', 'AGBL1', 'ALS2CR11', 'ANGPTL5',
 'APOBEC2', 'APOH', 'ARGFXP2', 'ART5', 'ASPDH', 'ATP2B2', 'AWAT2',
 'B3GALT1', 'BARX1', 'BTG4', 'BTN1A1', 'C11orf42', 'C12orf42',
 'C12orf53', 'C14orf86', 'C15orf53', 'C15orf55', 'C17orf54',
 'C17orf66', 'C1orf127', 'C1orf189', 'C1orf230', 'C20orf107',
 'C2orf52', 'C2orf82', 'C3orf65', 'C4orf22', 'C5orf47', 'C8orf46',
 'C8orf56', 'C8orf74', 'C9orf53', 'C9orf70', 'CACNA1E', 'CALML6',
 'CASQ1', 'CBLN4', 'CCDC163P', 'CCIN', 'CD1B', 'CD300E', 'CD300LG',
 'CDH12', 'CGB2', 'CHD5', 'CLCNKA', 'CLEC4F', 'CNKSR2', 'CPB1',
 'CPB2', 'CPNE4', 'CPO', 'CRB2', 'CREG2', 'CRYBB1', 'CRYBB3',
 'CSNK1A1P', 'CXorf59', 'CXorf61', 'CXorf65', 'CYP2A6', 'CYP4F22',
 'DAND5', 'DCAF4L1', 'DDX4', 'DEPDC4', 'DGCR9', 'DHH', 'DLGAP1',
 'DMGDH', 'DMRT3', 'DMRTC1B', 'DNAH8', 'DPF1', 'DPPA4', 'DUSP5P',
 'EBF3', 'ELANE', 'ELAVL2', 'EMR4P', 'EMX2', 'FABP5L3', 'FAM131C',
 'FAM163A', 'FAM196B', 'FAM19A3', 'FAM25A', 'FAM43B', 'FAM57B',
 'FAM66E', 'FBN3', 'FETUB', 'FGL1', 'FLJ14107', 'FLJ33360',
 'FLJ34503', 'FLJ37307', 'FMN2', 'FMO6P', 'FOX1', 'FOXN1', 'FTCD',
 'FZD9', 'GALNT13', 'GALNTL6', 'GCM1', 'GDF7', 'GLIS1', 'GPR120',
 'GPR1', 'GPR158', 'GPR78', 'GRIA3', 'GRIK4', 'GRIN2B', 'GRIP2',
 'GRM4', 'GRM7', 'GRPR', 'GSTM1', 'HIGD2B', 'HOXA10', 'HOXB8',
 'HOXD9', 'HPCAL4', 'HRNBP3', 'HS6ST3', 'HTR6', 'HTR7', 'HUS1B',
 'IFNA1', 'IGLL3', 'IL23R', 'IL26', 'ILDR2', 'ISL2', 'ITGAD',
 'ITLN2', 'IZUMO1', 'JPH3', 'KCNA3', 'KCNA1', 'KCNF1', 'KCNP1',
 'KCNJ13', 'KCNJ2', 'KCNJ1', 'KEL', 'KIR3DL1', 'KISS1', 'KLF1',
 'KLF17', 'KLKB1', 'KLRC2', 'KRT12', 'KRT81', 'KRTAP5-7',
 'KRTAP5-8', 'LASS1', 'LGALS12', 'LGALS7', 'LHX5', 'LOC100134368',
 'LOC100192378', 'LOC100302640', 'LOC121838', 'LOC127841',
 'LOC144486', 'LOC283867', 'LOC285735', 'LOC285740', 'LOC285768',

```
'LOC400794', 'LOC401387', 'LOC401463', 'LOC441455', 'LOC90586',
'LOC91450', 'LPO', 'LRRC26', 'LRRC67', 'LRRIQ4', 'LRRTM4', 'LRTM2',
'MBOAT4', 'MDS2', 'MED12L', 'MGC16025', 'MOGAT2', 'MOGAT3', 'MUSK',
'MYH7B', 'MYOT', 'NAP1L6', 'NAT8B', 'NBP7F', 'NCRNA00167',
'NCRNA00173', 'NECAB2', 'NHLH2', 'NIPAL4', 'NKX3-2', 'NLRP9',
'NOX5', 'NRG3', 'NUP210L', 'OCA2', 'OOEP', 'OR2A1', 'OR4C6',
'OR52N4', 'OR7D2', 'OXGR1', 'PART1', 'PCDHB17', 'PCDHGA3', 'PCP2',
'PDC', 'PDE6A', 'PENK', 'PEX5L', 'PHEX', 'PI16', 'PIWIL1', 'PLAC2',
'PM20D1', 'POM121L8P', 'POPDC3', 'PPEF1', 'PPP1R1A', 'PRAP1',
'PRDM12', 'PRKY', 'PRR18', 'PTCHD3', 'PTGER1', 'PTH2R', 'PURG',
'RBM44', 'REG3A', 'REN', 'RFPL2', 'RHBDL3', 'RHBG', 'RIMBP2',
'RIPPLY2', 'RPH3A', 'RPL23AP64', 'RS1', 'SEMG1', 'SERPINA6',
'SERPINB13', 'SGCG', 'SLC26A5', 'SLC30A2', 'SLC35F4', 'SLC5A11',
'SLC5A5', 'SLC7A10', 'SLC7A9', 'SMAD5OS', 'SMR3B', 'SOST', 'SOX30',
'SP7', 'SPATA9', 'SPATS1', 'SPDYE2', 'SPINK7', 'SPRR2C', 'SSTR3',
'ST70T4', 'ST8SIA5', 'STAR', 'SUGT1P1', 'SUMO4', 'SYCP2L',
'SYNPO2L', 'SYT6', 'TAC1', 'TAS2R4', 'TCAP', 'TCP10L', 'TCP11',
'TDGF1', 'TDRD9', 'TEX15', 'THEM5', 'TIGD4', 'TKTL2', 'TMC1',
'TMEM155', 'TMEM35', 'TPH1', 'TRIM40', 'TRIM67', 'TRIM71', 'TRPC5',
'TTBK1', 'TTC9B', 'TTLL2', 'TUBA3E', 'TULP1', 'UPB1', 'UROCI',
'UTS2R', 'WBP2NL', 'WISP3', 'WNK3', 'WNK4', 'WNT10B', 'WNT3A',
'XKR4', 'XKR5', 'ZFR2', 'ZNF114', 'ZNF541', 'ZNF660',
'STON1-GTF2A1L'], dtype=object)
```

1.0.12 select a few genes

say: ARID1A,ARID2,ARNT,ASPSR1,GNA11,GNAQ,PML,PMX1,PNUTL1,STAG2,STAT5B,ZNF384,ZNF521

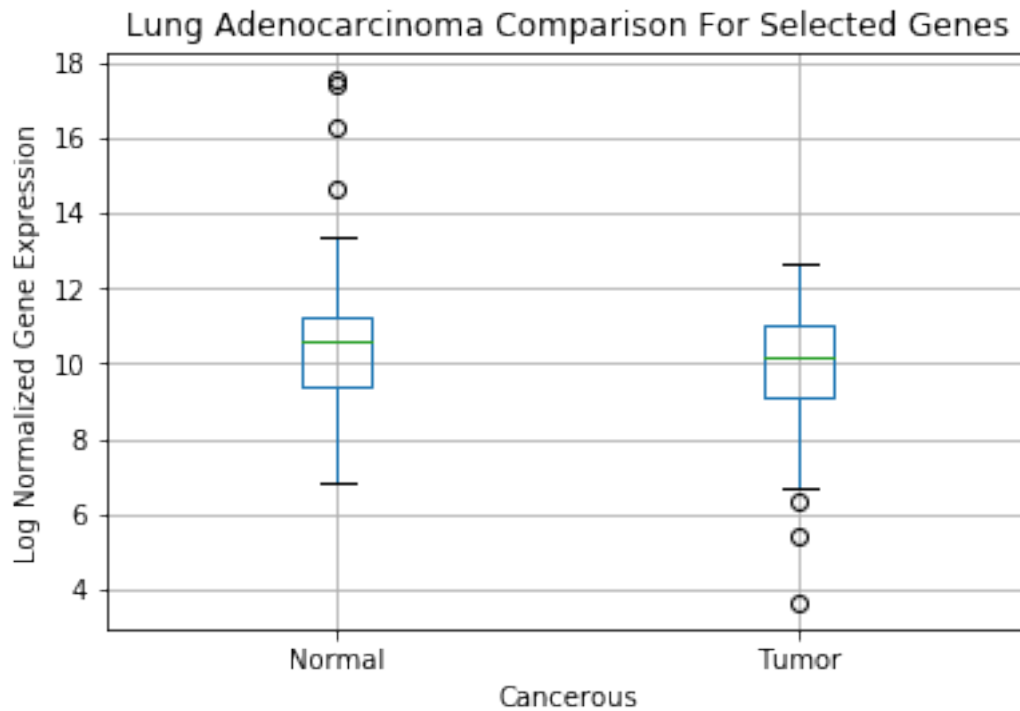
```
In [39]: selected = ["ARID1A", "ARID2", "ARNT", "ASPSR1", "GNA11", "GNAQ", "PML", "PMX1", "PNUTL1", "STAG2", "STAT5B", "ZNF384", "ZNF521"]
df_selected = df_stack[df_stack.get('Gene').isin(selected)]
```

1.0.13 making a boxplot

1.0.14 with matplotlib

```
In [40]: import matplotlib.pyplot as plt
```

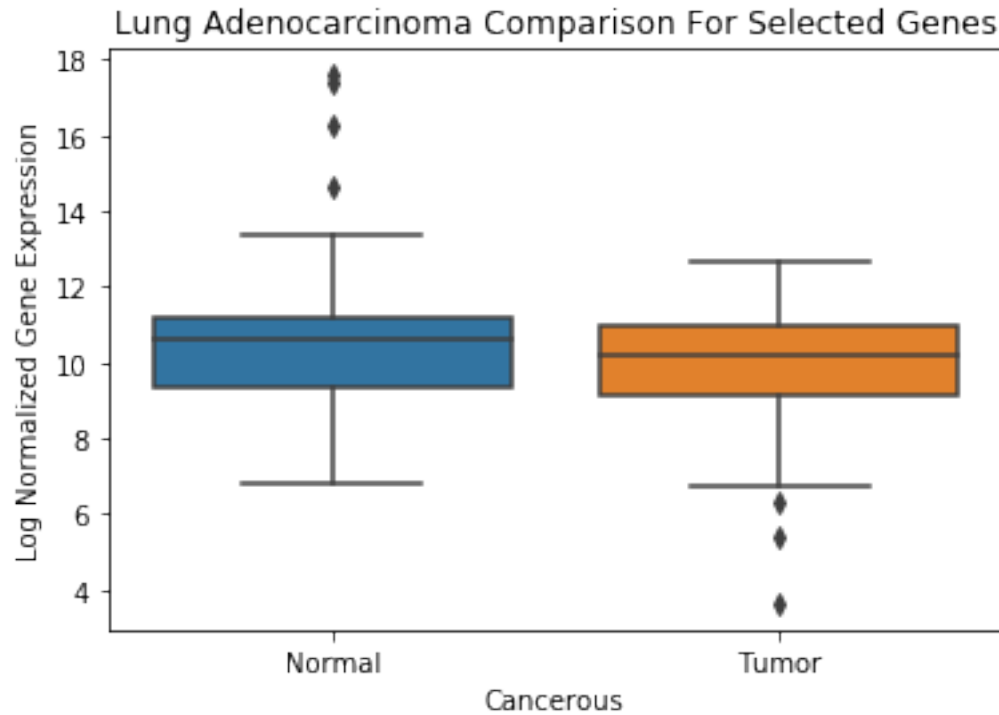
```
df_selected.boxplot(column='Expression_Norm', by='Cancerous')
plt.suptitle("")
plt.title("Lung Adenocarcinoma Comparison For Selected Genes")
plt.xlabel("Cancerous")
plt.ylabel("Log Normalized Gene Expression")
plt.show()
```

1.0.15 with seaborn

In [41]: `import seaborn as sns`

```
axes = sns.boxplot(x = 'Cancerous', y='Expression_Norm', data=df_selected, )
#axes = sns.swarmplot(x = 'Cancerous', y='Expression_Norm', data=df_selected,color="g")
plt.title("Lung Adenocarcinoma Comparison For Selected Genes")
plt.xlabel("Cancerous")
plt.ylabel("Log Normalized Gene Expression")
plt.show()
```



1.0.16 with plotly

```
In [42]: import plotly.offline as py
         from plotly import graph_objs as go
         py.init_notebook_mode(connected=True)

         py.iplot({
             "data": [go.graph_objs.Box(x=df_selected['Cancerous'],y=df_selected['Expression_Norm']),
             "layout" : go.Layout(dict(title="Lung Adenocarcinoma Comparison For Selected Genes",
                                     xaxis=dict(title="Cancerous"),
                                     yaxis=dict(title="Log Normalized Gene Expression")))]
         })
```