Abstract: Gene expression analysis was conducted on a dataset containing gene expression values for both clear cell renal carcinoma tumor cell samples and normal cell samples. The data was taken from The Cancer Genome Atlas (TCGA). The analysis involved data wrangling, filtering, exploratory data analysis, differential expression and correlation analysis, and gene coexpression analysis via unsupervised machine learning (hierarchical clustering). The analysis was able to both demonstrate techniques for conducting basic gene expression analysis and discover evidence pertaining to lifestyle factors which can increase the risk of kidney cancer. A possible link between obesity and clear cell renal carcinoma was found. ANGPTL4 is over expressed in kidney cancer tumor cells relative to expression levels found in normal cells (with expression levels in tumor cells nearly double the levels found in normal cells). ANGPTL4 has been implicated in glucose intolerance and obesity in previous studies. Another goal of this project was to show how open-ended data analysis of gene expression datasets can be used to guide molecular biology research and hypothesis development. Here we show how data analysis techniques can be used to provide valuable insights into gene expression profiles, which can spearhead experimental studies and strengthen grant applications for research funding.

Background and Objectives: The Cancer Genome Atlas (TCGA) is a collaboration between the National Institute of Cancer (NIC) and the National Human Genome Research Institute (NHGRI).[i] The TCGA dataset comprises more than two petabytes of genomic data, with expression profiles and multi-dimensional maps of key genomic changes in 33 different types of cancer. See Id. This data is publicly available to assist the cancer research community in improving the prevention, diagnosis, and treatment of cancer. See Id. The dataset used in this analysis contains gene expression values taken from normal (healthy) cells and tumor cells (from patients with clear cell renal carcinoma).

Clear cell renal carcinoma is a form of kidney cancer which primarily afflicts men in their sixties and seventies.[ii] It is called clear cell because the cells exhibit a clear cytoplasm surrounded by a distinct cell membrane containing round and uniform nuclei. See Id. In general, renal cell carcinomas are a kidney cancer that originates in the lining of the proximal convoluted tubule, a part of the network of small tubes in the kidney responsible for transporting urine. See Id. Initial treatment involves partial or complete surgical removal of the affected kidney, with a five-year post-surgery survival rate of between 65 and 90 percent. See Id. The greatest risk factors for renal cell carcinomas are lifestyle related e.g. obesity, smoking, and hypertension. See Id.

Problem statement and target audience: To aid in the development of treatment strategies for cancer, it is useful to understand the differences in gene expression between cancer cells and normal (healthy) cells. By better understanding the genes whose expression is either enhanced or suppressed in cancer cells, we both enrich our understanding of the biological pathways effected by deleterious mutations and identify targets for drugs and other types of treatments.

Data analysis can also be used to accelerate molecular biology research. For example, in a recent paper by L. La Paglia et al., titled “Potential Role of ANGPTL4 in the Cross Talk between Metabolism and Cancer through PPAR Signaling Pathway” we find a buildup of research which began at least as early as 1990.[iii] In other words, this study was the culmination of nearly 30 years of research by many different research institutions throughout the world. In this analysis, we identified ANGPTL4 as a gene of interest in the context of clear cell renal carcinoma within the span of days. While wet lab research is always required to characterize genes, understand their function, pathway, transcriptional/post-transcriptional

regulation, common mutations, and their role in human disease, bioinformatics is increasingly driving molecular biology research, identifying new questions relevant to enhancing our understanding of human disease and our genome, and even hypothesis development.

In this vein, it is important that novice bioinformaticians have ready access to resources to help them gain an intuition for genomic analysis, understanding the common statistical tests used in analysis, alternative methods which may be available, the strengths and weaknesses of those methods, and how to apply those techniques programmatically.

DATA PREPARATION

Importing dataset: Because of the large size of the dataset, it cannot be uploaded onto Github. However, the dataset can be accessed at:

https://tcga.xenahubs.net/download/TCGA.KIRC.sampleMap/HiSeqV2.gz

The data was loaded into a Jupyter Notebook as a csv file using the Pandas read_csv function. Here is an image of the data head:

Out[22]:

sample	TCGA-BP-4162-01	TCGA-CJ-5677-11	TCGA-DV-5566-01	TCGA-BP-5191-01	TCGA-BP-5200-01	TCGA-BP-4347-01	TCGA-BP-4770-01	TCGA-B0-5696-11	TCGA-BP-4762-01	TCGA-BP-4158-01	...	TCGA-B0-5104-01	TCGA-A3-3313-01	TCGA-B2-5633-01	TCGA-CJ-4872-01	TCGA-CJ-5684-01	TCGA CJ 4886 0
ARHGEF10L	10.5030	10.8969	10.7612	10.2063	10.0616	10.0193	8.4364	11.1427	9.9150	10.7621	...	10.5586	9.5556	10.2652	9.4981	10.2751	9.952
HIF3A	5.5283	6.4943	5.7842	5.0063	5.3326	7.6102	6.0422	5.9789	3.7802	7.2039	...	6.2061	4.8253	5.1680	8.6195	6.2080	6.311
RNF17	3.8036	0.0000	0.0000	0.0000	0.0000	0.3386	0.0000	0.0000	0.0000	0.0000	...	0.0000	0.0000	0.0000	0.0000	0.0000	0.000
RNF10	11.4379	12.2130	11.5478	12.3439	11.5149	11.4065	12.1963	12.3946	11.3734	11.3251	...	11.7338	11.7709	11.4968	11.9561	11.7518	11.569
RNF11	11.4180	11.8248	11.3190	10.3413	11.2923	10.9971	11.8936	11.7217	12.2329	11.2993	...	11.7629	10.2373	11.4637	11.4391	11.3992	11.493

5 rows × 606 columns

The index was set to 'sample' (the first column – containing all the gene names).

Data wrangling:

Good practices for preparation of gene expression datasets include setting the column with gene names as the index, either separating the data by condition or delineating between different conditions in your analysis and dealing with missing values.

- First, all rows containing zero values were dropped to deal with missing values.
  - First, all zero values were converted to NaN.
  - Then, all rows containing NaN were dropped.
- Next, the data was split into two Pandas dataframes using regular expressions (one with cancer cell gene expression values, the other with normal cell gene expression values).
  - The NIH codes column headers to identify the research institution, patient, and condition.
  - The last two numbers of the column headers denote the condition. Numbers ending with 11 denote samples taken from healthy subjects, while numbers ending in 01 denote samples taken from subjects with cancer.
- The following code was used for both operations:

Drop rows with zero values

```
In [23]: df_replace = data.replace(0.0000, np.nan)
         df_dropped = df_replace.dropna(axis=0, how='any')
         df_dropped.shape
Out[23]: (12580, 606)
```

Split data into two dataframes:

- The first dataframe (df1) contains expression values for normal cells
- The second dataframe (df2) contains expression values for cancer cells

```
In [24]: df1 = df_dropped.filter(regex = 'sample|11$', axis =1)
         # Creating new dataframe with only columns containing expression values for normal cells
```

Because the original dataset was normalized and log2 transformed, no additional preprocessing was required. It's important to note, alternative techniques for filtering gene expression data may involve dropping rows where the z-score (of mean expression values) is below a preset threshold or using the IQR method (and dropping rows where mean expression values fall outside the interquartile range). While those methods were not used here, a copy of the Capstone Project Jupyter Notebook was created to demonstrate the code we would use to employ these alternative techniques:

```
In [45]: from scipy import stats
         df_dropped = df_dropped[(np.abs(stats.zscore(df_dropped)) < 3).all(axis=1)]
         df_dropped.shape
```

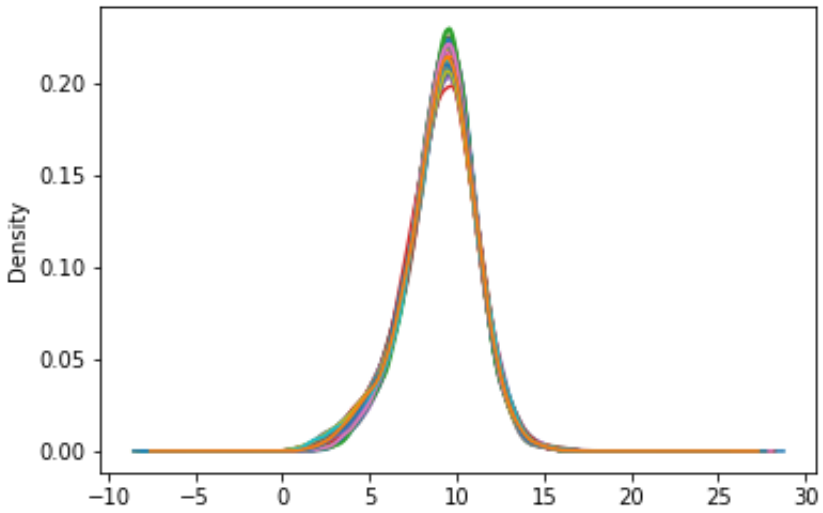
This code drops all rows with a z-score less than 3.

```
In [14]: Q1 = df_dropped.quantile(0.25)
         Q3 = df_dropped.quantile(0.75)
         IQR = Q3 - Q1
         df = df_dropped[~((df_dropped < (Q1 - 1.5 * IQR)) |(df_dropped > (Q3 + 1.5 * IQR))).any(axis=1)]
         df_dropped.shape
```

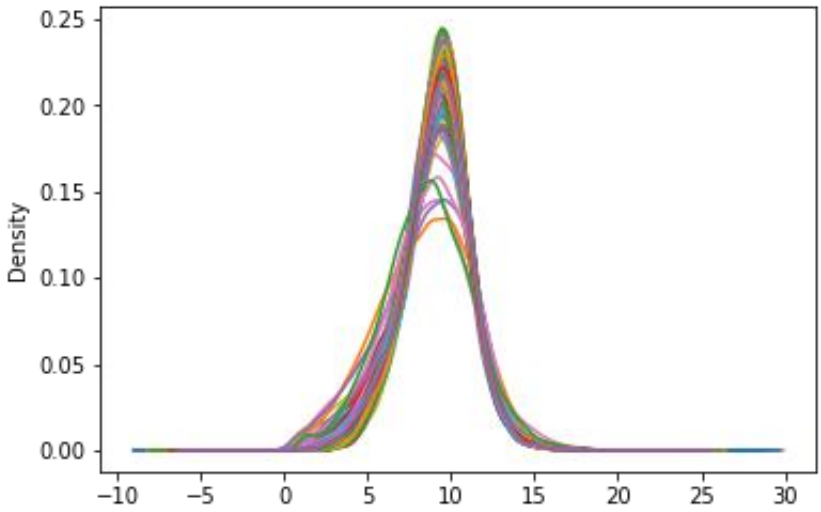
This code shows an implementation of the IQR method.

Exploratory data analysis:

First, density plots were generated to confirm the shape of the distribution:



Density plot for dataframe containing normal cell gene expression values.



Density plot for dataframe containing cancer cell gene expression values.

As shown, the data in both dataframes is normally distributed. To confirm that biological differences exist between the two dataframes, scatter plots were generated. The following code was used to create new columns for each dataframe, containing the mean and variance for both dataframes:

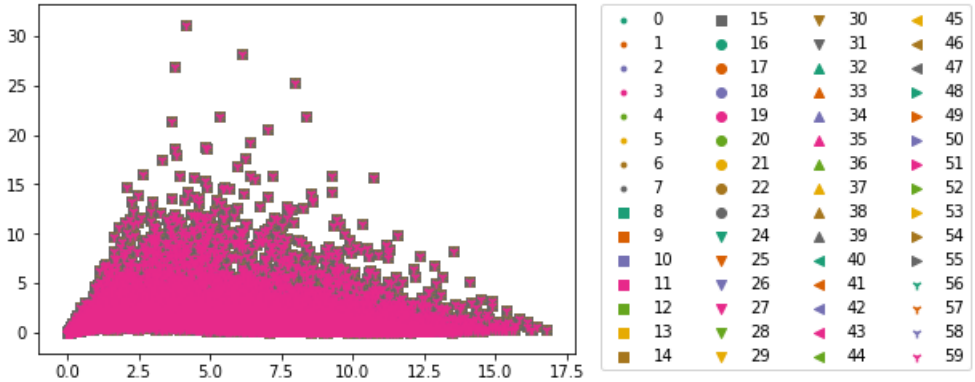
Add columns with row-wise mean & variance to gene exp dataframes

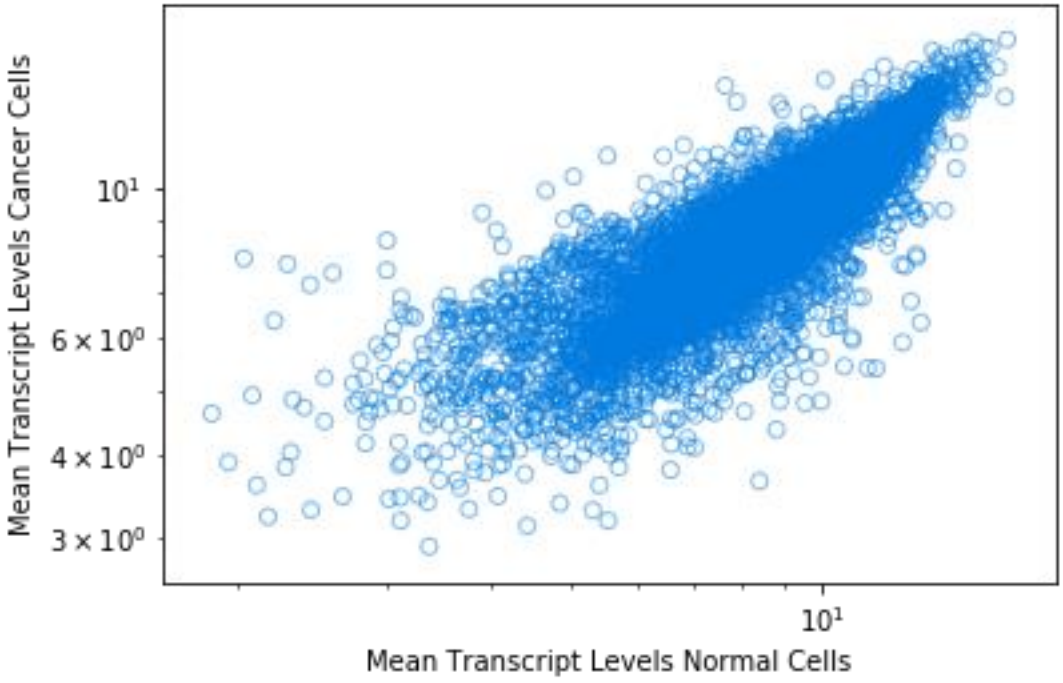
In [29]:
```
import warnings
warnings.simplefilter("ignore")
from scipy.stats import ttest_ind, ttest_ind_from_stats


df1['mean1'] = df1[df1.columns].mean(axis=1)
df1['variance1'] = df1[df1.columns].var(axis=1)
df2['mean2'] = df2[df2.columns].mean(axis=1)
df2['variance2'] = df2[df2.columns].var(axis=1)
```

Scatter plots:

The first scatter plot plots the mean against the variance of the entire dataset. The second scatter plot uses the data separated by category and plots the mean expression levels for normal cells against mean expression levels for cancer cells. Although the data has been normalized, we still look for points with enough deviation between categories or between mean and variance.

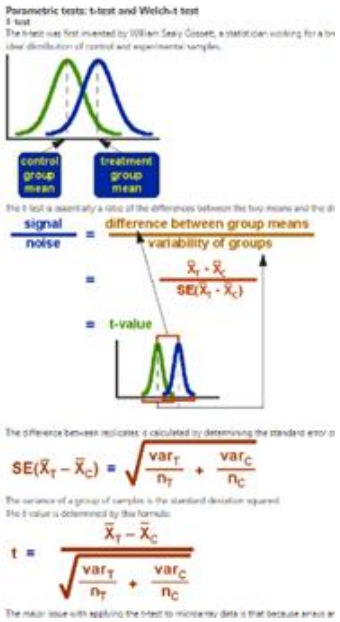




We see genes with significant differences in mean expression levels and between mean expression levels and variance, suggesting that true biological differences exist between the cancer and normal gene expression dataframes.

Differential expression analysis: for this analysis, the t-statistic method was used. However, other methods are available, such as the Empirical Bayesian method.

T-statistic method:



The above image shows the standard t-test as applied to differential expression analysis.[iv]

The t-test is a simple statistical method for detecting differentially expressed genes. It is particularly useful when comparing two conditions (e.g. cancer cells vs normal cells). The error variance (the square of which appears in the denominator of the t-test formula) is hard to estimate when the number of samples is small; however, the dataset used in this analysis contains many replicates, and the data has been normalized. Therefore, application of the t-test method is appropriate in this case.

Another popular method is the empirical Bayes model for differential expression analysis. The advantage of this test is it can work with smaller sample sizes, non-parametric models are available, and it allows for false discovery rate control.

2.1 A hierarchical model for measured intensities

In a typical microarray experiment, two conditions are compared for gene expression. Let us denote by X_{gr} and Y_{gr} the intensities of gene g from the rth replicate in the two conditions, respectively. Measurements between the two conditions are assumed to be independent. The proposed model is an extension of the EBarrays framework (Newton *et al.*, 2001; Kendziorski *et al.*, 2003). Extensions to the original two types of model formulation are considered in turn below.

GG. Here, a Gamma distribution is used to model the measured intensities of a given gene. Explicitly, the probability density of X_{gr} (resp. Y_{gr}) with shape and rate parameters a_g and θ_{gx} (resp. θ_{gy}) is given by

$$p(x \mid a_g, \theta_{gx}) = \frac{1}{\Gamma(a_g)} \theta_{gx}^{a_g} x^{a_g - 1} \exp(-x\theta_{gx}) \quad \text{for } x > 0. \tag{1}$$

To borrow strength across genes we assume an exchangeable Gamma(a_0, ν) prior for the rate parameters, and a Lognormal(η, ξ)prior for the shape parameters. The Gamma prior is used for simplicity as it is conjugate to the sampling distribution (Newton *et al.*, 2001) while the Lognormal prior is suggested by a histogram plot of the empirical shape parameters estimated by the method of moments (see Supplementary material). The hyperparameters a_0, ν, η and ξ are assumed unknown and will be estimated as part of our approach.

The above image outlines the empirical Bayes method for differential expression analysis.[v]

Differential expression analysis:

For demonstration purposes, we confine our analysis to the top 25 differentially expressed genes.

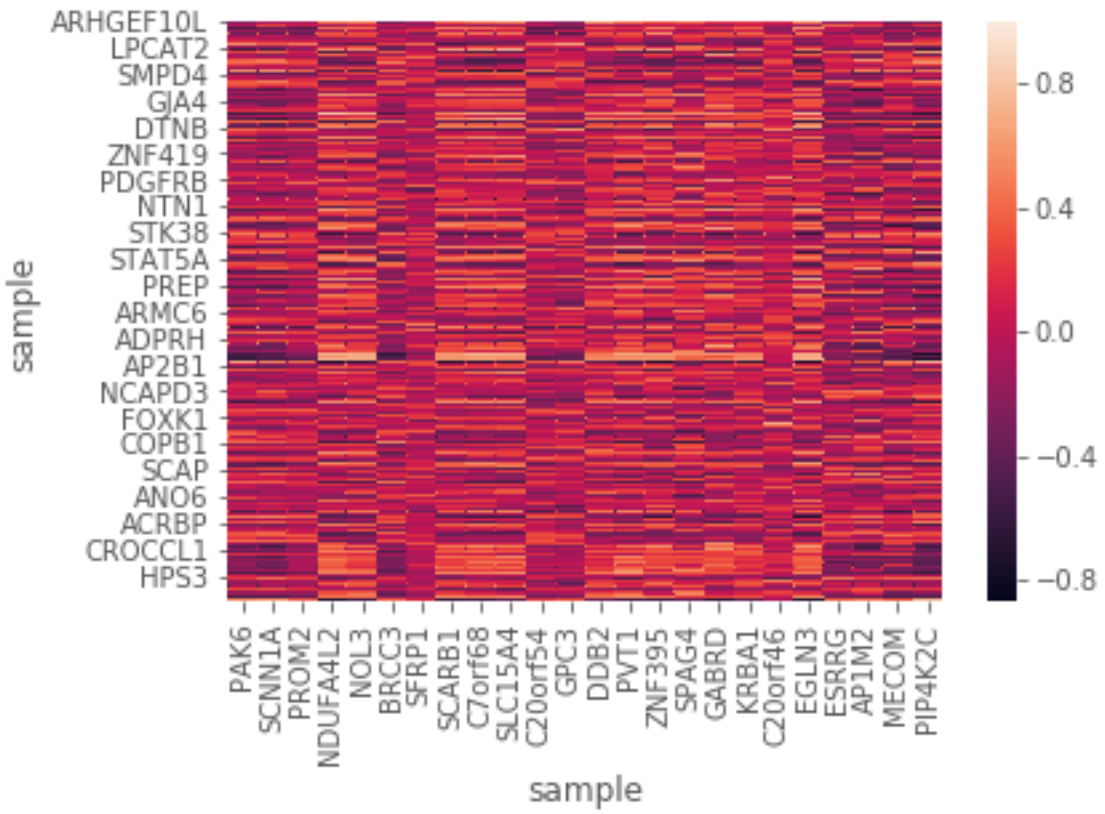
The top 25 differentially expressed genes

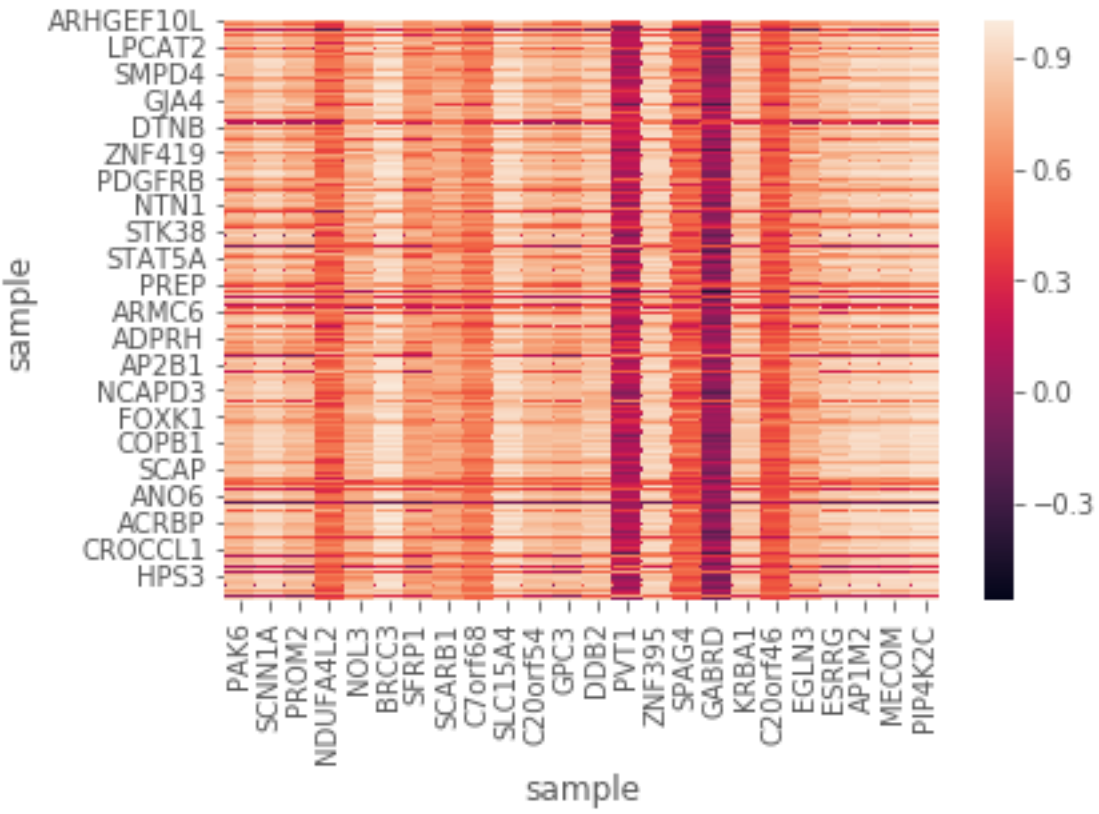
```
results = diffexp.iloc[1:25]
results
```

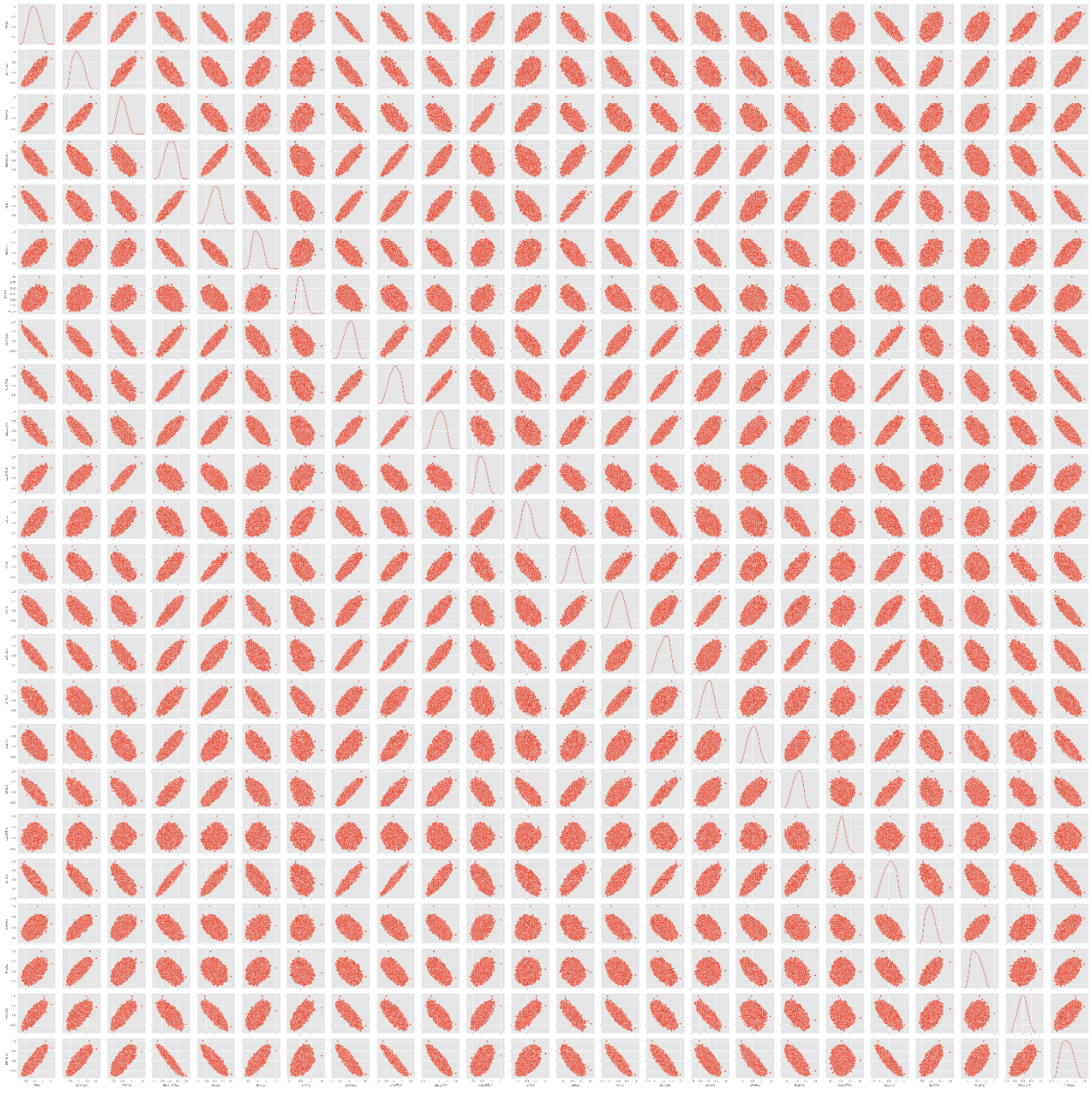
```
sample
IFI27L1      0.003882
SUPT4H1      0.003976
ZGPAT        0.004133
METTL9       0.004940
NCSTN        0.009173
MTMR1        0.010325
SLC9A8       0.012103
MMP24        0.012659
KIAA0141     0.013397
C2orf81      0.014385
RNFT2        0.014591
ZNF681       0.014675
ZNF541       0.018087
H19          0.020194
METT10D      0.020647
LOC254559    0.021607
DIP2C        0.022508
PSMD9        0.023762
P2RX5        0.028422
SCHIP1       0.030097
FANK1        0.030894
SFRS1        0.032160
ICT1         0.032761
SPOPL        0.033188
Name: ttest, dtype: float64
```

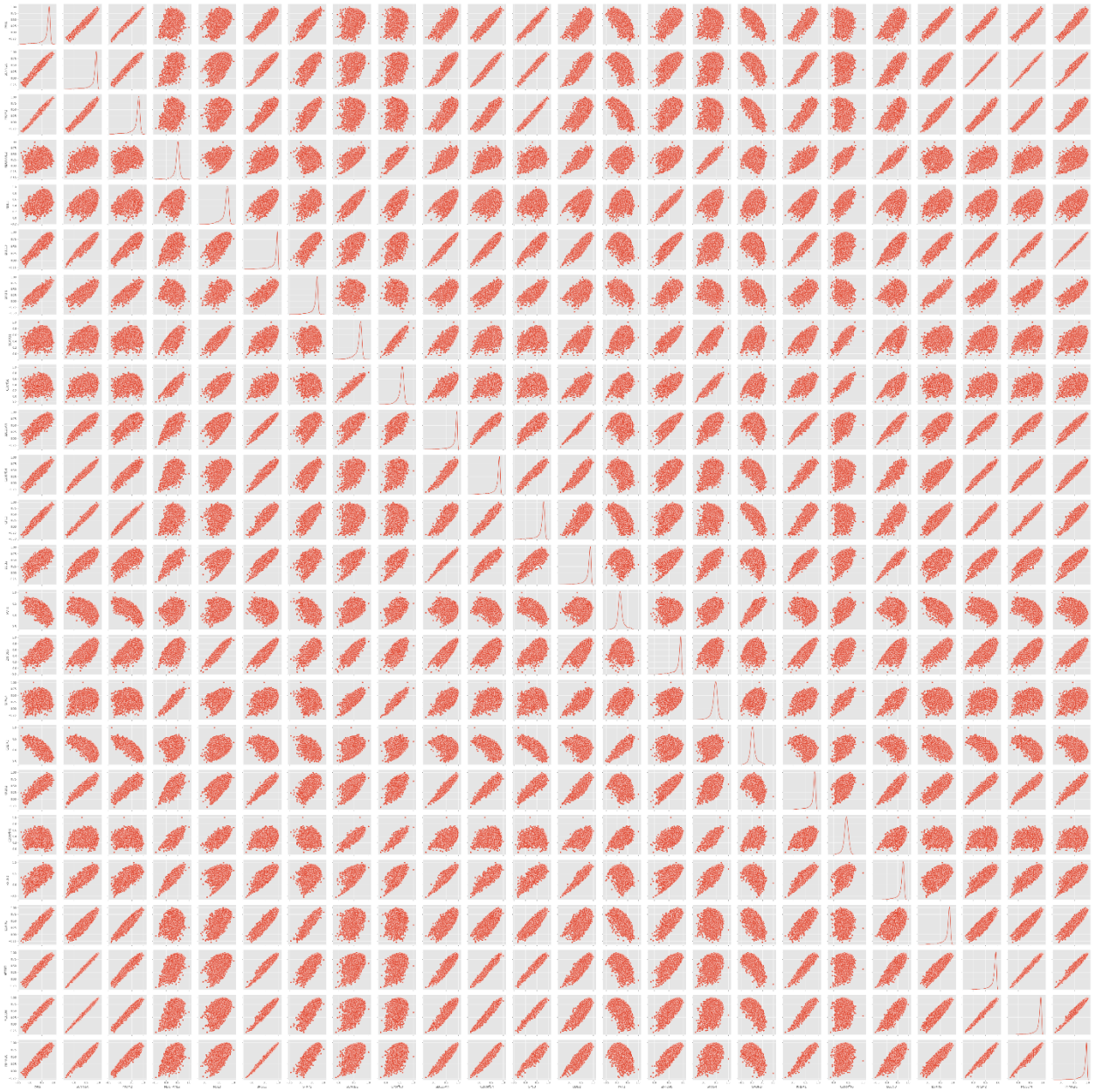
One common challenge in clustering that arises in many analytical contexts is truncating the data enough to generate comprehensible plots. In this case, we're not interested in the expression of all genes contained in the dataset (after all, we began with over 20,000 genes). We're interested in understanding the differences between cancer cells and healthy cells. Therefore, we can limit our analysis to differentially expressed genes, and truncate even further by only focusing on the top differentially expressed genes. Then, we move onto finding correlations (either positive or negative) between our top differentially expressed genes and the rest of the dataset. We might assume that genes with a markedly different expression profile (when comparing cancer to normal cells) is suggestive of gene which are behaving abnormally, possibly due to DNA mutations. However, we'd also like to know which other genes our abnormal genes are influencing. We do this through a correlation analysis. The main benefit of this approach in this case is it allows us to focus machine learning on only genes of biological interest, rather than trying to cluster the entire dataset (which would likely generate an incomprehensible dendrogram, with overlapping and unreadable gene names).

Correlation analysis:




ARHGEF10L
LPCAT2
SMPD4
GJA4
DTNB
ZNF419
PDGFRB
NTN1
STK38
STAT5A
PREP
ARMC6
ADPRH
AP2B1
NCAPD3
FOXK1
COPB1
SCAP
ANO6
ACRBP
CROCCL1
HPS3
sample
PAK6
SCNN1A
PROM2
NDUFA4L2
NOL3
BRCC3
SFRP1
SCARB1
C7orf68
SLC15A4
C20orf54
GPC3
DDB2
PVT1
ZNF395
SPAG4
GABRD
KRBA1
C20orf46
EGLN3
ESRRG
AP1M2
MECOM
PIP4K2C
sample
0.9
0.6
0.3
0.0
−0.3





We notice many negatively correlated genes in the cancer expression dataframe. Conversely, we see many positively correlated genes in the normal cell expression dataframe. This is not surprising, since oncogenes produce deleterious proteins which often inhibit important cellular functions such as cell cycle control and apoptosis. Pairs plots were also generated to further demonstrate the trends which appear in the heatmaps.

Next, the top correlated genes were processed for hierarchical clustering:

Create dataframes with correlated values for differentially expressed genes

In [66]:
```
diff_exp1 = correlations1[['PAK6', 'SCNN1A', 'PROM2', 'NDUFA4L2', 'NOL3', 'BRCC3', 'SFRP1', 'SCARB1', 'C7orf68', 'SLC15A4', 'C20or
```

In [67]:
```
diff_exp2 = correlations2[['PAK6', 'SCNN1A', 'PROM2', 'NDUFA4L2', 'NOL3', 'BRCC3', 'SFRP1', 'SCARB1', 'C7orf68', 'SLC15A4', 'C20or
```

Create object with only highly correlated genes

In [123]:
```
top_corr1 = diff_exp1[(diff_exp1 > abs(0.90)).any(1)]
top_corr1 = diff_exp1[(diff_exp1 < abs(1.0)).any(1)]
```

In [69]:
```
top_corr1.shape
```

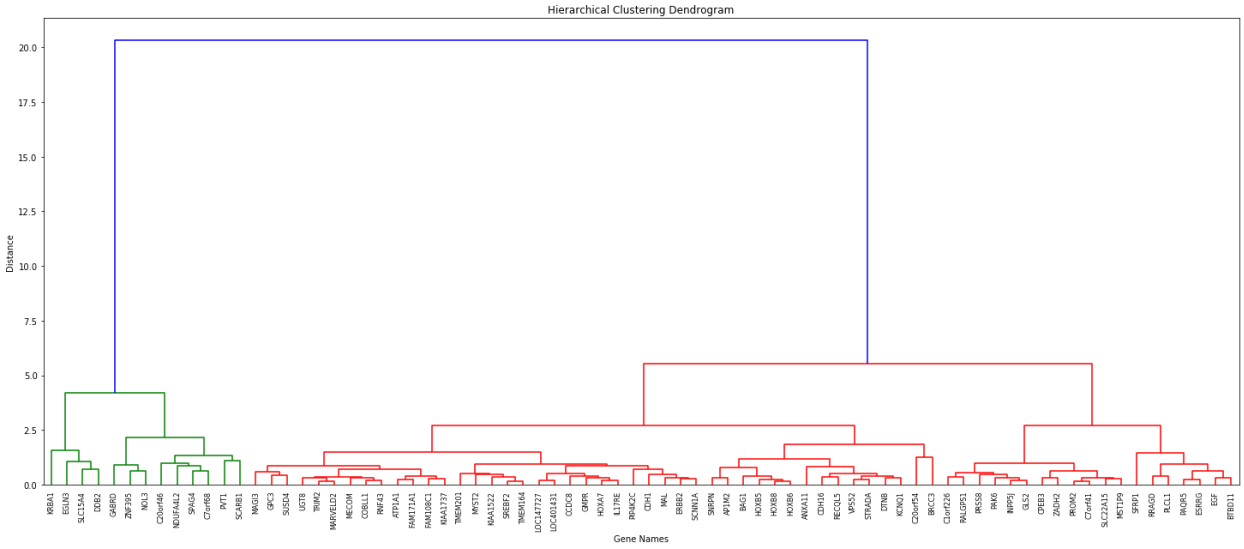
Out[69]: (8973, 24)

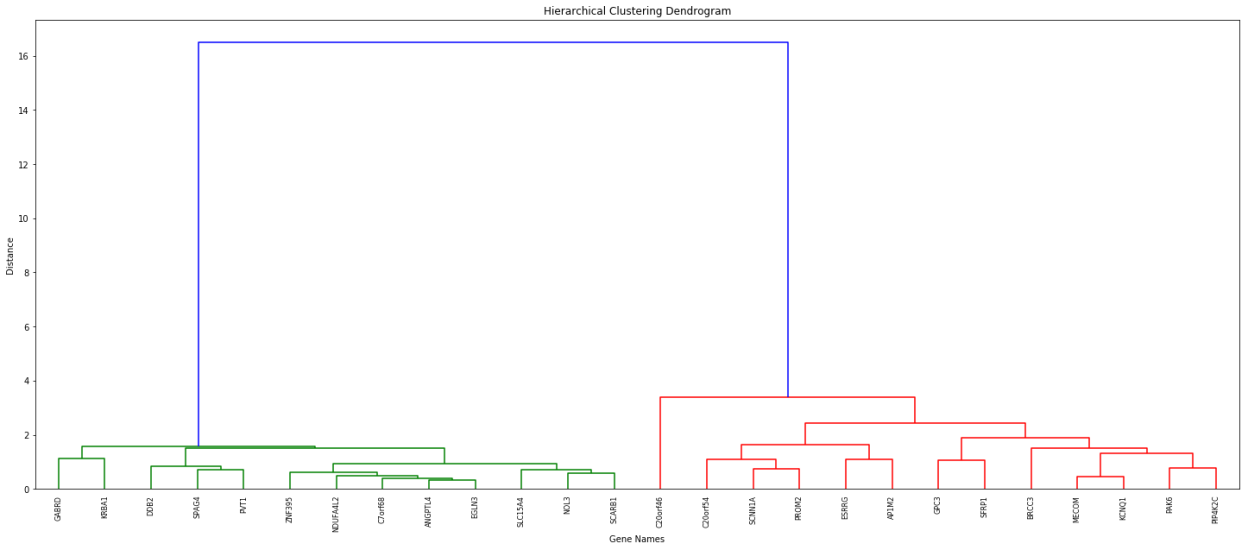
In [124]:
```
top_corr2 = diff_exp2[(diff_exp2 > abs(0.90)).any(1)]
top_corr2 = diff_exp2[(diff_exp2 < abs(1.0)).any(1)]
```

In [126]:
```
top_corr2.shape
```

Out[126]: (12580, 24)

Unsupervised learning:

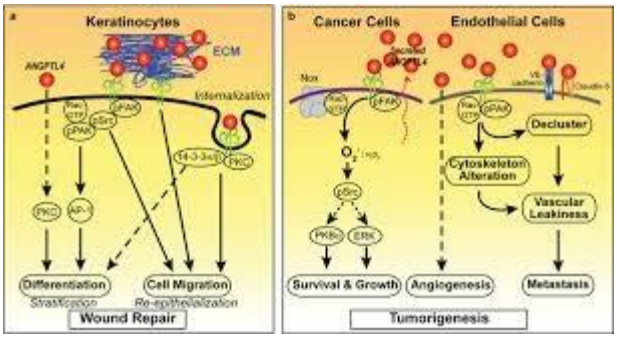




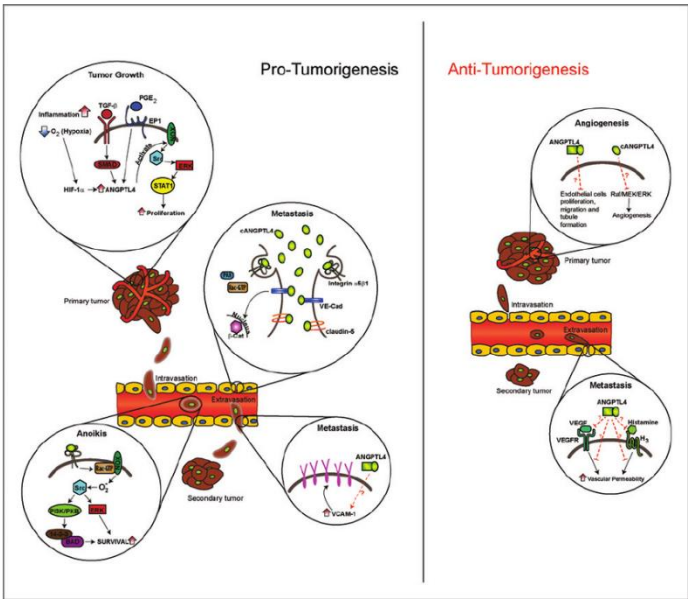
Let's analyze the green clusters from both dendrograms:

- The only gene that does not appear in our list of top differentially expressed genes is ANGPTL4 (in the cancer set).
- Moreover, ANGPTL4 does not appear in our correlation heatmaps.
- The fact that it clustered with differentially expressed genes and genes highly correlated with differentially expressed genes suggests that ANGPTL4 may be active in the same cell signal pathways that some of our differentially expressed/correlated genes function in.
- At this point, we have a robust cohort of interesting candidate genes we can select for further analysis. We (somewhat arbitrarily) select ANGPTL4 for further analysis.

ANGPTL4, or angiopoietin like 4, encodes a glycosylated, secreted protein that regulates glucose homeostasis, lipid metabolism, and insulin sensitivity.[vi] It has been implicated in both promoting and inhibiting tumorigenesis.[vii]



The above image demonstrates the role ANGPTL4 can play in tumorigenesis.[viii]



This image shows how ANGPTL4 can both promote and inhibit tumorigenesis. [ix]

Conclusion:

NULL Hypothisis: ANGPTL4 is not overexpressed in clear cell renal carcinoma tumor cells relative to expression levels in normal cells

Alternate Hypothesis: ANGPTL4 is overexpressed in clear cell renal carcinoma tumor cells relative to expression levels in normal cells

```
In [14]: df1.loc['ANGPTL4'].mean(axis=0)
Out[14]: 7.894029166666665

In [15]: df2.loc['ANGPTL4'].mean(axis=0)
Out[15]: 13.540759626168226

In [16]: from scipy import stats
         stats.ttest_ind(df1.loc['ANGPTL4'], df2.loc['ANGPTL4'])
Out[16]: Ttest_indResult(statistic=-23.8749431619333, pvalue=2.8860803388857175e-89)
```

- Our p-value is < 0.05, providing strong evidence against the NULL hypothesis
- Therefore, we can reject the NULL hypothesis
- This shows that ANGPTL4 is overexpressed in clear cell renal carcinoma tumor cells compared to expression levels in normal cells

Our analysis allowed us to develop a hypothesis concerning expression of ANGPTL4. It may seem obvious that ANGPTL4 is overexpressed in kidney cancer cells (given that its expression in cancer cells is nearly double the expression levels found in healthy cells), nonetheless, we formulate a hypothesis and perform hypothesis testing. As shown, we have an extremely small p-value, which provides excellent evidence that ANGPTL4 is overexpressed in kidney cancer cells.

Here we show basic pythonic techniques for conducting gene expression analysis, and we demonstrate how open-ended data analysis can identify interesting genes, which can then be selected for more intensive analysis. We also show alternative techniques which can be used in gene expression analysis. It's important to note, if we're working with raw data (as opposed to already normalized and transformed data), normalization and transformation will be required. Python has many methods to normalize datasets. One method commonly used with Pandas dataframes is scaling e.g. the

MinMaxScaler or RobustScaler methods. Z-scores can also be used for normalization. We can also employ these techniques with minimal use of packages:

```
In [34]: # min-max scaling
         df_scaled = (df-df.min())/(df.max()-df.min())
```

```
In [35]: # Z-score normalization
         df_scaled2 = (df - df.mean())/df.std()
```

Transformation of data is also very straightforward in Python:

```
In [36]: df = np.log2(df)
```

This code demonstrates application of a log2 transformation for the entire gene dataset.

Recommendations:

- All genes which appear in the cluster dendrograms should be further investigated.
- For any gene in our cohort that has not been characterized previously, wet lab studies should be done to better understand their function and role in human disease.
- Focusing on ANGPTL4, epidemiological studies should be conducted to further elucidate on the link between obesity and kidney cancer.
    - Given the function of ANGPTL4, it may make sense to better understand the role sugar/carbohydrate consumption plays in promoting tumorigenesis.
    - Is it exclusively obesity, or does high sugar/carbohydrate consumption play an important role in cancer and tumorigenesis?

---

[i] The Cancer Genome Atlas: About TCGA, retrieved July 31, 2018, available at, https://cancergenome.nih.gov/abouttcga.
[ii] Renal Cell Carcinoma, Wikipedia, last edited June 14, 2018, available at, https://en.wikipedia.org/wiki/Renal_cell_carcinoma#Risk_factors.
[iii] PPAR Research, Volume 2017, Article ID 8187235, available at, https://www.hindawi.com/journals/ppar/2017/8187235/#B1.
[iv] L. Mulvey, Differential Gene Expression and Hypothesis Testing, retrieved July 31, 2018, available at, http://compbio.pbworks.com/w/page/16252887/Differential%20Gene%20Expression%20and%20Hypothesis%20Testing.
[v] K. Lo & R. Gottardo, Flexible empirical Bayes models for differential gene expression, Bioinformatics, Vol. 23, Issue 3, pp. 328-335 (Feb. 1, 2007).
[vi] ANGPTL4 angiopoietin like 4, NCBI Gene, updated July 29, 2018, available at, https://www.ncbi.nlm.nih.gov/gene/51129.
[vii] Jie Tan, Ming & Teo, Ziqiang & Sng, Ming Keat & Zhu, Pengcheng & Tan, Nguan Soon. (2012). Emerging Roles of Angiopoietin-like 4 in Human Cancer. Molecular cancer research : MCR. 10. 677-88. 10.1158/1541-7786.MCR-11-0519
[viii] Pengcheng Zhu, Yan Yih Goh, Hwee Fang Alison Chin, Sander Kersten, Nguan Soon Tan, Angiopoietin-like 4: a decade of research, Bioscience Reports (Dec. 22, 2011).
[ix] Jie Tan et al. (2012).