Distribution of Log Fold Change

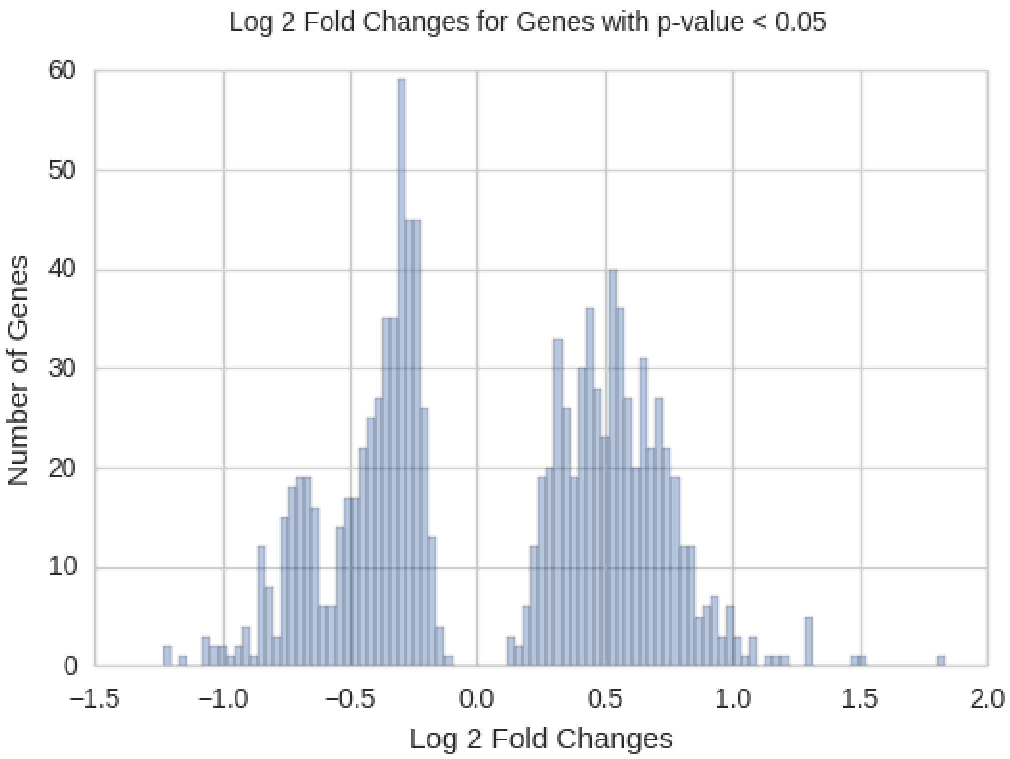
February 6, 2017

1 Distribution of Log Fold Change

For the purpose of replicating the *mRNA-Seq expression and MS3 proteomics profiling of human post-mortem BA9 brain tissue for Parkinson Disease and neurologically normal individuals* study by **Dumitriu A** et al. two different data sets were used, first the data sets of differentially expressed genes from the analysis using DESeq2 (an R Bioconductor Package) and the proteomics data set provided by the author of the study.

To reproduce the same plot I used three different Python libraries `pandas` to work with the data sets, `seaborn` for the aesthetic plots and `matplotlib_venn` to generate Venn Diagram. I also tried to do same thing in `R-Studio`.

```
In [7]: %matplotlib inline
        import seaborn as sns
        import pandas as pd
        from matplotlib_venn import venn2

        # Load the RNA-seq differentially expressed genes data
        datPD = pd.read_table('parkinsonDE.txt')

        # Slice the data by the adjusted p-value
        lowPVal = datPD[datPD['padj'] < 0.05]
        sns.set_style('whitegrid')

        # Generate the distribution plot of the differentially expressed genes
        log2FCData = lowPVal['log2FoldChange']
        log2FCPlot = sns.distplot(log2FCData, kde=False, bins=100)

        # Label the distribution plot
        log2FCPlot.set(xlabel='Log 2 Fold Changes', ylabel='Number of Genes')
        sns.plt.suptitle('Log 2 Fold Changes for Genes with p-value < 0.05')

        #ax.set_xtickslabel([-1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, 2.0])

Out[7]: <matplotlib.text.Text at 0x7eff7271f3c8>
```

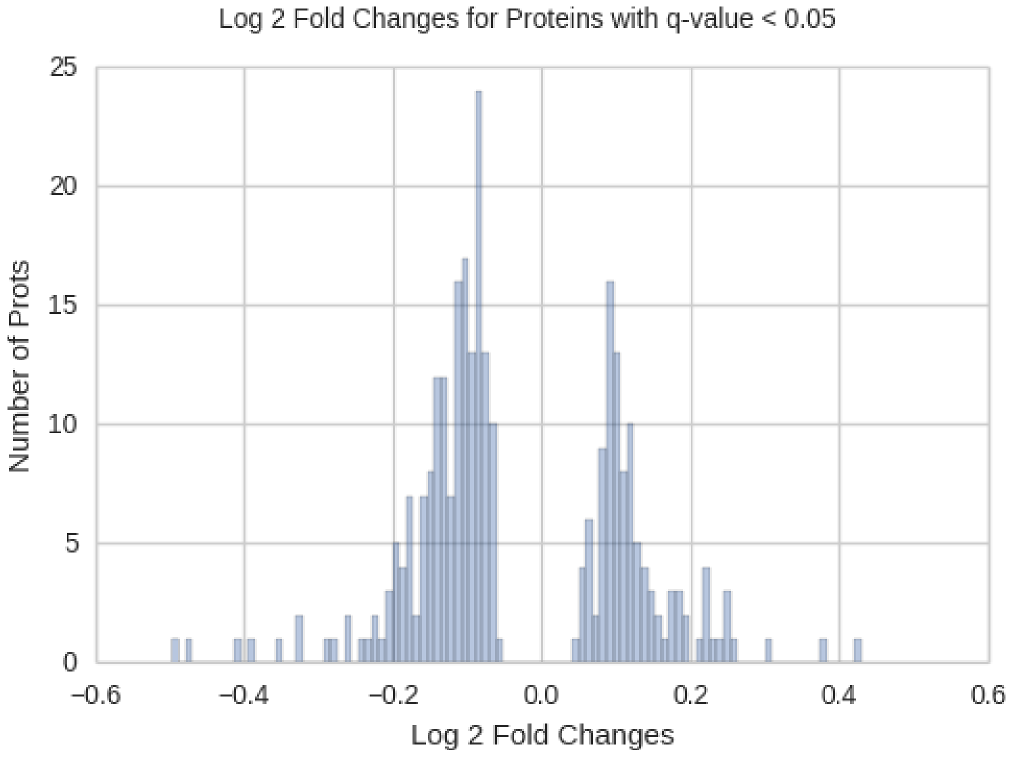


```
In [40]: totalGenes = datPD.shape[0]
         totalPVLGenes = lowPVal.shape[0]
         PVLGenesRat = totalPVLGenes/totalGenes * 100
         upRegGenes = len(lowPVal[lowPVal['log2FoldChange'] > 0])
         downRegGenes = len(lowPVal['log2FoldChange']) - len(lowPVal[lowPVal['log2F
         upRegGenesRat = upRegGenes/len(lowPVal['log2FoldChange']) * 100
         downRegGenesRat = 100 - upRegGenesRat

         print('Total number of genes: ' + str(totalGenes))
         print('Total number of genes with p-value < 0.05: ' + str(totalPVLGenes) +
         print('Number of up regulated genes: ' + str(upRegGenes) + ', ' + str(roun
         print('Number of down regulated genes: ' + str(downRegGenes) + ', ' + str
```

```
Total number of genes: 17580
Total number of genes with p-value < 0.05: 1095, 6.23%
Number of up regulated genes: 570, 52.05%
Number of down regulated genes: 525, 47.95%
```

Looking at the codes above we know that out of the total **17580** only **1095** or **6.23%** of them have p-values lower than 0.05. And out of the genes with p-value lower than 0.05 we know that **570** or **52.05%** of them are up regulated while **525** or **47.95%** of them are down regulated.

```
In [8]: # Load the MS3 Proteomics data
        protDatPD = pd.read_table('protPDE.csv')

        # Slice the data by the FDR q-value
        lowQVal = protDatPD[protDatPD['qvalue'] < 0.05]

        # Generate the distribution plot of the proteomics data
        log2FCPData = lowQVal['log2FoldChange']
        log2FCPPlot = sns.distplot(log2FCPData, kde=False, bins=100)

        # Label the distribution plot
        log2FCPPlot.set(xlabel='Log 2 Fold Changes', ylabel='Number of Prots')
        sns.plt.suptitle('Log 2 Fold Changes for Proteins with q-value < 0.05')

Out[8]: <matplotlib.text.Text at 0x7eff72501320>
```



```
In [43]: totalProts = protDatPD.shape[0]
         totalPVLProts = lowQVal.shape[0]
         PVLProtsRat = totalPVLProts/totalProts * 100
         upRegProts = len(lowQVal[lowQVal['log2FoldChange'] > 0])
         downRegProts = len(lowQVal['log2FoldChange']) - len(lowQVal[lowQVal['log2F
         upRegProtsRat = upRegProts/len(lowQVal['log2FoldChange']) * 100
         downRegProtsRat = 100 - upRegProtsRat
```

```
        print('Total number of Proteins: ' + str(totalProts))
        print('Total number of Proteins with q-value < 0.05: ' + str(totalPVLProts
        print('Number of up regulated Proteins: ' + str(upRegProts) + ', ' + str(r
        print('Number of down regulated Proteins: ' + str(downRegProts) + ', ' + s
```

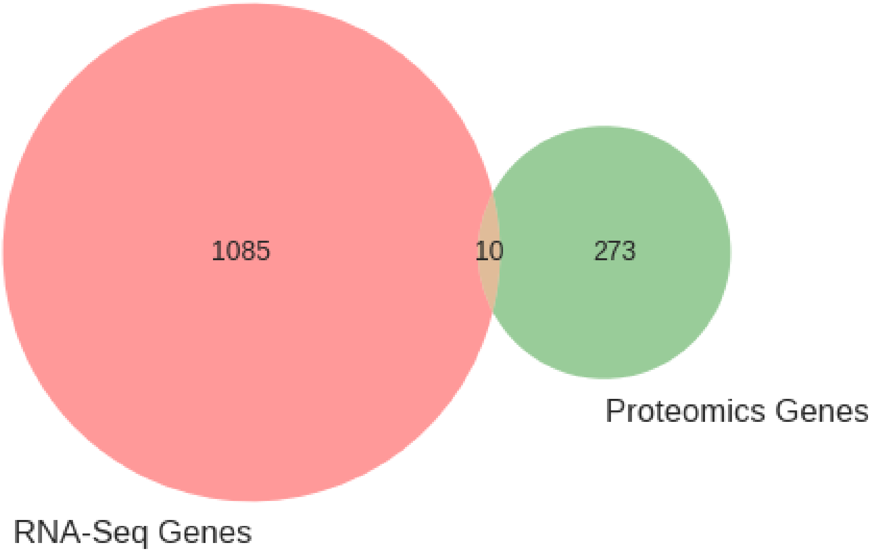
```
Total number of Proteins: 3558
Total number of Proteins with q-value < 0.05: 283, 7.95%
Number of up regulated Proteins: 106, 37.46%
Number of down regulated Proteins: 177, 62.54%
```

Looking at the codes above we know that out of the total of **3558** only **283** or **7.95%** of them have p-values lower than 0.05. And out of the genes with p-value lower than 0.05 we know that **106** or **37.46%** of them are up regulated while **177** or **62.54%** of them are down regulated.

```
In [45]: # List the names of RNA-seq and proteomics genes with p-val < 0.05 and q-v
         geneNames = list(lowPVal['symbol'])
         protNames = list(lowQVal['Symbol'])

         # Array of the intersection between
         intersectGPs = []

         # Compare the list of proteins and genes
         for protName in protNames:
             if protName in geneNames:
                 intersectGPs.append(protName)

         print('Here are the genes/proteins that exist in both subset: ' + ', '.joi
```

```
Here are the genes/proteins that exist in both subset: ACTA2, PRUNE2, ALDH1A1, SLC4
```

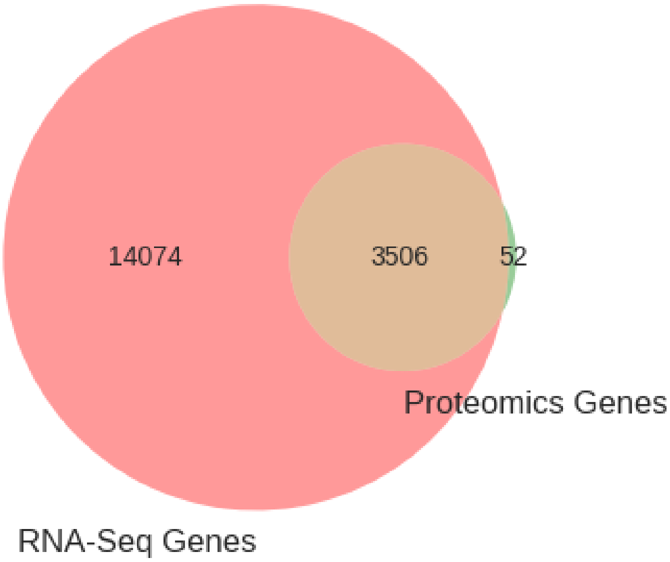
```
In [47]: # Create subsets for both proteomic genes and RNA-Seq genes
         protRNA = [len(geneNames)-len(intersectGPs), len(protNames)-len(intersectG

         # Generate the Venn diagram
         protRNASubVenn = venn2(protRNA, ['RNA-Seq Genes', 'Proteomics Genes'])
```



Comparing the RNA-Seq Genes and Proteomics Genes, it's found that they have **10** similar genes, they are; *ACTA2, PRUNE2, ALDH1A1, SLC4A8, CRELD1, VAPB, GFM1, NDUFS1, MTX3, OPA1*

```
In [49]: # List the names of all RNA-seq and proteomics genes in data set respecti
         allGeneNames = list(datPD['symbol'])
         allProtNames = list(protDatPD['Symbol'])

         # Array of the intersection between the them
         allIntersectGPs = []

         # Compare the list of the total RNA-seq genes and Proteomics Genes
         for allProtName in allProtNames:
             if allProtName in allGeneNames:
                 allIntersectGPs.append(allProtName)
```

```
In [50]: # Create subsets for the total of both proteomic genes and RNA-Seq genes
         totProtRNA = [len(allGeneNames)-len(allIntersectGPs), len(allProtNames)-le
                   len(allIntersectGPs)]

         # Genereate the Venn diagram
         totProtRNAVenn = venn2(totProtRNA, ['RNA-Seq Genes', 'Proteomics Genes'])
```


14074
3506
52
Proteomics Genes
RNA-Seq Genes