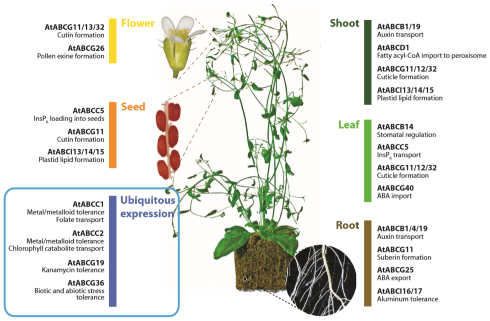
CMSE 410 Midterm Report
Yashwanth Manne
03/27/2020

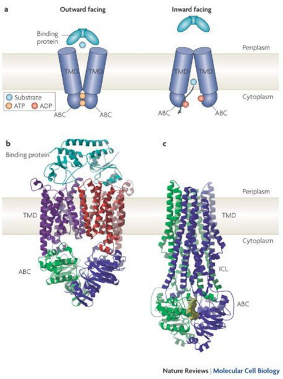
Linking of ATP-binding cassette (ABC) transporters and lipid transfer proteins (LTPs) in Arabidopsis

Background

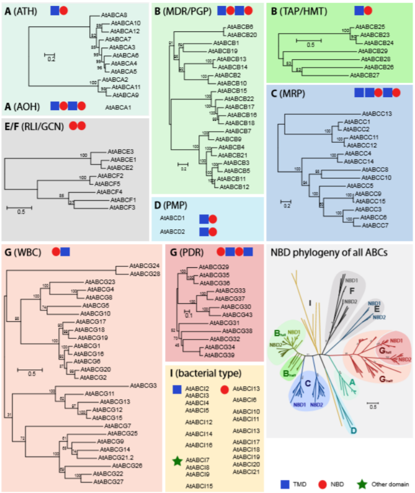


Arabidopsis thaliana is the principal genetic model and standard reference organism in plant and crop science. This is due to its short generation time, small size and space requirements, prolific seed production, and small, genetically tractable genome.[1]

ATP-binding cassette (ABC) transporters are one of the largest protein families in all living organisms. They function as a method of transmembrane transport driven by ATP hydrolysis. In prokaryotes, they not only serve as importers (prokaryotes only) and exporters but are also thought to function in DNA repair and translation.[2] In Arabidopsis, 22 out of 130 ABC protein types have been functionally analyzed.[3] All of these have been localized to membranes and are involved in roles including but not limited to detoxification, organ growth, plant nutrition, plant development, abiotic stress response, pathogen resistance, plant-environment interaction, pathogen response, surface lipid deposition, phytate accumulation in seeds, and transport of the phytohormones auxin and abscisic acid.

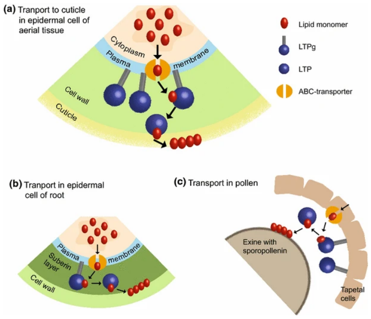


Nature Reviews | Molecular Cell Biology



Generally, each ABC transporter contains two transmembrane domains (TMDs) and two cytosolic/nucleotide-binding domains (NBDs). In plants, the encoding of each subunit varies by subgroups.[3] All may be coded by individual genes, two genes may each encode an NBD and TMD pair and form heterodimers, a single gene may encode an NBD and TMD pair and form homodimers, or a single gene can encode all four domains. Note that there are 8 different ABC subgroups (A, B, C, D, E, F, G, I) and the I subgroup has bacterial origins.

Lipid Transfer Proteins (LTPs) are small, compact proteins folded around a hydrophobic cavity, enabling it to transport other hydrophobic molecules like lipids. In fact, it has been shown that the majority of lipid traffic is done by LTPs rather than vesicles.[4] It is thought to be key in the colonization of land as it is encoded by gene families in all land plants but not algae or other organisms.[5] There is evidence that LTPs are involved in the transfer and deposition of monomers required for the assembly of water-proof lipid barriers and signaling during pathogen attacks. Additionally, it has been theorized that LTPs facilitate the transfer of barrier materials and adhesion between barriers and extracellular materials.



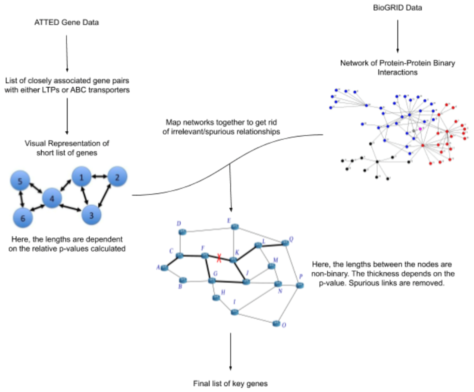
Problem

It has been shown that ABC transporters often work in parallel with LTPs that 'pick up' the compound from ABC transporters.[6] This project aims to identify possible gene pairs of ABC transporters and LTPs to provide a branching point for future research of cellular transport mechanisms in plants. This is vital as transport mechanisms are vital in most plant pathways and its understanding can aid not just future research but also the production of high-value plant products or crops.

Approach

In general, the project aims to take co-expression data of Arabidopsis genes and rule out spurious relations by cross-referencing with protein-protein interactions.

1. Obtain Arabidopsis data from ATTED & BioGRID
2. Load data using Python's Jupyter Notebooks and Pandas, OS libraries
3. Remove excess annotations and irrelevant gene data
4. Plot preliminary figures using Matplotlib, Numpy, and Networkx libraries
5. Create weighted graph using Networkx
6. Use Networkx's astar_shortest_path algorithm to get all gene interactions less than cutoff
7. Compare to list of top 50 ABC-LTP gene pairs with lowest Mutual Rank (MR) from ATTED data
8. Check if any of output pairs have studies done on them
9. Attempt to recreate with tissue-specificity


ATTED Gene Data

List of closely associated gene pairs with either LTPs or ABC transporters

Visual Representation of short list of genes

Here, the lengths are dependent on the relative p-values calculated

BioGRID Data

Network of Protein-Protein Binary Interactions

Map networks together to get rid of irrelevant/spurious relationships

Here, the lengths between the nodes are non-binary. The thickness depends on the p-value. Spurious links are removed.

Final list of key genes

Methods & Software

The main libraries used in this project are Pandas, Networkx, Matplotlib, Numpy, OS, Seaborn, Collections.

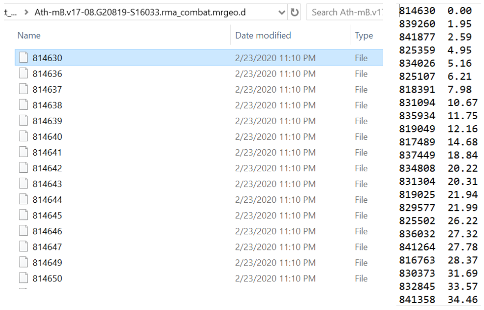
Functions:

- Pandas read_csv('',sep ='\t')
  - I: Tab-separated textfile
  - O: Pandas DataFrame
- Taxonomy Name/ID Status Report
  - I: Strings of names/ NCBI taxonomy ID
  - O: Tab-separated text file with 'code', 'taxid', 'primary taxid', and 'taxname'
- Networkx from_pandas_edgelist()
  - I: DataFrame, source, target
  - O: Networkx graph
- Networkx add_edge()
  - I: Source node, target node, weight
  - O: Networkx graph
- Networkx draw_netwokx_nodes()
  - I: Networkx Graph, position, node size
  - O: plot of just nodes
- Networkx draw_netwokx_edgess()
  - I: Networkx Graph, position, edgelist, width, edge color, line style
  - O: plot with edges between nodes
- Networkx draw_netwokx_labels()
  - I: Networkx Graph, position, font size, font family
  - O: Nodes are labelled
- Networkx astar_path_length()
  - I: Graph, source node, target node
  - O: Length of shortest path between source and target using Astar heuristic algorithm thats optimal for big datasets

Data

ATTED

ATTED-II is a plant coexpression database created from numerous DNA microarray analysis studies.[7,8] The ATTED-II data seems to be a rma.mrgeo.d file. Specifically, the folder that I downloaded, "Ath-mB.v17-08.G20819-S16033.rma_combat.mrgeo.d," contains multiple files, with each being a coexpressed gene list for that specific query gene. Each file has the gene ID, a mutual rank of the gene (MR), and the Pearson's correlation coefficient (PCC) of each gene to a given gene. The creators of the data recommended that MR be used rather than the PCC for data analysis. Data is shown to the right.

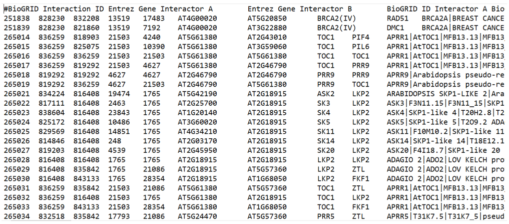


```
Name      Date modified         Type
814630    2/23/2020 11:10 PM    File
814636    2/23/2020 11:10 PM    File
814637    2/23/2020 11:10 PM    File
814638    2/23/2020 11:10 PM    File
814639    2/23/2020 11:10 PM    File
814640    2/23/2020 11:10 PM    File
814641    2/23/2020 11:10 PM    File
814642    2/23/2020 11:10 PM    File
814643    2/23/2020 11:10 PM    File
814644    2/23/2020 11:10 PM    File
814645    2/23/2020 11:10 PM    File
814646    2/23/2020 11:10 PM    File
814647    2/23/2020 11:10 PM    File
814649    2/23/2020 11:10 PM    File
814650    2/23/2020 11:10 PM    File
```

```
814630  0.00
839260  1.95
841877  2.59
825359  4.95
834026  5.16
825107  6.21
818391  7.98
831094  10.67
835934  11.75
819049  12.16
817489  14.68
837449  18.84
834808  20.22
831304  20.31
819025  21.94
829577  21.99
825502  26.22
836032  27.32
841264  27.78
816763  28.37
830373  31.69
832845  33.57
841358  34.46
```

BioGRID

Biological General Repository for Interaction Datasets (BioGRID), which is a curated database of genetic, protein, and chemical interactions as well as post-translational modifications.[9] The "BIOGRID-ORGANISM-3.5.181.tab2.zip" file that I downloaded contains the protein-protein interactions of all organisms but I will specifically be using the file for Arabidopsis: "BIOGRID-ORGANISM-Arabidopsis_thaliana_Columbia-3.5.181.tab2". This file is formatted as a Tab 2.0 Delimited Text file and contains all interaction and associated annotation data. Data is shown below.



```
#BioGRID Interaction ID Entrez Gene Interactor A        Entrez Gene Interactor B        BioGRID ID Interactor A Bio
251838  828230  832208  13519   17483   AT4G00020       AT5G20850       BRCA2(IV)       RAD51   BRCA2A|BREAST CANCE
251839  828230  821860  13519   7192    AT4G00020       AT3G22880       BRCA2(IV)       DMC1    BRCA2A|BREAST CANCE
265014  836259  818903  21503   4240    AT5G61380       AT2G43010       TOC1    PIF4    APRR1|AtTOC1|MFB13.13|MFB13
265015  836259  825075  21503   10390   AT5G61380       AT3G59060       TOC1    PIL6    APRR1|AtTOC1|MFB13.13|MFB13
265016  836259  836259  21503   21503   AT5G61380       AT5G61380       TOC1    TOC1    APRR1|AtTOC1|MFB13.13|MFB13
265017  836259  819292  21503   4627    AT5G61380       AT2G46790       TOC1    PRR9    APRR1|AtTOC1|MFB13.13|MFB13
265018  819292  819292  4627    4627    AT2G46790       AT2G46790       PRR9    PRR9    APRR9|Arabidopsis pseudo-re
265019  819292  836259  4627    21503   AT2G46790       AT5G61380       PRR9    TOC1    APRR9|Arabidopsis pseudo-re
265021  834224  816408  19474   1765    AT5G42190       AT2G18915       ASK2    LKP2    ARABIDOPSIS SKP1-LIKE 2|Ara
265022  817111  816408  2463    1765    AT2G25700       AT2G18915       SK3     LKP2    ASK3|F3N11.15|F3N11_15|SKP1
265023  838604  816408  23843   1765    AT1G20140       AT2G18915       SK4     LKP2    ASK4|SKP1-like 4|T20H2.8|T2
265024  825172  816408  10486   1765    AT3G60020       AT2G18915       SK5     LKP2    ASK5|SKP1-like 5|T2O9.2 ADA
265025  829569  816408  14851   1765    AT4G34210       AT2G18915       SK11    LKP2    ASK11|F10M10.2|SKP1-like 11
265026  814846  816408  248     1765    AT2G03170       AT2G18915       SK14    LKP2    ASK14|SKP1-like 14|T18E12.1
265027  819203  816408  4539    1765    AT2G45950       AT2G18915       SK20    LKP2    ASK20|F4I18.7|SKP1-like 20
265028  816408  816408  1765    1765    AT2G18915       AT2G18915       LKP2    LKP2    ADAGIO 2|ADO2|LOV KELCH pro
265029  816408  835842  1765    21086   AT2G18915       AT5G57360       LKP2    ZTL     ADAGIO 2|ADO2|LOV KELCH pro
265030  816408  843133  1765    28354   AT2G18915       AT1G68050       LKP2    FKF1    ADAGIO 2|ADO2|LOV KELCH pro
265031  836259  835842  21503   21086   AT5G61380       AT5G57360       TOC1    ZTL     APRR1|AtTOC1|MFB13.13|MFB13
265032  836259  816408  21503   1765    AT5G61380       AT2G18915       TOC1    LKP2    APRR1|AtTOC1|MFB13.13|MFB13
265033  836259  843133  21503   28354   AT5G61380       AT1G68050       TOC1    FKF1    APRR1|AtTOC1|MFB13.13|MFB13
265034  832518  835842  17793   21086   AT5G24470       AT5G57360       PRR5    ZTL     APRR5|T31K7.5|T31K7_5|pseud
```

In [1]:

```
%%time
# Import necessary Libraries
import os
import pandas as pd
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
import seaborn as sns
import collections
# Reading in BioGRID as DataFrame
path = os.path.join('C:\\Users\\ysman\\OneDrive\\Desktop\\project_data\\BIOGRID-ORGANISM-Arabidopsis_thaliana_Columbia-3.5.181.tab2.txt'
bioGRID_file = open(path, "r")
testFile = open(os.path.join('C:\\Users\\ysman\\OneDrive\\Desktop\\project_directory\\data\\test.txt'), 'r')
bioGRID_DF = pd.read_csv(bioGRID_file, sep = '\t')
# Simplified DataFrame to only include interactions
simplebGRID = bioGRID_DF[['Entrez Gene Interactor A','Entrez Gene Interactor B']]
# Identifying Organisms Present in BioGRID
OrganismTypesA = list(bioGRID_DF['Organism Interactor A'].unique())
OrganismTypesB = list(bioGRID_DF['Organism Interactor B'].unique())
OrganismTypesA.sort()
OrganismTypesB.sort()
OrganismTypes = list(set([*OrganismTypesA, *OrganismTypesB]))
OrganismTypes.sort()
# Use NCBI's Taxonomy Name/ID Status Report - Plug in OrganismTypes and get .txt
organismIDs = pd.read_csv('../../project_data/tax_report.txt', sep = '\t')
organismIDs.drop(columns = ['|','|.1','|.2', 'code', 'primary taxid'],inplace = True)
# Categorize into different subsets based on organism ID. We know '3702' is Arabidopsis
mask1 = bioGRID_DF['Organism Interactor A'] == 3702
mask2 = bioGRID_DF['Organism Interactor B'] == 3702
onlyArabDF = bioGRID_DF[mask1& mask2]
oneArabDF = bioGRID_DF[~mask1|~mask2]
noArabDF = bioGRID_DF[~mask1 & ~mask2]
# Get list of genes so I can import the necessary ATTED Data. Note that the ATTED data has a text file by Entrez gene ID
# WholeData:
wholeGenesA = list(bioGRID_DF['Entrez Gene Interactor A'].unique())
wholeGenesB = list(bioGRID_DF['Entrez Gene Interactor B'].unique())
wholeGenesA.sort()
wholeGenesB.sort()
wholeGenes = list(set([*wholeGenesA, *wholeGenesB]))
wholeGenes.sort()
# Only Arabidopsis Subset
ArabGenesA = list(onlyArabDF['Entrez Gene Interactor A'].unique())
ArabGenesB = list(onlyArabDF['Entrez Gene Interactor B'].unique())
ArabGenesA.sort()
ArabGenesB.sort()
ArabGenes = list(set([*ArabGenesA, *ArabGenesB]))
ArabGenes.sort()
# Read in only the Overlapping Genes
# Get a list of all genes in ATTED
atted = pd.read_csv('../../project_data/Ath-mB.v17-08.G20819-S16033.rma_combat.mrgeo.d/814630', sep = '\t', header = None)
atted = atted.sort_values(by = 0)
a1 = np.array(atted[0])
attedGenes = list(a1)
# Reading in of Overlapping Genes
attedpath = 'C:\\Users\\ysman\\OneDrive\\Desktop\\project_data\\Ath-mB.v17-08.G20819-S16033.rma_combat.mrgeo.d\\'
overlapGenes = []
for i in range (len(wholeGenes)):
    if os.path.exists(attedpath+'{}'.format(wholeGenes[i])):
        overlapGenes.append(wholeGenes[i])
DF = {0:attedGenes}
for x in overlapGenes:
    tempAtted = pd.read_csv('../../project_data/Ath-mB.v17-08.G20819-S16033.rma_combat.mrgeo.d/{}'.format(x), sep = '\t', header= None)
    tempAtted = tempAtted.sort_values(by = 0)
    templist = list(tempAtted[1])
    DF.update({x:templist})
attedData = pd.DataFrame(DF,dtype='float64')
```

```
Wall time: 5min 20s
```

In [2]:

```
print(bioGRID_DF.columns)
print(bioGRID_DF.dtypes)
bioGRID_DF.head()
```

```
Index(['#BioGRID Interaction ID', 'Entrez Gene Interactor A',
       'Entrez Gene Interactor B', 'BioGRID ID Interactor A',
       'BioGRID ID Interactor B', 'Systematic Name Interactor A',
       'Systematic Name Interactor B', 'Official Symbol Interactor A',
       'Official Symbol Interactor B', 'Synonyms Interactor A',
       'Synonyms Interactor B', 'Experimental System',
       'Experimental System Type', 'Author', 'Pubmed ID',
       'Organism Interactor A', 'Organism Interactor B', 'Throughput', 'Score',
       'Modification', 'Phenotypes', 'Qualifications', 'Tags',
       'Source Database'],
      dtype='object')
#BioGRID Interaction ID          int64
Entrez Gene Interactor A         int64
Entrez Gene Interactor B         int64
BioGRID ID Interactor A          int64
BioGRID ID Interactor B          int64
Systematic Name Interactor A    object
Systematic Name Interactor B    object
Official Symbol Interactor A    object
Official Symbol Interactor B    object
Synonyms Interactor A           object
Synonyms Interactor B           object
Experimental System             object
Experimental System Type        object
Author                          object
Pubmed ID                        int64
Organism Interactor A            int64
Organism Interactor B            int64
Throughput                      object
Score                           object
Modification                    object
Phenotypes                      object
Qualifications                  object
Tags                            object
Source Database                 object
dtype: object
```

Out[2]:

	#BioGRID Interaction ID	Entrez Gene Interactor A	Entrez Gene Interactor B	BioGRID ID Interactor A	BioGRID ID Interactor B	Systematic Name Interactor A	Systematic Name Interactor B	Official Symbol Interactor A	Official Symbol Interactor B	Synonyms Interactor A	...	Pubme I
0	251838	828230	832208	13519	17483	AT4G00020	AT5G20850	BRCA2(IV)	RAD51	BRCA2A|BREAST CANCER 2 like 2A|EDA20|EMBRYO SA...	...	1501444
1	251839	828230	821860	13519	7192	AT4G00020	AT3G22880	BRCA2(IV)	DMC1	BRCA2A|BREAST CANCER 2 like 2A|EDA20|EMBRYO SA...	...	1501444
2	265014	836259	818903	21503	4240	AT5G61380	AT2G43010	TOC1	PIF4	APRR1|AtTOC1|MFB13.13|MFB13_13|PRR1|PSEUDO-RES...	...	1463416
3	265015	836259	825075	21503	10390	AT5G61380	AT3G59060	TOC1	PIL6	APRR1|AtTOC1|MFB13.13|MFB13_13|PRR1|PSEUDO-RES...	...	1463416
4	265016	836259	836259	21503	21503	AT5G61380	AT5G61380	TOC1	TOC1	APRR1|AtTOC1|MFB13.13|MFB13_13|PRR1|PSEUDO-RES...	...	1463416

5 rows × 24 columns

In [3]:

```
simplebGRID.head()
```

Out[3]:

	Entrez Gene Interactor A	Entrez Gene Interactor B
0	828230	832208
1	828230	821860
2	836259	818903
3	836259	825075
4	836259	836259

In [4]:
```
organismIDs
```

Out[4]:

	taxid	taxname
0	3055	Chlamydomonas reinhardtii
1	3702	Arabidopsis thaliana
2	3847	Glycine max
3	4081	Solanum lycopersicum
4	4098	Nicotiana tomentosiformis
5	4577	Zea mays
6	9606	Homo sapiens
7	9823	Sus scrofa
8	9913	Bos taurus
9	10090	Mus musculus
10	10116	Rattus norvegicus
11	10298	Human alphaherpesvirus 1
12	12242	Tobacco mosaic virus
13	39947	Oryza sativa Japonica Group
14	284812	Schizosaccharomyces pombe 972h-
15	316407	Escherichia coli str. K-12 substr. W3110
16	559292	Saccharomyces cerevisiae S288C

In [5]:
```
attedData.head()
```

Out[5]:

	0	814630	814637	814641	814643	814644	814646	814647	814649	814651	...	3767983	3768737	3768753	3768908	3769417	3769951
0	814630.0	0.00	13293.00	3853.85	1178.08	11468.52	9484.30	1937.88	8729.24	17812.21	...	12788.20	15990.21	12623.05	13868.44	5235.79	13968.73
1	814636.0	318.41	1345.84	3478.57	1796.04	160.36	11708.48	1387.92	3212.26	10825.32	...	6007.37	13585.46	9601.16	4391.35	5081.69	9042.38
2	814637.0	13293.00	0.00	2975.28	3094.54	3910.94	16486.87	3222.50	2356.92	16362.67	...	16634.63	15844.40	13778.88	10607.97	2558.52	15105.20
3	814638.0	2012.21	6477.10	4150.36	11929.09	10871.97	18827.22	17722.02	8126.23	14922.36	...	11786.21	17986.79	10007.81	7227.17	11842.46	14021.38
4	814639.0	391.97	10638.69	7853.09	7441.45	8371.53	10912.49	1848.11	10229.29	18331.34	...	15761.09	10034.88	9979.45	10097.74	3848.97	12749.52

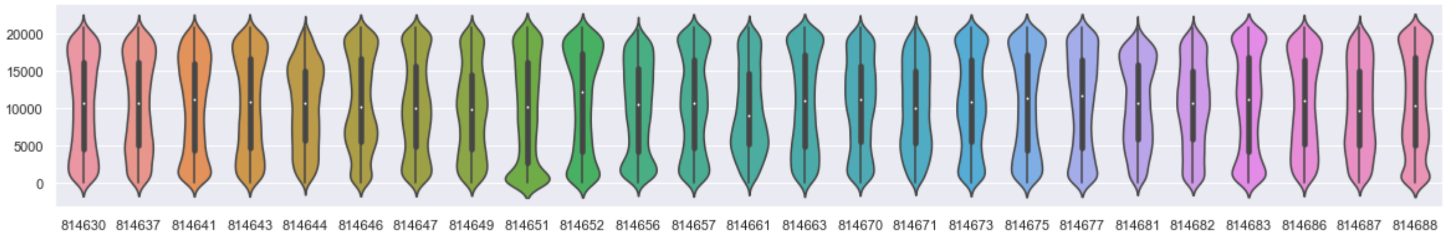
5 rows × 8782 columns

Preliminary Data Analysis

The BioGRID data shows that there are 15 other organisms interacting with Arabidopsis. Of the 56198 total interactions, 55814 are Arabidopsis:Arabidopsis interactions, 384 are Arabidopsis:Other interactions, and 0 that don't involve Arabidopsis. There are 10550 unique genes in the dataset but only 10367 genes involved in only Arabidopsis:Arabidopsis interactions. There are some missing annotations in the dataset, but they are not relevant to this project and can be ignored.

There are 20819 total genes in the ATTED dataset out of which 8782 are also present in the BioGRID set. The dataset contains no missing values.

In [6]:
```
# Create Violin Plots
sns.set()
vioColnames= list(attedData.columns)
vioColnames = vioColnames[1:]
vioColnames
ax = plt.figure(figsize = (20,3))
ax =sns.violinplot(data = attedData[vioColnames[:25]])
```



This plot shows that the data is normalized and the distribution of the Mutal Rank values of the ATTED genes are relatively similar across multiple genes.

In [7]:

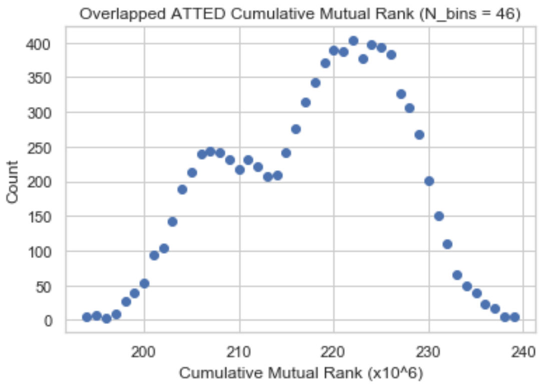
```
# Create BioGRID Networkx Graph
G = nx.from_pandas_edgelist(simplebGRID, source = 'Entrez Gene Interactor A', target = 'Entrez Gene Interactor B')
# ATTED Degree-Rank based on MR
sumMR_A = attedData.sum()
sumMR_A = sumMR_A.drop(0)
sumMR_A = list(sumMR_A)
rank_sumMR_A = sorted(sumMR_A, reverse = True)
# BioGRID Degree-Rank based on MR
deg_Bio = [G.degree(gene) for gene in wholeGenes]
rank_deg_bio = sorted(deg_Bio, reverse = True)
# ATTED Degree-Count based on MR
binsize = 1000000
binMR= (np.array(sumMR_A)/binsize).astype('int')
attedCount = collections.Counter(binMR)
MR_A, cnt_A = zip(*attedCount.items())
n_bins = len(MR_A)
# BioGRID Degree-Count
deg_Bio_count = collections.Counter(deg_Bio)
deg_B,cnt_B = zip(*deg_Bio_count.items())
```

In [8]:

```
# Visualize MR distribution of ATTED (pseduo histogram)
sns.set_style('whitegrid')
plt.scatter(MR_A,cnt_A)
plt.title('Overlapped ATTED Cumulative Mutual Rank (N_bins = {})'.format(n_bins))
plt.xlabel('Cumulative Mutual Rank (x10^{:.0f})'.format(np.log10(binsize)))
plt.ylabel('Count')
```

Out[8]:

```
Text(0, 0.5, 'Count')
```

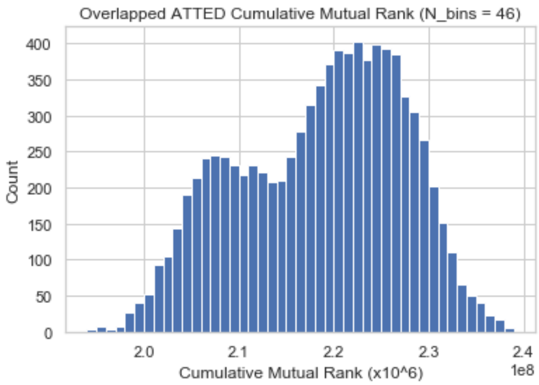


In [9]:

```
# Visualize MR distribution of ATTED (Histogram)
plt.hist(sumMR_A,bins = np.arange(194000000,240000000,1000000))
plt.title('Overlapped ATTED Cumulative Mutual Rank (N_bins = {})'.format(n_bins))
plt.xlabel('Cumulative Mutual Rank (x10^{:.0f})'.format(np.log10(binsize)))
plt.ylabel('Count')
```

Out[9]:

```
Text(0, 0.5, 'Count')
```



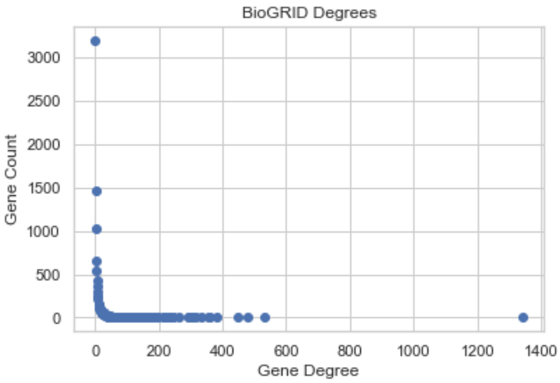
The Cumulative Mutual Rank follows a bimodal distribution with peaks at 207 million and 222 million with about 250 and 400 genes having cumulative MRs in that range.

In [10]:

```
# Visualize degree distribution of BioGRID (psuedo bar plot)
sns.set_style('whitegrid')
sns.axes_style('ticks')
plt.scatter(deg_B,cnt_B)
plt.title('BioGRID Degrees')
plt.xlabel('Gene Degree')
plt.ylabel('Gene Count')
```

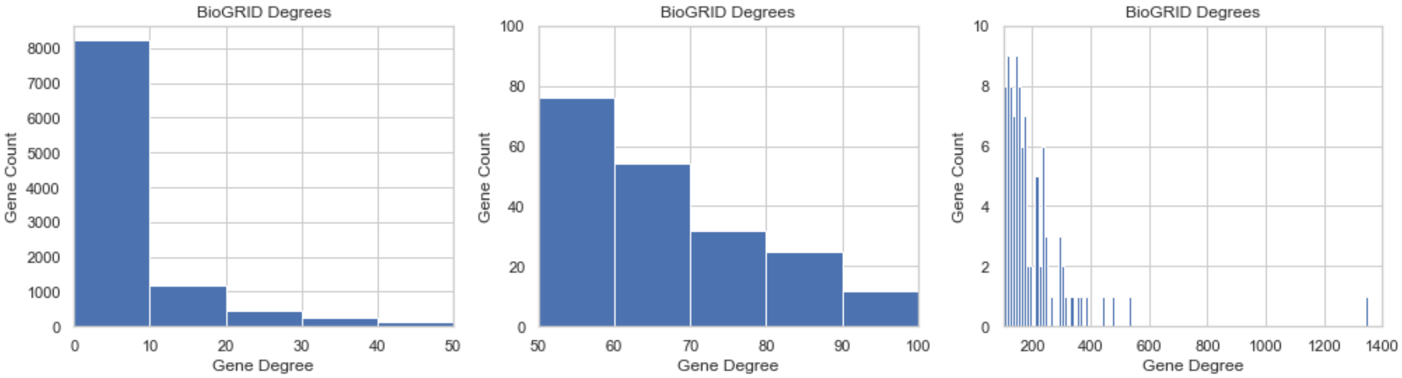
Out[10]:

```
Text(0, 0.5, 'Gene Count')
```



In [11]:

```
# Histogram Separated into 3 sections to see
fig, axs = plt.subplots(1, 3, figsize=(14, 4))
for i in range(3):
    axs[i].hist(deg_Bio, bins = np.arange(0,1500,10))
    axs[i].set_title('BioGRID Degrees')
    axs[i].set_xlabel('Gene Degree')
    axs[i].set_ylabel('Gene Count')
axs[0].set_xlim(0,50)
axs[1].set_xlim(50,100)
axs[1].set_ylim(0,100)
axs[2].set_xlim(100,1400)
axs[2].set_ylim(0,10)
plt.tight_layout()
```



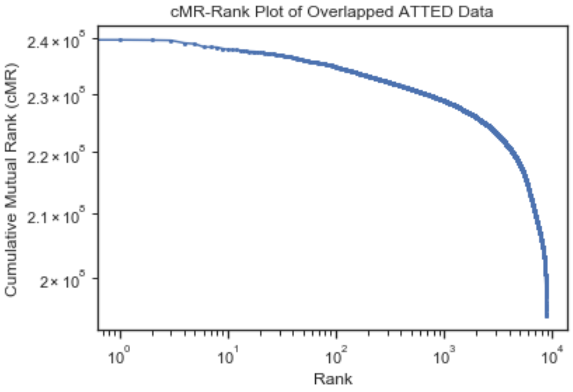
Most genes have between 0-10 connections but there is one gene with 1341 connections.

In [12]:

```
sns.set_style('ticks')
plt.loglog(rank_sumMR_A, 'b-', marker='o', markersize = 2)
plt.title("cMR-Rank Plot of Overlapped ATTED Data")
plt.ylabel("Cumulative Mutual Rank (cMR)")
plt.xlabel("Rank")
```

Out[12]:

```
Text(0.5, 0, 'Rank')
```

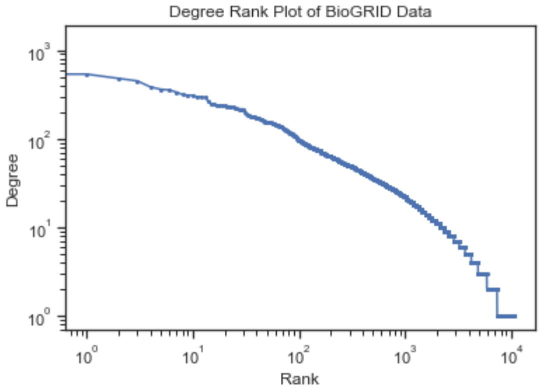


In [13]:

```
sns.set_style('ticks')
plt.loglog(rank_deg_bio, 'b-', marker='o', markersize = 2)
plt.title("Degree Rank Plot of BioGRID Data")
plt.ylabel("Degree")
plt.xlabel("Rank")
```

Out[13]:

```
Text(0.5, 0, 'Rank')
```



Toy Data

In [14]:

```
# Create Toy Data from Actual BioGRID Data
toyGRID = simplebGRID.loc[:10]
# Add Weights
np.random.seed(1)
toyGRID['MR']=np.random.randint(1,10,len(toyGRID))
toyGRID
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy (http://pand
as.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
  """
```

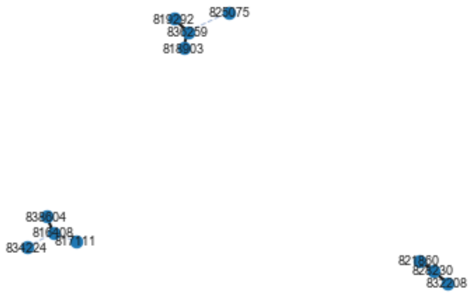
Out[14]:

	Entrez Gene Interactor A	Entrez Gene Interactor B	MR
0	828230	832208	6
1	828230	821860	9
2	836259	818903	6
3	836259	825075	1
4	836259	836259	1
5	836259	819292	2
6	819292	819292	8
7	819292	836259	7
8	834224	816408	3
9	817111	816408	5
10	838604	816408	6

In [15]:

```
%%time
# Initializes Networkx Graph
toyG = nx.Graph()
# Adds edges for each interaction
for i in range(toyGRID.shape[0]):
    toyG.add_edge(toyGRID['Entrez Gene Interactor A'].loc[i], toyGRID['Entrez Gene Interactor B'].loc[i], weight = toyGRID['MR'].loc[i])
# Categorizes edges based on weight
elarge = [(u, v) for (u, v, d) in toyG.edges(data=True) if d['weight'] > 5]
esmall = [(u, v) for (u, v, d) in toyG.edges(data=True) if d['weight'] <= 5]
# Positions for all nodes
toypos = nx.spring_layout(toyG)
# Draw Nodes
nx.draw_networkx_nodes(toyG, toypos, node_size=70)
# Draw Edges
nx.draw_networkx_edges(toyG, toypos, edgelist=elarge, width=2)
nx.draw_networkx_edges(toyG, toypos, edgelist=esmall, width=1, alpha=0.5, edge_color='b', style='dashed')
# Add Labels
nx.draw_networkx_labels(toyG, toypos, font_size=9, font_family='sans-serif')
plt.axis('off')
plt.show()
print('Distance between gene 819292 and gene 818903 is {}.'.format(nx.astar_path_length(toyG, 819292, 818903)))
print('This should output 13.')
```

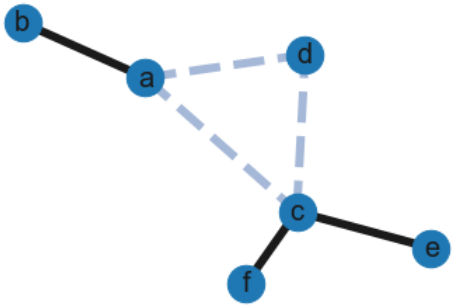
```
C:\ProgramData\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:579: MatplotlibDeprecationWarning:
The iterable function was deprecated in Matplotlib 3.1 and will be removed in 3.3. Use np.iterable instead.
  if not cb.iterable(width):
```



```
Distance between gene 819292 and gene 818903 is 13.
This should output 13.
Wall time: 354 ms
```

In [16]:

```
# CLear Online Example
G = nx.Graph()

G.add_edge('a', 'b', weight=0.6)
G.add_edge('a', 'c', weight=0.2)
G.add_edge('c', 'd', weight=0.1)
G.add_edge('c', 'e', weight=0.7)
G.add_edge('c', 'f', weight=0.9)
G.add_edge('a', 'd', weight=0.3)

elarge = [(u, v) for (u, v, d) in G.edges(data=True) if d['weight'] > 0.5]
esmall = [(u, v) for (u, v, d) in G.edges(data=True) if d['weight'] <= 0.5]

pos = nx.spring_layout(G)  # positions for all nodes

# nodes
nx.draw_networkx_nodes(G, pos, node_size=700)

# edges
nx.draw_networkx_edges(G, pos, edgelist=elarge,
                       width=6)
nx.draw_networkx_edges(G, pos, edgelist=esmall,
                       width=6, alpha=0.5, edge_color='b', style='dashed')

# labels
nx.draw_networkx_labels(G, pos, font_size=20, font_family='sans-serif')

plt.axis('off')
plt.show()
print('Distance between a and e is {}.'.format(nx.astar_path_length(G, 'a','e')))
print('It should out put 0.9.')
```



```
Distance between a and e is 0.8999999999999999.
It should out put 0.9.
```

Future Goals

- Get list of ABC transporter genes & list of LTPs genes and input permutations of these into the astar_path_length.
- Collect top pairs.
- Check literature for any evidence confirming or denying associa
- Gephi to draw out the graphs
- Separate data into tissue-specific and run once more.

Sources

1. Krämer, Ute. “Planting Molecular Functions in an Ecological Context with Arabidopsis Thaliana.” ELife, vol. 4, 25 Mar. 2015, doi:10.7554/elife.06100.
2. Davidson AL, Dassa E, Orelle C, Chen J (Jun 2008). "Structure, function, and evolution of bacterial ATP-binding cassette systems". Microbiology and Molecular Biology Reviews. 72 (2): 317–64, table of contents. doi:10.1128/MMBR.00031-07.
3. Kang, Joohyun, et al. “Plant ABC Transporters.” The Arabidopsis Book, vol. 9, 6 Dec. 2011, doi:10.1199/tab.0153.
4. Salminen, Tiina A., et al. “Lipid Transfer Proteins: Classification, Nomenclature, Structure, and Function.” Planta, vol. 244, no. 5, 25 Aug. 2016, pp. 971–997., doi:10.1007/s00425-016-2585-4.
5. Edqvist, Johan, et al. “Plant Lipid Transfer Proteins: Are We Finally Closing in on the Roles of These Enigmatic Proteins?” Journal of Lipid Research, vol. 59, no. 8, 19 Mar. 2018, pp. 1374–1382., doi:10.1194/jlr.r083139.
6. DeBono, Allan, et al. “Arabidopsis LTPG Is a Glycosylphosphatidylinositol-Anchored Lipid Transfer Protein Required for Export of Lipids to the Plant Surface.” The Plant Cell, vol. 21, no. 4, Apr. 2009, pp. 1230–1238., doi:10.1105/tpc.108.064451.
7. Takeshi Obayashi, Yuichi Aoki, Shu Tadaka, Yuki Kagaya, Kengo Kinoshita, ATTED-II in 2018: A Plant Coexpression Database Based on Investigation of the Statistical Property of the Mutual Rank Index, Plant and Cell Physiology, Volume 59, Issue 1, January 2018, Page e3, doi:10.1093/pcp/pcx191
8. Obayashi T., Hayashi S., Saeki M., Ohta H. & Kinoshita K. (2009) ATTED-II provides coexpressed gene networks for Arabidopsis. Nucleic Acids Research 37,D987–D991.

9. Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, Mike Tyers, BioGRID: a general repository for interaction datasets, Nucleic Acids Research, Volume 34, Issue suppl_1, 01 January 2006, Pages D535–D539, doi:10.1093/nar/gkj109