Vahideh Anghardi

Gene Expression Data Analysis of Persister Cancer Cells

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Tahimy Landestoy, Juan C. Mier, Prashant Karki,
Dr. Vahideh Angardi, Dr. Mehmet Orman
Department of Chemical and Biomolecular Engineering

Background

What are persisters?

- Reversible drug-tolerant, slow-growing cells[1]
- Use non-mutational mechanism to survive lethal concentration of drug [2-3]

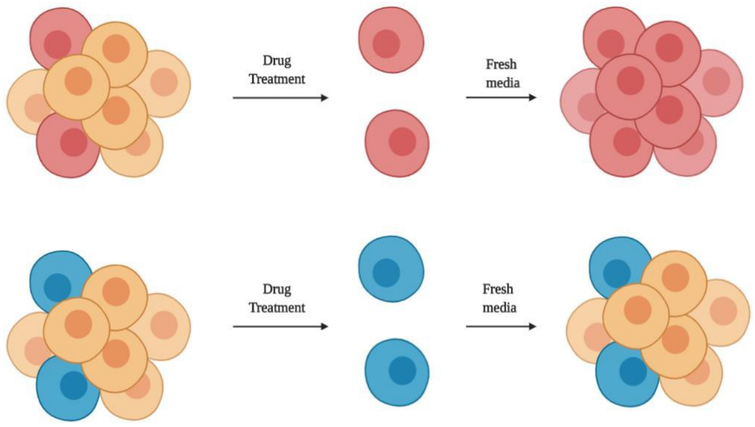
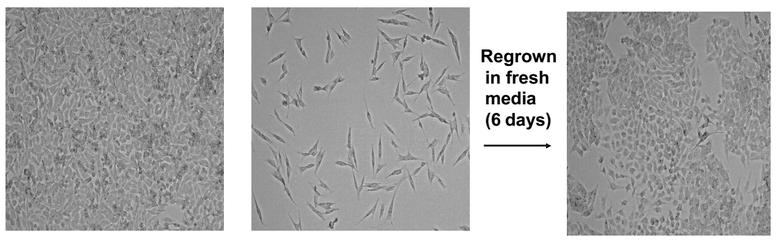


Fig. 1 Resistant cells vs. Persister cells



Untreated Day 3 | VFM Treatment Day 3 | Resuscitated Persister cells

Fig. 2: Generating persister cells from parent (A375) melanoma cells.

Why to study persisters?

- Associated with recurrence and multi-drug resistance in cancer cells [1-3]
- Mechanism associated with persistence is unclear [1]
- Anti-persister therapies can enhance the efficacy of current treatment approaches [4]

Objectives

Persister cells have unique metabolism compared to general cancer population [1]

With our study we aim to:

- Analyze database for metabolic pathways associated with persister cells using Python and machine learning programming
- Identify specific genes that are up or downregulated in persister cells.

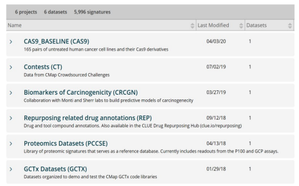
Method

Where did the data come from?

ConnectivityMap

Unravel biology with the world's largest perturbation-driven gene expression dataset.

BROAD INSTITUTE



- The Broad Institute from MIT, in collaboration with Harvard, and other leading biotech labs
- Database with cells taken from patients with and without cancer
- Cells were cured with different perturbations
- Changes were recorded in Transcriptional Expression Data for set of 1000 genes (Landmark Genes, L1000 assay)

Why did I use Python?

- Database easy access, Data Extraction, Interface Customization, and Quick system integration

Data analysis

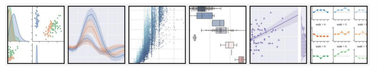
NumPy
Linear algebra library

pandas
Extremely powerful version of Excel

Data Visualization

matplotlib Version 3.1.2
Similar feel to MATLAB

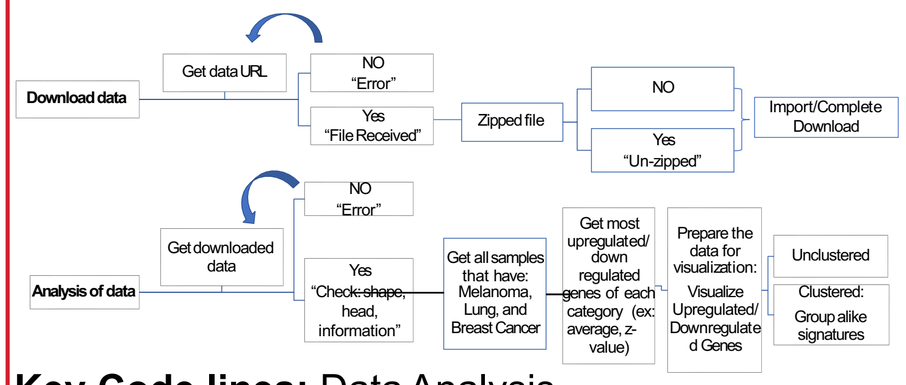
seaborn: statistical data visualization



Statistical plotting library

Approach

STEPS on the Code:



Key Code lines: Data Analysis

Importing the libraries & checking data information

```
In [62]: from cmapPy.pandasGEXpress.parse import parse
         import pandas as pd
         import numpy as np
```
Libraries

```
In [63]: __filename__ = "GSE70138_Broad_LevelS_COMPZ.gctx"
         geneX_gctoo = parse(__filename__)
```
Downloaded data

```
In [64]: geneX_gctoo.data_df.shape
Out[64]: (12328, 118050)
```
Data shape/size

- CID ("Column ID") := Gene Signatures
- RID ("Row ID") := Genes

Defining rows= genes columns= drugs

```
In [125]: geneX_df.head()
```

	cid	REPA001_A375_24H:A03	REPA001_A375_24H:A04	REPA001_A375_24H:A05	REPA001_A375_24H:A06	REPA001_A375_24H:A07	REPA001_A375_24H:A08
rid							
780		4.264143	-0.382211	-0.571711	0.564376	0.658348	-0.004232
7849		0.057249	0.304313	-0.754999	-0.589973	-0.226854	-0.363419
2978		-1.012480	-0.674992	0.414515	-0.227603	0.267899	0.239820
2049		0.308898	-0.335931	-0.502323	-1.775247	-0.666601	0.080279
2101		-0.934070	0.324732	0.495425	-0.107543	-0.091924	0.643074

5 rows × 118050 columns

Get the most upregulated and downregulated genes of each category

```
import numpy as np

def getMostAndLeastUpregulated(gene_data, numberOfGenes = None, method= lambda x: np.average(x)):
    if numberOfGenes == None:
        numberOfGenes = len(gene_data.index)

    gene_regulation = pd.DataFrame(data=np.zeros(len(gene_data.index)), index = gene_data.index, columns= ["Regulation"])

    for gene in gene_data.index:
        gene_regulation.loc[gene,'Regulation'] = method(gene_data.loc[gene,:])

    gene_regulation.sort_values(by='Regulation',inplace=True,ascending=False)

    return gene_regulation.iloc[:numberOfGenes,:], gene_regulation.iloc[-numberOfGenes:,:]
```

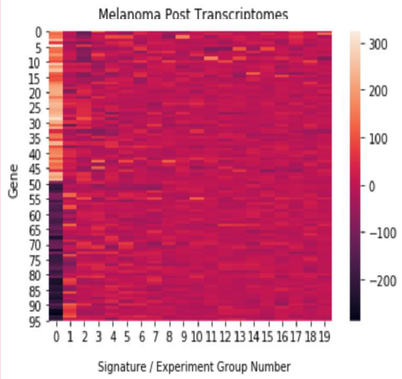
Lambda function= average of all rows

Use 50 most downregulated/ upregulated genes

Results and Conclusions

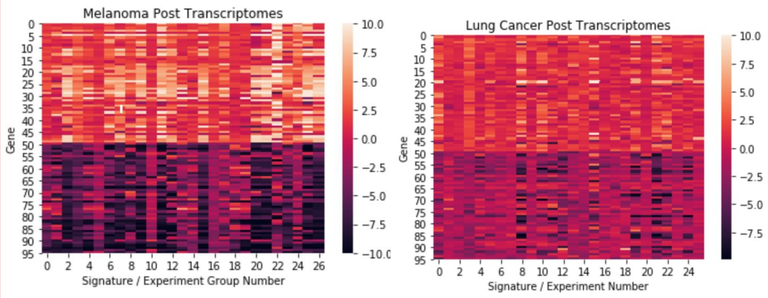
Key Code lines: Data Visualization

UNCLUSTERED



- Check to see if there are any genes that are up/down regulated
- Used principal component analysis (PCA) to reduce dimensionality
- Data not well pixelated; need to cluster

CLUSTERED



- Optimization of the data: removed non-effective signatures
- Columns represent the perturbation and rows represent gene signatures.
- Perturbations inducing similar responses were clustered together.
- Statistical significance test were conducted using z-scores.

Acknowledgements

- Summer Undergraduate Research Fellowship (SURF) Program
- The Office of Undergraduate Research
- Dr. Mehmet Orman Lab
- The Connectivity Map (CMap) at the Broad Institute

Bibliography/References

1. Hangauer, M. J. et al. Drug-tolerant persister cancer cells are vulnerable to GPX4 inhibition. Nature 551, 247-250, doi:10.1038/nature24297 (2017).
2. Sharma, S. V. et al. A chromatin-mediated reversible drug-tolerant state in cancer cell subpopulations. Cell 141, 69-80, doi:10.1016/j.cell.2010.02.027 (2010).
3. Ramirez, M. et al. Diverse drug-resistance mechanisms can emerge from drug-tolerant cancer persister cells. Nature communications 7, 10690, doi:10.1038/ncomms10690 (2016).
4. GPX4 Inhibition Selectively Targets Drug-Tolerant Persister Cells. (2017), *8*(1), OF9-OF9. https://doi.org/10.1158/2159-8290.cd-rw2017-21