



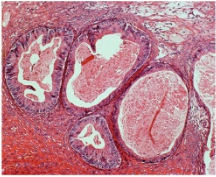
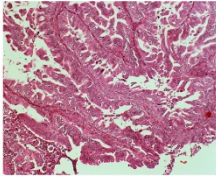
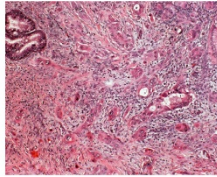
DATABRICKS NOTEBOOKS USER STORY



A clinical development researcher wants to determine whether specific clusters of variants correlate with groups of individuals and/or specific phenotypic/disease attributes (e.g. specific tumor somatic mutations occurring more often in specific histopathologically-defined cancer types).

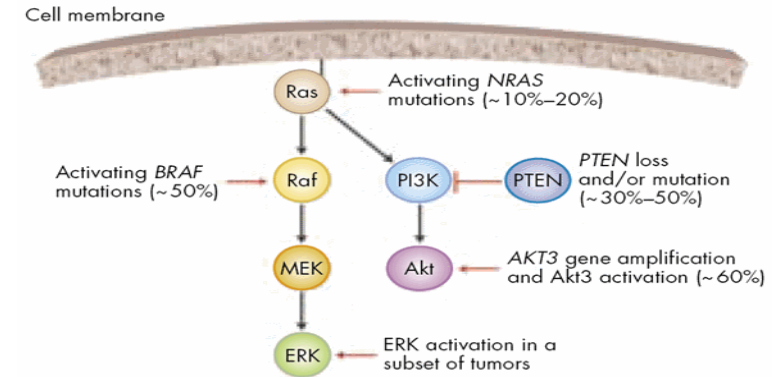
GENOTYPE/PHENOTYPE CLUSTERING

Histopathological Tumor Classification

Mucinous tumor of borderline malignancy	Mucinous intraepithelial carcinoma	Mucinous invasive carcinoma
Complex architecture	Basically borderline tumor	Expansile type or infiltrative type
No stromal invasion	No definite stromal invasion	Obvious invasion (>5mm)
Grade 1 atypia	Grade 2-3 atypia	
		

Traditionally, cancers have been described by their histopathological description, comprising organ and cell type of origin, e.g. ovarian clear cell carcinoma

Molecular Tumor Classification



However, with the advent of NGS-based profiling of tumors it is now common to determine the set of DNA variants that are present in a patient's tumor

- An individual's cancer can now be described by this molecular characterization, thus determining its vulnerability to specific targeted drug therapies
- What has emerged is the realization that:
 - Tumors with the same histopathological origin can have very different molecular variant profiles
 - Tumors with different histopathological origins can have similar molecular variant profiles

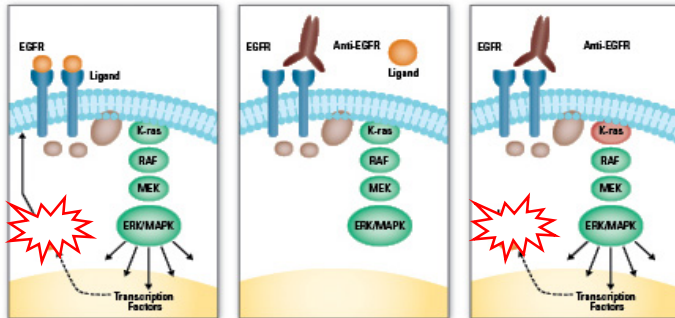
POTENTIAL USES

- Assess variant distribution across cancer types to aid in upstream target identification and downstream indication selection
 - Simplest case: examining frequency of individual variants relevant to targeted drug
- Assess variant co-occurrence across cancer types for development of combination therapies
 - More complex case: examining frequency of small combinations of variants
- Assess variant co-occurrence across cancer types for discovery of potential interactions to aid in patient stratification (e.g. EGFR/KRAS)
 - Most complex case: examining frequency of larger groups of variants, possibly pathway-based, to aid in rational, rather than serendipitous, discovery of clinically relevant interactions

VARIANT INTERACTIONS: EGFR/KRAS

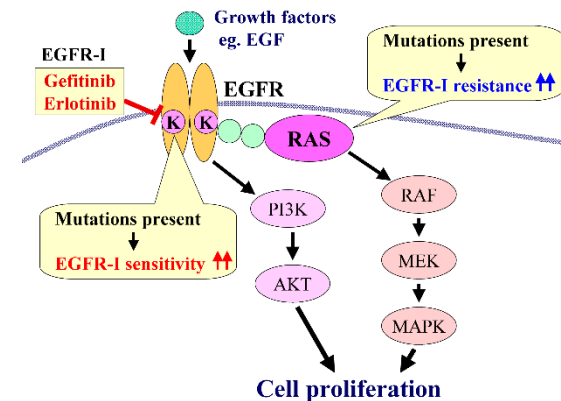
- EGFR activating mutations are drivers of many cancers and the target of several antibody and small molecule drugs (e.g. panitumumab, erlotinib)
- EGFR inhibition is much less effective in tumors with downstream KRAS mutations
 - This dependency on wild-type KRAS varies with cancer type:

Antibody treatment in colorectal adenocarcinoma



In CRC, EGFR protein (over)expression is seen in tumor cells and anti-EGFR antibodies have become standard of care. However, these drugs are not effective in tumors with KRAS mutations, and KRAS testing has become commonplace.

TKI inhibitor in NSCLC

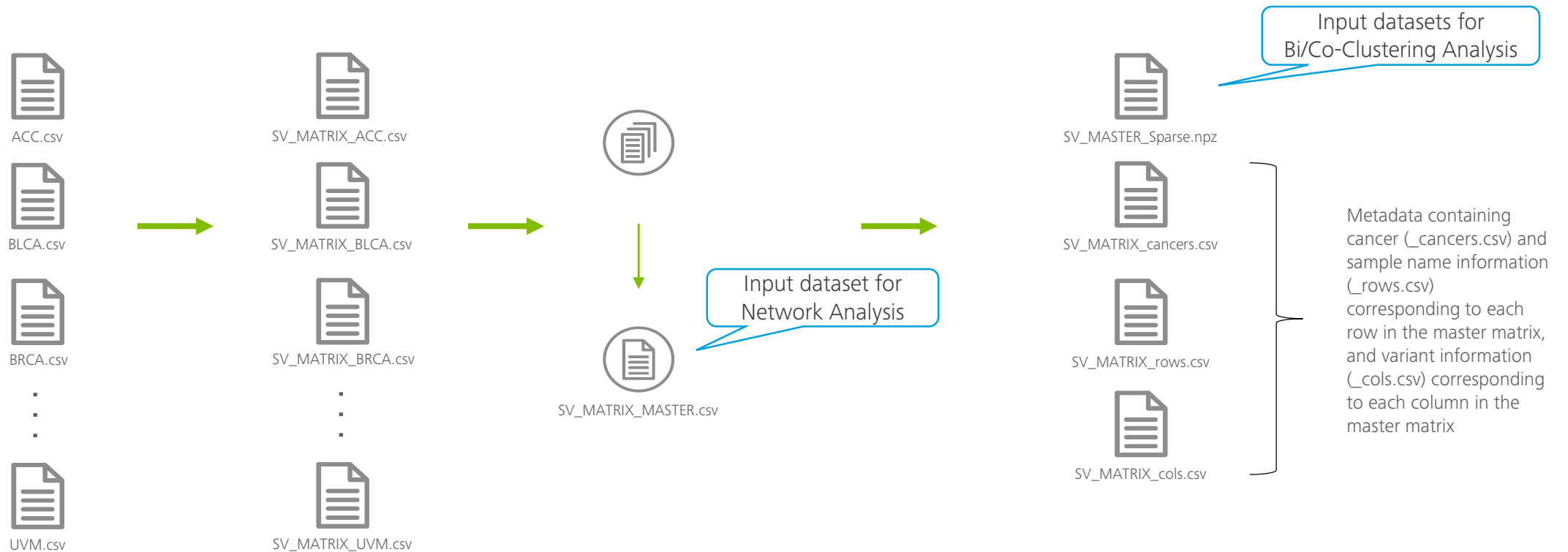


In NSCLC, EGFR mutations are seen in 15-20% of tumors, and TKI inhibitors are effective in these cases. In these tumors EGFR and KRAS mutations tend to be mutually exclusive, so role of KRAS testing is controversial.

DATA PRE-PROCESSING

TCGA data was collected (in this case, via GenePool) in 33 different CSV files, each containing information on one specific cancer type:

- Sample IDs: subject sample ID
- Variant (mutation) ID: denoted by the chromosomal location and REF – ALT variant information



- Each row unique to one variant
- All samples listed by row and separated by a semicolon

Each row corresponds to one given sample and variant pair

Data from 33 cancers **aggregated** into one csv file

- 3 columns: SAMPLE, CANCER, VARIANT
- 1,333,890 rows

- Each row corresponds to a sample: 8,174 total
- Each column represents a variant: 1,169,592 total
- Matrix entry is 1 if where variant V occurs in sample S, and 0 otherwise
- Master dataset represented in **sparse** matrix form
- Saved as npz file, in CSR file format (Compressed Sparse Row) in Python
- Can be converted to dense format

SUMMARY OF MASTER DATASETS

- Total of 8,174 samples, from individuals suffering from 33 cancer types
- 1,169,592 unique gene variants
- 1,333,890 unique sample/variant combinations

Summary Statistics

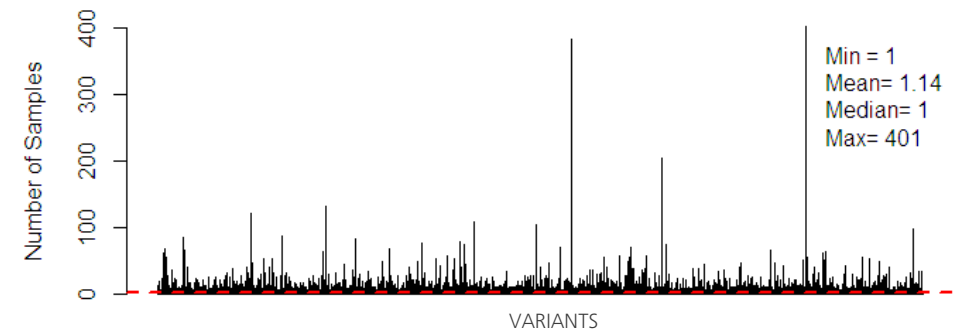
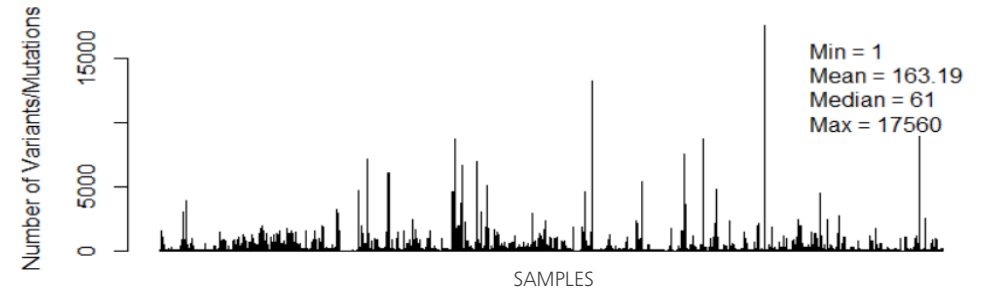
- Number of variants per sample
 - Min=1 | Mean = 163 | Median=61 | Max=17,560
- Number of samples per variant
 - Min=1 | Mean=1.14 | Median =1 | Max=401

Network Analysis Dataset

- Information in form of sample/variant pairs (which variants occur in which samples)

Bi/Co-Clustering Dataset

- Matrix entry is 1 where variant V occurs in sample S, and 0 otherwise
- Saved in **sparse** form as npz file, in CSR file format (Compressed Sparse Row) in Python
 - Information stored on which rows/column entries are 0, and which are not
 - Compressed format
 - Fast: takes ~10 seconds to load
- Can be converted to **dense** form
 - “Natural” matrix representation
 - Slow: can take more than 4h to load, and is difficult to store
 - Quicker to load sparse format, then covert to dense format
 - Metadata available to map cancers and samples to rows and variants to columns
 - Approximately 0.014% of the matrix is populated by 1s



BI/CO-CLUSTERING

Notebook name:
GENE_VARIANT_BI/COCLUSTERING

Python

What



- Algorithm that takes as input a (large) matrix and identifies sub-matrices with common patterns
- Rows and columns of data matrix are re-arranged to yield block diagonal or checkerboard matrix structures
- Simultaneously clusters rows and columns of a data matrix - clusters of rows and columns are known as bi-clusters

Why



- Fast – works with sparse matrices
- Easy to interpret and allows for investigation of association between sets of variants and samples
- Data-driven approach to translational research and hypothesis generation

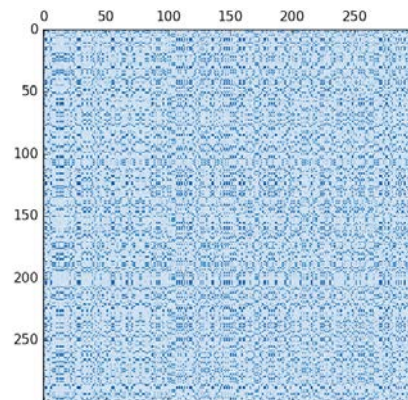
How



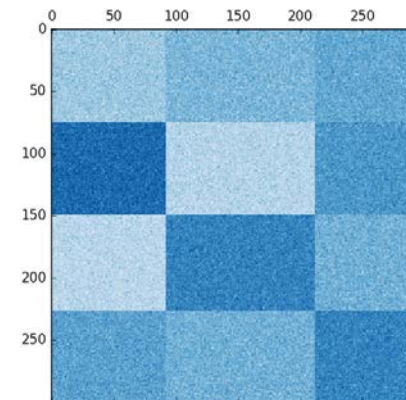
- Spectral clustering: dimensionality reduction via eigenvalue decomposition + k-means clustering
- Partitions the rows and columns of a matrix so that a corresponding checkerboard/block diagonal matrix provides a good approximation to the original matrix
- Documentation for the [sklearn.cluster.bicluster](#) module in Python

Co-clustering:
Block diagonal structure

Bi-clustering:
Checkerboard structure

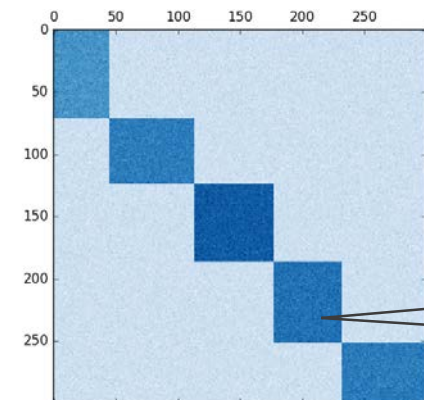


Original matrix



After bi-clustering:
checkerboard structure

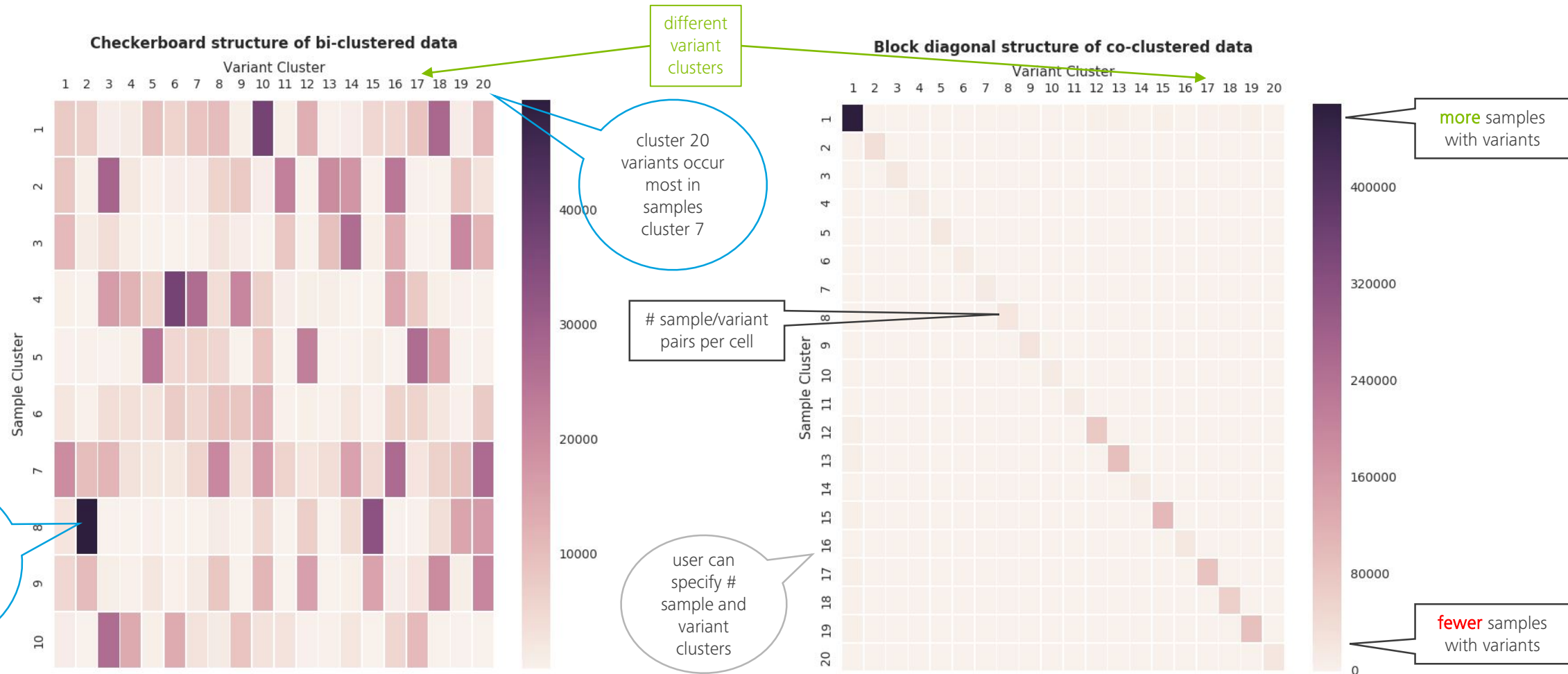
or



After co-clustering:
block diagonal structure

Cell represents
a co-cluster

BI/CO-CLUSTERING: READING OUTPUT



BI-CLUSTERING

- Imposes structure to the original, "shuffled," matrix
- Each row belongs to all column clusters, and each column belongs to all row clusters
- Assumes that the input data matrix has a hidden checkerboard structure.

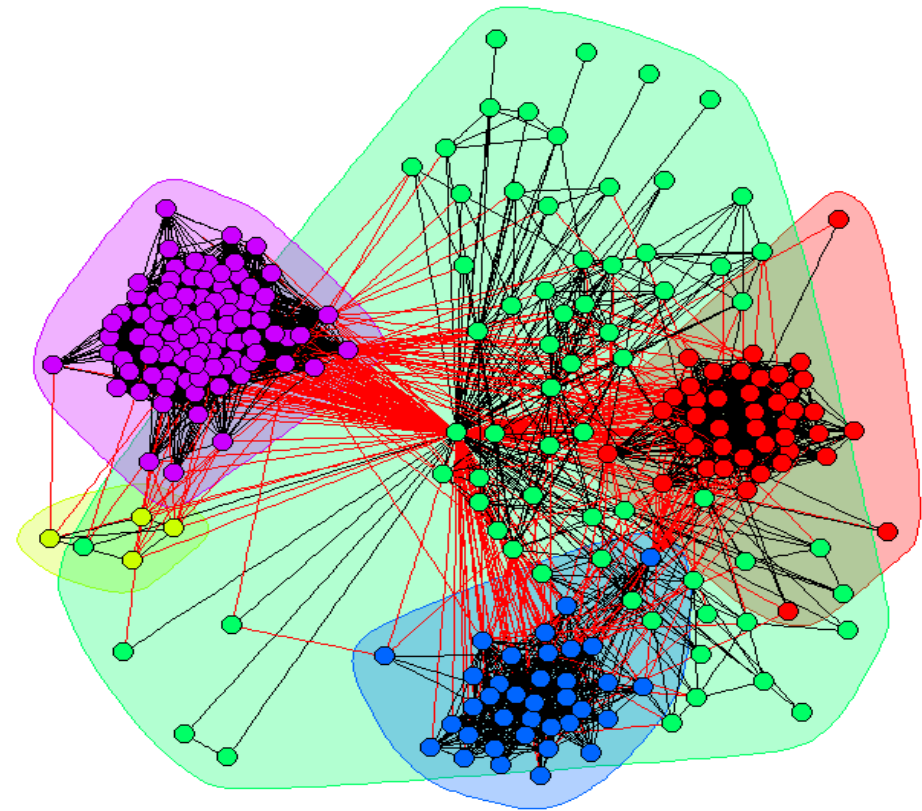
CO-CLUSTERING

- Imposes additional structure to the original, "shuffled," matrix
- Each row and each column to belong to one bi-cluster and one bi-cluster only
- As such, it finds biclusters with values higher than those in the corresponding other rows and columns

NETWORK ANALYSIS WITH COMMUNITY DETECTION

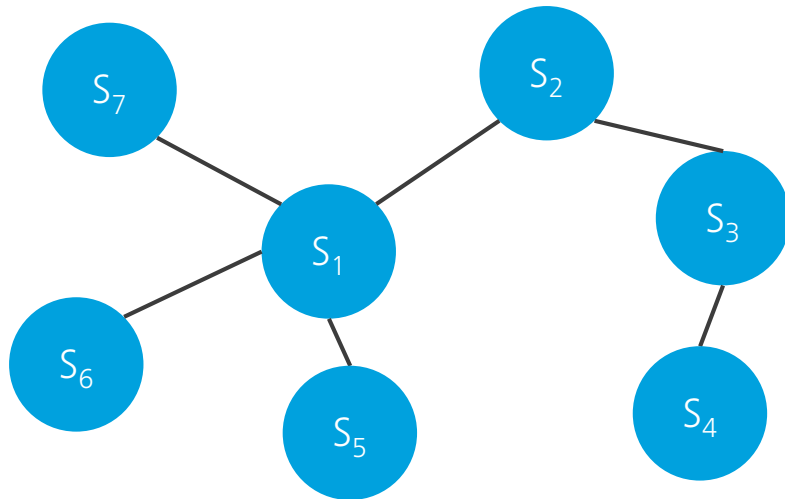
Graph Theory

- A bottom-up analysis is often required to **understand complex systems** such as biological systems where multiple individual components act through a complex interconnected web of molecular pathways.
- Network analysis provides a very **practical and intuitive approach** to investigate a system as a whole as well as individual components.
- This can be done by examining elementary constituents individually and how these are connected.
- For example if we need to investigate the association between multiple mutations/variants in samples and different cancer types, it makes sense to view this system as an interconnected network of samples connected by means of the presence or absence of a specific mutation/variant in them.
- Some of the fields where network analysis has potential applications are:
 - Drug target identification
 - Determining a protein's or gene's function
 - Designing effective strategies for treating various disease



GRAPH THEORY, SOME DEFINITIONS, AND COMMUNITY DETECTION

- A **graph** (or network) G can be defined as a pair of vertices (V) and edges (E) where V is a set of vertices representing the nodes and E is a set of edges representing the connections between the nodes.
- The **degree of a node** in an undirected graph is the number of connections or edges the node has to other nodes.
- **Degree Centrality** shows that an important node is involved in a large number of interactions. [Analogy: [hub](#)]
- **Closeness Centrality** indicates important nodes that can communicate quickly with other nodes of the network.
- **Betweenness Centrality** shows that nodes which are intermediate between neighbors rank higher. Without these nodes, there would be no way for two neighbors to communicate with each other. Thus, betweenness centrality shows important nodes that lie on a high proportion of paths between other nodes in the network. [Analogy: [network broker](#)]



S_1 has a degree of 4.
It has the highest **degree centrality**.

S_1 can access the other 6 nodes much easier (i.e., with shorter steps) than any other node.
As such, S_1 has the highest **closeness centrality**.

The number of shortest paths that pass through S_1 is the highest amongst all nodes.
As such, S_1 has the highest **betweenness centrality**.

A network is said to have a **community (or cluster) structure** if the nodes (a.k.a. vertices) can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally.

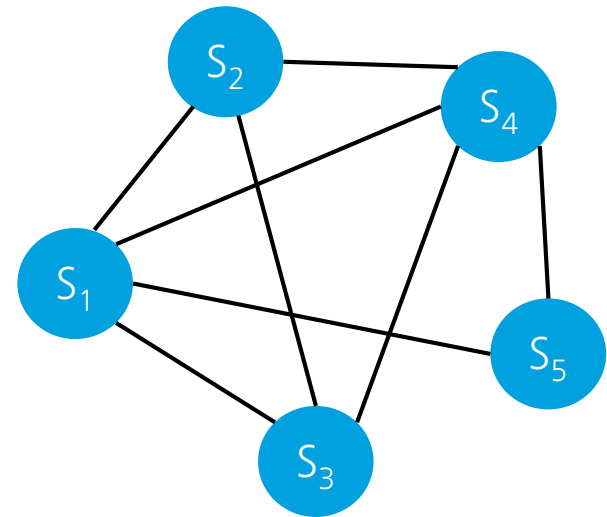
DATA PRE-PROCESSING FOR NETWORK ANALYSIS

- Generate all the binary combinations (for the edges in the graph) between samples based on the presence or absence of a common variant between them.
- That is, if a variant V1 is present in samples S1 and S2, there will be a connection between S1 and S2. If another variant V2 is present in samples S2 and S3, there will be another row indicating the connection between S2 and S3.

VARIANT_ID	CANCER_TYPE	SAMPLE
V ₁	ACC	S ₁
V ₁	ACC	S ₂
V ₁	GBM	S ₃
V ₁	GBM	S ₄
V ₂	LGG	S ₄
V ₂	GBM	S ₅
V ₃	GBM	S ₆

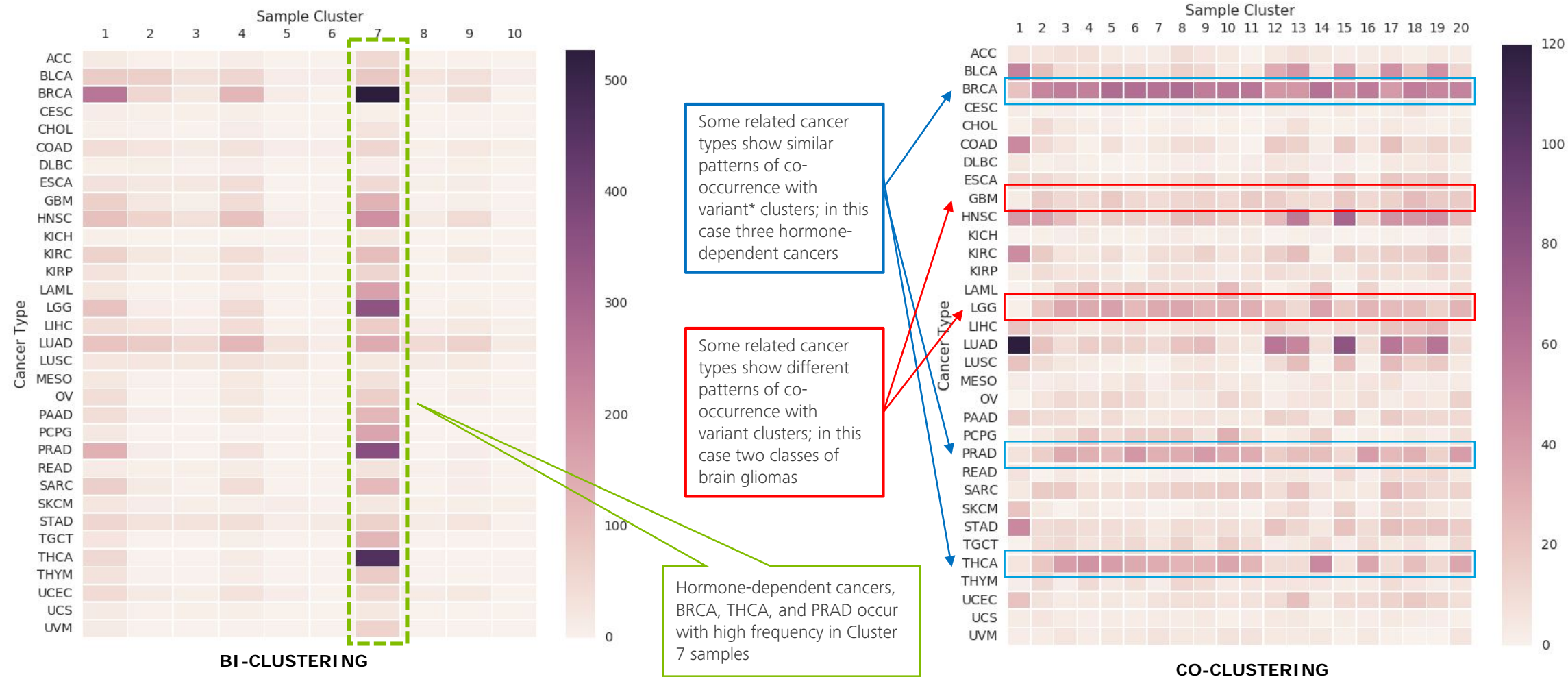


SAMPLE_01	SAMPLE_02
S ₁	S ₂
S ₂	S ₃
S ₃	S ₄
S ₁	S ₃
S ₁	S ₄
S ₂	S ₄
S ₄	S ₅



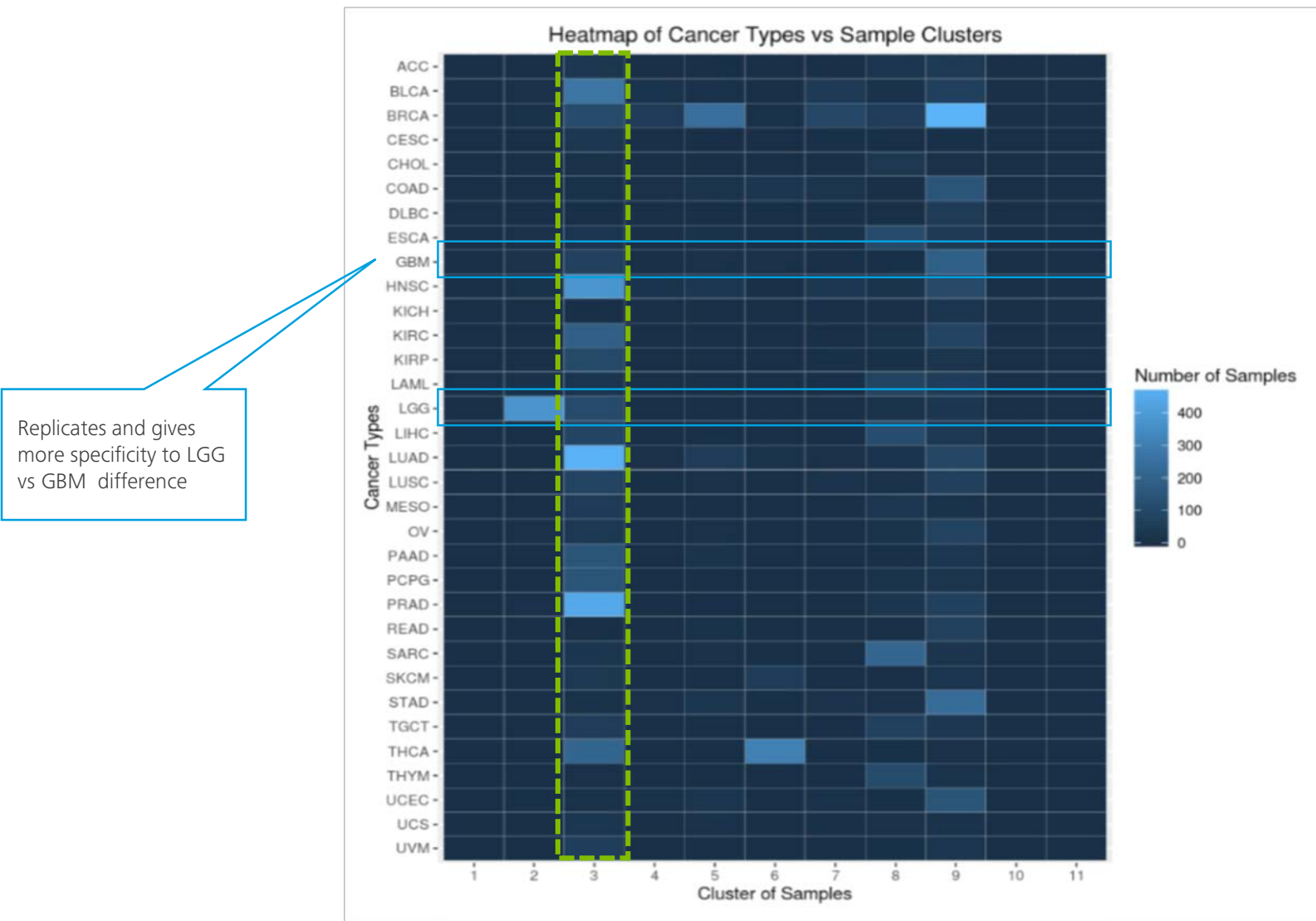
In the above example with the three samples alone, S2 is acting as a **connecting node** between S1 and S3. This **'influence'** of a node within the network can be found out using the **betweenness score** – see notebook for analysis.

CORRELATING CANCER TYPES TO VARIANT CLUSTERS: BI/CO-CLUSTERING



(*Above, 1:1 map between samples and clusters)

CORRELATING CANCER TYPES TO SAMPLE CLUSTERS: NETWORK ANALYSIS



APPENDIX

NETWORK DATA PRE-PROCESSING

- All the variants which occurred in more than 1 sample will generate pairwise combinations.
- How many rows will we then obtain in the pairwise dataset?
 - Let n be the number of variants which occurred in more than 1 sample
 - Let $nrow(V_i)$ denote the number of occurrences of a specific variant V_i across samples
 - Then the total number of rows in the output dataset of pairwise combinations of samples will be:
$$= \sum_{i=1}^n \binom{nrow(V_i)}{2}$$
 - This can act as a quality check to validate the output dataset from pre-processing
- Number of vertices = the total number of unique samples in the pairwise combination dataset → 8114

	SAMPLE_01	SAMPLE_02
1	S ₁	S ₂
2	S ₂	S ₃
3	S ₃	S ₄
....
.....
.....
n	S ₈₀₀₀	S ₈₁₁₄