Multi-omics Data Integration for Bladder Cancer Subtype Classification Using GATv2 Graph Neural Networks

Authors:

Haraja, Oussama
Grigoryev, Fedor
Komlev, Savelii
Martinez, Clara
Sharafutdinov, Emil

Supervisor:

Ph.D Zehraoui, Farida
IBISC Laboratory at Université D’Evry - Paris Saclay

08/11/2023

Glossary

Deep Learning: Subfield of machine learning that focuses on the use of artificial neural networks with multiple layers (deep neural networks) to model and analyze complex patterns in data. Deep learning algorithms, often referred to as deep neural networks, automatically learn hierarchical representations of data, enabling them to perform tasks such as image and speech recognition, natural language processing, and other sophisticated tasks.

Neural Network: Computational model inspired by the structure and functioning of the human brain. It consists of interconnected nodes (artificial neurons) organized into layers. Each connection between nodes has a weight that is adjusted during the learning process. Neural networks are employed in machine learning tasks, particularly in deep learning, to recognize patterns, make predictions, and perform various cognitive tasks.

Omics: Comprehensive approaches in biological and biomedical research that involve the study of various biological components or molecules within a biological system. Common examples include genomics (study of genes), transcriptomics (study of RNA transcripts), proteomics (study of proteins), and metabolomics (study of metabolites). The term "omics" is used to indicate a holistic and systematic investigation of biological information at a specific level.

Supervised Learning: Machine learning where the algorithm is trained on a labeled dataset, which means the input data is paired with corresponding desired output labels. The goal is for the algorithm to learn a mapping from inputs to outputs so that it can make accurate predictions or classifications on new, unseen data.

Unsupervised Learning: Machine learning paradigm where the algorithm is given unlabeled data and must find patterns, relationships, or structures within the data without explicit guidance. Unlike supervised learning, there are no predefined output labels, and the algorithm explores the inherent structure of the data on its own. Common tasks include clustering, dimensionality reduction, and density estimation.

Abbreviations

- GCN : Graph Convolutional Network,
- GNN : Graph Neural Network,
- NGS : Next-Generation Sequencing,
- GAT : Graph Attention Network,
- ML : Machine Learning,
- Ba/Sq: Basal/Squamous bladder cancer subtype,
- LumP: Luminal Papillary bladder cancer subtype,
- LumU: Luminal Unstable bladder cancer subtype,
- LumNS: Luminal Non-Specified bladder cancer subtype,
- NE-like: Neuroendocrine-like bladder cancer subtype.

Table of Contents

Introduction

Cancer is a complex genetic disease that is characterized by intricate interactions between genes and environments. Cancer subtyping is a crucial aspect for clinical practice, because it allows for tailoring treatment strategies to individual patients. The neoplasm can be addressed as a dysregulated molecular network (National Cancer Institute, 2021), therefore suggesting graph-based methodology for oncological research.

The introduction of next-generation sequencing (NGS) technologies has allowed for molecular profiling of cancer for better therapeutic decisions. The recent advances in the industry resulted in the generation of a vast amount of high throughput multi-omics data that helps the researchers and clinicians to have a broader and more holistic look at the patient's tumor. However, there is no golden rule for integration of multi-omics data. This problem holds pivotal importance in obtaining a comprehensive understanding of cancers and other complex diseases such as Alzheimer's and Parkinson's (Tanvir, 2023). As omics data provide a single view susceptible to noise and bias, the combination of multiple omics data types is necessary for accurate cancer prognosis prediction (Chai, 2019).

Major applications to use omics-data to predict cancer subtypes and patient categorization rely on graph-based learning models, and notably Graph Convolutional Networks (GCN). For instance, a recent study has integrated multi-omics data using graph convolutional networks for patient classification and biomarker identification for Alzheimer's disease patient classification (Wang, 2021)

However, existing Graph Neural Network (GNN) approaches have limitations that the graph attention network (GAT) model can overcome. GCN-based frameworks, though effective in extracting salient features from different omics data, struggle to determine the relative significance of neighboring samples in downstream analyses such as cancer subtype prediction and patient stratification (Tanvir, 2023).

This project presents GATv2, a novel multi-omics integration-based bladder cancer subtype prediction leveraging a Graph Attention Network (GAT) model that incorporates graph-based learning with an attention mechanism for analyzing multi-omics data. GATv2 introduces a multi-head attention mechanism, optimizing information extraction for individual patients through unique attention coefficients assigned based on pairwise metrics with their neighboring patients. In the comprehensive multi-omics integration, we encompass five distinct data types: protein expression, gene expression, miRNA expression, DNA methylation, and clinical data. This integrative approach aims to provide a nuanced and holistic understanding of bladder cancer heterogeneity for more accurate subtype predictions.

Part 1 : Literature Review

The challenge of multi-omics data

Understanding human health and diseases demands a nuanced and complex interpretation of molecular pathways at different levels. The confluence of multi-omics data, including the genome, epigenome, transcriptome, proteome, and metabolome, has led to the redefinition of the fields of medicine and biology. Indeed, the integration of multi-omics data, coupled with clinical information, has emerged as a crucial driver for extracting valuable insights into cellular functions, underscoring the importance of a comprehensive, system-level approach (Subramanian et al., 2020).

To develop a deeper comprehension of complex biological systems, it is crucial to identify intricate molecular relationships. Numerous studies have demonstrated the importance of integrating multi-omics data over single omics analysis (Subramanian et al., 2020). However, the promises of this integration come with inherent challenges, particularly in the effective integration of multi-omics data.

A key aspect in the integration of multi-omics data is navigating the interconnected equilibrium between an abundant number of variables and a relatively limited sample size. This central balance generates complexities, leading to confusion when identifying important patterns and associations. Recent advancements in deep learning techniques have shown promise in tackling these challenges, offering a pathway to unlock the full potential of multi-omics data integration (Chai et al., 2019). The application of these advanced learning methods presents a hopeful avenue to navigate the inherent complexity of multi-omics data, providing a more effective approach to uncover the underlying molecular signatures associated with cancer.

Traditional approaches

Machine Learning (ML) is a branch of Artificial Intelligence, uses the principle of inference to solve the problem of learning from data samples (Bishop, 2006). This learning-process is split into two stages:

(i) Estimation of unknown dependencies within a system from a given dataset.
(ii) Utilization of these estimated dependencies to predict new outputs of the system.

Despite the robustness of ML models, its models face challenges in handling the high dimensionality and complexity introduced by multi-omics data. In response, models utilizing graph-based learning have been proposed to extract hidden representations and network structures from various omics data types, enhancing applications such as cancer prediction and patient categorization.

Networks are universal descriptors of systems interacting elements. In biomedicine and healthcare, graphs represent various concepts and entities such as signaling pathways, molecular interactions or healthcare systems. Indeed, graph representation learning can contribute to the identification of genetic variants influencing complex traits, unraveling the intricacies of single-cell behaviors, and supporting patients in diagnosis and treatment decisions (Li, 2022).

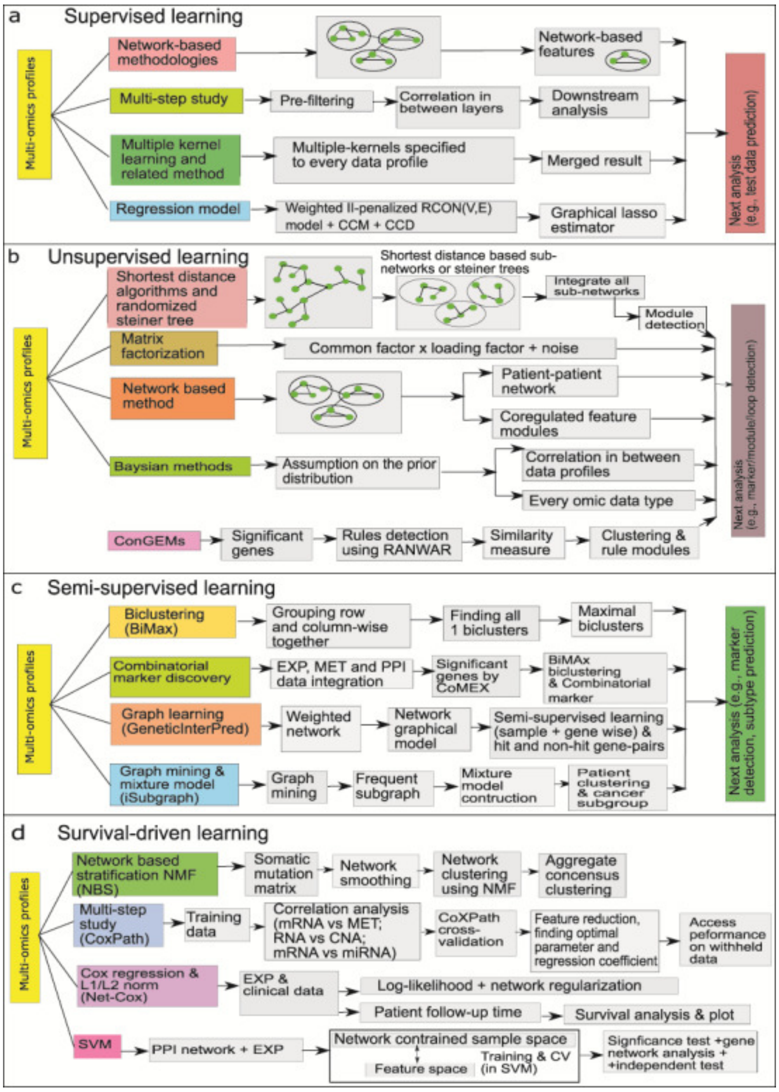

a
Supervised learning
Multi-omics profiles
Network-based methodologies
Network-based features
Multi-step study
Pre-filtering
Correlation in between layers
Downstream analysis
Multiple kernel learning and related method
Multiple-kernels specified to every data profile
Merged result
Regression model
Weighted ll-penalized RCON(V,E) model + CCM + CCD
Graphical lasso estimator
Next analysis (e.g., test data prediction)
b
Unsupervised learning
Multi-omics profiles
Shortest distance algorithms and randomized steiner tree
Shortest distance based sub-networks or steiner trees
Integrate all sub-networks
Module detection
Matrix factorization
Common factor x loading factor + noise
Network based method
Patient-patient network
Coregulated feature modules
Baysian methods
Assumption on the prior distribution
Correlation in between data profiles
Every omic data type
ConGEMs
Significant genes
Rules detection using RANWAR
Similarity measure
Clustering & rule modules
Next analysis (e.g., marker/module/loop detection)
c
Semi-supervised learning
Multi-omics profiles
Biclustering (BiMax)
Grouping row and column-wise together
Finding all 1 biclusters
Maximal biclusters
Combinatorial marker discovery
EXP, MET and PPI data integration
Significant genes by CoMEX
BiMAx biclustering & Combinatorial marker
Graph learning (GeneticInterPred)
Weighted network
Network graphical model
Semi-supervised learning (sample + gene wise) & hit and non-hit gene-pairs
Graph mining & mixture model (iSubgraph)
Graph mining
Frequent subgraph
Mixture model contruction
Patient clustering & cancer subgroup
Next analysis (e.g., marker detection, subtype prediction)
d
Survival-driven learning
Multi-omics profiles
Network based stratification NMF (NBS)
Somatic mutation matrix
Network smoothing
Network clustering using NMF
Aggregate concensus clustering
Multi-step study (CoxPath)
Training data
Correlation analysis (mRNA vs MET; RNA vs CNA; mRNA vs miRNA)
CoXPath cross-validation
Feature reduction, finding optimal parameter and regression coefficient
Access peformance on withheld data
Cox regression & L1/L2 norm (Net-Cox)
EXP & clinical data
Log-likelihood + network regularization
Patient follow-up time
Survival analysis & plot
SVM
PPI network + EXP
Network contrained sample space
Feature space
Training & CV (in SVM)
Signficance test +gene network analysis + +independent test

Figure 1: Overview of learning process with (a) supervised learning, (b) unsupervised learning, (c) semi-supervised learning and (d) represents survival-based learning. (Mallik, 2020)

In cancer classification and prognosis, Graph Convolutional Networks (GCNs) have particularly emerged as a powerful tool to gain valuable insights from complex biological data. For instance, the studies of Li et al. (2022) and Yin et al. (2022), exemplify the application of GCNs in cancer subtype analysis through the integration of multi-omics data.Li et al. (2022) introduced MoGCN, a novel cancer subtype analysis method using Graph Convolutional Networks (GCNs). MoGCN integrates multiple omics datasets, capturing intricate relationships with graph convolutional layers to identify distinct cancer subtypes. This exemplifies how GCNs enhance multi-omics integration, showcasing potential for personalized therapeutic strategies. Yin et al. (2022) proposed a molecular subtyping approach with a robust graph neural network, emphasizing GCNs' adaptability to diverse omics data. Their work illustrates how GCNs unravel complexities in cancer heterogeneity with enhanced precision.

However, Graph Convolutional Network (GCN) also exhibit notable drawbacks for cancer subtype prediction (Tanvir et al., 2023):

(a) Lack of consideration for various omics data
(b) Failure to estimate the relative significance of neighboring nodes (samples or patients)
(c) Generic approach for all different omics data.

A novel multi-omics integration approach : Graph Attention Network

Graph Attention Network (GAT) represents a pioneering multi-omics integration method that surpasses conventional feature-based, network embedding, and state-of-the-art Graph Neural Network (GNN)-based methodologies, such as signed graph convolutional network (SGCN). SiGAT, showcasing superior performance, substantiates these advancements through experimental evaluations on real datasets (Huang, 2019).

In contrast to Graph Convolutional Networks (GCN), GAT introduces a dynamic approach to information aggregation through attention mechanisms. The traditional GCN relies on static neighborhood aggregation as a "convolution" operation. In GAT, attention weights are dynamically learned, providing a nuanced perspective on the significance of specific neighbor nodes in influencing the target node (Eliassen, 2023).

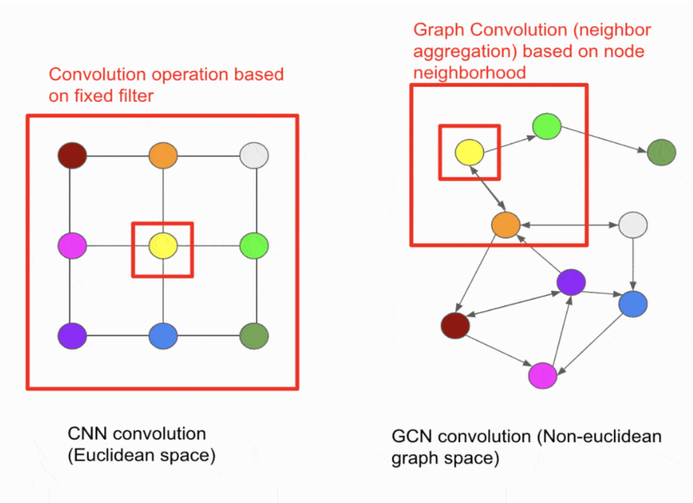
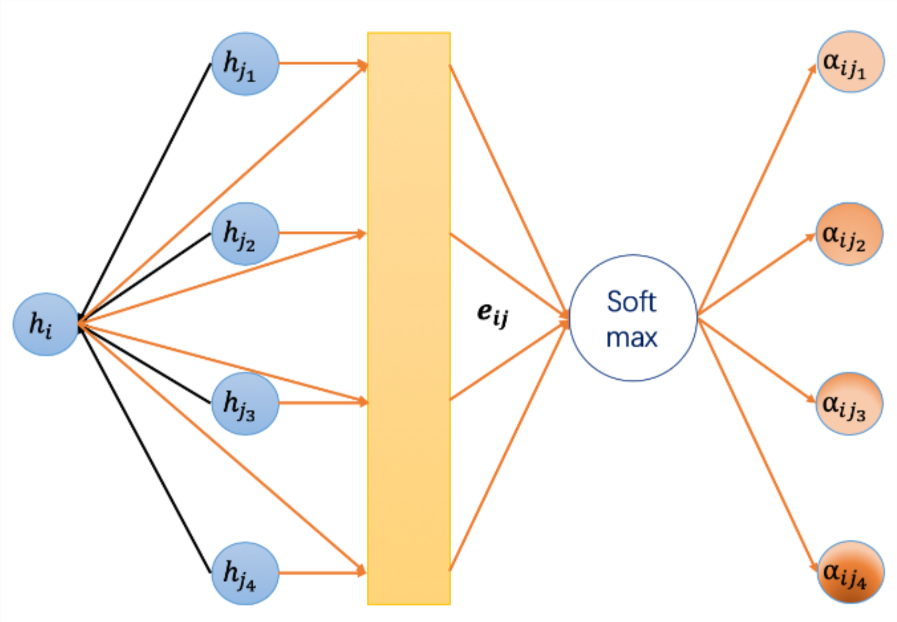

Convolution operation based on fixed filter
Graph Convolution (neighbor aggregation) based on node neighborhood
CNN convolution (Euclidean space)
GCN convolution (Non-euclidean graph space)

Figure 2: Comprehensive comparison between standard CNN and GCN operators. (Eliassen, 2023) This comparison emphasizes the distinctive information processing mechanisms inherent in GAT.

A crucial divergence between GAT and GCN lies in how they aggregate information from the one-hop neighborhood. Graph Convolutional Networks employ a graph convolution operation, producing the normalized sum of node features of neighbors. On the other hand, GAT introduces an attention mechanism, which dynamically assigns weights to the convolution weights based on the relevance of specific edges.

The equations (1) to (4) elucidate the intricate process involved in computing node embeddings in GAT, showcasing a unique approach to aggregation and normalization.



$$z_i^{(l)} = W^{(l)} h_i^{(l)}, \quad (1)$$

$$e_{ij}^{(l)} = \text{LeakyReLU}(\vec{a}^{(l)^T} (z_i^{(l)} || z_j^{(l)})), \quad (2)$$

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^{(l)})}, \quad (3)$$

$$h_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} z_j^{(l)} \right), \quad (4)$$

Figure 3: Graph Attention Network (GAT) Architecture and Equations. (Zhang, 2023)
Equation (1): represents a linear transformation of the lower-layer embedding using a learnable weight matrix (Huang, 2019).
Equation (2): computes pair-wise un-normalized attention scores between two neighbors utilizing an attention mechanism (Zhang, 2023).
Equation (3): applies softmax to normalize attention scores on each node’s incoming edges (Zhang, 2023).
Equation (4): Similar to GCN, aggregates embeddings from neighbors scaled by attention scores (Zhang, 2023).

GAT's innovative design, incorporating attention mechanisms for dynamic aggregation, distinguishes it from traditional convolutional operations, enhancing its ability to capture intricate relationships within multi-omics datasets.

In the exploration of Graph Attention Networks (GAT), Brody, Alon, and Yahav (2021) shed light on a crucial limitation inherent in GAT's attention mechanism. The authors elucidate that GAT computes a restricted form of attention, termed static attention, where the ranking of attention scores is unconditioned on the query node. This intrinsic limitation hinders GAT from expressing certain graph problems, demonstrating its inadequacy in fitting even controlled training data. To overcome this identified limitation, the authors propose a straightforward solution by modifying the order of internal operations, introducing GATv2. GATv2, a dynamic graph attention variant, emerges as a strictly more expressive model than GAT. The universal approximator attention function in GATv2 overcomes the static attention constraints of GAT, enhancing its capacity to tackle a broader range of graph problems. Notably, GATv2 exhibits superior performance compared to GAT across 12 Open Graph Benchmark (OGB) datasets and other benchmarks, showcasing heightened capabilities while upholding comparable parametric costs (Brody, 2021).

The essence of attention lies in its capacity to compute a distribution across a set of input key vectors when provided with an additional query vector. When an attention function consistently assigns equal or greater weight to one key over others, independent of the query, it is deemed static. The inherent limitation of static attention becomes evident as it fails to model scenarios where various keys hold distinct relevance to different queries, restricting its adaptability (Brody, Alon, & Yahav, 2021).

In the pursuit of overcoming this limitation, the introduction of a dynamic graph attention network involves a fundamental alteration to the internal operations of GAT, giving rise to GATv2 (Brody, Alon, & Yahav, 2021). GATv2 stands as a refined version of GAT, equipped with a notably more expressive attention mechanism. Addressing a critical flaw in the standard GAT scoring function (Equation (2)), GATv2 ensures a richer expressive capacity by applying a layer after the nonlinearity (LeakyReLU) and the W layer after concatenation. This adjustment essentially incorporates a multi-layer perceptron (MLP) to compute scores for each query-key pair, enhancing the overall dynamism and expressive power of the attention mechanism in GATv2 (Brody, Alon, & Yahav, 2021; Velicković et al., 2018).

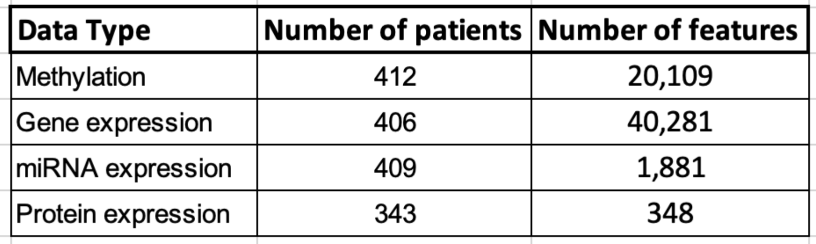
Part 2 : Materials and Methods

Data Preparation

The TCGA project focuses on in-depth cancer profiling and therefore generates multitude omics data for downstream analysis. We were dealing with public data on BRCA that consisted of four modalities of omics data for ~400 patients, as well as their clinical annotation.

The latter was a combination of two components. The first part included demographic details (age, ethnicity, gender, profession), results from basic physician assessments (height, weight, smoking habits), and certain historical data related to disease and treatment (age at initial pathologic diagnosis, history of neoadjuvant treatment). The second dataset specifically documented cases where patients were identified to have developed a new cancer subsequent to undergoing treatment for bladder cancer.

Table1: Data size



Data Type	Number of patients	Number of features
Methylation	412	20,109
Gene expression	406	40,281
miRNA expression	409	1,881
Protein expression	343	348

Surely, the integration of all these modalities is inherently challenging due to their high dimensionality - the feature space sizes of the high throughput methods are in the tens of thousands, while the sample cohort is limited to ~400 patients.

This puts us far away from the assumptions of the convex optimisation problem. Generally, in order for the machine learning algorithm to effectively generalize on the data we need the reverse situation: times more objects than variables.

Dimensionality Reduction

In order to achieve such a state we have implemented two consecutive dimensionality reduction strategies - after preliminary manual filtering of the most noisy features, we proceed with the use of a MLP-based autoencoder for each modality of data.

There are two principally different approaches to reduce the number of predictors in a dataset: feature selection and feature projection. Feature selection is the process of selecting a subset of relevant features for use in model construction. Feature projection transforms the data from the high-dimensional space to a space of fewer dimensions. The data transformation may be linear, as in principal component analysis (PCA), or nonlinear as in dimensionality reduction via autoencoders (Sarangi, Susanta, et al).

In this work we used both feature selection and projection. For feature selection we used differential expression (DE) analysis to identify the genes, which are over- or under-expressed between the samples of different cancer subtypes. For feature projection we implemented autoencoders. We chose autoencoders over more common techniques like PCA because autoencoders, being neural network-based models, have the advantage of capturing non-linear relationships in the data. And we expected OMICs data to have non-linearity.

Differential expression as biologically-relevant threshold

Gene expression analysis, one of the most widely used methodologies in modern biology, provides information about the transcriptional behavior of biological systems. It is a highly effective tool that is utilized in many fields of biology and medicine. It can be used to identify genes that are differentially expressed between tissues or between two or more biological conditions of interest (Skubitz KM and Skubitz AP, 2002), to carry out classification or discrimination analysis in heterogeneous diseases such cancer (Bhattacharjee et al., 2001), to comprehend the relationships connecting gene profiles and covariates such as survival or the aggressiveness of tumors (Veer et al., 2002; Beer et al., 2002), to discover new drugs or optimize their production (Clarke et al., 2001; Pagliarulo et al., 2002; Johnson et al., 2002), to diagnose diseases (Heller et al., 1997), and to tailor therapeutics for particular pathologies (Thiery et al., 2006).

We performed DE analysis using the python version of DESeq2 (Love et al., 2014), PyDESeq2 (0.4.3) [https://pydeseq2.readthedocs.io/en/latest/] (Muzellec et al., 2023). DESeq2 is a widely used and robust tool designed specifically for identifying differentially expressed genes in expression data, and it is designed to account for the inherent variability of the data, adjusting sample-to-sample variation and gene-specific biological variations. DESeq2 applies a negative binomial distribution model to account for the inherent variability, taking library size and gene-specific dispersions into consideration. Additionally, the tool employs a generalized linear model approach with shrinkage estimation of dispersion. Using the DESeq2 statistical framework, differential expression analysis was conducted on the normalized counts. Adjusted p-values were assessed with multiple testing using the false discovery rate (FDR) method. Additionally, Log2(FoldChange) is calculated to give an idea how the features were differentially expressed (over- or under-expressed).

PyDESeq2 provided us with a way to reduce the number of features in gene expression and methylation data based on significant statistical values, adjusted p-value and Log2(FoldChange). This will give our dimensionality reduction more interpretability and biological significance.

PyDESeq2 applies the differential expression analysis (DEA) approach developed by Love et al. (2014). In essence, this involves modeling raw counts through a negative binomial distribution. Initially, dispersion parameters are individually estimated for each gene by employing a negative binomial generalized linear model (GLM). Subsequently, these dispersions are adjusted towards a universal trend curve. The obtained dispersions are then utilized to model log-fold changes (LFC) on a per-gene basis between cohorts and to conduct Wald tests for assessing differential expression.

Before running the DEA on the gene expression data (40,281 features) and methylation data (20,109 features) to reduce the number of features based on significant statistical values, we had to preprocess the data in a way to make them compatible with the tool. PyDESeq2 only takes counts that are presented as integers. The data we have is float, and since it is normalized we can only multiply the whole count table by the same factor (x1000000). The data was then filtered on features presenting differences in expression between samples, which was done by excluding the columns that have a sum of counts of all samples for the same feature equal to 0. The data was then transformed from float to integer values. Those steps will not influence the analysis that much since the count will be renormalized again with DESeq2 and then compared. We also filtered the labels to have the same samples in the count files as in labels and to have the same column IDs in both labels and count files. It was done through merging both files based on IDs.

We used `DeseqDataSet` to create a dds object that contains the count of the features and annotations. The tool estimates the dispersion and log fold-change (LFC), it will fit the model based on comparison between samples taking into consideration the groups of samples that are determined by the classes provided to the tool in the labels. Since we ran the analysis on all the cancer subtypes, the tool executes a paired wise comparison between classes which we should extract from the object one by one. The output of the summary of `DeseqStats` on dds object is adata (annotated data), which is impossible to process, therefore, the pydeseq2 packages contains the command `results_df` that enables us to transform it into a computable dataframe. We then extracted the features that are differentially represented between the samples in all the paired-wise comparison by filtering based on absolute LFC (>10 for gene expression data and >2 for methylation count) and the adjusted P-values (< 0.01), the list of features obtained will be then used to reduce the initial count tables.

AutoEncoders

Each modality of data has been independently transformed to a lower-dimensional space with the use of MLP-based autoencoders, figure 4 (Song et al 2021), using pytorch (Paszke 2019). The general architecture of the autoencoders was the same and consisted of several fully connected layers. However, to match the heterogeneity in data, the hyperparameters of the models have been tuned independently, on a validation subset of the sample. The higher-dimensional data, necessitating more complex autoencoders (with the number of weights in the first and last FC layers proportional to the dimensionality), naturally led to the challenge of overfitting. To address this, we introduced regularization to the model by incorporating Dropout layers. Additionally, for enhanced learning effectiveness and stability with RNAseq data, we implemented a learning rate decay.

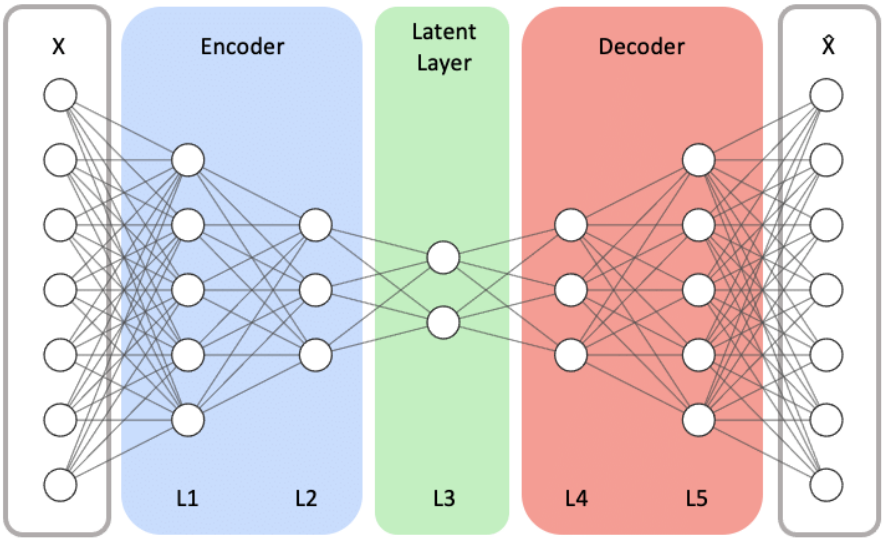


Figure 4: Autoencoder Architecture (Zhang, 2023)

Alternative approach to dimensionality reduction for RNA data is using pre-trained autoencoders and fine-tuning them on the available data, and there indeed exist some implementations, for example trained on TCGA pan-cancer metacohort. However, in our project we have decided to keep a more DIY-approach to the problem.

An important note on our data preparation is also our approach to tackle the absence of data across several modalities. The most common approach in the literature is a simple subsetting the sample by filtering out the patients that lack some of the modalities (Tanvir, 2023). However, in our case, that is severely detrimental to the training sample size as 15% of the samples lack protein expression data. In order to overcome this, we had to fill the absent protein data with median expression values for every protein. This surely simplifies the real data distribution and will result in the loss of an accuracy, but on the other hand, allows for incorporation of incomplete multi-omic samples and we prioritize the usability and applicability.

Graph Attention Network Protocol

General approaches and software

For this task we decided to proceed with the pytorch implementation of GATv2 operator from the paper “How Attentive are Graph Attention Networks?”. We used **`torch`** in this project as it is a widely utilized framework and is convenient to build a model. Many other frameworks are built on top of it, for example, we were able to feed a torch model to a network visualization tool to see the graph. We use **`sklearn`** (Pedregosa 2011) to compute metrics measures the model and to compute similarity indexes between the patients.

We use **`pyvis`** (Perrone 2020) and **`plotly`** (Plotly Technologies 2015) as a network visualization tool. This tool creates interactive graphs, so it is playable for the user and provides more context to the network. For other visualization tasks, such as plots and heatmaps, we use matplotlib and seaborn

Model architecture: GATv2 for multi-omic integration

For the GATv2 model the architecture remained the same for all 4 models, consisting of two graph attention layers, which was suggested in the literature (Tanvir 2023). The hidden layer dimension for each GAT model and learning rate were selected based on grid search-based hyperparameter tuning. Each layer was wrapped with ELU activation function which tends to converge cost to zero faster and produce more accurate results in comparison to ReLU. Dropout régularisation technique was used for both GATv2 layers to prevent overfitting.

The resulting embeddings from all the models were concatenated and forwarded through two linear multilayer perceptrons with also ELU as activation function and dropout as regularization technique. Raw logits were then in the output for the CrossEntropyLoss function in the output.

Gradients then propagated from last perceptron layers across all GATv2 heads allowing us to train all 4 models simultaneously and work on result improvement at the same time.

Model Training and Validation Strategy

Addressing Class Imbalance During Training

Weighted multi-class Cross Entropy Loss was used in order to train the model. For pytorch implementation, weight (*Tensor*) – a manual rescaling weight given to each class. If given, it has to be a Tensor of size C, where C is the number of classes. Weights for each class were computed as N / (M + N/70), where N is total number of samples, M is number of a specific class. N/70 was added to reduce dominance of imbalanced classes.

Pairwise metrics

Gaussian kernel was used to compute similarity indexes between patients based on clinical data.

$$k(x, y) = \exp(-\gamma\|x - y\|^2)$$

We were trying to use other kernels available in sklearn as well, but after autoencoders part some of the features became negative, which made them non fedable to other indexes. **`similarity_df`** was produced for all clinical data of all pairs of patients.

We then plotted the distribution of the similarity indexes and noticed that most of them are below 0.6 threshold (out of 1.0). To avoid severe overfitting we needed to keep the connection density low, but keep the number of edges for the most sitmilar patients. Based on the graph below we believe it was a reasonable decision to keep only nodes, which had a similarity index above 0.9, so we proceeded with it.

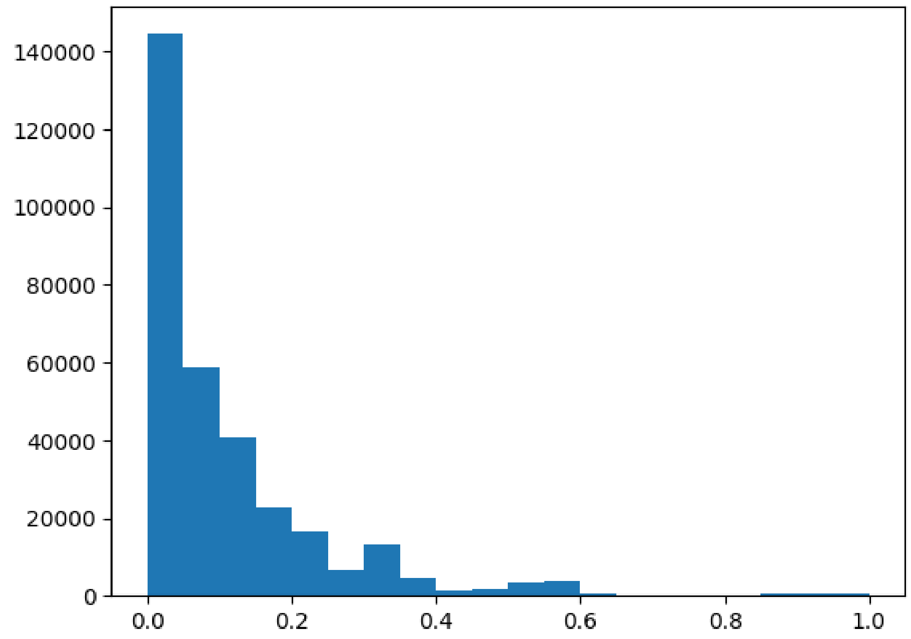


Figure 5: Similarity indexes distribution

We use multi-headed GATv2 architecture. We will come back to it, but for now it is important to mention that each head of the model is responsible for one modality from our datasets. We had protein expression, gene expression, methylation and miRNA expression datasets.

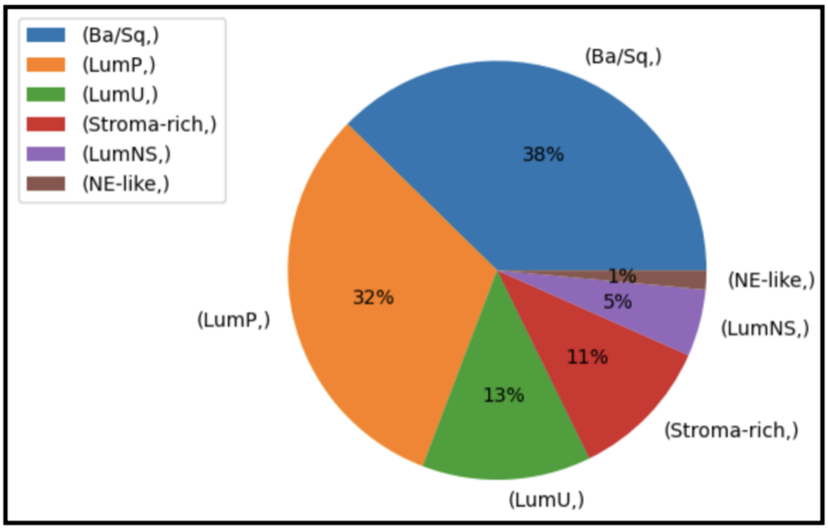
For each dataset a separate graph was created. All graphs had the same topology, meaning they all had the same edges and the same number of nodes - patients. The only difference was in the types of data that was attached to the node. For the protein expression, each node contained protein expression data for the specific patient, for miRNA expression it was miRNA expression for the patient and so on.

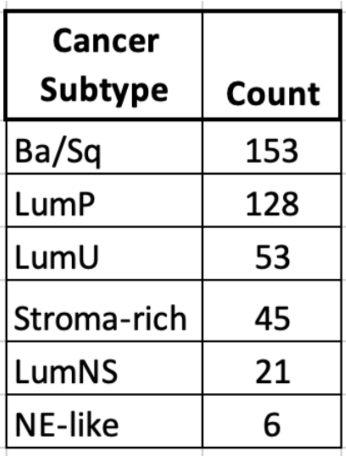
Number of classes was 6, so it was a multi-class one-label classification problem with 6 classes - for different types of cancer. Each node - each patient - had a label. Important to note here that the class imbalance is significant in the given dataset.

We split our dataset in two portions - train and test, where 89% of patients went to the train part and the rest went to the test.

Addressing severe class imbalance at validation

The dataset analyzed in this work hosts severe class imbalance. We can observe on Figure 6 that the class NE-like account for only 6 samples in the dataset.





Cancer Subtype	Count
Ba/Sq	153
LumP	128
LumU	53
Stroma-rich	45
LumNS	21
NE-like	6

Figure 6: Addressing the class imbalance

To address this issue we used performance metrics, which are not sensitive to class imbalance: precision, recall (sensitivity), F1-score.

Precision is the fraction of relevant instances among the retrieved instances. In the context of cancer subtypes classification, high precision indicates that when the model predicts a specific cancer subtype, it is likely to be correct. High precision is crucial when false positives (incorrectly predicting a subtype) may have significant consequences, such as unnecessary treatments or interventions.

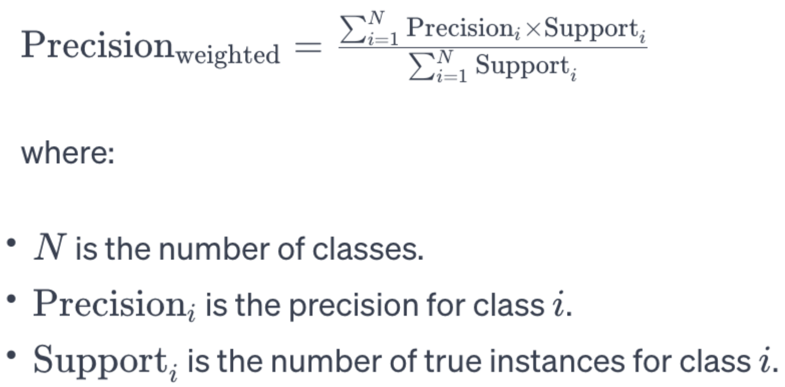
Recall (sensitivity) is the fraction of relevant instances that were retrieved. A high recall indicates that the model is effective at identifying instances of a specific cancer subtype. Considering recall is important in cases where false negatives (failing to predict a subtype when it is present) can have serious implications for patient outcomes.

F1-score is the harmonic mean of precision and recall intended to give a balanced single-value estimation of a model's performance.

Since those metrics were developed to estimate classification accuracy on binary tasks, we had to adapt them to our multiclass problem. The most common way to do this is to calculate binary scores for each class separately in a “one vs all” fashion (estimate the ability of a model to discriminate one class from all others). Then the scores for each class are averaged to obtain a single value for all the classes.

Because some of the classes like “Ne-like” or “LumNS” were extremely underrepresented, we used a weighted averaging strategy to obtain the overall scores for each model. This strategy involved assigning a weight to the score for each class, which was proportional to the number of samples supporting this class in the test set. This way the total scores of a model were penalized less for errors in underrepresented classes (Pedregosa, F., et al).

Example of weighted precision calculation:



$$\text{Precision}_{\text{weighted}} = \frac{\sum_{i=1}^{N} \text{Precision}_i \times \text{Support}_i}{\sum_{i=1}^{N} \text{Support}_i}$$

where:

- N is the number of classes.
- Precision_i is the precision for class i.
- Support_i is the number of true instances for class i.

Weighted recall and F1 scores were calculated in the same way.

Hyperparameter optimization with Tree-structured Parzen Estimator

TPE, or Tree-structured Parzen Estimator, is a Bayesian optimization algorithm commonly used for hyperparameter tuning. It belongs to the category of sequential model-based optimization (SMBO) algorithms. The main idea behind TPE is to model the objective function using a probabilistic model and iteratively refine this model to guide the search towards promising regions in the hyperparameter space (Bergstra, James, et al).

Part 3 : Results

Dimensionality Reduction

Differential Expression

The DE analysis helped us reduce the number of features significantly in this work. We were able to reduce the genes from 40281 to 3158 (7,84%) that are significant. Additionally, we reduced the methylation features from 20109 to 4509 (11,19%). Biologically, these features are considered potential biomarkers for the subtypes and the ones controlling the development of the bladder cancer into one of the subtypes.

We constructed UMAPs of both gene expression and methylation annotated data using PyDESeq2 (Figure 8). Obtained clusterization does not separate all the cancer subtypes from each other equally well: some subtypes are mixed. That is probably due to the fact that, biologically, those samples still belong to the same cancer, therefore, they have much in common. Additionally, the UMAPs were constructed based on a strictly reduced number of features, which could have resulted in information loss. Furthermore, differentially expressed features were selected based on pairwise comparisons between different subtypes, and not in a one-vs-all fashion.

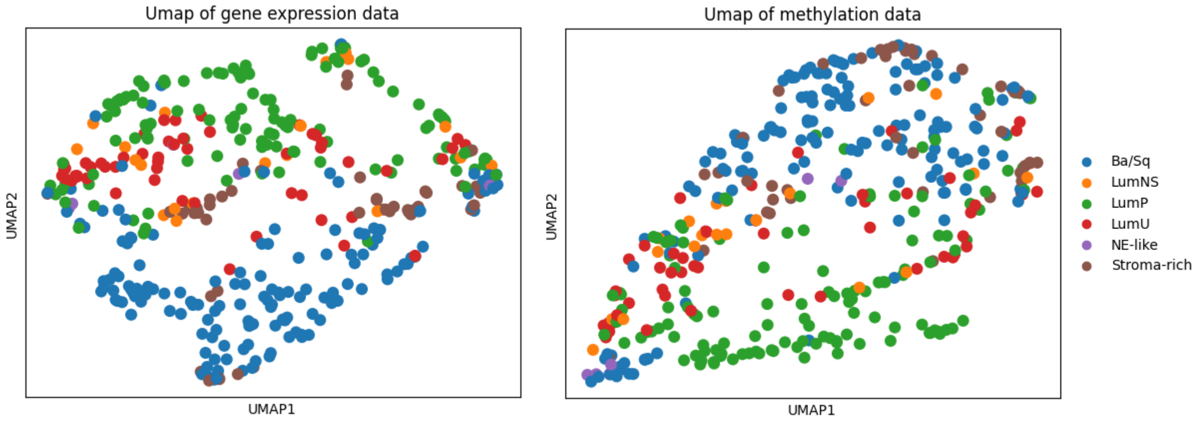


Figure 8: Umap on Gene expression and methylation data.

We can see that LumNS, LumP and LumU mostly cluster with each other separately from Ba/Sq. Two most abundant subtypes Ba/Sq and LumP are clearly separated in both modalities. Stroma-rich subtype clustered with Ba/Sq in the Umap of methylation data, but not for gene expression.

Autoencoders

Four autoencoders were used independently for dimensionality reduction. The example shows the training curve on reducing the dimensions of omics data can be seen on Figure 9. The same procedure has been used throughout all the modalities.

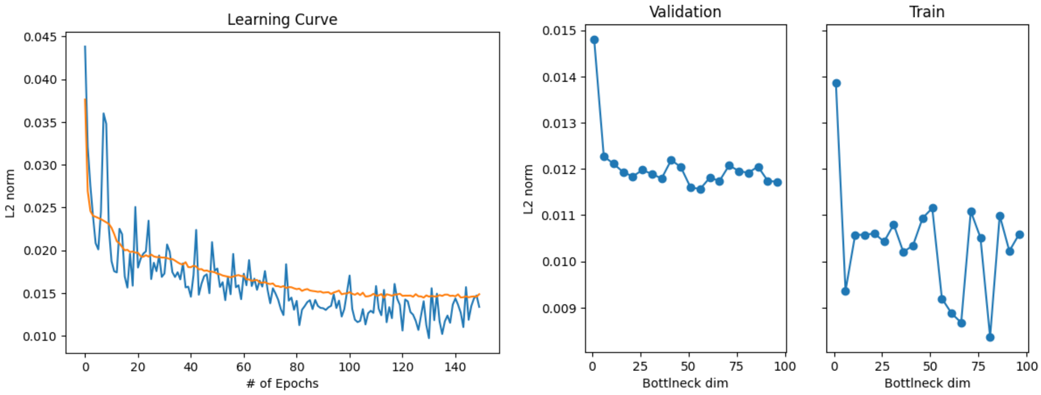
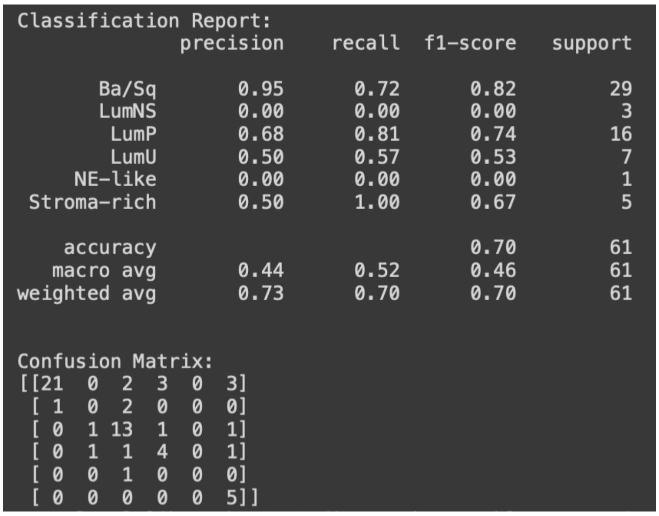


Figure 9: Autoencoder training process
In the provided example we pick 22 as a bottleneck output dimensionality hyperparameter.

We also can comparatively assess the dim reduction quality (Figure 10) by comparing the outputs of some predictor model on them (in our case we utilized xgboost classifier with the default parameters).



```
Classification Report:
              precision    recall  f1-score   support

       Ba/Sq       0.95      0.72      0.82        29
       LumNS       0.00      0.00      0.00         3
        LumP       0.68      0.81      0.74        16
        LumU       0.50      0.57      0.53         7
     NE-like       0.00      0.00      0.00         1
 Stroma-rich       0.50      1.00      0.67         5

    accuracy                           0.70        61
   macro avg       0.44      0.52      0.46        61
weighted avg       0.73      0.70      0.70        61


Confusion Matrix:
[[21  0  2  3  0  3]
 [ 1  0  2  0  0  0]
 [ 0  1 13  1  0  1]
 [ 0  1  1  4  0  1]
 [ 0  0  1  0  0  0]
 [ 0  0  0  0  0  5]]
```

Figure 10: Embedding quality assessment

Model Training

Training and Validation process

We used 400 epochs to train models with different modalities and other hyperparameters. At each step loss function for train and test datasets was computed and put on the graph to represent the learning curve. F1 score was selected as a main score and we kept the model with the best F1 performance within the loop of 400 epochs. This model then was saved as the most performative one. Best hyperparameters were selected using grid search in the hyperparameter space based on best F1-score on the validation dataset.

Average class representations (probabilities for each class) of the best model (all modalities integrated):

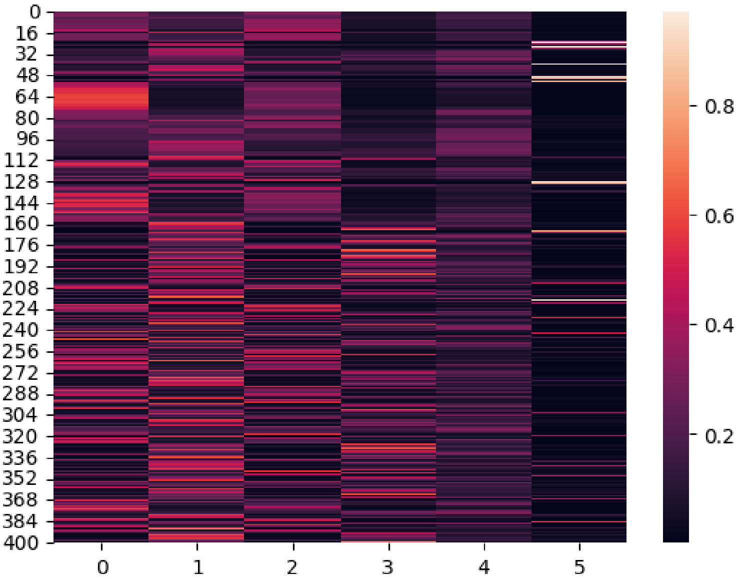


Figure 11: Integrative Model probability scores output

The most probable classes were taken as ones that were predicted by the model. Labels from 0 to 5 represent the targeted cancer subtypes in the order of decreasing abundance.

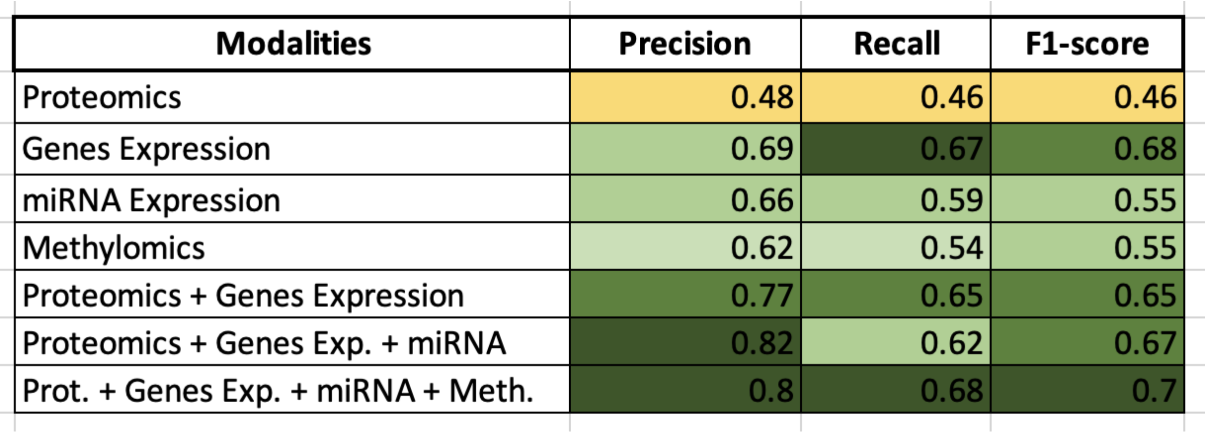
The averaged performance scores of the integrated model:

```
F1 score: 0.7026798840952391
Precision score: 0.80179438522074
Recall score: 0.6829268292682927
```

Different omics modalities’ integration results in non-linear shifts in classification quality

We inspected how integrating different OMICs data modalities into the model one by one affected the model’s performance. From a ML perspective, a particular combination of modalities can be viewed as a hyperparameter.

Table 2: Effect of integration of modalities



Modalities	Precision	Recall	F1-score
Proteomics	0.48	0.46	0.46
Genes Expression	0.69	0.67	0.68
miRNA Expression	0.66	0.59	0.55
Methylomics	0.62	0.54	0.55
Proteomics + Genes Expression	0.77	0.65	0.65
Proteomics + Genes Exp. + miRNA	0.82	0.62	0.67
Prot. + Genes Exp. + miRNA + Meth.	0.8	0.68	0.7

Single Modality Models

Looking at each modality separately (Table 2), Proteomics data gives the lowest results across all three metrics, while Transcriptomics gives the best. Performance on miRNA or Methylomics is in between the aforementioned modalities. Poor model’s performance on the protein expression data exclusively could arise from several causes. Firstly, only 348 proteins’ expression level was measured compared to the thousands and tens of thousands of predictors available for other modalities. Secondly, and probably more importantly, the protein dataset had significantly less samples for training than the others. Best performing Genes Expression model, on the contrary, had 40.000+ predictors and 16% more samples to train on. Apart from that, we know that a major proportion of cancer heterogeneity arises from the differences in gene expression (Cancer Genome Atlas Network, 2012).

Multiple Modalities Models

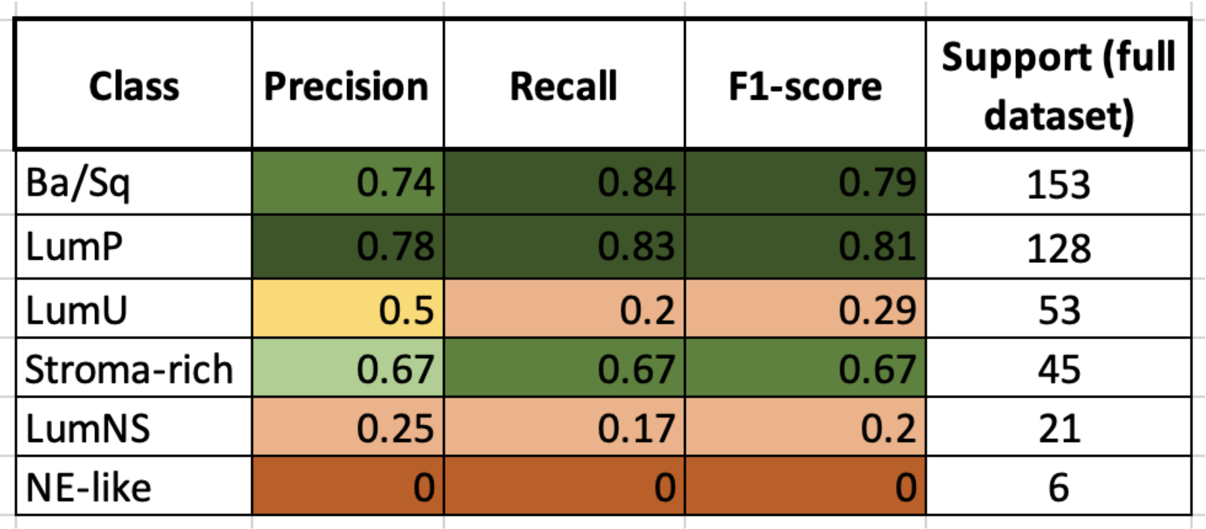
In comparison to the best single-modality model (Gene Expression) the integration of Proteomics and Transcriptomics data increases the precision by 8% at the cost of reducing the recall by only 2% and the class-weighted F1-score by 3% (Table 2). The overall positive change shows that integrating those two types of OMICs data gives new information about the cancer subtypes, and thus increases the precision of classification. However, apart from adding new information, the integration also presents more noise to the model, which is reflected on the recall drop. Expanding the model further by adding the miRNA modality pushes precision to the highest observed value of 82%. It again happens at the cost of the model's sensitivity, but this time the balanced F1-score goes up almost to the level observed for single RNA-seq modality. So far, adding new modalities allowed the model to be more confident when it predicts a cancer subtype for a specific patient (higher precision), but less sure that it identifies all the samples of a specific subtype in a dataset.

Notably, adding Methylatilomics to the model, and thus integrating all the available data modalities results in the most balanced model with highest recall, F1-measure, and almost highest precision. It seems that presenting methylation data allowed the model to both recover the sensitivity lost after adding Proteomics and miRNA on top of Transcriptomics and at the same time preserve the precision induced by modalities integration.

Classification quality depends on cancer subtype

We looked at how well the model performs at classifying each cancer subtype.

Table 3: Classification quality



Class	Precision	Recall	F1-score	Support (full dataset)
Ba/Sq	0.74	0.84	0.79	153
LumP	0.78	0.83	0.81	128
LumU	0.5	0.2	0.29	53
Stroma-rich	0.67	0.67	0.67	45
LumNS	0.25	0.17	0.2	21
NE-like	0	0	0	6

Model performs best on the two most abundant classes: Basal and Luminal Papillary bladder cancers, which is not surprising (Table 3). Interestingly, the linear dependency between the classification quality and the number of supporting classes fades in the case of LumU and Stroma-rich subtypes. For those subtypes the quality is far better for the Stroma-rich class, despite the LumU having a bit more samples. The poor performance for the LumU class might be due to the severe genomic instability of this subtype.

LumNS and NE-like show the worst performance. For NE-like subtype zero scores could be almost entirely explained by the scarcity in samples: only 6 samples were available for both training and validation. For LumNS poor classification quality might be not only due to the lack of training examples, but also because this subtype is an abstraction composed of unspecified samples, which could probably be distributed to other categories.

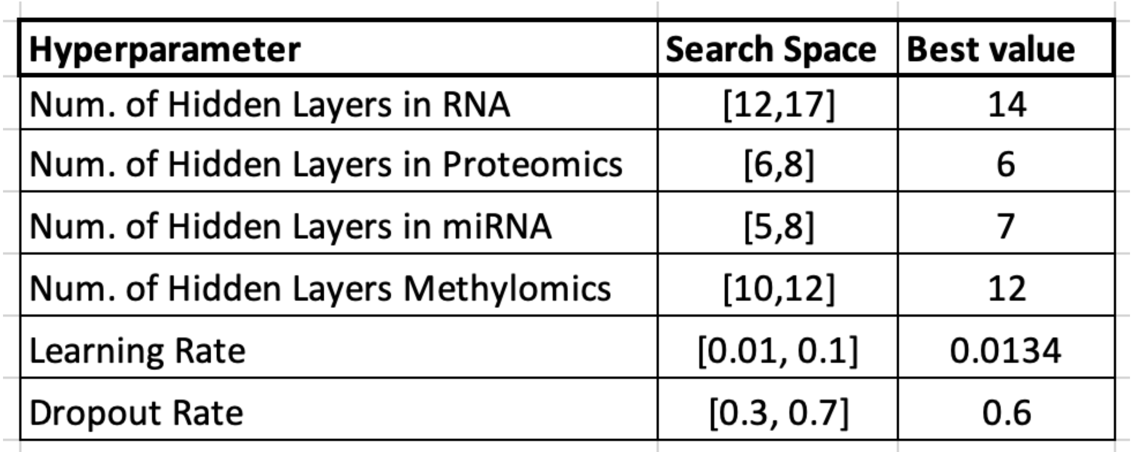
The aforementioned observations correspond well to the UMAP plots we built for differential expression and methylation analysis. The cancer subtypes, which were grouped in separate clusters on UMAP, are the ones that achieved good classification quality scores: Basal and Luminal Papillary, Stroma-rich.

Hyperparameter optimization with TPE

We used Tree-structured Parzen Estimator (TPE), which is a type Bayesian optimization algorithm, to optimize the best model (all omics integrated). Best parameters were selected based on the F1-score in 5-fold cross-validation.

The searching space was defined as described in Table 4.

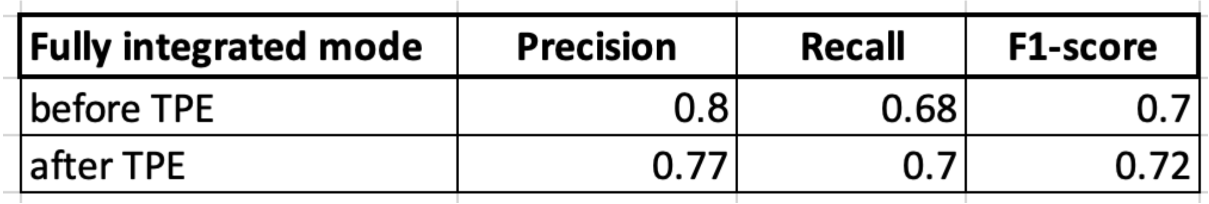
Table 4: Search space



Hyperparameter	Search Space	Best value
Num. of Hidden Layers in RNA	[12,17]	14
Num. of Hidden Layers in Proteomics	[6,8]	6
Num. of Hidden Layers in miRNA	[5,8]	7
Num. of Hidden Layers Methylomics	[10,12]	12
Learning Rate	[0.01, 0.1]	0.0134
Dropout Rate	[0.3, 0.7]	0.6

As a result, F1-score increased by 2% (Table 5).

Table 5: F1 score increase



Fully integrated mode	Precision	Recall	F1-score
before TPE	0.8	0.68	0.7
after TPE	0.77	0.7	0.72

Interestingly, the optimization procedure increased the F1-score by raising up the recall, but dropping precision a bit in the process. Ultimately, the model became more balanced.

Conclusion

By comparing models integrating different omics modalities using GATv2, we showed that the best balanced performance (all core metrics are high) was achieved for the all-modality integrated model. It achieved high precision without compromising recall or F1-score as other integrated models did. For example, a model without Methylomics data illustrated slightly better precision, but at the cost of dropping classification sensitivity. Importantly, even a single-modality model based on Gene Expression only demonstrated potent results. This fact combined with the low number of data samples we used for training illustrates how capable GATv2 architecture is.

Integrated model's performance was not uniform across different cancer subtypes: higher metrics were achieved for more represented classes. Severely underrepresented classes like LumNS and NE-like resulted in very low classification quality for these subtypes. The distribution of quality scores across subtypes corresponded with how well those subtypes clustered on UMAP during the differential expression analysis. TPE-based hyperparameter optimization of the all-integrated model increased the F1-score by balancing out precision and recall better.

Code & Data Availability

All the code and data used in the project are available at
https://github.com/Unknown-Negotiator/Multiomics-Bladder-Cancer-Subtype-Classification-using-GNN

Bibliography

Bergstra, James, et al. "Algorithms for hyper-parameter optimization." Advances in neural information processing systems 24 (2011).

Bhattacharjee A, Richards WG, Staunton J, Li C, Monti S, Vasa P, Ladd C, Beheshti J, Bueno R, Gillette M, Loda M, Weber G, Mark EJ, Lander ES, Wong W, Johnson BE, Golub TR, Sugarbaker DJ, Meyerson M. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc Natl Acad Sci U S A. 2001 Nov.

Brody, S., Alon, U., & Yahav, E. (2021). How Attentive are Graph Attention Networks? ArXiv, abs/2105.14491.

Chai, H., Zhou, X., Zhang, Z., Rao, J., Zhao, H., & Yang, Y. (2019). Integrating multi-omics data through deep learning for accurate cancer prognosis prediction. https://doi.org/10.1101/807214

Eliassen, T. (2023). Evolve GAT-A Dynamic Graph Attention Model. Retrieved from https://medium.com/stanford-cs224w/evolve-gat-a-dynamic-graph-attention-model-d3a416bb7c33

Huang, J., Shen, H., Hou, L., & Cheng, X. (2019). Signed Graph Attention Networks. International Conference on Artificial Neural Networks.

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2014). Machine learning applications in cancer prognosis and prediction. Computational and structural biotechnology journal, 13, 8–17. https://doi.org/10.1016/j.csbj.2014.11.005

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal, 13, 8–17. https://doi.org/10.1016/j.csbj.2014.11.005

Li, M. M., Huang, K., & Zitnik, M. (2022). Graph representation learning in biomedicine and healthcare. Nature biomedical engineering, 6(12), 1353–1369. https://doi.org/10.1038/s41551-022-00942-x

Li, X., Ma, J., Leng, L., Han, M., Li, M., He, F., & Zhu, Y. (2022). MoGCN: A Multi-Omics Integration Method Based on Graph Convolutional Network for Cancer Subtype Analysis. Frontiers in genetics, 13, 806842. https://doi.org/10.3389/fgene.2022.806842

Love, M.I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. 2014.

Mallik, S., & Zhao, Z. (2020). Graph- and rule-based learning algorithms: a comprehensive review of their applications for cancer type classification and prognosis using genomic data. Briefings in bioinformatics, 21(2), 368–394. https://doi.org/10.1093/bib/bby120

Meeks, J. J., Al-Ahmadie, H., Faltas, B. M., Taylor, J. A., 3rd, Flaig, T. W., DeGraff, D. J., Christensen, E., Woolbright, B. L., McConkey, D. J., & Dyrskjøt, L. (2020). Genomic heterogeneity in bladder cancer: challenges and possible solutions to improve outcomes. Nature reviews. Urology, 17(5), 259–270. https://doi.org/10.1038/s41585-020-0304-1

Muzellec B, Teleńczuk M, Cabeli V, Andreux M. PyDESeq2: a python package for bulk RNA-seq differential expression analysis. Bioinformatics. 2023.

National Cancer Institute (2021). What Is Cancer? Retrieved from https://www.cancer.gov/about-cancer/understanding/what-is-cancer

Paszke, A. et al., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035. Available at: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2023, November 15). Precision-Recall-F measure metrics. scikit-learn. https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics

Sarangi, Susanta, Md Sahidullah, and Goutam Saha. "Optimization of data-driven filterbank for automatic speaker verification." Digital Signal Processing 104 (2020): 102795.

Skubitz KM, Skubitz AP. Differential gene expression in renal-cell cancer. J Lab Clin Med. 2002 Jul.

Song, Y.; Hyun, S.; Cheong, Y.-G. Analysis of Autoencoders for Network Intrusion Detection. Sensors 2021, 21, 4294. https://doi.org/10.3390/s21134294

Subramanian, I., Verma, S., Kumar, S., Jere, A., & Anamika, K. (2020). Multi-omics Data Integration, Interpretation, and Its Application. Bioinformatics and biology insights, 14, 1177932219899051. https://doi.org/10.1177/1177932219899051

Tanvir, R. B., Islam, M. M., Sobhan, M., Luo, D., & Mondal, A. M. (2023). MOGAT: An Improved Multi-Omics Integration Framework Using Graph Attention Networks. https://doi.org/10.1101/2023.04.01.535195

Van 't Veer LJ, Dai H, Van de Vijver MJ, He YD, Hart AA, Mao M, Peterse HL, Van der Kooy K, Marton MJ.