Revealing Hidden Patterns in Cancer Data: An Unsupervised Learning Approach to Misdiagnosis Detection

Yaniv Lavi Ilan (ID. 215971441), Regev Yehezkel Imra (ID. 326724770)

Bar-Ilan University

May 2025

Abstract

This study applies unsupervised learning techniques to cancer gene expression data from the Xena project. Using clustering and anomaly detection combined with the cancer types labels, we identify potentially misdiagnosed samples and biologically meaningful cluster structures. We test multiple clustering algorithms while reducing dimensionality via UMAP. To evaluate clustering performance, we use statistical tools such as the Kruskal-Wallis test, Shapiro-Wilk test, and paired t-tests over bootstrapped evaluations of a custom defined metric. Our results highlight the efficacy of K-Means clustering and reveal significant associations between cluster anomalies and misdiagnosed samples, as confirmed by Fisher's exact test. We further identify the most misdiagnosed cancer types and propose a robust correction based on the cluster composition. Furthermore, we observe the emergence of different subclusters within certain cancer types, suggesting that some cancer types may include hidden biological differences that are only visible in a higher-dimensional space. This study not only demonstrates the utility of unsupervised learning methods, but also lays the groundwork for future research on cancer subtypes and improving diagnostic frameworks. The code and supplementary files are available at: `https://github.com/YanivIlan/Unsupervised_Project/tree/main`.

1 Introduction

Cancer is a group of diseases characterized by uncontrolled cell growth and genetic mutations. It remains a leading cause of death worldwide, with its complexity continuing to challenge accurate diagnosis and treatment. At the molecular level, different cancer types—and even subtypes within the same type—can be distinguished by their gene expression profiles. These high-dimensional datasets, capturing the activity levels of tens of thousands of genes, offer valuable insight into the disease, classification, and potential misdiagnoses - which are going to tackle.

One ongoing issue is the limited understanding of cancer subtypes. Many subtypes exhibit distinct gene expression patterns but are grouped under a broader diagnosis and receive the same treatment. This can lead to ineffective or inappropriate treatment for some patients, as biologically distinct conditions are managed as if they were the same. We are going to look into this issue.

Unsupervised learning has emerged as a powerful approach for analyzing such complex biomedical data. Unlike supervised methods, which rely on labeled data, unsupervised methods identify patterns and structure directly from the data itself [1]. This is particularly useful in biological settings, where labels may be noisy, incomplete, or inconsistent.

In this work, we apply unsupervised learning techniques to a large gene expression dataset from the Xena cancer genomics platform. We first preprocess the data to significantly reduce the number of dimensions from 20530. We then split the data into a training and test sets. On the train set we apply four different clustering methods: K-Means, GMM, DBSCAN, HDBSCAN [2]. For each algorithm we find the best number of dimensions and clusters, using the UMAP algorithm over a grid search for the number of clusters, and opted for the combination

of the parameters that maximized our metric. We then used the optimal parameters we got, and on the test set performed statistical tests (See 2.6) for comparison between the different clustering algorithms, to decide what is the best one for the dataset. We then performed anomaly detection, and used the Fisher's exact test to test whether anomalous samples are more likely to be misdiagnosed (See 2.9), and offer a solid correction to some misdiagnoses. Then, for each clustering algorithm we manage to find multiple distinct subclusters within some cancer types. Each step is accompanied with a 2-dimensional easy to intepret representation, including a robust visualization of the clusters against the real labels. This is done using the TSNE algorithm [3].

2 Methods

2.1 The Dataset

We used gene expression data from the UCSC Xena platform, an open-access resource that hosts large-scale genomics datasets from multiple cancer studies. The platform provides standardized, preprocessed data from sources such as The Cancer Genome Atlas (TCGA). In this project, we utilized two datasets:

The first, the EB++ Adjusted Pan-Cancer RNASeqV2 dataset. The dataset consists of 11060 samples, which are in this case patients with a certain type of cancer. For each sample it contains 20530 log2 - normalized gene expression values [4]. This normalization helps stabilize variance and improve interpretability. The data is numerical and continuous. Link to the dataset: `https://api.gdc.cancer.gov/data/3586c0da-64d0-4b74-a449-5ff4d9136611`

The second, the TCGA phenotype dataset. This dataset contains the labels for the samples. Each sample is assigned a cancer type. This data is categorical, and consists of 33 unique labels. Link to the dataset: `https://api.gdc.cancer.gov/data/0fc78496-818b-4896-bd83-52db1f533c5c`

2.2 Data Preprocessing

In order to decrease the number of dimensions from 20530 to a mild number, while maintaining the original distribution as much as possible, the following preprocessing steps were performed in this order:

- Removed genes that include null values. We could afford doing that as there are not a lot of null values in the data.
- Removed constant-like genes. We remained with the top 10% most variate genes - 1634 in total. This preprocessing step is commonly used in genomic data analysis. The rationale is that genes with a small variance across all samples do not contribute meaningfully to sample differentiation drive any patterns in the data.
- Z-scored normalized the data.
- Cleaned the remaining noise in the data. This was performed by applying the PCA algorithm on the remaining 1634 dimensional data. We set it to keep 90% of the original variance.

At the end of this process we were left with 413 dimensions and 11060 samples.

2.3 UMAP (Dimensionality Reduction)

We use the UMAP-learn package for python. UMAP (Uniform Manifold Approximation and Projection) is a dimensionality reduction technique for numerical data, designed to preserve both local and global structure in high-dimensional datasets. UMAP is particularly effective for visualizing and clustering biological data, where the number of features is large and distances are often non-linear.

2.4 Train test split

We split the data into two groups. The train set, consists of 60%, was used to find the best parameters - the number of clusters and dimensions for each clustering algorithm. The test set, consisting of the rest 40% was used for performing external statistical tests, using the parameters we got from the training set.

2.5 Bootstrapping Evaluation

To ensure the robustness of our clustering evaluation, we employed bootstrapping. For each clustering algorithm, we fix the optimal parameters best parameters we got from the training set, and generate 50 multiple resampled subsets of the test set by sampling with replacement. On each bootstrap sample, we perform clustering using those parameters and and evaluate the clustering using a custom metric, enabling us to estimate the distribution, mean, variance of the results vector.

2.6 Statistical Evaluation

To compare the performance of different clustering algorithms, we applied several statistical tests on our boostrap results.

- For cluster evaluation, we used the Kruskal–Wallis H-test [5] to evaluate whether the distribution of evaluation scores differed significantly across clustering algorithms. Following a significant Kruskal–Wallis result, we conducted pairwise comparisons between algorithms. To determine whether the assumptions of normality held for each algorithm's results, we used the Shapiro–Wilk test [6]. Then, when the results of two algorithms were found to be approximately normal, we applied the paired t-test; otherwise, we used the Wilcoxon signed-rank test, a non-parametric alternative.
- We applied Fisher's exact test [7] to investigate the relationship between detected anomalies and misdiagnosed cancer samples (see 2.9), testing whether anomalous samples were significantly more likely to deviate from their cluster's dominant cancer type.

2.7 Finding the optimal number of clusters and dimensions

We tested four clustering algorithms: K-Means, GMM, DBSCAN, HDBSCAN. In order to find the optimal combination of the number of clusters and the dimensionality of the UMAP-reduced space, we performed a grid search over different values: number of clusters (2 through 15, and from 20 to 50 with increments of 4) and number of dimensions (1 through 10, as well as 50, 100, 150, and 200). We have evaluated the performances using the metric $0.8 \cdot \text{sil} - 0.2 \cdot \text{db}$, where sil represents the silhouette score, and db represents the Davies–Bouldin index, and opted for the the parameters that maximized this metric.

- **Silhouette Score [8]** The Silhouette Score evaluates how similar an object is to its own cluster compared to other clusters, with values ranging from -1 to 1. A higher score indicates that the sample is well matched to its own cluster and poorly matched to neighboring clusters.
- **Davies–Bouldin index [9]** The Davies–Bouldin index measures the average similarity between each cluster and its most similar one. Lower index values indicate a better clustering result. The index is improved by increased separation between clusters and decreased variation within clusters.

We used this metric to counter the bias of the silhouette score torwards a small number of cluster, while still making it impactful.

In addition to the grid search, we also tested two other variations for the K-Means and the GMM algorithms. For K-Means, we first fixed the number of dimensions as the number that maximized the average silhouette score over the whole clusters number range. Then, we selected the number of clusters using the elbow method, requiring

to cut the K-Means Loss function by at least 50% of the initial loss. For GMM, we chose he number of clusters in advance by Bayesian Information Criterion [10].

For DBSCAN and HDBSCAN we fix some internal parameters that do not depened on the number of clusters. For HDBSCAN we set $min\_cluser\_size = 20$, and $min\_samples = 10$ .For DBSCAN, we first fix $min\_samples = 5$, and throughout the grid search for each dimension we used a KNN graph [11] and applied the KneeLocator method [12] to estimate the optimal ϵ.

2.8 Anomaly Detection

Anomalies were identified using three methods:

- **K-Means** For each cluster, we computed the Euclidean distance of each sample to its assigned centroid. Samples whose distance exceeded the cluster mean plus three standard deviations were flagged as anomalies.
- **GMM** Anomalies were identified based on log-likelihood under the fitted GMM. Samples with likelihoods below the 2nd percentile were considered anomalous.
- **HDBSCAN** HDBSCAN provides an outlier score for each sample based on local density. We labeled as anomalies those samples whose score was above the 98th percentile, indicating sparse local neighborhoods or low density support.

2.9 Cluster Interpretation

Clusters with more than 90% purity of a single cancer type were labeled "dominant". Minority cases in these clusters were labeled potential misdiagnoses. This was done using the labels.

3 Results

3.1 Finding the optimal number of clusters and dimensions

After performing the grid search on our metric exactly as written in 2.7 on the training set, we got the following results:

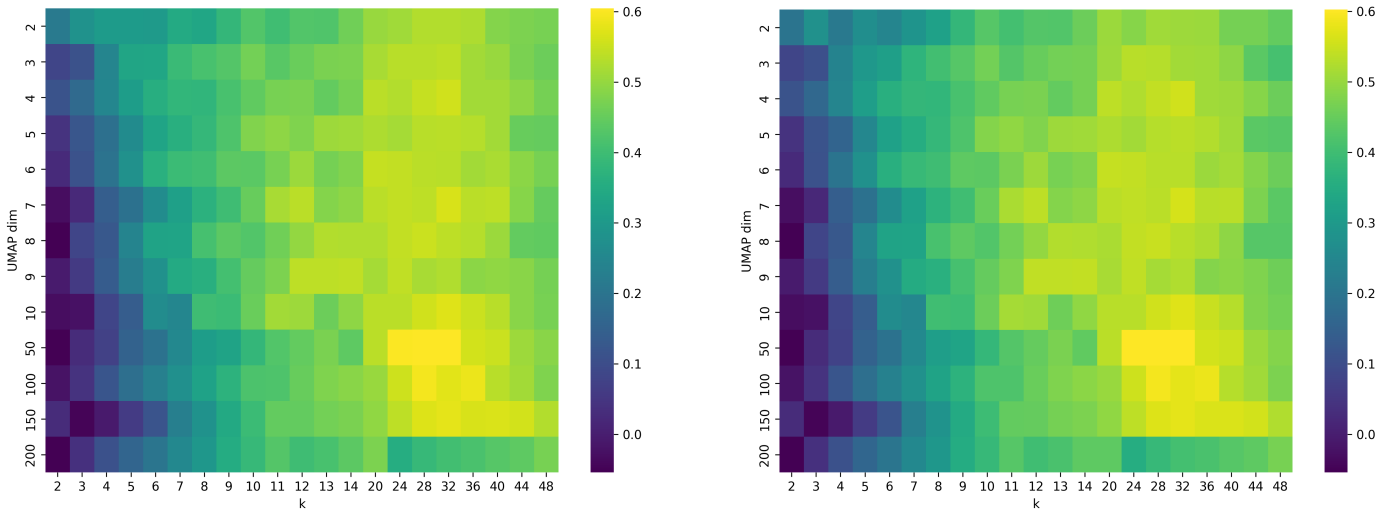
Table 1: Optimal number of clusters and UMAP dimensions for each clustering algorithm

Parameter	**K-Means**	**GMM**	**DBSCAN**	**HDBSCAN**
UMAP Dimensions	50	50	100	50
Number of Clusters	32	32	42	59

Table 2: Evaluation metrics for each clustering algorithm on the train set

Algorithm	**Silhouette Score**	**DB Index**	**Combined Score**
K-Means	0.762	0.274	0.604
GMM	0.761	0.276	0.602
DBSCAN	0.641	0.755	0.501
HDBSCAN	0.714	0.590	0.553

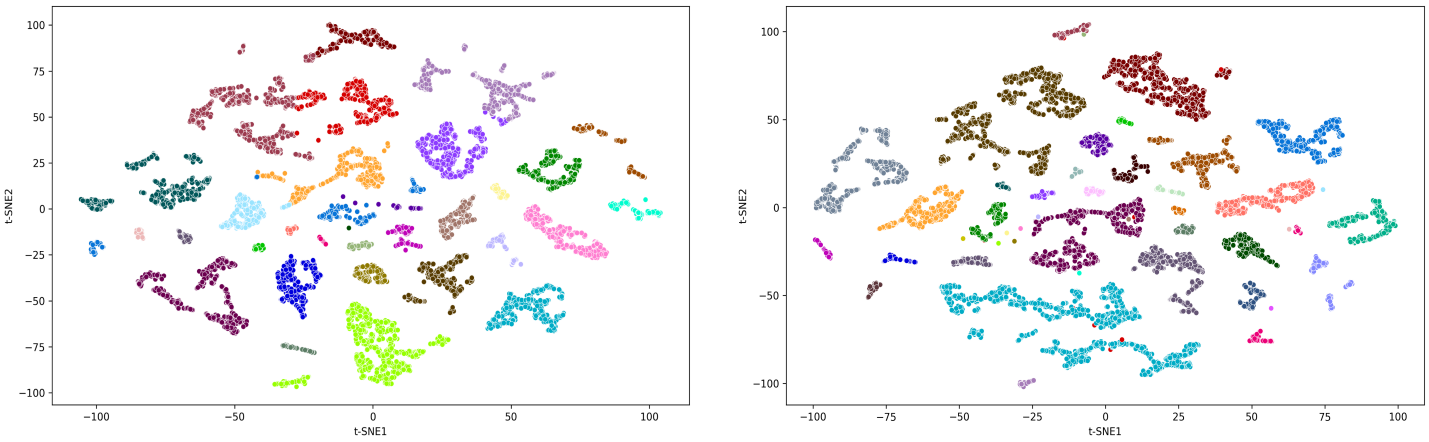
For the K-Means variate we described, we got that the optimal parameters are 2 dimensions and 4 clusters with a combined score of 0.359, and for the GMM BIC variate the result is 6 dimensions and 20 clusters with a combined score of 0.559. See 1 for some heatmaps, and for more results check the GitHub repository.



(A) (B)

Figure 1: Heamaps of clustering evaluation using the combined metric, the number of clusters against the number of dimensions. (A) K-Means (B) GMM.

We then visualized those initial clustering results in 2 dimensions using the TSNE algorithm:



(A) (B)

Figure 2: Visualization of the clustering algorithms with the optimal parameters. (A) K-Means (B) DBSCAN.

3.2 Finding the best clustering algorithm

For each clustering algorithm we used the optimal parameters we got from 3.1 and performed bootstrapping and statistical tests on the test set exactly as mentioned in 2.5 and 2.6. The metric we evaluated the bootstraps on is the average between the silhouette score and the mutual information with the external labels, as we believe it produces a balanced and more robust result. For the bootstraps result see 3.

Table 3: Average averaged score scores over bootstrapped samples (mean ± std)

Metric	**K-Means**	**GMM**	**DBSCAN**	**HDBSCAN**	**K-Means E**	**GMM-BIC**
Averaged score	0.7766 ± 0.01	0.7762 ± 0.01	0.691 ± 0.01	0.75 ± 0.01	0.45 ± 0.007	0.76 ± 0.01

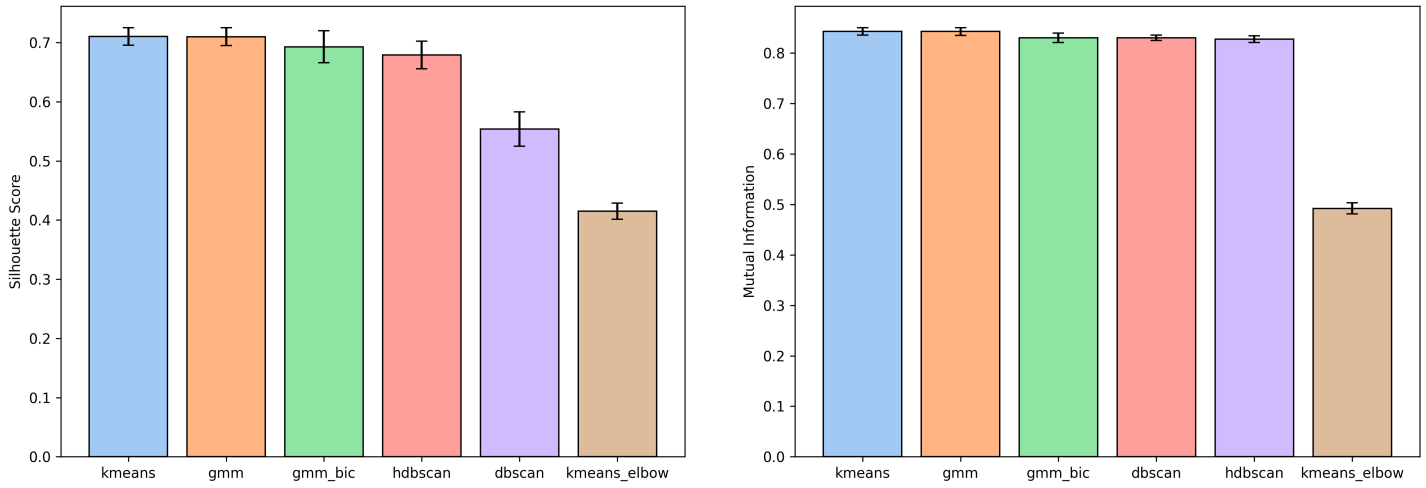
We then used the Kruskal–Wallis test to assess whether the clustering scores differed significantly between

algorithms. The result was highly significant, with $H = 243.07$, $p < 10e - 49$, indicating that at least one algorithm performs better then the others. We then used the Shapiro-Wilk test to check whether we can assume normality and use the paired t-test. The results:

Table 4: Shapiro-Wilk test p-values for normality of bootstrapped clustering scores

Algorithm	**K-Means**	**GMM**	**DBSCAN**	**HDBSCAN**	**K-Means E**	**GMM-BIC**
p-value	0.721	0.730	0.482	0.636	0.223	0.112

As we can see, all of the p-values are greater than 0.05. Thus, for every algorithm's measurements, the null hypothesis that the data is normally distributed, was not rejected. This allows us to continue with a pair-wise paired t-test between every two algorithms to figure out which is the best one overall. Performing this procedure yields that K-Means is the best algorithm. It is better than GMM with $p < 4.28e - 08$, better than DBSCAN with $p < 1.31e - 35$, better than HDBSCAN with $p < 8.22e - 14$, better than K-Means-E with $p < 5.16e - 72$ and better than GMM-BIC with $p < 1.04e - 06$ of the paired t-test. For the silhouette score and mutual information measurements see 3.



(A) (B)

Figure 3: Results of 50 bootstraps evaluation for each cluster including the error bars. (A) Silhouette score (B) Mutual information with the external labels.

3.3 Anomalies and Misdiagnoses

We first applied K-Means, GMM, and HDBSCAN clustering on the full dataset, using the optimal parameters identified in section 3.1. For each clustering result, we performed the corresponding anomaly detection method as described in section 2.8, resulting in a list of samples flagged as anomalies. (The complete lists, along with two-dimensional t-SNE visualizations highlighting anomalies, are available in the GitHub repository.)

To incorporate external information, we used the ground truth cancer type labels. We defined a cluster as dominant if more than 90% of its samples belonged to the same cancer type. This 90% threshold ensures a strong majority, reducing the influence of noise while allowing for minor biological or technical variation.

Within each dominant cluster, we labeled any sample whose true cancer type did not match the cluster's dominant type as a *misdiagnosis*. Our main question was whether samples identified as anomalies—using only internal, label-free methods—tend to be *misdiagnosed* according to the external labels.

To assess this association, we conducted Fisher's exact tests between the set of anomalous samples and the set of misdiagnosed samples for each clustering method. Due to the test's results only being valid for samples within

dominant clusters, we present their numbers in advance to avoid false conclusions. Keep in mind that there are overall 33 unique cancer types in the dataset.

Table 5: Dominant clusters counts for each algorithm

Algorithm	**K-Means**	**GMM**	**HDBSCAN**	**DBSCAN**	**GMM-BIC**
N. dominant clusters	20	20	37	26	10

And the Fisher's exact results:

Table 6: Fisher's Exact Test results: association between anomalies and misdiagnoses

Algorithm	**Anomaly + Mis**	**Anomaly Only**	**Mis Only**	**Normal**	**Odds Ratio**	**P-Value**
K-Means	17	126	113	10804	12.9	$7.08e-13$
GMM	9	213	126	10712	3.592	$1.58e-3$
HDBSCAN	16	206	129	10709	6.448	$3.17e-8$

Here "Mis" means "Misdiagnosis" due to lack of space. Those results suggest a strong association between samples flagged as anomalies and those identified as potential misdiagnoses in dominant clusters. **This result is highly significant in the K-Means algorithm, where anomalous samples were nearly 13 times more likely to be misdiagnosed compared to non-anomalous samples. This highly significant result suggests that K-Means clustering on a dimensions reduced dataset, followed by internal anomaly detection, may be effective at flagging new mislabeled or diagnostically ambiguous samples, even when relying just on existing data.** Note that those results can be improved if we relax the dominant threshold, for example, to 80%, which is still robust enough. Doing so for K-Means makes this result applicable to 23 clusters instead of 20. We will elaborate more on the applicability of this result in the Discussion.

3.4 Cancer-Type-Specific Findings

Now we investigate which cancer types are most frequently misdiagnosed within dominant clusters across the different clustering algorithms. We also assess whether certain algorithms are consistently challenged by the same cancer types.

Table 7: Most misdiagnosed cancer type for each algorithm

Algorithm	**Most Misdiagnosed Type**	**#Misdiagnosed Samples**	**# Samples Of This Type**
K-Means	Cholangiocarcinoma	41	45
GMM	Cholangiocarcinoma	41	45
DBSCAN	Cholangiocarcinoma	41	45
HDBSCAN	Lung Squamous Cell Carcinoma	20	553

For those types, based on our clustering procedure, let's understand what their real type should be. We do this by simply asking whether a potentially misdiagnosed sample lies in a dominated cluster, and take the corresponding type as the real type. If a sample lies in a cluster which isn't dominated we still present the results.

For the K-Means algorithm, the real cancer type of 40 out of the 41 initially Cholangiocarcinoma diagnosed samples is actually Liver Hepatocellular Carcinoma, and the remaining sample is skin Cutaneous Melanoma (GMM and DBSCAN yield similar results, see the GitHub repository). For HDBSCAN, the real type of 15 out of the 20 initially Lung Squamous Cell Carcinoma diagnosed samples is actually Lung Adenocarcinoma. Note that script can easily be modified to analyze every cancer type like we did here using the full misdiagnoses list available in the GitHub repository, resulting in a full correction for the whole dataset.

Now, Let's visualize those results. For each algorithm we reduced the dimensionality and clustered using the optimal parameters. Then applied the TSNE algorithm for a 2-D embedding. In the figure, each background

region represents a cluster assigned by the algorithm, and each small dot corresponds to cancer type label. Samples marked with "X" shape are Cholangiocarcinoma diagnosed.

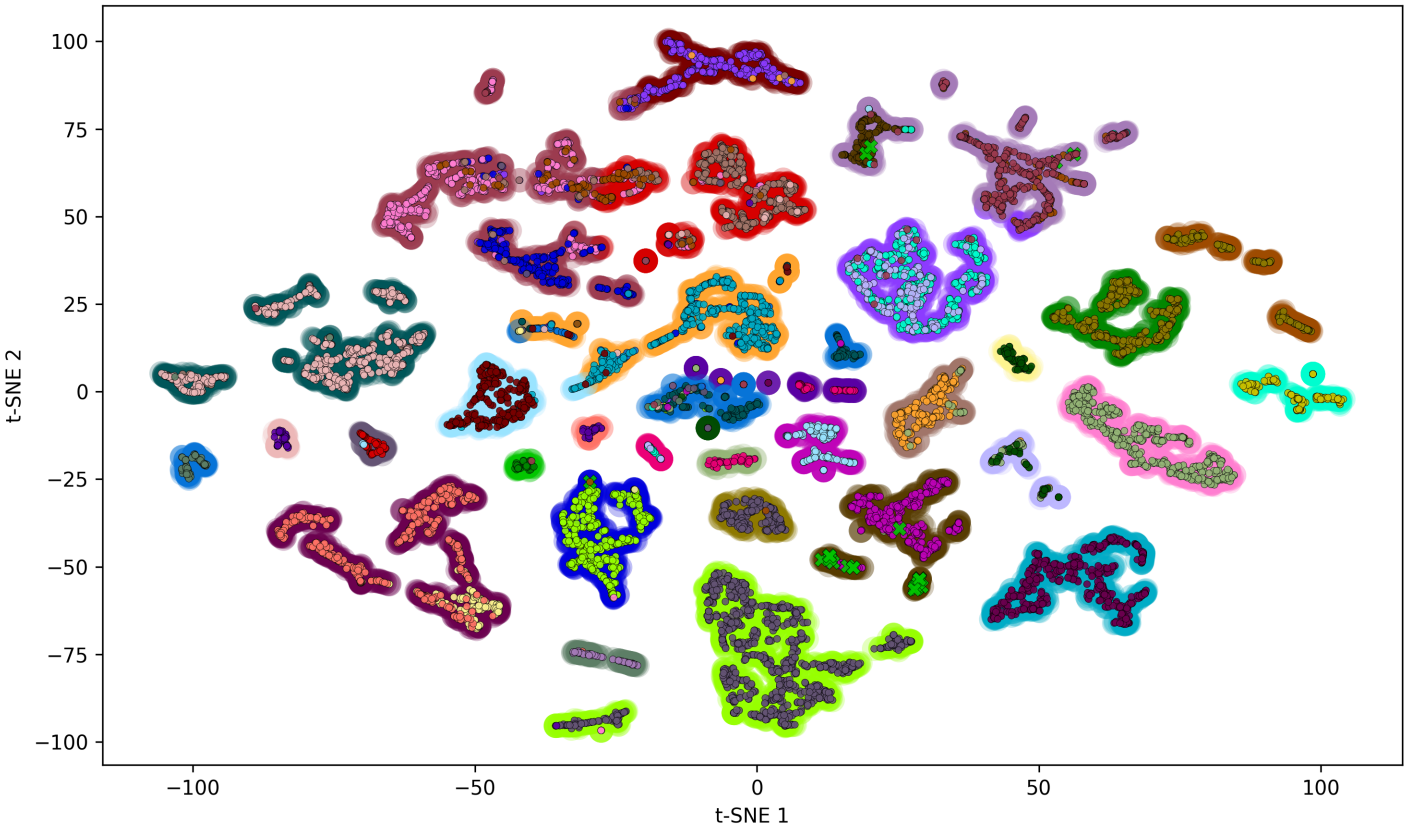


Figure 4: K-Means 2-Dimensional visualization highlighting Cholangiocarcinoma diagnosed samples.

This figure also strengthens our claim that the clustering combined with dimension reduction methods can be used to perform this kind of analysis. Due to lack of space we can not put all the figures here, so for the HDBSCAN figure and more check the GitHub repository.

3.5 Subtype Discovery

While some cancer types are known to be heterogeneous, unsupervised learning offers an opportunity to reveal potential subtypes purely from gene expression data, without relying on prior information. For each clustering algorithm, we examined clusters which were dominated by a single cancer type. Then, if a cancer type dominated more than one cluster, we considered it a potential indication of underlying biological subtypes.

Here are some examples we extracted from the files. (See the GitHub repository for the complete list). Using DBSCAN Breast Invasive Carcinoma was split into four distinct clusters, each with >95% dominance and Acute Myeloid Leukemia was split into two clusters with 100% dominance, and more. using K-Means, GMM and DBSCAN, Head and neck Squamous Cell Carcinoma was split into three distinct clusters, and Thymoma was split into two distinct clusters with 100% dominanace.

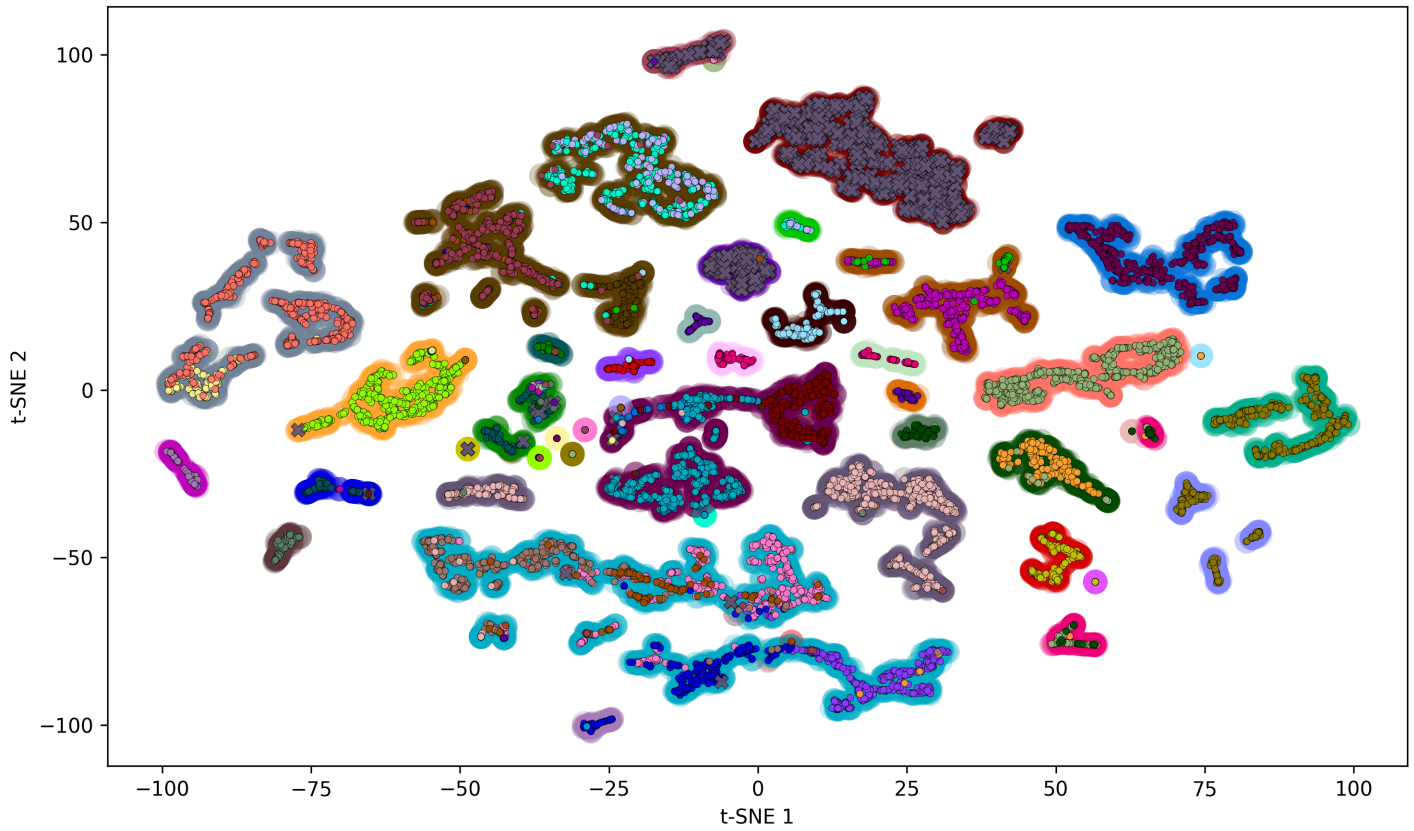


Figure 5: DBSCAN clustering. Samples diagnosed with Breast Invasive Carcinoma have an "X" shape.

In this visualization we can clearly see the distinct clusters that carry the same cancer type type. Those results show that using unsupervised learning methods can serve as a tool for biological investigation of the different cancer subtypes. The next thing to analyze when it comes to really understanding what genes derive each of those clusters and making them different from one another is feature importance, but, we were not apply to perform it in this project. See 3.6.

3.6 Challenges in Biological Interpretation

Due to UMAP's non-linearity, interpreting clusters in terms of gene contributions was infeasible. Due to that we also tried testing dimension reduction using the PCA algorithm from the Scikit-learn library [13], but the algorithm completely failed to preserve the global structures of the data and to reveal informative natural groupings. this is because gene expression data often has a complex and nonlinear structure. Future work may involve alternative dimensionality reduction approaches to address this.

4 Discussion

This study demonstrates the potential of unsupervised learning to uncover meaningful biological structure and highlight diagnostic inconsistencies in large-scale cancer data. By applying dimensionality reduction, multiple clustering algorithms, and statistical validation, we were able to evaluate and compare clustering performance and derive biological insights. Among the clustering methods tested, K-Means consistently outperformed others across internal metrics and statistical tests.

A major focus of our work was identifying potentially misdiagnosed samples using anomaly detection techniques tailored to each clustering method. Our results showed a strong association between algorithmic anomalies and

real-world diagnostic inconsistencies. For example, Fisher's exact test for K-Means revealed that anomalous samples were nearly 13 times more likely to be misdiagnosed compared to the rest, with a highly significant.

This suggests that simple internal criteria, such as distance to centroid or likelihood score, can serve as valuable tools for surfacing mislabeled or biologically atypical cases—without relying on new data. This could be a powerful mechanism in real-world clinical pipelines for flagging outliers for further pathological review.

In addition to identifying potential diagnostic errors, our analysis revealed candidate molecular subtypes within several cancer types. For example, both HDBSCAN and DBSCAN successfully separated cancers like Breast Invasive Carcinoma into distinct clusters, each exhibiting $> 90\%$ purity. This suggests the presence of biologically distinct subgroups within a single cancer type. These findings highlight the potential of unsupervised clustering to uncover not only broad diagnostic categories but also more refined biological variations that may reflect underlying genetic or differences.

Altogether, our results emphasize that unsupervised learning, when combined with robust dimensionality reduction and proper evaluation, offers an efficient method to validate diagnoses, reveal hidden structure, as well as raising hypotheses and questions for future biological investigation. As we said, in order to understand what drives the dominant clusters and the different cancer types, future work may involve finding alternative dimensionality reduction methods that enable the use of feature extraction.

References

[1] Geeks For Geeks Contibutors, "Unsupervised learning." https://www.geeksforgeeks.org/unsupervised-learning.

[2] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017.

[3] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[4] Monti Lab, "Rna-seq data scaling and normalization." https://montilab.github.io/BS831/articles/docs/RNAseq_ScalNorm.html.

[5] DATAtab Team, "Datatab: Online statistics calculator." https://datatab.net, 2025. DATAtab e.U., Graz, Austria.

[6] Ahmad Humaizi, "10.0 shapiro-wilk test." https://medium.com/@maizi5469/10-0-shapiro-wilk-test-5be38fd3c2a6, 2024.

[7] Wikipedia contributors, "Fisher's exact test — wikipedia, the free encyclopedia." https://en.wikipedia.org/wiki/Fisher%27s_exact_test, 2024.

[8] Wikipedia contributors, "Silhouette (clustering) — wikipedia, the free encyclopedia." https://en.wikipedia.org/wiki/Silhouette_(clustering).

[9] Wikipedia contributors, "Davies–bouldin index — wikipedia, the free encyclopedia." https://en.wikipedia.org/wiki/DaviesBouldin_index.

[10] Geeks For Geeks Contibutors, "Bayesian information criterion (bic)." https://www.geeksforgeeks.org/bayesian-information-criterion-bic/.

[11] Scikit-learn Developers, "sklearn.neighbors.kneighbors_graph — scikit-learn documentation." https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.kneighbors_graph.html.

[12] K. Arvai, "kneed: Knee point detection in python." https://pypi.org/project/kneed/.

[13] Scikit-learn Developers, "sklearn.decomposition.pca — scikit-learn documentation." https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.