

CancerSubtyper: A Web-Based Deep Learning Platform for Cancer Subtyping Through DNA Methylation Data

Yat Fei Cheung

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Application

Liqing Zhang, Chair
Chang-Tien Lu
Na Meng

August 14, 2025
Blacksburg, Virginia

Keywords: Bioinformatics, Deep Learning, Web Development

Copyright 2025, Yat Fei Cheung

CancerSubtyper: A Web-Based Deep Learning Platform for Cancer Subtyping Through DNA Methylation Data

Yat Fei Cheung

(ABSTRACT)

Cancer subtyping plays a critical role in understanding tumor heterogeneity, predicting patient outcomes, and guiding personalized therapies. While DNA methylation data offers an informative molecular source for subtyping, leveraging its signals across cohorts remains challenging due to high dimensionality, batch effects, and lack of standardized tools. In this thesis, we present *CancerSubtyper*, a web-based deep learning platform that enables both supervised classification and semi-supervised discovery of cancer subtypes using methylation data. The platform incorporates two models: *BCtypeFinder*, designed for subtype prediction with domain adaptation; and *CancerSubminer*, a flexible model for subtype discovery or refinement, with potential applicability across different cancer types. Users can upload labeled and unlabeled datasets, select models, and visualize results through various interactive plots including uniform manifold approximation and projection (UMAP), boxplots, and survival curves. We evaluate the platform using The Cancer Genome Atlas (TCGA) breast cancer cohort, demonstrating distinct cancer subtypes, batch correction, and clinical relevance via Kaplan-Meier survival analysis. *CancerSubtyper* offers a reproducible and user-oriented platform that streamlines cancer subtype analysis using DNA methylation data. It bridges a methodological gap by enabling both classification and discovery tasks, supports batch correction across datasets, and facilitates result interpretation through interactive visualizations. This platform empowers researchers to conduct comparative and clinically meaningful

subtyping analyses without requiring extensive programming expertise. The platform is designed for accessibility and requires no programming expertise, making it a practical tool for researchers and clinicians to explore cancer subtypes and contribute to personalized care.

CancerSubtyper: A Web-Based Deep Learning Platform for Cancer Subtyping Through DNA Methylation Data

Yat Fei Cheung

(GENERAL AUDIENCE ABSTRACT)

Cancer is not a homogeneous disease. Even among patients diagnosed with the same type, such as breast cancer, the internal structure of their tumors can vary significantly. Understanding these differences, or “subtypes,” plays crucial roles because each subtype may respond differently to treatment. In this project, we developed a user-friendly web platform called *CancerSubtyper*, which helps researchers identify and study cancer subtypes using a biological signal known as DNA methylation. DNA methylation regulates gene expression at the transcriptional level, and abnormal methylation patterns are commonly associated with tumor development and progression. Our system uses artificial intelligence to detect patterns in DNA methylation data and group tumors into biologically and clinically meaningful subtypes. We tested the platform on a public dataset and demonstrated distinct subtype separation and clinical relevance through survival analysis.

Contents

List of Figures	viii
List of Tables	x
1 Introduction	1
2 Methods	4
2.1 Platform Overview	4
2.2 System Architecture	5
2.3 Data Input and Preprocessing	8
2.4 Cancer Subtyping	8
2.5 Batch Effect Correction	10
2.6 Computational Complexity and Scalability	10
2.7 Prediction Result with Interactive Visualization	12
3 Results	14
3.1 Experimental Setup	14
3.2 Cancer Subtyping Results	15
3.3 Investigating CpG Clusters Associated with Each Subtype	20

3.4	CpG Methylation Intensity Across Subtypes	24
3.5	Survival Analysis by Predicted Subtypes	27
4	Discussions	30
5	Conclusions	36
	Appendices	40
	Appendices	40
A	Model Architecture and Hyperparameters	40
A.1	BCtypeFinder Architecture	40
A.2	CancerSubminer Architecture	40
A.3	Hyperparameter Settings	41
B	Preprocessing Pipeline Details	42
B.1	Input Files	42
B.2	Preprocessing Steps	42
C	Platform Usage Instructions	44
C.1	Creating an Analysis Profile	44
C.2	Running Jobs	44
C.3	Viewing Results	45

C.4 Export Options	45
Bibliography	46

List of Figures

2.1	System architecture of the CancerSubtyper platform. The platform integrates a React-based frontend, a FastAPI backend, asynchronous job management with Celery and Redis, and model execution via encapsulated scripts. All services are containerized using Docker and deployed on a Linux-based server.	7
3.1	UMAP visualization of the target dataset before and after applying BCtypeFinder. Upper panels show the original dataset colored by batch and subtype, respectively. Lower panels show the dataset after batch correction, demonstrating improved subtype separation and reduced batch effects.	17
3.2	Subtype prediction comparison between BCtypeFinder and baseline machine learning models (SVM, RF, LogReg). BCtypeFinder yielded a more balanced prediction across subtypes, particularly for Her2 and Luminal B.	18
3.3	UMAP visualization of the target dataset before and after applying CancerSubminer with manual 5-cluster configuration. Baseline clustering methods (K-means and NeMo) are shown for comparison. CancerSubminer provided better separation and balanced clustering across batches.	19
3.4	Subtype prediction comparison between CancerSubminer, K-means clustering, and NeMo clustering under the 5-cluster setup. CancerSubminer achieved a more balanced and biologically meaningful subtype assignment.	20

3.5	Interactive heatmap generated by the platform showing Spearman correlations between the top 30 CpG clusters within a selected batch and subtype. The heatmap highlights co-methylated clusters, which may reflect coordinated epigenetic regulation associated with subtype-specific pathways. Users can select the desired subtype and batch through a dropdown interface. . . .	22
3.6	Interactive CpG metadata table displayed alongside the heatmap. Each row represents a CpG site from the top 30 clusters, with columns showing the cluster ID, CpG identifier, chromosomal location, strand, UCSC gene annotation, and genome build. The table supports interactive filtering and searching for targeted exploration.	23
3.7	Beta value distribution for a selected CpG cluster across five breast cancer subtypes. The interactive boxplot allows users to explore subtype-specific methylation intensity. Each color corresponds to a distinct subtype. Users may select different clusters and toggle between batches to explore customized methylation patterns.	26
3.8	Kaplan–Meier survival curves for a selected batch, stratified by predicted subtypes. Subtypes exhibit distinct survival trajectories, with Luminal A showing the most favorable outcomes. Log-rank test yields a statistically significant p -value of 2.4×10^{-3}	28

List of Tables

2.1	Asymptotic costs of major steps (time and peak memory).	12
3.1	Datasets used for model evaluation.	15
A.1	Shared hyperparameters used in BCtypeFinder and CancerSubminer.	41

List of Abbreviations

BRCA Breast Invasive Carcinoma

CpG Cytosine-phosphate-Guanine site

CSV Comma-Separated Values

DNA Deoxyribonucleic Acid

KM Kaplan-Meier

LogReg Logistic Regression

RF Random Forest

SVM Support Vector Machine

TCGA The Cancer Genome Atlas

UMAP Uniform Manifold Approximation and Projection

Chapter 1

Introduction

Molecular cancer subtyping is the process of classifying tumors into biologically distinct groups based on molecular features such as gene expression profiles or epigenetic patterns, rather than solely relying on traditional histopathological observations [3]. This approach is critical because patients with the same cancer type—such as breast cancer—can experience vastly different clinical outcomes and responses to therapy [14]. A major contributor to this variation is tumor heterogeneity, the presence of genetically and phenotypically diverse cell populations within and across tumors [1].

Accurate molecular subtyping has become essential for informing personalized treatment strategies and improving prognosis [16]. For instance, the widely used PAM50 model introduced intrinsic breast cancer subtypes that are now integral to clinical decision-making [16]. However, effective subtyping requires access to high-dimensional molecular data and robust analytical tools.

Among various molecular data types, DNA methylation has emerged as a valuable biomarker for cancer subtype prediction [7]. Methylation is an epigenetic mechanism in which methyl groups are added to cytosines at CpG dinucleotides, regulating gene expression without altering the DNA sequence [9]. This modification acts like a dimmer switch, modulating gene activity, and is often dysregulated in cancerous cells [18]. Tumors typically exhibit hypermethylation in tumor suppressor gene regions and hypomethylation in oncogenes, creating subtype-specific methylation patterns that differ from both healthy tissue and other tumor

types [17, 19].

DNA methylation data is inherently high-dimensional, often consisting of measurements from hundreds of thousands of CpG sites. Traditional machine learning methods struggle with such complexity unless extensive preprocessing or feature selection is performed [2]. Deep learning techniques offer a compelling alternative. These models are capable of learning latent, nonlinear representations from high-dimensional inputs and can identify subtle molecular patterns without manual feature engineering [10]. Recent models have demonstrated promising performance in both cancer subtype classification and discovery [8, 12, 19].

Despite the progress in algorithmic development, there is a lack of web-based platforms that support deep learning-driven cancer subtyping using DNA methylation data. Existing tools such as MethSurv [15] and MEXPRESS [13] primarily focus on single-gene analysis, differential methylation, or survival association, rather than multi-class subtype classification. Other platforms, like HiTAIC [5], specialize in tissue-of-origin classification and are not designed to uncover intra-cancer subtype structures. Many existing tools do not support batch effect correction, which is essential for integrating datasets collected from different laboratories, technologies, or time points. Without correcting for batch effects, which are systematic technical variations unrelated to biological differences, analyses can produce misleading subtype groupings that reflect technical artifacts instead of true biological signals [4, 11]. In addition, most platforms present results only as static plots, which limits the ability to explore, compare, and interpret subtyping results interactively. These limitations, along with the technical complexity of running deep learning pipelines, pose significant challenges for researchers without a computer science background. In particular, biologists and clinicians often lack the programming expertise or computational infrastructure required to perform large-scale subtyping analyses, underscoring the need for an accessible and interactive solution.

To address these gaps, this work introduces CancerSubtyper, an interactive, web-based platform for cancer subtyping based on DNA methylation data. CancerSubtyper supports two models: BCtypeFinder, a supervised classifier for cancer subtype prediction, and CancerSubminer, an integrative cancer subtyping method combining supervised and unsupervised learning. Both models incorporate adversarial domain adaptation for batch correction and produce interactive visualizations, including UMAP projections, Kaplan-Meier survival curves, and heatmaps.

The platform offers a streamlined, end-to-end workflow: users upload methylation datasets, select and run model, and receive subtype predictions and interactive plots—all without requiring any programming expertise. By lowering the barrier to advanced molecular subtyping analysis, CancerSubtyper empowers researchers with limited computational resources to derive deeper insights from complex methylation datasets.

Chapter 2

Methods

2.1 Platform Overview

CancerSubtyper is a web-based platform developed to support cancer subtyping using deep learning models trained on DNA methylation data. The system allows users to create analysis profiles by uploading a pair of datasets: a source dataset containing methylation profiles with subtype labels and a target dataset containing unlabeled methylation data. Additionally, users may upload an optional metadata file that includes clinical variables such as overall survival time and event status. Each analysis profile is annotated with the tumor type, a project name, and a short description. Under a single profile, users can execute multiple jobs, each applying a selected model to the uploaded data.

The two available models on the platform are BCtypeFinder[6] and CancerSubminer. Upon submission, each job proceeds through an automatic pipeline that performs data preprocessing, batch effect correction, and model execution. The system then generates a collection of interactive visualizations and downloadable result files. Users can interact with the results through filtering and exploration by batch or subtype and can inspect individual data points by hovering for details. Visualizations are exportable as PNG or SVG, while processed features, cluster assignments, and subtype predictions are available in CSV format.

2.2 System Architecture

CancerSubtyper is implemented as a modular, scalable full-stack web platform designed to make deep learning-based cancer subtyping accessible to researchers without requiring programming expertise. The system is organized into four major components: the frontend interface, backend services, job execution pipeline, and data storage.

The frontend is developed using React.js and styled with TailwindCSS through the DaisyUI component library. It provides a clean and responsive user interface for uploading data, configuring subtyping jobs, viewing results, and downloading outputs. The design prioritizes usability and accessibility, allowing users to complete the full analysis workflow through intuitive point-and-click interactions.

The backend is built with FastAPI, a high-performance Python web framework that handles authentication, API requests, file uploads, and job orchestration. All user metadata, job configurations, and result summaries are stored in a PostgreSQL database, which ensures efficient retrieval and persistence.

To manage computation-intensive subtyping tasks, the platform uses Celery for asynchronous job execution, with Redis acting as the message broker. Upon job submission, the backend serializes the task and sends it to a Celery worker, which executes the selected subtyping model using the uploaded data and configuration parameters.

The subtyping models used in the platform, namely BCtypeFinder and CancerSubminer, were developed externally and are integrated into the platform as encapsulated components. The platform handles data preprocessing, file management, and output collection, wrapping each model's execution within a standardized pipeline. This design ensures consistency across different models and allows for future extensions, such as the integration of new models or analytical methods, without requiring substantial changes to the platform infrastructure.

All services are containerized using Docker, which allows for reproducible deployment and easy environment configuration. The system is currently deployed on a Linux-based virtual machine, although the architecture supports horizontal scaling to accommodate larger workloads or multi-user usage.

By combining a user-friendly web interface with a robust backend and scalable job management system, CancerSubtyper enables end-to-end cancer subtyping analysis through an accessible and reproducible platform architecture.

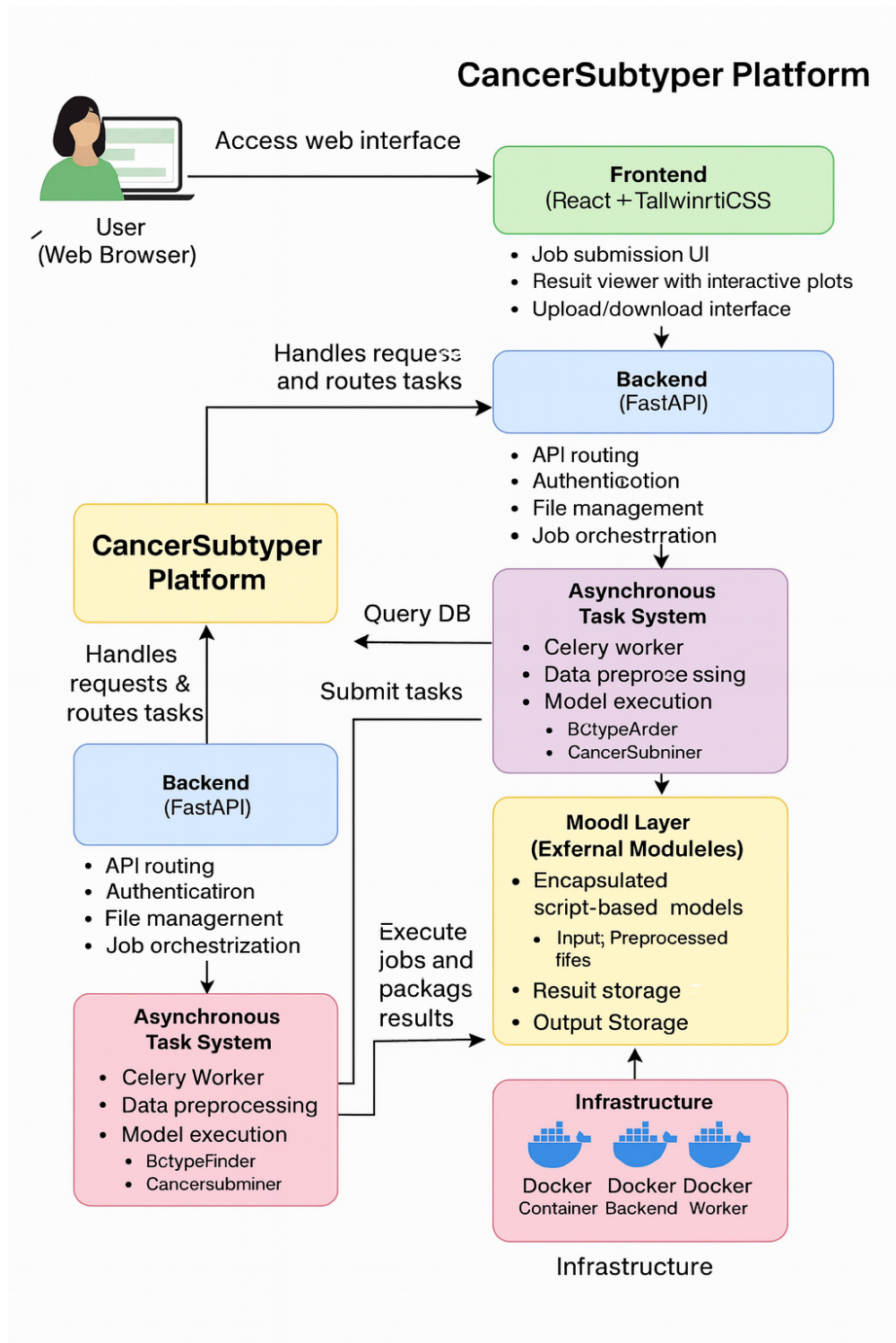


Figure 2.1: System architecture of the CancerSubtyper platform. The platform integrates a React-based frontend, a FastAPI backend, asynchronous job management with Celery and Redis, and model execution via encapsulated scripts. All services are containerized using Docker and deployed on a Linux-based server.

2.3 Data Input and Preprocessing

CancerSubtyper supports three types of input files: a source dataset, a target dataset, and an optional metadata file. The source and target datasets are both expected to be CSV files containing methylation beta values, with the source dataset also including known subtype labels. The metadata file contains clinical variables such as survival time and event status, also formatted as CSV.

Upon submission, each dataset undergoes a standardized preprocessing pipeline. The system first removes CpG sites that have more than 20 percent missing values. Remaining missing values are imputed using the median across samples. To reduce dimensionality, the platform applies K-means clustering to group CpG sites into 3,000 clusters, a number determined empirically using the elbow method. Features are aggregated at the cluster level to form a compact representation of the data. Finally, batch effect correction is applied using a combination of domain adaptation and subtype alignment methods. These steps ensure that downstream analysis is robust to cohort-specific technical variability.

2.4 Cancer Subtyping

CancerSubtyper implements two deep learning-based cancer subtyping approaches: (1) BC-typeFinder and (2) CancerSubminer. Both models leverage semi-supervised learning strategies to enhance subtype prediction using labeled source data and unlabeled target data. A key feature of both models is the use of adversarial domain adaptation to correct for batch effects, which are systematic technical differences that arise when data is collected across different laboratories, platforms, or time periods. Without correction, these effects can obscure true biological variation and compromise the reliability of subtype assignments. By

learning domain-invariant representations, the models improve consistency across datasets and support more accurate, biologically meaningful subtyping.

In both approaches, the final outputs include predicted subtype labels for the target dataset, along with several interpretive visualizations to aid in understanding subtype structure and clinical relevance.

BCtypeFinder is specifically designed for cancer subtyping and follows a three-stage training pipeline. The model begins by pre-training a feature extractor and classifier using the labeled source data. It then enters an adversarial training phase, during which a domain discriminator is trained to minimize distributional differences between the source and target datasets. In the final stage, the model performs semi-supervised fine-tuning, incorporating both pseudo-labeling and subtype alignment to refine predictions for the target data. In addition to subtype assignments, the model generates correlation heatmaps, beta value distribution plots, UMAP projections, model comparison tables, and Kaplan-Meier survival curves.

CancerSubminer is an integrative subtyping model that combines supervised and unsupervised learning within a unified training pipeline. It begins with supervised pre-training on the labeled source dataset, similar to BCtypeFinder. K-means clustering is then applied to the source data to capture subtype structure and reassign low-confidence samples based on cluster consistency. This is followed by adversarial training to produce a domain-invariant representation shared across source and target datasets. In the final step, pseudo-labeling and centroid alignment are used to refine subtype assignments for the target data. In addition to the visual outputs shared with BCtypeFinder, CancerSubminer also produces K-means clustering plots and NeMo projections to support subtype discovery and interpretation.

2.5 Batch Effect Correction

Batch effects are non-biological sources of variation that can arise from differences in sample preparation, sequencing platforms, or laboratory protocols. These artifacts can distort subtyping models and lead to inaccurate or irreproducible results. CancerSubtyper integrates batch correction directly into the training pipeline for both BCtypeFinder and CancerSubminer. Each model includes an adversarial training component in which a domain discriminator attempts to distinguish between source and target samples. The feature extractor is trained to confuse the discriminator, thereby learning a domain-invariant representation. In parallel, subtype alignment is performed by matching cluster centroids between source and target data. These two strategies together ensure that the model focuses on biologically relevant signals rather than technical variation.

2.6 Computational Complexity and Scalability

The CancerSubtyper workflow involves several computationally intensive steps, and it is important to understand how these scale with larger datasets and parameter choices. At a high level, the runtime and memory demands are driven by three components: preprocessing of methylation matrices, deep learning model training, and batch effect correction through domain adaptation and subtype alignment.

During preprocessing, basic quality control and filtering require only a single pass over the data and scale linearly with the number of CpG sites and samples. Median imputation is similarly linear in data size and introduces minimal overhead. The main cost in preprocessing is the large-scale CpG clustering step. Clustering hundreds of thousands of CpGs into several thousand clusters requires repeated distance calculations in the sample space, and both

runtime and memory increase in proportion to the number of CpGs, the number of clusters, and the number of samples. This step can become a bottleneck when full Illumina 450K or EPIC arrays are used without filtering, although in practice variance-based preselection and mini-batch updates substantially reduce the load.

Training of the deep learning models dominates the overall compute time. Each forward and backward pass scales with the number of input features after clustering and with the size of the mini-batch. Across multiple training phases—pretraining, adversarial training, semi-supervised refinement, and fine-tuning—the total number of epochs largely determines the wall-clock time. Memory usage is influenced by the number of CpG clusters (which define the input dimension), the mini-batch size, and the depth of the network layers. On modern GPUs, using a few thousand clusters with batch sizes in the low hundreds is well within memory limits.

Batch effect correction introduces additional computation through the adversarial discriminator and subtype alignment. The discriminator adds a small network that typically increases runtime by 10–30% compared to baseline training. Subtype alignment requires centroid calculations over the latent space, which are modest compared to the main neural network operations. If kernel alignment penalties are used, there is an additional quadratic term in the batch size, so batch sizes are chosen conservatively.

Table 2.1 summarizes the asymptotic time and memory complexity of each major step. In practice, the main bottlenecks are CpG clustering on very large input matrices and the long multi-stage training schedules. These are mitigated by pre-filtering CpGs, limiting the number of clusters to the 1,000–3,000 range, using mixed precision and gradient accumulation, and adopting learning rate schedules with early stopping.

Table 2.1: Asymptotic costs of major steps (time and peak memory).

Stage	Time	Peak Memory
QC & filtering	$\mathcal{O}(pm)$	streaming $\mathcal{O}(bm)$; full $\mathcal{O}(pm)$
Median imputation	$\mathcal{O}(pm)$	$\mathcal{O}(p)$ per group
CpG k-means (k)	$\mathcal{O}(pk m_s)$	data $\mathcal{O}(p m_s)$, centroids $\mathcal{O}(k m_s)$
Cluster aggregation	$\mathcal{O}(p)$	$\mathcal{O}(km)$
Supervised train (per E)	$\mathcal{O}(E m_s P)$	$\mathcal{O}(P) + \mathcal{O}(Bf) + \text{activations}$
Adversarial train (per epoch)	$\mathcal{O}((m_s + m_t)(P + P_d))$	as above
SSL (per epoch)	$\mathcal{O}(m_t P + m_s P)$	as above
Subtype alignment	$\mathcal{O}(SDh)$ per batch	negligible vs. MLP

2.7 Prediction Result with Interactive Visualization

For each completed job, the CancerSubtyper platform generates a suite of interactive visualizations that facilitate the interpretation and validation of results. These visual outputs help users examine subtype separability, assess batch correction effectiveness, and explore the biological and clinical relevance of the model predictions.

Both BCtypeFinder and CancerSubminer produce UMAP projections that visualize sample distributions in a two-dimensional space. These are shown both before and after batch correction, with coloring by either subtype or batch to reveal whether samples group by biological category or technical origin. BCtypeFinder additionally includes comparison results against traditional machine learning models such as support vector machines (SVM), random forest (RF), and logistic regression (LogReg), enabling users to benchmark deep learning performance under a consistent setup.

CancerSubminer supports two modes of operation based on the availability of subtype structure: automatic subtype discovery, which identifies subtypes in an entirely unsupervised manner, and manual subtype refinement, where users specify the number of clusters to guide the analysis. The discovery mode is intended for exploring unknown or novel subtype structures, while the refinement mode allows researchers to evaluate or adjust existing subtype definitions. To support unsupervised cancer subtyping, the platform provides results based on both K-means clustering and NeMo, a state-of-the-art nonlinear manifold learning method for cancer subtype analysis [19]. In both modes, the model outputs a K-means clustering plot and a NeMo projection. The K-means plot reveals how samples are grouped based on the learned feature representation, while the NeMo projection offers a low-dimensional manifold view that helps uncover transitional patterns and sample relationships that may not be evident through conventional clustering.

To provide additional insight into the molecular characteristics of predicted subtypes, the platform generates correlation heatmaps of the top 30 CpG clusters. These heatmaps depict pairwise Pearson correlations between CpG clusters selected for their high variance and relevance to subtype separation. The platform also presents beta value distribution plots for each predicted subtype using boxplots, which illustrate the spread and consistency of methylation levels. Furthermore, a Kaplan-Meier survival plot is provided to evaluate survival differences across subtypes, based on metadata that includes time-to-event and survival status. These visualizations are consistently generated across all jobs and models to ensure a standardized and interpretable output format.

All visualizations are interactive and downloadable in both PNG and SVG formats. In addition, processed data outputs—including subtype predictions, batch-corrected feature matrices, CpG cluster assignments, and cluster centroids—are available in CSV format to support further analysis and reproducibility.

Chapter 3

Results

3.1 Experimental Setup

To evaluate the performance of CancerSubtyper, we adopted the same experimental configuration as the original BCtypeFinder study [6]. This design allows consistent benchmarking and supports a fair comparison across models implemented in the platform.

The labeled source dataset was obtained from the TCGA breast cancer (TCGA-BRCA) cohort, which includes 1,060 primary tumor samples with DNA methylation data profiled using Illumina Human Infinium 450K and 27K arrays. Each sample is annotated with intrinsic subtype labels derived from the PAM50 classification system [16], which defines five subtypes: Luminal A, Luminal B, Her2-enriched, Basal-like, and Normal-like.

For the unlabeled target data, three publicly available GEO breast cancer datasets were used: GSE69914, GSE75067, and GSE72245. These datasets were chosen to represent independent experimental batches and include a total of 611 samples. Although subtype labels were available in GSE72245, they were withheld during model inference to ensure objective evaluation.

All data underwent the same preprocessing pipeline. CpG sites with more than 20% missing values were filtered out, and remaining missing values were imputed using the median. K-means clustering was then applied to the CpG features to reduce dimensionality. Specifically,

CpG sites were grouped into 3,000 clusters, and each cluster was represented by its median beta value. This clustering-based representation was used as the model input.

Both BCtypeFinder and CancerSubminer were evaluated under this shared setup. BCtypeFinder was tested in its default supervised configuration. For CancerSubminer, we tested two modes: automatic, where the number of clusters was estimated based on internal model heuristics, and manual, where we fixed the number of subtypes to five to match the PAM50 labels. Domain adaptation and batch correction were enabled in both models to mitigate technical differences across datasets.

A summary of all datasets used is shown in Table 3.1. Hyperparameters and training conditions were consistent with those described in the BCtypeFinder paper [6].

Table 3.1: Datasets used for model evaluation.

Dataset	# of CpGs	# of Samples	Subtype Labels	Used for Training
TCGA-BRCA (27K)	27K	267	Yes	Yes
TCGA-BRCA (450K)	450K	793	Yes	Yes
GSE69914	450K	305	No	No
GSE75067	450K	188	No	No
GSE72245	450K	118	Yes	No

3.2 Cancer Subtyping Results

The CancerSubtyper platform provides a unified interface for applying integrated subtyping models to DNA methylation data. This section demonstrates how the platform facilitates subtype prediction, batch correction, and visualization using two supported models: BCtypeFinder and CancerSubminer.

For supervised classification, the platform supports BCtypeFinder, which applies deep learning with domain adaptation to leverage labeled source data for subtype prediction in unlabeled target datasets. In this study, it was configured to use the PAM50 subtype labels from the TCGA dataset. The results were visualized using UMAP to assess batch effects and subtype separability before and after applying BCtypeFinder.

Figure 3.1 displays the UMAP visualization results. The upper panels show the raw uncorrected dataset, where batch effects are evident in the batch-colored plot and subtype clusters are heavily entangled. After applying domain adaptation via BCtypeFinder, the lower panels demonstrate improved batch alignment and clearer subtype separation. Notably, subtypes such as Luminal A and Basal-like formed distinct clusters across batches.



Figure 3.1: UMAP visualization of the target dataset before and after applying BCtypeFinder. Upper panels show the original dataset colored by batch and subtype, respectively. Lower panels show the dataset after batch correction, demonstrating improved subtype separation and reduced batch effects.

To support benchmarking, the platform includes a comparison of subtype prediction outputs from traditional machine learning models, including Support Vector Machines (SVM), Random Forest (RF), and Logistic Regression. These models were trained and evaluated on the same preprocessed data within the platform. As shown in Figure 3.2, BCtypeFinder yielded a more balanced subtype distribution, particularly for underrepresented classes such as Her2

and Luminal B. In contrast, baseline models tended to overpredict the dominant Luminal A subtype.

Classification Results

This table presents the classification results, alongside predictions from baseline machine learning and deep learning models. It allows for direct comparison across methods to evaluate model performance.

Search... All Columns

Sample	Batch	BCTypeFinder	SVM	Random Forest	Logistic Regression
GSM1712367	GSE69914	LumA	LumA	LumA	LumA
GSM1712370	GSE69914	LumA	LumA	LumA	LumA
GSM1712371	GSE69914	LumA	LumA	LumA	LumA
GSM1712373	GSE69914	LumA	Normal-like	LumA	LumA
GSM1712374	GSE69914	LumB	LumA	LumA	LumA
GSM1712375	GSE69914	Her2	Normal-like	Normal-like	Normal-like
GSM1712376	GSE69914	LumB	LumA	LumA	LumA
GSM1712377	GSE69914	LumA	LumA	LumA	LumA
GSM1712379	GSE69914	LumA	LumA	LumA	LumA
GSM1712380	GSE69914	LumA	LumA	LumA	LumA

Page 1 of 62

Figure 3.2: Subtype prediction comparison between BCTypeFinder and baseline machine learning models (SVM, RF, LogReg). BCTypeFinder yielded a more balanced prediction across subtypes, particularly for Her2 and Luminal B.

In addition to supervised classification, the CancerSubtyper platform supports semi-supervised subtype discovery through CancerSubminer. In this evaluation, the number of subtypes was fixed to five to match the PAM50 categories, allowing for subtype refinement while preserving interpretability.

Figure 3.3 shows UMAP visualizations for the uncorrected dataset (upper left) and the CancerSubminer output (upper right). The baseline clustering methods K-means and NeMo are also shown for comparison (lower panels). While the original dataset exhibits strong batch effects and unclear subtype structure, CancerSubminer facilitated improved cluster

definition and reduced batch variation. Compared to K-means and NeMo, the platform's output via CancerSubminer demonstrated more biologically consistent and well-separated clusters.



Figure 3.3: UMAP visualization of the target dataset before and after applying CancerSubminer with manual 5-cluster configuration. Baseline clustering methods (K-means and NeMo) are shown for comparison. CancerSubminer provided better separation and balanced clustering across batches.

To support interpretability and quantitative comparison, the platform also provides clustering result summaries. As shown in Figure 3.4, CancerSubminer facilitated subtype as-

segment with a more balanced sample distribution across the five clusters. This output avoided the dominance of a single cluster—a common issue in K-means—and provided subtype groupings that align with known breast cancer biology.

Subtyping Results
This table displays the classification results for the given job. Each entry represents a prediction based on the selected model.

Sample	Batch	CancerSubminer	KMeans	NeMo
TCGA-A2-A0YK-01A-22D-A10A-05	Source	Cluster 1	Cluster 5	Cluster 4
TCGA-8H-A208-01A-11D-A161-05	Source	Cluster 2	Cluster 3	Cluster 5
TCGA-LL-A441-01A-11D-A244-05	Source	Cluster 2	Cluster 1	Cluster 3
TCGA-AO-A0JB-01A-11D-A10P-05	Source	Cluster 2	Cluster 4	Cluster 2
TCGA-AC-A2FK-01A-12D-A17Z-05	Source	Cluster 5	Cluster 1	Cluster 3
TCGA-C8-A8HR-01A-11D-A36K-05	Source	Cluster 5	Cluster 1	Cluster 3
TCGA-LD-A9QF-01A-32D-A41Q-05	Source	Cluster 3	Cluster 1	Cluster 3
TCGA-EW-A1P1-01A-31D-A14H-05	Source	Cluster 1	Cluster 1	Cluster 3
TCGA-8H-A28Q-01A-11D-A22B-05	Source	Cluster 5	Cluster 1	Cluster 3
TCGA-A2-A1G6-01A-11D-A13K-05	Source	Cluster 5	Cluster 1	Cluster 3

Page 1 of 168

Figure 3.4: Subtype prediction comparison between CancerSubminer, K-means clustering, and NeMo clustering under the 5-cluster setup. CancerSubminer achieved a more balanced and biologically meaningful subtype assignment.

3.3 Investigating CpG Clusters Associated with Each Subtype

To support biological interpretation of predicted subtypes, the CancerSubtyper platform provides a dedicated module for exploring CpG methylation clusters associated with each subtype. This feature allows users to interactively examine co-methylation patterns across the genome and assess their potential relevance to cancer subtyping. Unlike traditional

static outputs, the platform supports real-time filtering, selection, and visualization of CpG correlation structures tailored to specific molecular groups or batches.

Internally, the platform groups CpG sites into clusters during preprocessing using K-means clustering, based on the filtered and imputed beta value matrix. These clusters represent aggregated methylation features that reduce dimensionality while preserving signal relevant to subtyping. For correlation analysis, the Spearman correlation matrix is computed for the beta values of CpG clusters within each group of interest—defined by the combination of subtype and batch. From this matrix, the 30 most variable clusters are automatically selected using variance-based ranking. This selection ensures that only the most informative and dynamic features are visualized, while also maintaining computational efficiency.

The result is a 30x30 Spearman correlation matrix that highlights patterns of co-methylation between CpG clusters. This matrix is visualized as an interactive heatmap (Figure 3.5) directly in the platform. Each cell in the heatmap reflects the degree of correlation between a pair of clusters, ranging from -1 to 1, with color gradients indicating the strength and direction of correlation. Clusters with strong positive correlations (e.g., yellow blocks) suggest coordinated methylation behavior, which may indicate biological relevance such as shared regulatory pathways or chromatin proximity.

Users of the platform can dynamically filter the data by selecting a specific subtype, batch, or combination thereof. This flexibility allows for tailored exploration of both global trends and cohort-specific methylation structures. For example, users may choose to focus only on samples from the GSE69914 batch with the Luminal B subtype, and the platform will automatically regenerate the correlation matrix and visualization specific to that subset. This functionality enables fine-grained investigation of how methylation patterns differ across molecular and technical contexts.

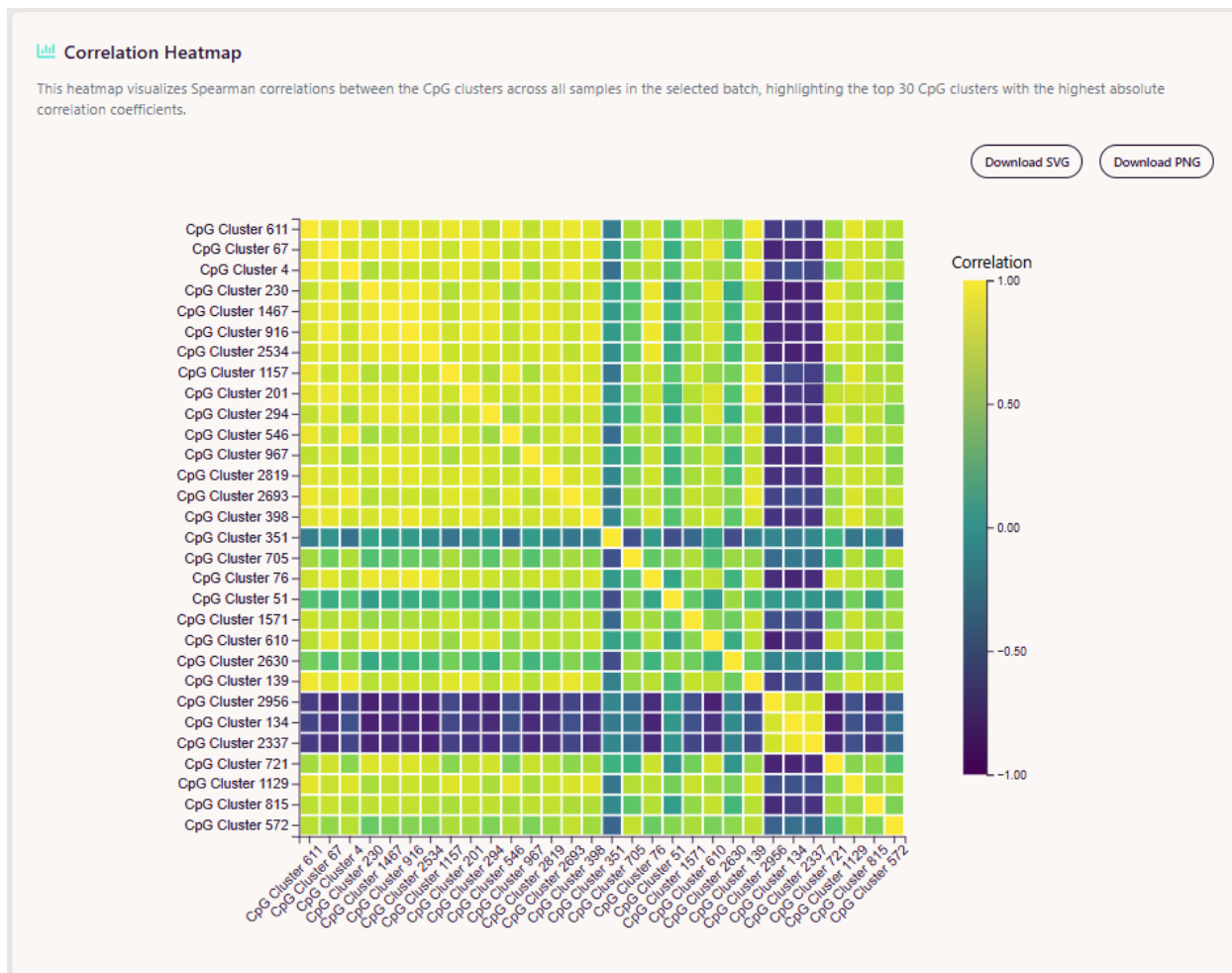


Figure 3.5: Interactive heatmap generated by the platform showing Spearman correlations between the top 30 CpG clusters within a selected batch and subtype. The heatmap highlights co-methylated clusters, which may reflect coordinated epigenetic regulation associated with subtype-specific pathways. Users can select the desired subtype and batch through a dropdown interface.

In parallel with the heatmap, the platform also provides an interactive metadata table listing all CpG sites that belong to the selected clusters (Figure 3.6). This table displays comprehensive genomic annotations for each CpG, including its cluster assignment, CpG ID, chromosomal position, strand orientation, UCSC gene mapping, and the reference genome build (GRCh37). The table is searchable and sortable, enabling users to filter by gene name, chromosome, or any other attribute of interest.

This feature serves as a bridge between methylation-based clustering and biological interpretation. For example, in one session, users identified clusters enriched for CpG sites located in or near genes such as *GPR39*, *S100B*, and *CBLC*, all of which have prior associations with breast cancer progression and tumor biology. The ability to cross-reference CpG clusters with known gene functions allows researchers to prioritize candidate features for downstream validation or hypothesis generation.

Details for CpG clusters extracted from the model

This table lists detailed attributes of the top 30 CpG clusters included in the correlation heatmap above.

Search... All Columns

Cluster	CpG	Position	Chromosome	Strand	UCSC Gene	Genome Build
4	cg07152925	48024683	21.0	-	S100B	GRCh37
4	cg26927807	2016532	19.0	+	BTBD2	GRCh37
4	cg07785936	133174635	2.0	+	GPR39	GRCh37
4	cg03292149	45279765	19.0	+	CBLC;CBLC	GRCh37
4	cg22780475	45281526	19.0	+	CBLC;CBLC	GRCh37
4	cg12188860	144416485	8.0	-	TOP1MT	GRCh37
4	cg15652212	142981776	7.0	+	TMEM139	GRCh37
4	cg12999109	6333587	19.0	-	ACER1;ACER1	GRCh37
4	cg14528319	14607713	19.0	+	GIPC1;GIPC1;GIPC1;...	GRCh37
4	cg15479752	35940862	19.0	+	FFAR2	GRCh37

< Prev Page 1 of 39 Next >

Figure 3.6: Interactive CpG metadata table displayed alongside the heatmap. Each row represents a CpG site from the top 30 clusters, with columns showing the cluster ID, CpG identifier, chromosomal location, strand, UCSC gene annotation, and genome build. The table supports interactive filtering and searching for targeted exploration.

From an implementation standpoint, the correlation heatmap generation is fully automated within the platform’s backend. As shown in the code snippet used for analysis, the process

involves reading the subtype-labeled beta value matrix, grouping data by subtype and batch, calculating the Spearman correlation matrix, and selecting the most variable features using variance-based filtering. This process is applied consistently across both source and target datasets, ensuring comparable correlation outputs regardless of model or data origin. The result files are stored in CSV format and visualized on the frontend through dynamic rendering components that allow users to download either PNG or SVG versions of the heatmap for reporting or publication.

By offering this visualization module, the platform not only supports subtype classification but also facilitates deeper biological interpretation of the underlying methylation signals. This makes CancerSubtyper a more comprehensive tool for researchers and clinicians aiming to study the epigenetic landscape of cancer.

3.4 CpG Methylation Intensity Across Subtypes

To facilitate the investigation of epigenetic variation among molecular subtypes, the CancerSubtyper platform provides an interactive boxplot module that visualizes the distribution of CpG methylation levels across subtypes. This feature enables users to assess methylation intensities at the cluster level and examine how they vary between distinct cancer groups, offering insights into potential subtype-specific regulatory patterns.

Each CpG cluster represents an aggregate of CpG sites grouped through unsupervised clustering during preprocessing. The methylation level of each sample is represented by its beta value, which ranges from 0 (completely unmethylated) to 1 (fully methylated). The platform integrates these values with corresponding subtype and batch labels, allowing users to explore methylation behavior from both biological and technical perspectives.

Figure 3.7 illustrates a representative boxplot rendered by the platform for one such CpG cluster. The x-axis shows the five canonical PAM50 subtypes—Normal-like, Luminal A, Luminal B, Basal-like, and Her2—while the y-axis displays the distribution of beta values for each subtype. The platform automatically assigns distinct colors to each subtype for improved clarity and pattern recognition. Users can easily switch between CpG clusters using a dropdown selector, and optionally filter the view by batch to isolate specific experimental cohorts.

This visualization supports comparative analysis of methylation profiles across subtypes. For instance, in the example shown, the Luminal B and Luminal A subtypes exhibit elevated median methylation levels in the selected CpG cluster, while the Basal-like group shows a broader interquartile range and more dispersed values. Such variations suggest potential epigenetic divergence and may hint at underlying differences in gene regulation or chromatin accessibility. The ability to observe and compare these distributions visually enables researchers to generate biologically relevant hypotheses about methylation-driven subtype characteristics.

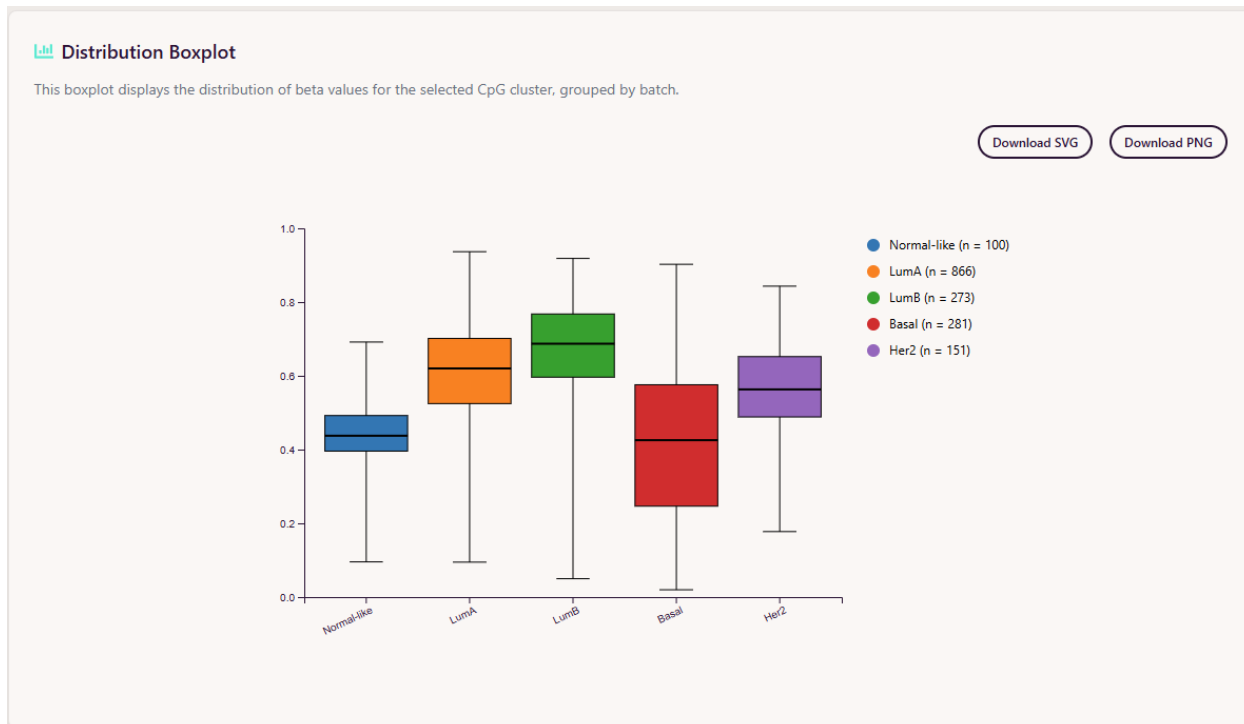


Figure 3.7: Beta value distribution for a selected CpG cluster across five breast cancer subtypes. The interactive boxplot allows users to explore subtype-specific methylation intensity. Each color corresponds to a distinct subtype. Users may select different clusters and toggle between batches to explore customized methylation patterns.

The backend implementation aggregates beta values from both source and target datasets, merging them with their respective subtype labels. The platform stores this preprocessed dataset and uses it as the basis for generating subtype-stratified boxplots. While the underlying code handles data merging and label alignment (as shown in the backend script), the emphasis of the platform is on delivering an intuitive, user-friendly frontend for visual analysis.

Importantly, this visualization feature is consistently available across both subtyping models integrated in the platform—BCtypeFinder and CancerSubminer. Regardless of which model is used to generate subtype predictions, the same methylation intensity exploration module can be applied to assess downstream biological patterns. This consistency supports cross-

model comparisons and helps ensure that results are interpretable and actionable.

Overall, this module supports both clinicians and researchers in understanding epigenetic trends across breast cancer subtypes, facilitating hypothesis generation, biomarker discovery, and broader interpretation of methylation-based classification results.

3.5 Survival Analysis by Predicted Subtypes

To support the evaluation of clinical significance associated with predicted cancer subtypes, the CancerSubtyper platform provides an interactive Kaplan–Meier (KM) survival analysis module. This component facilitates stratification of patient samples based on predicted subtypes and visualizes survival outcomes across different molecular categories. The visualization enables users—clinicians and researchers alike—to assess whether subtype assignments correspond to meaningful differences in overall survival (OS), thereby reinforcing the biological and translational utility of the classification models.

Figure 3.8 shows a representative KM plot produced within the platform. The curves represent the survival distributions for the five canonical PAM50 subtypes: Luminal A, Luminal B, Her2-enriched, Basal-like, and Normal-like. Each curve traces the proportion of patients who remain alive over time, as captured by the metadata provided by the user. The x-axis represents survival time, while the y-axis indicates the probability of survival. Subtypes are color-coded for clarity and annotated in the legend alongside their sample sizes.

In this instance, the Luminal A subtype shows the most favorable survival trajectory, consistent with clinical expectations. Conversely, Luminal B and Her2 subtypes show steeper declines, suggesting more aggressive disease behavior and lower survival probabilities. These observations support the clinical validity of the subtype predictions provided by the platform.

The CancerSubtyper platform automatically computes a log-rank test to assess the statistical significance of survival differences among the subtypes. The resulting p -value of 2.4×10^{-3} indicates that the observed divergence in survival outcomes is unlikely to be due to chance, confirming the relevance of the subtype separations from a prognostic standpoint.

This analysis module is dynamically linked to batch selection. Users can choose a specific batch for which valid metadata—containing both survival time and event status—is available. The platform processes this metadata and merges it with the predicted subtype labels to generate the KM plot. Only batches with survival annotations are eligible for this analysis, ensuring the plot accurately reflects real-world clinical data. Notably, there is no “all batches” option, as combining cohorts with disparate metadata would compromise statistical validity.

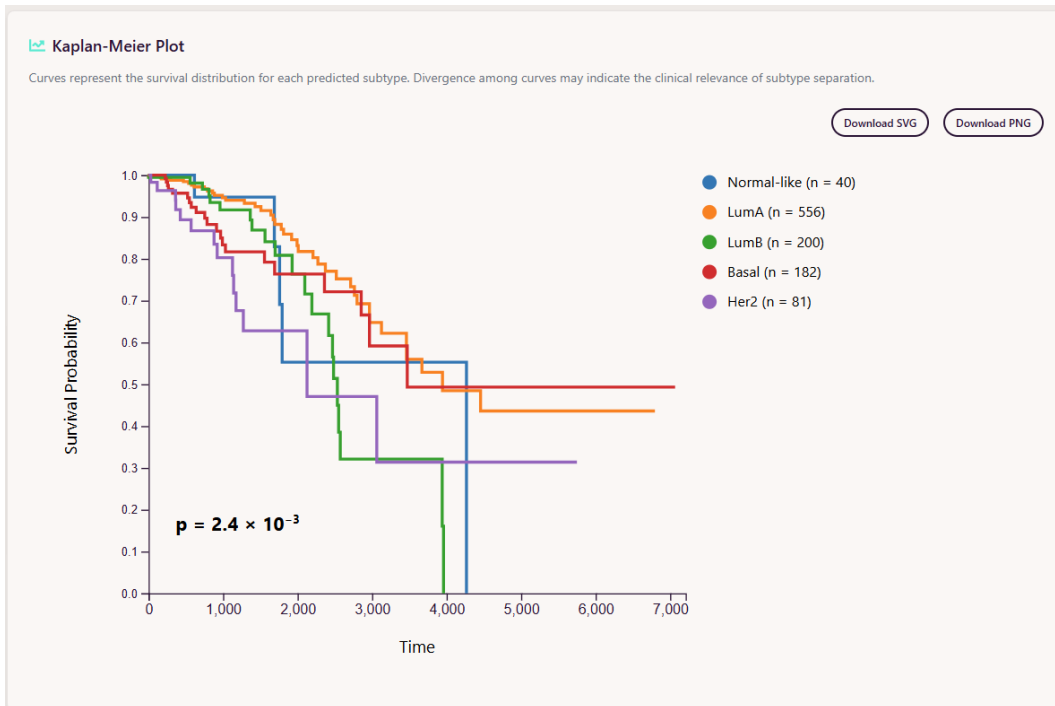


Figure 3.8: Kaplan–Meier survival curves for a selected batch, stratified by predicted subtypes. Subtypes exhibit distinct survival trajectories, with Luminal A showing the most favorable outcomes. Log-rank test yields a statistically significant p -value of 2.4×10^{-3} .

This KM analysis is supported for both integrated models—BCtypeFinder and CancerSubminer. Regardless of which model generated the subtype predictions, the survival plot can be rendered for any batch with available clinical follow-up. By providing this functionality, the platform not only classifies molecular subtypes but also supports downstream validation of their prognostic relevance, offering end-users a more holistic view of subtype outcomes. This feature is especially valuable for researchers seeking to link molecular subtyping with patient survival, as well as for clinicians interested in evaluating whether subtype assignments may inform treatment decisions or risk stratification.

Chapter 4

Discussions

The results presented in this chapter demonstrate that the CancerSubtyper platform effectively enables deep learning-based cancer subtyping using DNA methylation data. Both models integrated into the system, BCtypeFinder and CancerSubminer, produced meaningful outputs under a variety of evaluation settings, supporting the platform’s utility for both subtype prediction and discovery.

BCtypeFinder achieved balanced subtype predictions across the canonical breast cancer subtypes, with stronger performance in identifying Her2 and Luminal B subtypes compared to traditional machine learning models. Its domain adaptation process successfully aligned features across batches, as confirmed by the corrected UMAP visualizations where subtype clusters were more compact and batch-specific grouping was eliminated. These outcomes suggest that deep learning models that incorporate domain adaptation can overcome technical variability in cross-cohort studies more effectively than conventional classifiers.

CancerSubminer also performed well under both its automatic and manual clustering modes. In automatic mode, it discovered unsupervised clusters that aligned with known subtype structure, while preserving batch-invariant feature representations. In manual mode, where the number of clusters was fixed to five, the model further refined subtype boundaries and showed consistent distributions with clearer subtype separation. The ability to switch between discovery-driven and guided refinement makes CancerSubminer particularly flexible for exploratory studies and downstream biological interpretation.

The visualization components—CpG correlation heatmaps, beta value distributions, and Kaplan-Meier plots—further reinforced the interpretability of the results. In particular, the KM plot confirmed that predicted subtypes carried significant prognostic differences, with Luminal A associated with the highest survival probability. These visualizations help validate that the models are not just performing statistical pattern recognition, but are identifying biologically and clinically meaningful subgroups.

Computational Complexity and Scalability

In addition to predictive accuracy, an important consideration for CancerSubtyper is the computational feasibility of its workflow. Although Chapter 2 described the theoretical time and memory complexity of each stage, here we interpret how those requirements were experienced in practice.

Across experiments, preprocessing was generally fast and lightweight, with the exception of CpG clustering. When full Illumina 450K or EPIC arrays were used without filtering, clustering several hundred thousand CpGs into thousands of clusters could dominate preprocessing time and memory usage. This step became the primary bottleneck for very large datasets, though the cost was substantially reduced when CpGs were prefiltered by variance or when mini-batch clustering was applied.

Model training constituted the largest share of total runtime. The multi-stage design of BCtypeFinder and CancerSubminer—covering pretraining, adversarial training, semi-supervised refinement, and fine-tuning—means that even relatively compact networks required many hundreds of epochs to converge. On a modern GPU, training with a few thousand CpG clusters and mini-batch sizes of 128 to 256 was well within memory limits. The number of epochs, rather than the per-epoch cost, was the key determinant of runtime.

In practice, adopting learning-rate schedules with early stopping proved effective in reducing unnecessary training cycles.

Batch effect correction introduced a modest but noticeable overhead. The adversarial discriminator increased runtime by roughly 10–30% compared to baseline training, while subtype alignment calculations added little extra cost. The additional overhead was justified by the improved subtype separability observed in corrected embeddings, which confirmed that domain adaptation successfully mitigated cohort effects. When kernel-based penalties were tested, the quadratic scaling with batch size required us to use smaller mini-batches, which slowed training but remained manageable.

Overall, these results suggest that the CancerSubtyper workflow is computationally tractable for typical methylation datasets on a single modern GPU, provided that CpG clustering is constrained to a few thousand features. The main scalability challenges arise when attempting to process entire unfiltered arrays or when running long multi-phase training schedules. Both challenges can be mitigated through careful feature selection, the use of mixed-precision training, and the adoption of adaptive learning-rate strategies. These practical considerations will be especially important when extending the platform to larger pan-cancer studies or multi-omics datasets.

Model Hyperparameter Sensitivity

The performance of both BCtypeFinder and CancerSubminer depends on a small set of key hyperparameters, most notably the learning rate, the number of CpG clusters, and the strength of the adversarial loss used for domain adaptation. Understanding the sensitivity of the models to these settings is important for ensuring robustness and reproducibility.

Among all parameters, the learning rate proved to be the most sensitive. Rates that were too high caused unstable training and divergence, while rates that were too low led to slow convergence and poor embedding quality. Stable performance was generally observed within the range of 5×10^{-4} to 2×10^{-3} under the Adam optimizer. The number of CpG clusters also played a central role, as it defined the input dimensionality of the models. When fewer than 500 clusters were used, critical subtype-specific features were lost, resulting in weaker separation and poorer survival stratification. On the other hand, using more than 5,000 clusters increased runtime and memory cost without consistent gains in accuracy. Empirically, settings between 1,000 and 3,000 clusters offered the best tradeoff between biological signal and computational efficiency.

The adversarial loss required careful balancing as well. If the discriminator was weighted too strongly, embeddings aligned across batches but subtype boundaries collapsed, reducing biological interpretability. If weighted too weakly, batch effects persisted and biased predictions. In practice, running the discriminator and generator with matched update frequencies achieved a stable balance. Small adjustments to this ratio did not dramatically affect results, but extreme deviations degraded performance.

Hyperparameters in this study were tuned using a systematic grid search over plausible ranges, with validation on source data and, where possible, high-confidence target predictions. Both models were generally robust to modest deviations in these settings, producing consistent results across multiple runs. Future extensions of the platform may incorporate more advanced tuning strategies such as Bayesian optimization or Hyperband, which could reduce the computational cost of hyperparameter selection and further improve robustness.

Model Generalization Across Cancer Types

Another important question is how well the proposed models would generalize beyond the breast cancer datasets used in this study. DNA methylation landscapes vary substantially across different tumor types, and models trained on one cancer type may not transfer directly to another. As implemented, BCtypeFinder and CancerSubminer were evaluated exclusively on breast cancer data from TCGA-BRCA and external cohorts, so broader validation remains necessary.

For other cancers, several adjustments would be needed to maintain accuracy and biological relevance. The most straightforward strategy is to retrain each model on cancer-specific labeled data, using the same preprocessing and clustering pipeline but adapting the classifier to the new subtype labels. When labeled data is limited, transfer learning can be applied by reusing the lower layers of the feature extractor—which capture general methylation patterns—and fine-tuning only the higher layers for the new cancer type. Feature re-engineering may also be required, as some cancers exhibit different distributions of CpG variability; in such cases, the number of clusters or the prefiltering strategy could be adjusted to better capture subtype-specific signals. CancerSubminer is particularly well suited for generalization because of its semi-supervised design, which allows it to leverage unlabeled data and adapt to new subtype structures.

These strategies suggest that, while CancerSubtyper was benchmarked on breast cancer, its architecture is extensible to other tumor types. Systematic evaluation on additional TCGA cohorts and pan-cancer datasets will be an important future direction for confirming the generalizability of the platform.

Limitations and Future Directions

While the platform shows strong performance, there are a few limitations to acknowledge. Due to memory constraints, the target dataset was downsampled, which may affect the generalizability of results. In addition, CancerSubminer’s clustering results, particularly in automatic mode, may vary slightly between runs due to stochastic training elements. Lastly, KM survival analysis depends on the availability and quality of clinical metadata, which may not always be accessible.

Looking forward, the platform could be extended to support other cancer types and data modalities, such as gene expression or multi-omics integration. Additional model types and clustering strategies could be incorporated to provide alternative perspectives on subtype structure. From a usability perspective, integrating additional interactivity and interpretability features into the web interface may further enhance user experience. Overall, CancerSubtyper presents a robust, extensible foundation for accessible and reproducible cancer subtyping research.

Chapter 5

Conclusions

This thesis presents CancerSubtyper, a web-based platform for deep learning-based cancer subtyping using DNA methylation data. The platform is designed to enable researchers to upload labeled or unlabeled methylation data, run classification or discovery models, and interpret results with the help of interactive visualizations—all without needing to write any code. Two deep learning models, BCtypeFinder and CancerSubminer, were integrated into the system to support supervised and semi-supervised subtyping workflows, respectively.

The evaluation was carried out using cancer data from the TCGA BRCA cohort, under a consistent experimental setup. The BCtypeFinder model demonstrated strong performance in assigning intrinsic subtypes, particularly in recovering less dominant subtypes like Her2 and Luminal B. Its corrected UMAP visualizations showed clean separation by subtype, and its predictions aligned well with downstream survival analysis results. In comparison with traditional machine learning methods, BCtypeFinder produced a more balanced distribution of predicted subtypes and was less affected by batch artifacts.

CancerSubminer, tested in both automatic and manual subtype estimation modes, was able to discover and refine biologically meaningful clusters. In automatic mode, the model effectively removed batch effects and recovered interpretable groupings without relying on label information. In manual mode, subtype separation was improved, and the model leveraged known structure to produce more consistent results. When compared to NeMo and KMeans clustering, CancerSubminer produced the most compact and balanced cluster assignments.

The platform’s visual outputs, including UMAP projections, CpG heatmaps, beta value distributions, and Kaplan-Meier plots, support the interpretability and biological relevance of the predicted subtypes. The statistically significant survival differences observed across predicted groups further validate the utility of the subtyping pipeline.

Major Contributions

The main technical contributions of this thesis are summarized as follows:

1. **Development of CancerSubtyper:** Designed and implemented the first web-based platform for DNA methylation-based cancer subtyping that integrates deep learning models into a user-friendly and accessible workflow.
2. **Integration of dual modeling strategies:** Incorporated two complementary deep learning approaches—BCtypeFinder for supervised classification with domain adaptation, and CancerSubminer for semi-supervised discovery and refinement of cancer subtypes.
3. **Batch effect correction:** Applied adversarial domain adaptation and subtype alignment directly within model training, enabling robust cross-cohort and cross-platform analysis.
4. **Dimensionality reduction via CpG clustering:** Implemented large-scale CpG clustering to reduce hundreds of thousands of features into compact, biologically meaningful representations suitable for deep learning.
5. **Interactive visual analytics:** Delivered an integrated suite of interactive visualization modules (UMAP projections, CpG heatmaps, beta value plots, Kaplan–Meier

survival curves) to enhance biological interpretation and clinical relevance.

6. **Reproducible and scalable system architecture:** Deployed a full-stack, containerized platform using React, FastAPI, Celery, Redis, and Docker, ensuring reproducibility, scalability, and ease of deployment in research environments.

Limitations and Future Work

Despite these strengths, there are still several limitations. The models were evaluated on a downsampled subset of the full BRCA data due to hardware constraints, which may limit generalizability. Additionally, current support is limited to DNA methylation data and cancer subtyping. Expanding support to other omics data types and cancer types would be a logical next step. Finally, real-world deployment will require further testing under different user environments and dataset scenarios.

Looking ahead, future work may include incorporating multimodal data (e.g., RNA-seq or clinical variables), adding support for more cancer types, and developing improved clustering algorithms tailored to epigenetic data. The CancerSubtyper framework is also well-positioned to serve as a foundation for a broader bioinformatics platform supporting interactive model deployment, pipeline sharing, and community-driven collaboration.

Closing Remarks

In summary, this work demonstrates the feasibility and utility of combining deep learning and web technologies for cancer subtyping research. By lowering the barrier for computational analysis, CancerSubtyper has the potential to empower researchers to explore tumor

heterogeneity and translate methylation data into clinically meaningful insights.

Appendix A

Model Architecture and Hyperparameters

A.1 BCtypeFinder Architecture

BCtypeFinder consists of three major components: a feature extractor, a subtype classifier, and a domain discriminator. The model initiates by training on a labeled source dataset using a supervised learning approach to extract subtype-specific features. The architecture is composed of fully connected layers, each followed by ReLU activations and dropout layers to prevent overfitting. During domain adaptation, a domain discriminator is adversarially trained to align the source and target distributions, encouraging domain-invariant features.

A.2 CancerSubminer Architecture

CancerSubminer extends the BCtypeFinder framework with added components for subtype refinement and clustering. It consists of the following stages:

- **Pre-training:** The model is trained using cross-entropy loss on a labeled source dataset to initialize the feature extractor and classifier.
- **Clustering with Reassignment:** K-means clustering is applied to reassign low-

Table A.1: Shared hyperparameters used in BCtypeFinder and CancerSubminer.

Hyperparameter	Value
Optimizer	Adam
Learning Rate	0.001
Batch Size	64
Dropout Rate	0.3
Epochs	100
Confidence Threshold	0.95
K-means Initialization	K-means++
Number of Clusters (K)	5 (manual) / auto (silhouette-based)

confidence samples to majority subtypes within clusters, refining the subtype boundaries.

- **Adversarial Training:** A domain discriminator is trained to distinguish source and target domains, while the feature extractor aims to confuse it, thereby learning domain-invariant features.
- **Fine-tuning:** Pseudo-labels are iteratively updated for the target dataset and refined using subtype alignment and centroid matching.

A.3 Hyperparameter Settings

The following hyperparameters were used for training both models:

Appendix B

Preprocessing Pipeline Details

B.1 Input Files

The system accepts three primary input files:

- **Source Data:** DNA methylation data with subtype labels.
- **Target Data:** DNA methylation data without labels.
- **Metadata File:** Includes survival time and event status, used for Kaplan-Meier analysis.

B.2 Preprocessing Steps

1. **CpG Site Filtering:** CpG sites with more than 20% missing values are removed.
2. **Missing Value Imputation:** Median imputation is used for remaining missing values.
3. **CpG Clustering:** K-means clustering is applied to group CpG sites into 3000 clusters. The elbow method is used to estimate the number of clusters.
4. **Feature Extraction:** Cluster means are computed and used as model input features.

5. **Normalization:** Features are standardized across samples.

6. **Batch Effect Correction:** Domain adaptation via adversarial training and subtype alignment using centroid matching are applied.

This preprocessing pipeline ensures that both models operate on consistent and biologically meaningful features, while minimizing technical noise introduced by batch differences.

Appendix C

Platform Usage Instructions

C.1 Creating an Analysis Profile

To start an analysis, users must create a new profile on the CancerSubtyper platform. Each profile requires:

- A name and description
- Tumor type
- A source dataset (.csv.gz) containing subtype labels
- A target dataset (.csv.gz) without labels
- (Optional) A metadata file with survival time and event columns

C.2 Running Jobs

Once a profile is created, users may launch a job by selecting a model (BCtypeFinder or CancerSubminer). For CancerSubminer, users can choose to manually specify the number of subtypes or allow automatic estimation.

C.3 Viewing Results

After job completion, users can interactively explore:

- UMAP plots colored by subtype or batch
- Heatmaps of CpG cluster correlation
- Boxplots of beta value distribution
- Kaplan-Meier survival curves
- Cluster projections (CancerSubminer only)

C.4 Export Options

Users may download:

- All visualizations in PNG or SVG format
- Processed feature files, subtype predictions, and model comparison tables in CSV format

Bibliography

- [1] P. L. Bedard, A. R. Hansen, M. J. Ratain, and L. L. Siu. Tumour heterogeneity in the clinic. *Nature*, 501:355–364, 2013.
- [2] Christoph Bock. Analysing and interpreting dna methylation data. *Nature Reviews Genetics*, 13:705–719, 2012.
- [3] Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490:61–70, 2012.
- [4] et al. Chen. Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods. *PLOS One*, 2011.
- [5] et al. Chen. Hitaic: High-throughput tissue-of-origin analysis using integrated clustering. *Cell Reports Methods*, 2023.
- [6] Joung Min Choi, Kevin Choi, and Liqing Zhang. Bctypefinder: A deep learning framework for robust cross-cohort breast cancer subtype classification using dna methylation data. *Journal of Computational Biology*, 2024.
- [7] Andrew P. Feinberg and Bert Vogelstein. Epigenetics at the epicenter of modern medicine. *JAMA*, 299(11):1345–1350, 2008.
- [8] Ying Gao et al. Methycapsnet: Dna methylation-based cancer subtype classification with capsule network. *Briefings in Bioinformatics*, 23(6):bbac360, 2022.
- [9] Rudolf Jaenisch and Adrian Bird. Epigenetic regulation of gene expression: how the genome integrates intrinsic and environmental signals. *Nature Genetics*, 33:245–254, 2003.

- [10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.
- [11] Jeffrey T. Leek et al. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11:733–739, 2010.
- [12] Jonathan J. Levy, Matthew D. Titus, Jeffrey T. Leek, and Kasper D. Hansen. Methyl-net: an automated and modular deep learning approach for dna methylation analysis. *Bioinformatics*, 36(23):5445–5450, 2020.
- [13] Ronald L. B. T. C. Lin et al. Mexpress: a web tool for the integration and visualization of gene expression and dna methylation data. *BMC Genomics*, 16:636, 2015.
- [14] Andriy Marusyk and Kornelia Polyak. Tumor heterogeneity: causes and consequences. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1805(1):105–117, 2010.
- [15] Vivek Modhukur et al. Methsurv: a web tool to perform multivariable survival analysis using dna methylation data. *Epigenomics*, 10(3):277–288, 2018.
- [16] Joel S. Parker, Maggie Mullins, Maggie C. Cheang, Samuel Leung, David Voduc, Tammi Vickery, et al. Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of Clinical Oncology*, 27(8):1160–1167, 2009.
- [17] A. Portela and M. Esteller. Epigenetic modifications and human disease. *Nature Biotechnology*, 28:1057–1068, 2010.
- [18] David T. Ting et al. Epigenetics in cancer: a primer for clinicians. *The Lancet Oncology*, 17:e491–e492, 2016.
- [19] et al. Zhou. Deep learning model for cancer prognosis using dna methylation. *Bioinformatics*, 2021.