

***A Framework for Employing Probabilistic Topic Models on Gene Expression
Data***

***Αλγόριθμοι Πιθανοτικής Θεματικής Μοντελοποίησης για Ανάλυση Δεδομένων Γονιδιακής
Έκφρασης***

Thaleia Ntiniakou



Chania, 2019

***A Framework for Employing Probabilistic Topic Models on Gene Expression
Data***

**Αλγόριθμοι Πιθανοτικής Θεματικής Μοντελοποίησης για Ανάλυση Δεδομένων Γονιδιακής
Έκφρασης**

*A thesis submitted in fulfillment of the requirements
for the degree of Master of Science in*

Electronic and Computer Engineering

awarded by the

School of Electrical and Computer Engineering

at Technical University of Crete

Chania, November 2019

Thaleia Ntiniakou

COMMITTEE:

Supervisor : Georgios Chalkiadakis, Associate Professor, TUC

Committee Member : George Paliouras, Researcher NCSR Demokritos

Committee Member : Michalis Zervakis, Professor TUC



Technical University of Crete
School of Electrical and Computer Engineering

Abstract

One of the most important problems in computational biology is extracting knowledge and identifying patterns in real world biological datasets. In particular, microarray analysis experiments measure *gene expression*, the fundamental process by which gene products such as proteins are created, and which gives rise to the gene phenotype. Gene expression data can be analyzed to uncover genes or groups thereof, which are accountable for the development of specific diseases.

In this thesis, we employ *Probabilistic Topic Modeling (PTM)*, a category of unsupervised learning algorithms, for gene expression data analysis. PTM was first introduced and applied for extracting latent “topics” in text documents. Here we use them to uncover the genetic patterns responsible for biological processes and trigger specific diseases.

More precisely, this thesis contributes a generic framework that allows the use of any PTM algorithm of choice for gene expression data analysis. Our framework allows the incorporation of data preprocessing and transformation techniques, to permit the preprocessing of gene expression data into the “bag of words” paradigm, a format that the majority of Probabilistic Topic Models require as input. Following this potential data transformation, the PTM algorithm of choice is employed to extract probabilistic topics—that is, the hidden probability distributions (themes) over the genes (words), which govern the creation of biological samples (documents). The extracted topics are subsequently utilized for performing dimensionality reduction, particularly feature selection and feature extraction, of the most important features (genes), that characterize the dataset.

Finally, the framework comes complete with modern topics’ visualization techniques.

We populate our framework with various data transformation algorithms, and with two PTM techniques: *Latent Dirichlet Allocation (LDA)*, a well-established PTM technique, and *Latent Process Decomposition (LPD)*, an algorithm introduced specifically for the microarray setting. One of the data transformation algorithms we employ is novel, designed specifically for the task at hand. Moreover, we propose the novel use of two scoring methods (“KL-divergence” and “Relevance Score”) to assist our feature selection efforts.

We conduct a systematic evaluation of our techniques for feature selection and feature extraction tasks in this setting, using two real-world gene expression datasets—a recent dataset associated to muscle tissue conditions, and a frequently used breast cancer-related dataset. Overall, our results indicate that PTM algorithms can be quite successful in dimensionality reduction tasks in this setting, exhibiting performance that is usually at least comparable to that of the baseline algorithms used for evaluation; with the performance of LPD in feature extraction tasks being particularly noteworthy. Moreover, interesting conclusions on the efficacy of our various data transformation algorithms when combined with LDA are drawn in the process. Finally, this thesis demonstrates and helps underscore the fact that PTMs allow for the easy visualization of the hidden underlying genetic patterns at work in gene expression processes, and can therefore provide much needed assistance to biologists attempting to identify interesting classes of genes (i.e., carrying out gene annotation and enrichment analysis tasks).

Abstract in Greek

Ένα από τα πιο σημαντικά προβλήματα στην υπολογιστική βιολογία είναι η εξαγωγή γνώσης και ο εντοπισμός μοτίβων σε πραγματικά βιολογικά δεδομένα. Τα πειράματα με μικροσυστοιχίες γονιδίων, για παράδειγμα, αποσκοπούν στη μέτρηση της *γονιδιακής έκφρασης*, μιας θεμελιώδους διαδικασίας μέσω της οποίας δημιουργούνται παράγωγα των γονιδίων, όπως οι πρωτεΐνες, και η οποία δημιουργεί τον φαινότυπο του γονιδίου. Η ανάλυση δεδομένων γονιδιακής έκφρασης μπορεί να οδηγήσει στον εντοπισμό γονιδίων ή ομάδων γονιδίων τα οποία σχετίζονται με την εμφάνιση συγκεκριμένων ασθενειών.

Σε αυτή την μεταπτυχιακή εργασία, χρησιμοποιούμε *Πιθανοτική Θεματική Μοντελοποίηση (Probabilistic Topic Modeling - PTM)*, μια κατηγορία αλγορίθμων μη επιβλεπόμενης μάθησης, για την ανάλυση δεδομένων γονιδιακής έκφρασης. Αν και αυτές οι μέθοδοι πρωτοεφαρμόστηκαν για την εξαγωγή λανθάνοντων “θέματων” σε κείμενα, εδώ χρησιμοποιούνται για την ανακάλυψη γονιδιακών μοτίβων υπεύθυνων για βιολογικές διαδικασίες που μπορούν να πυροδοτήσουν συγκεκριμένες παθήσεις.

Πιο αναλυτικά, η παρούσα μεταπτυχιακή μελέτη συνεισφέρει ένα γενικό πλαίσιο εργασίας, το οποίο επιτρέπει την χρήση οποιουδήποτε PTM αλγορίθμου για ανάλυση δεδομένων γονιδιακής έκφρασης. Το πλαίσιο αυτό επιτρέπει την ενσωμάτωση τεχνικών προ-επεξεργασίας και μετασχηματισμού των γονιδιακών δεδομένων, ώστε να εκφραστούν σε συμφωνία με το πρότυπο αναπαράστασης κειμένου “σύνολο λέξεων (bag of words)”, το οποίο απαιτούν ως είσοδο οι περισσότεροι PTM αλγόριθμοι. Μετά από αυτόν τον (ενδεχόμενο) μετασχηματισμό των δεδομένων εισόδου, το πλαίσιο επιτρέπει την εκτέλεση του όποιου επιλεγμένου PTM αλγορίθμου ώστε να εξαχθούν τα “πιθανοτικά θέματα” (probabilistic topics), δηλαδή οι κρυφές πιθανοτικές κατανομές που ακολουθούν τα γονίδια (λέξεις), και οι οποίες διέπουν την δημιουργία βιολογικών δειγμάτων (κείμενα). Τα θέματα που έχουν εξαχθεί στην συνέχεια χρησιμοποιούνται για την μείωση των διαστάσεων του χώρου γνωρισμάτων, και πιο συγκεκριμένα την επιλογή και εξαγωγή των πλέον σημαντικών γνωρισμάτων (γονιδίων) που χαρακτηρίζουν τα βιολογικά δείγματα. Τέλος, το προτεινόμενο πλαίσιο επιτρέπει τη χρήση μοντέρνων εργαλείων για την οπτικοποίηση των εξαχθέντων θεμάτων.

Έχουμε ήδη υλοποιήσει και εντάξει στο προτεινόμενο πλαίσιο ένα σύνολο από τεχνικές μετασχηματισμού δεδομένων, καθώς και δύο αλγορίθμους PTM: τον *Latent Dirichlet Allocation (LDA)*, μια εδραιωμένη τεχνική PTM, και τον *Latent Process Decomposition (LPD)*, έναν αλγόριθμο που προτάθηκε σχετικά πρόσφατα στη βιβλιογραφία, συγκεκριμένα για ανάλυση μικροσυστοιχιών γονιδίων. Μία από τις μεθόδους μετασχηματισμού που χρησιμοποιούμε είναι εντελώς καινοτόμα, και σχεδιασμένη στα πλαίσια αυτής της εργασίας, συγκεκριμένα για το πρόβλημα που έχουμε να αντιμετωπίσουμε. Επιπλέον, προτείνουμε την καινοτόμα χρήση δυο γνωστών μετρικών (της “KL-divergence” και του “Relevance Score”), για να συνδράμουν στην επιλογή των γνωρισμάτων.

Διεξάγουμε μια συστηματική αξιολόγηση των τεχνικών για επιλογή και εξαγωγή γνωρισμάτων σε αυτό το πρόβλημα, χρησιμοποιώντας δυο πραγματικά σύνολα δεδομένων γονιδιακής έκφρασης— ένα σετ δεδομένων που σχετίζεται με ασθένειες μυϊκού ιστού, καθώς και ένα ευρέως χρησιμοποιούμενο σετ δεδομένων σχετικό με τον καρκίνο του μαστού. Τα αποτελέσματά μας εν γένει υποδεικνύουν ότι οι αλγόριθμοι PTM μπορεί να είναι αρκετά αποτελεσματικοί όσον αφορά την μείωση των διαστάσεων των δεδομένων σε αυτό το πρόβλημα, παρουσιάζοντας επιδόσεις που είναι συνήθως τουλάχιστον συγκρίσιμες με εκείνες γνωστών εναλλακτικών αλγορίθμων που χρησιμοποιήθηκαν για την αξιολόγηση. Η απόδοση του αλγορίθμου LPD συγκεκριμένα όσον αφορά το πρόβλημα “εξαγωγής γνωρισμάτων” (feature selection) είναι ιδιαίτερα αξιοσημείωτη. Επιπροσθέτως, η εργασία μας καταλήγει σε ενδιαφέροντα συμπεράσματα σχετικά με την αποτελεσματικότητα των διάφορων μεθόδων μετασχηματισμού των δεδομένων όταν αυτές συνδυάζονται με τον αλγόριθμο LDA. Τέλος, η μεταπτυχιακή αυτή εργασία εκτός των άλλων αναδεικνύει το γεγονός πως η χρήση PTM αλγορίθμων συμβάλλει στην οπτικοποίηση των κρυμμένων και υποβόσκοντων γενετικών μοτίβων που ενεργοποιούνται στην διαδικασία της γονιδιακής έκφρασης. Με βάση και αυτό το γεγονός, η χρήση του προτεινόμενου πλαισίου πιθανοτικής θεματικής μοντελοποίησης μπορεί να παρέχει σημαντική βοήθεια στους βιολόγους που επιχειρούν να αναγνωρίσουν ενδιαφέρουσες τάξεις γονιδίων (πραγματοποιώντας εργασίες γονιδιακού σχολιασμού και εμπλουτισμού).

Acknowledgments

First I would like to thank my supervisor, Professor Georgios Chalkiadakis, for all his support, inspiration and encouragement he provided throughout my studies at the Technical University of Crete. By urging me to pursue my dreams and entrusting me to work with multifaceted topics in Machine Learning, Multi-agent systems, and Game Theory, Prof. Chalkiadakis helped me open my scientific horizons, and trust my own capabilities. Moreover, I would like to thank him for all the opportunities and help he offered me.

Secondly, I would like to express my gratitude to Dr. Georgios Paliouras and Dr. William Duddy, for providing guidance and knowledge for this work and my studies, and Dr. Drakoulis Yannoukakos without whom I would not have had this collaboration. Also, I would like to thank Professor Michalis Zervakis, who as a member of my committee helped me complete my master's.

Moreover, I would like to thank my lab colleagues and friends Athina, Dia, Dimitris and Antonis for their help and support. I would also like to thank my friends Anna, Marilou, Yana, Iphigenia, Aris, Giorgos, Manolis and Io who supported me during my stay in Chania. Also, my friends from Athens, Eirini, Elly, Peggy, Sofia, Margarita, whose remote support guided me through tough times. I am also grateful for the family I made in Chania, Evina, her grandparents Nona, Stelios and her mother Artemis who welcomed me in their home.

Last but not least, I would like to thank my parents, Tasos and Xanthippi and my brother Stelios, without whose support and encouragement I wouldn't have managed to fulfil my studies. I could not let out my godmother Pola who has guided me from the beginning of my studies and urged me to follow the field of Computer Science.

Contents

Abstract	v
Abstract in Greek	vi
1 Introduction	1
1.1 Motivation	2
1.2 Contributions	2
1.3 Thesis Structure	3
2 Background	5
2.1 Microarray Analysis	5
2.1.1 Microarray Data	6
2.1.2 Comprehending Microarray Data	7
2.2 Dimensionality Reduction	8
2.2.1 Feature Selection	8
2.2.2 Feature Extraction	9
2.2.2.1 Principal Component Analysis	9
2.3 Solving a Data Analysis Task	9
2.3.1 Classification	10
2.3.1.1 Support Vector Machines	10
2.3.1.2 k Nearest Neighbors	11
2.3.2 Clustering	12
2.3.2.1 K-Means	12
2.3.2.2 Hierarchical Agglomerative Clustering	12
2.4 Probabilistic Topic Modeling	13
2.4.1 Latent Dirichlet Allocation	14
2.4.1.1 The Generative Process	15
2.4.1.2 Posterior Inference	17
2.4.2 Hierarchical Dirichlet Process	17
2.4.3 Latent Process Decomposition	20
2.5 Related Work	21
2.5.1 Dimensionality Reduction	21

2.5.2	Cluster Analysis	22
2.5.3	Classification	22
3	A framework on employing PTMs on Gene Expression Data	23
3.1	Transforming the Data into the Bag of Words Paradigm	25
3.1.1	Median	26
3.1.2	Repetition	27
3.1.3	Bibin: A Novel Transformation Method	29
3.2	Probabilistic Topic Models on Gene Expression Data	31
3.3	Dimensionality Reduction	32
3.3.1	Feature Selection Using Extracted Topic Models	33
3.3.1.1	Topic-Term Distribution	34
3.3.1.2	Relevance Score	35
3.3.1.3	Deferentially Expressed Genes using Kullback-Leibler Divergence	35
3.3.2	Feature Extraction Using Extracted Topic Models	35
3.3.3	PTM-Clustering Analysis (Ca) Visualizations	37
4	Experimental Setting	39
4.1	Data Preprocessing	39
4.1.1	The Datasets	39
4.1.2	Corpora Variants	40
4.1.2.1	Median	41
4.1.2.2	Repetition	42
4.1.2.3	Bibin	43
4.2	Experimental Setup	44
4.2.1	Topic Modeling Algorithms	44
4.2.2	Feature Selection using the extracted Topic Models	51
4.2.2.1	Unsupervised Feature Selection using LDA	51
4.2.2.2	Unsupervised Feature Selection Using LPD	72
4.2.3	Feature Extraction	86
4.2.3.1	Feature Extraction using LDA	86
4.2.3.2	Feature Extraction using LPD	95
4.2.4	PTM-cluster Analysis.Visualizing the Topic Models	99
5	Conclusions and Future Work	103
5.1	Conclusions	103
5.2	Future Work	104
A	Appendix Title	105
A.1	Sys-myo Dataset	105
A.1.1	Data Collection	105
A.1.2	Quality Control	105
A.1.3	Data Normalization	105
A.1.4	Batch Correction	106

A.1.5	Filtering expressed genes and annotating with gene symbols	106
A.2	Distributions of Documents for each Topic	106
A.3	Feature Selection using Topic Models Results	166
Bibliography		233

List of Figures

2.1	The Central Dogma of Molecular Biology [Dogma,]	6
2.2	The most common gene expression data matrix formats obtained by [Berrar et al., 2009].	7
2.3	Example of the intuition of Latent Dirichlet Allocation. obtained by [Blei, 2012a].	14
2.4	Probabilistic Graphical Model of Latent Dirichlet Allocation.	16
2.5	The tasks of a topic model in Bioinformatics.[Liu et al., 2016]	21
3.1	A framework on employing PTMs on Gene expression data.	24
3.2	Workflow for Transforming the data into BoW.	25
3.3	Workflow for implementing the Median Method.	26
3.4	Example of transforming a gene profile to a Bag of Words with Repetition Method.	27
3.5	Workflow for implementing the Repetition Method	29
3.6	Example of transforming a gene profile to a Bag of Words with Bibin Method.	30
3.7	Workflow for Dimensionality Reduction	32
3.8	Feature Selection using Extracted Topic models.	34
3.9	Workflow for Feature Extraction	36
4.1	The distribution over the samples for each topic on the Muscle Dataset using the Bibin method with $t_B = 0.2$	46
4.2	The distribution over the documents for Topic 11 in the MD Dataset using Bibin with $t_B = 0.2$	47
4.3	The distribution over the documents for Topic 4 in the MD Dataset using Bibin with $t_B = 0.2$	47
4.4	The distribution of the topics over the samples for the TCGA dataset using the Median method with $t_M = 0.2$	47
4.5	The distribution over the samples for each topic on the Muscle Dataset on the original GEM obtained by LPD.	49
4.6	The distribution over the samples for each topic on the TCGA Dataset on the original GEM obtained by LPD.	50
4.7	Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset.	56
4.8	Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Repetition variant ($t_1 = \times$, RemoveBin= \checkmark , $t_2 = \times$) on the GEM.	60
4.9	HAC Dendrogramm for UFS on Median variant on the MD.	64
4.10	HAC Dendrogramm for UFS on Median variant on the MD.	65
4.11	HAC Dendrogramm for UFS on Median variant on the MD.	66

4.12	HAC part of the Dendrogram for UFS on Repetition variant of MD Dataset on the Topic-Term matrix and number of clusters 9.	67
4.13	Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the TCGA with Median Variant with $(t_M = 0.2, b = 200, n_c = 3)$ on the Topic-Term Matrix.	69
4.14	Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the TCGA with Bibin Variant with $(t_B = 0.2, b = 200, n_c = 3)$ on the Topic-Term Matrix.	70
4.15	Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the TCGA with Repetition Variant with $(t_1=\mathbf{X}, \text{Remove Bin 1} = \mathbf{X}t_2 = \mathbf{X}, b = 50, n_c = 4)$ on the Relevance Matrix with $\lambda = 0.8$	71
4.16	Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset. . .	77
4.17	Unsupervised Feature Selection using LPD and Univariate Feature Selection on TCGA Dataset where $K = 3$ on the GEM.	79
4.18	HAC Dendrogram for UFS-LPD on the TCGA Dataset.	81
4.19	HAC Dendrogram for UFS-LPD on the MD Dataset.	82
4.20	HAC Dendrogram for UFS-LPD on the MD Dataset.	83
4.21	HAC Dendrogram for UFS-LPD on the TCGA Dataset.	85
4.22	Feature Extraction using LDA and on Repetition variant of MD Dataset.	92
4.23	HAC Dendrogram for Feature Extraction using LDA and on Bibin variant of MD Dataset. . . .	93
4.24	HAC Dendrogram for Feature Extraction using LDA and on Median variant of TCGA Dataset. .	94
4.25	The LDA topics visualized with pyLDAvis for the MD dataset with Median variant $t_M = 0.0$. .	99
4.26	The LDA topics visualized with t-SNE for the MD dataset with Media variant $t_B = 0.2$	101

List of Tables

2.1	A simple example of a Gene Expression Matrix	7
3.1	Cluster Analysis Visualization.	37
4.1	An visualized example of the gene expression matrix	40
4.2	Conditions and Anatomic Parts found in the Muscle Disease Dataset.	40
4.3	Transforming the Muscle Disease Data with Median Method.	41
4.4	Transforming the TCGA Breast Cancer Data with Median Method.	41
4.5	Datasets that occur after applying Repetition transformation method on the Muscle Disease Dataset.	42
4.6	Datasets that occur after applying Repetition transformation method on the TCGA Dataset.	42
4.7	Transforming the Muscle Disease Data with Bibin Method.	43
4.8	Transforming the TCGA Data with Bibin Method.	43
4.9	Machine Learning Algorithms implemented for in this work.	44
4.10	Topic Models implemented in this work.	44
4.11	Corpus Information and number of topics K inferred by HDP the Median Method and its Variants in the Muscle Disease Dataset.	45
4.12	Corpus Information and number of topics K inferred by HDP for the Repetition Method and its Variants in the Muscle Disease Dataset.	45
4.13	Corpus Information and number of topics K inferred by HDP for the Bibin Method and its Variants in Muscle Disease Dataset.	45
4.14	Classification results (average over all h values) using SVM for all threshold-defined “Median” transformation method variants in the MD dataset. Each row presents the best LDA results (across all genes selection/scoring metrics tried). Though Simple SVM (without feature selection) achieves best results for $t_m < 0.2$, SelectKBest does best for $t_m = 0.2$, which is the t_m value indicated by the biologist specialists providing the dataset. LDA FS has a performance that is comparable to that of SelectKBest.	53
4.15	Classification results (average over all h values) using SVM for all threshold-defined “Bibin” transformation method variants in the MD dataset. Each row presents the best LDA results (across all genes selection/scoring metrics tried). Though Simple SVM (without feature selection) achieves best results for $t_m < 0.2$, LDA FS does best for $t_m = 0.2$, which is the t_m value indicated by the biologist specialists providing the dataset.	53

4.16	Classification results (average over all h values) using SVM for all “Repetition” transformation method variants in the MD dataset. Each row presents the best LDA results (across all genes selection/scoring metrics tried). LDA FS achieves best results for $t_1 = 0$ and $t_1 = 0.2$. In the variant where no data was removed LDA FS again outperforms both SelectKBest and simple classification with SVM. In two particular variants (at which Bin 1 was removed) simple SVM achieves an accuracy that is only slightly better than that of LDA FS.	54
4.17	Classification accuracy results using SVM for all threshold-defined “Median” transformation method variants in the MD dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. Though Simple SVM (without feature selection) achieves best results for no t_M , SelectKBest does best for $t_m \geq 0.0$. LDA FS has a performance that is comparable to that of SelectKBest.	54
4.18	Classification accuracy results using SVM for all threshold-defined “Bibin” transformation method variants in the MD dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. LDA FS attains in the majority of certain combinations of h and K better accuracies than SelectKBest and simple SVM. The Data Type column indicates on which type of data (continuous or discrete variables) the SVM was employed.	54
4.19	Classification accuracy results using SVM for all “Repetition” transformation method variants in the MD dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. LDA FS attains in the majority of certain combinations of h and K better accuracies than SelectKBest and simple SVM. The Data Type (continuous or discrete variables) on which SVM was employed is also indicated.	55
4.20	Classification results (average over all h values) using SVM and KNN for all threshold-defined “Median” transformation method variants in the TCGA dataset. Each row presents the best LDA FS results (across all genes selection/scoring metrics tried). LDA FS accomplishes the best results using SVM as a classifier. When removing expression values below $t_m = 0.2$, LDA FS always achieves the highest classification accuracy.	57
4.21	Classification results (average over all h values) using SVM and KNN for all threshold-defined “Bibin” transformation method variants in the TCGA dataset. Each row presents the best LDA FS results (across all genes selection/scoring metrics tried). SelectKBest feature selection accomplishes the best results. LDA FS performance is entirely comparable to that of SelectKBest.	57
4.22	Classification results (average over all h values) using SVM and KNN for all “Repetition” transformation method variants in the TCGA dataset. Each row presents the best LDA FS results (across all genes selection/scoring metrics tried). When SVM is used for classification, LDA FS is consistently the best method, while when KNN is employed SelectKBest is usually the better method.	58
4.23	Classification accuracy results using SVM for all threshold-defined “Median” transformation method variants in the TCGA dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. LDA FS is consistently the best method.	58
4.24	Classification accuracy results using SVM for all threshold-defined “Bibin” transformation method variants in the TCGA dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. Classification results are almost the same so no conclusion can be made in this case.	58

4.25	Classification results (average over all h values) using SVM and KNN for all pipelined process-defined “Repetition” transformation method variants in the TCGA dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. SVM classification accuracy scores are higher for feature selection using LDA while KNN accuracies are higher for feature selection using SelectKBest algorithm.	59
4.26	Clustering scores using HAC for all threshold-defined “Median” transformation method variants in the MD dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_M and for each metric. SelectKBest achieves the highest NMI and RAND scores.	62
4.27	Clustering scores using HAC for all threshold-defined “Bibin” transformation method variants in the MD dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_B and for each metric. SelectKBest achieves the highest RAND scores. LDA FS achieves higher clustering scores where no data was removed.	62
4.28	Clustering scores using HAC for all pipelined process-defined “Repetition” transformation method variants in the MD dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_B and for each metric. SelectKBest achieves higher scores in most variants.	63
4.29	Clustering scores using HAC for all threshold-defined “Median” transformation method variants in the TCGA dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_M and for each metric. SelectKBest achieves the highest NMI and RAND scores.	67
4.30	Clustering scores using HAC for all threshold-defined “Bibin” transformation method variants in the TCGA dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_B and for each metric. SelectKBest achieves the highest NMI and RAND scores.	67
4.31	Clustering scores using HAC for all pipelined process-defined “Repetition” transformation method variants in the TCGA dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_B and for each metric-ranking. SelectKBest achieves higher scores in most of the variants.	68
4.32	Classification accuracy results (average over all h values) with SVM and KNN using LPD for feature selection (LPD FS) on the MD Dataset. Each row presents the classification accuracies for each metric and number of topics while having removed expression values below $t_{LPD} = 0$. LPD FS outperforms SelectKBest for each combination of classification-metric-number of topics.	73
4.33	Classification accuracy results (average over all h values) with SVM and KNN using LPD for feature selection (LPD FS) on the MD Dataset. Each row presents the classification accuracies for each metric and number of topics while having removed expression values below $t_{LPD} = 0.2$. LPD FS outperforms SelectKBest for each combination of classification-metric-number of topics.	74
4.34	Classification accuracy results (average over all h values) with SVM and KNN using LPD for feature selection (LPD FS) on the MD Dataset. Each row presents the classification accuracies for each metric and number of topics while not having removed any data ($t_{LPD} = \mathbf{X}$). LPD FS outperforms SelectKBest for each combination of classification-metric-number of topics.	75
4.35	Classification results with SVM and KNN using LPD for feature selection (LPD FS) on the MD dataset, for specific values of h and for each metric in which LDA FS achieved the best accuracy score. This table depicts classification accuracies for certain values of h in which LPD FS achieved the best accuracy score for each metric and number of topics, while $t_{LPD} = \mathbf{X}$. LPD FS outperforms SelectKBest for almost every combination of classifier-metric-number of topics.	75

4.36	Classification results with SVM and KNN using LPD for feature selection (LPD FS) on the MD dataset. This table depicts classification accuracies for certain values of h in which LPD FS achieved the best accuracy score for each metric and number of topics, while $t_{LPD} = 0$. LPD FS outperforms SelectKBest for each combination of classification-metric-number of topics.	76
4.37	Classification results using SVM and KNN for feature Selection using LPD in the MD dataset. This table depicts classification accuracies for certain values of h in which LPD FS achieved the best accuracy score for each metric and number of topics, while $t_{LPD} = 0.2$. LPD FS outperforms SelectKBest for each combination of classification-metric-number of topics.	76
4.38	Classification accuracy results (average over all h values) with SVM and KNN using LPD for feature selection (LPD FS) on the TCGA Dataset. Each row presents the classification accuracies for each metric and number of topics. SelectKBest outperforms LPD FS for each combination of classifier-metric-number of topics.	78
4.39	Classification results with SVM and KNN using LPD for feature selection (LPD FS) on the TCGA dataset. This table depicts classification accuracies for certain values of h in which LPD FS achieved the best accuracy score for each metric and number of topics. LPD FS and SelectKBest exhibit comparable performance in most cases.	78
4.40	Clustering scores using HAC for all threshold-defined variants of LPD in the MD dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_{LPD} and for each metric. LPD FS achieves the highest NMI and RAND scores.	80
4.41	Classification results using SVM and KNN for all “Median” transformation method variants in the MD dataset. Each row presents the classification accuracies obtained after feature extraction is performed with LDA and PCA, for each of the different values of t_M and the corresponding number of conditions (classes) the classifier should predict. Feature Extraction using PCA outperforms LDA in most cases.	87
4.42	Classification results using SVM and KNN for all “Bibin” transformation method variants in the MD dataset. Each row presents the classification accuracy obtained after feature extraction is performed with LDA and PCA, for each of the different values of t_B and the corresponding number of conditions (classes) the classifier should predict. Feature Extraction using PCA consistently outperforms LDA.	88
4.43	Classification results using SVM and KNN for all “Repetition” transformation method variants in the MD dataset. Each row presents the classification accuracies obtained after feature extraction is performed with LDA and PCA, for each of the different values of the thresholds ($t_1, \text{RemoveBin1}, t_2$). Feature Extraction using PCA outperforms LDA in most cases.	89
4.44	Classification results using SVM and KNN for all “Median” transformation method variants in the TCGA dataset. Each row presents the classification accuracies obtained after feature extraction is performed with LDA and PCA, for each of the different values of t_M . Feature Extraction using LDA outperforms PCA.	90
4.45	Classification results using SVM and KNN for all “Bibin” transformation method variants in the TCGA dataset. Each row presents the classification accuracies obtained after feature extraction is performed with LDA and PCA, for each of the different values of t_B . PCA outperforms LDA. . .	90

4.46	Classification results using SVM and KNN for all “Repetition” transformation method variants in the MD dataset. Each row presents the classification accuracies obtained after feature extraction is performed with LDA and PCA, for each of the different values of the thresholds (t_1 , RemoveBin1, t_2). Feature Extraction using LDA outperforms PCA.	91
4.50	Classification Results with SVM and KNN using LPD FE on the TCGA Dataset. Each row depicts the classification accuracy depending on the number of topics. LPD FE achieves the best results. .	95
4.47	Classification Results with SVM and KNN using LPD FE on the MD Dataset. Each row depicts the classification accuracy depending on the number of conditions (classes) to predict) and the number of topics, while no data has been removed ($t_{LPD} = \mathbf{X}$). PCA performs better than LPD FE, although LPD FE achieves comparable results.	96
4.48	Classification Results with SVM and KNN using LPD FE on the MD Dataset. Each row depicts the classification accuracy depending on the number of conditions (classes) to predict) the number of topics, while $t_{LPD} = 0.0$. PCA performs better than LPD FE, although LPD FE achieves higher classification results when predicting all conditions.	97
4.49	Classification Results with SVM and KNN using LPD FE on the MD Dataset. Each row depicts the classification accuracy depending on the number of conditions (classes) to predict) the number of topics, while $t_{LPD} = 0.2$. PCA performs better than LPD FE, although LPD FE achieves higher classification results when predicting all conditions regardless of the number of topics.	98

Introduction

The amount of biological data that is available since the last century, has seen an exponential increase. This raises two issues: the first concerns that the data should be stored and handled efficiently. Furthermore, the potential is increased to interpret these data and extract underlying undiscovered information. The later issue falls in the jurisdiction of *Computational Biology*, which aims at developing tools and approaches for analyzing the, in other respects, heterogeneous data, in a way that we will not simply describe them but create models that are able to perform predictive modeling.

An interesting application of computational methods in the field of biology is managing and interpreting rather complex experimental data that evolve from biological experiments. One of the most common type of experiments that produce this type of data are *microarray analysis*. The data that are generated by a microarray experiment pose two challenges: the data should be pre-processed in order to meet a form that can be later fed to a machine learning algorithm, and the analysis of the data that depends on what we are looking to find out. To interpret the results of microarray data, scientist need to identify expression patterns in genes, to classify biological samples or underlying biological processes.

When data needs to be analyzed to extract information, *Machine Learning (ML)*, a sub-field of *Computer Science* that aims at developing algorithms that can learn how to make predictions of data, as well as comprehending massive amounts of information [Bishop, 2006], is used. Biologists employ ML-extracted information to further analyse it and link specific classes of genes expression patterns to disease phenotypes, a process also known as *enrichment analysis*.

Two fundamental classes of ML techniques are used in the literature; *Supervised* and *Unsupervised Learning* [Mitchell et al., 1997]. The most commonly used one, *Supervised learning*, includes models that learn a classification or an estimation function. In recent years, *Unsupervised methods* are also used, to examine the data in order to find previously unperceived consistencies and relationships.

Unsupervised learning, is an intriguing category in machine learning, since no ground truth is provided to the model regarding the data and their category. Consequently, these algorithms can model the data in such way that can lead to bring to light patterns that are difficult to be perceived by eye. Probabilistic Topic Models (PTMs), are statistical models that fall in this category and intend to uncover hidden structure of data. These models were first introduced in text mining, to discover “latent topics” found in large collections of documents [Blei, 2012a]. The basic intuition behind these models, is that they intend to reproduce the process which generate the data, and extract topics, in a form of distributions over a vocabulary. These algorithms infer the proportion of each topic in a text document (newspaper or scientific article, chapter in a book etc.). However, since their emergence, they have been widely used in other applications as well. For example, in our recent

work [Georgara et al., 2018], we employed PTMs in order to learn latent preference relationships in *Hedonic Games with Dichotomous preferences*, a class of games in *Cooperative Game Theory*.

1.1 Motivation

In general, the development of increasing computational power is constantly leading to testing and establishment of machine learning models, that were proposed in the previous century. Another aspect of the growing development of machine learning approaches as solutions to problems, is that nowadays data is easily stored and accessed. Biological data are not excluded from this evolution. However, computational biology has not reached its limits, regarding the information scientists can derive by combining statistical and mathematical solutions to this field.

In this thesis, we turn our attention to Probabilistic Topic Models [Blei, 2012a], and how we can capitalize on their potentials in microarray experimental data. PTMs have been employed on microarray data throughout the bibliography [Liu et al., 2016] on various types of experiments and extracted different types of information. Microarray experiments produce *gene expression* data, that is information regarding the level of expression of thousands of genes. As a result, we were motivated to use PTMs on this type of data to distinguish genes (or groups of genes) whose expression level can be associated to a disease, or a biological process. Our incentive is to focus on how these models can be adapted to this setting, how we will be able to interpret the results, use them to perform various data analysis tasks, and provide specialists the ability to comprehend and analyse the extracted knowledge. Moreover, we apply these models on a dataset obtained by [Malatras et al., 2019], on which PTMs have not been yet applied. This dataset consists of samples from muscular tissues of patients that carry various myopathies, hence a multiclass problem. More technical information regarding the dataset can be found A.1.

1.2 Contributions

This thesis, motivated by the previously mentioned aspects, approaches probabilistic topic models in order to adopt them in detail in our problem. To do so we propose a generic framework. At first, we apply different well-established techniques to transform our data in the form that the majority of PTMs require. In this context, we use methods to discretize the continuous data. Consequently, we are able to compare the discretization techniques, and later on evaluate their impact on the extracted topic models. We propose a novel discretization method, Bibin that was designed to tackle the problem at hand. Its performance is comparable to the others, but as we will show, it depends on the variance of the dataset. Furthermore, we alter a common discretization technique *Repetition*, by using a pipe-lined process to examine how the removal of data affects the results. With our proposed framework, PTMs are used as a dimensionality reduction technique, and their use allows the easy visualization of important information (i.e., disease-related topics), and the subsequent handling of that information by biologists for further analysis and interpretation.

We contribute to this problem, by introducing scores applied on the results of topic models, to extract further results. In particular, we show promising results by introducing a novel way of selecting subsets on features based on the topic-term distribution which is extracted by topic modelling algorithms. By employing different topic modelling algorithms, we evaluate their impact and performance on both feature selection and feature extraction. We depict how the performance is altered by using different topic models, and different discretization techniques. We provide numerous visualizations of the results of the topic models on the muscle disease dataset, in order to extract knowledge based on the results of the topic models.

In a more abstract manner, this framework shows how microarray analysis can benefit from the power of Probabilistic Topic Models. In particular, to be able to use PTMs with microarray experiments, we provide a correlation of documents and words to biological samples and genes. We indicate how the expression values of genes can be translated to word frequencies. The proposed framework then employs probabilistic topic models to perform *Cluster Analysis* and *Dimensionality Reduction*. In a way we use our framework to perform biclustering, namely simultaneously cluster “similar” genes and samples. Moreover, through dimensionality reduction and particularly, feature selection, a ranking of important genes is provided, which can be further used for enrichment analysis.

1.3 Thesis Structure

This thesis is organized as follows. In Chapter 2 we provide the necessary background for this thesis, and discuss related work. Subsequently, in Chapter 3 we present our proposed framework and implemented data transformation, probabilistic topic modelling, and dimensionality reduction methods. Then, in Chapter 4 we evaluate our framework and accompanying methods. Finally, Chapter 5 concludes this thesis, and outlines future research directions.

Background

In this section, we provide the necessary background knowledge, and we will walk through some baseline notions of Microarray analysis, Machine learning, and Probabilistic Topic Models. In what follows, firstly we discuss briefly the definition of a microarray experiment, the notion, and interpretation of its results. Later on, we present dimensionality reduction techniques and machine learning algorithms that we utilized to analyze our data. We also explore the basic notation of Cooperative Game Theory which will be utilizing for dimensionality reduction. Ultimately, we briefly denote previous work on employing Probabilistic Topic Models on biological data.

2.1 Microarray Analysis

Proteins are the cells' and tissues' structural elements and execute many of the biological system's crucial functions. Protein production is regulated by genes coded in deoxyribonucleic acid (DNA), common to all cells in one being, and mostly immutable throughout one's lifetime [Parmigiani et al., 2003]. The definition of a gene is intricate in the field of classical genetics, but briefly recounted in [Pearson, 2006] a gene is an abstract concept, a unit of inheritance that ferries a characteristic from parent to child. The production of proteins from genes involves two main stages, transcription and translation, which compose the Central Dogma of Molecular Biology [Crick, 1958]. A single strand of messenger ribonucleic acid, or mRNA, is transferred from the gene-coding DNA section during transcription. mRNA is used as a template for assembling a chain of amino acids to create the protein after transcription. This flow of information is also known as *gene expression* (GE). The mechanism of gene expression is a procedure that regulates cellular processes such as growth, maintenance, and response to stimuli. Simply put, GE is a process by which information encoded within a gene is used to synthesize a functional gene product. What is important to conceive, is the fact that we can perform various measurements while this process is conducted, and identify how active a gene is at a particular point.

A DNA-Microarray experiment (or analysis) is a novel tool in molecular biology, capable of performing these measurements simultaneously on hundreds or even thousands of gene transcripts of an mRNA molecule of a given cell or tissue sample. During these experiments, biologists are able to identify which genes are expressed or not and their level of expression. In the case where a gene is expressed, it is possible that it mutates, a situation that can lead to triggering the cell to become abnormal. This information facilitates doctors in diagnosing and treating diseases.

Microarray analysis is used in order to conduct various types of studies that can assist in extracting knowledge [Tarca et al., 2006]. Firstly, the results of microarrays can identify co-expressed genes, either in a subpopu-

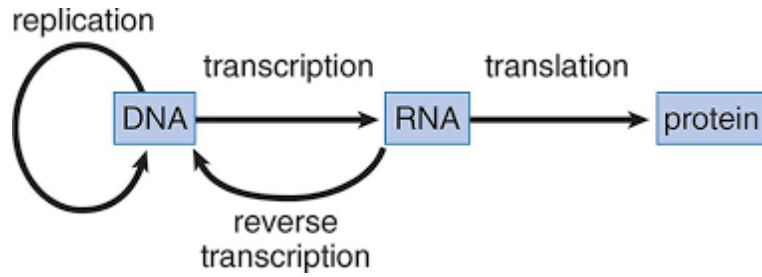


Figure 2.1: The Central Dogma of Molecular Biology [Dogma,]

lation or as genes that are always expressed together, or genes whose expression patterns facilitate in distinguishing between biological entities that are otherwise strenuous to discriminate. Moreover, these tools can assist in identifying gene activity patterns under various stress conditions, such as chemical or in general drug treatment. Although in many biological studies microarrays have been applied, it is not trivial to handle and analyze the large volumes of generated data. Later on, we will explain these types of studies which can be performed on Microarray Data, but first, we describe the results of a microarray experiment.

2.1.1 Microarray Data

By using a variety of experimental techniques, for example, DNA microarray or RNA-seq, we can concurrently measure the expression of multitude of genes. A microarray experiment produces at first raw data that need to be preprocessed. We will not be engaging in this procedure since the data being used in this thesis have been already preprocessed. After manipulating the data, the final form of a microarray experiment results to a matrix referred as *gene expression matrix* (GEM). Before defining a gene expression matrix, someone must understand the concept of the *expression profile*. As stated in [Berrar et al., 2009] an *expression profile* has two types; it depicts the expression values for an individual gene across many tissue samples *and* for many genes in one condition or tissue sample. In order to differentiate these two categories of gene expression, we adopt the following terms:

- *One gene over multiple samples*: The expression values for one gene over multiple samples is defined as *gene profile*.
- *Many genes over one sample*: An *array profile* is comprised of gene expression values for multitude of genes under a specific condition or tissue sample.

Considering the above terms:

Definition 1 (Gene Expression Matrix (GEM)). *A gene expression matrix (GEM) is a matrix, whose one dimension represents the number of biological samples on which gene expression was performed, and the second dimension corresponds to the number of genes measured simultaneously. Bearing in mind, that the number of tissue samples is M and the number of genes is N , a GEM is $M \times N$ matrix where given sample i and gene j , the matrix entry is a real value corresponding to the gene expression level or value (“gev”) of gene j in sample i , denoted by e_{ij} . The horizontal axis of the matrix is an array profile, whereas the vertical axis represents gene profiles.*

It can be observed in Figure 2.2 that a GEM can be illustrated in two forms; one that places the genes vs the samples, and its transposed form. As it emerges from the definition of a gene expression matrix, in this work, we will be adopting the diagram on the right side of Figure 2.2.

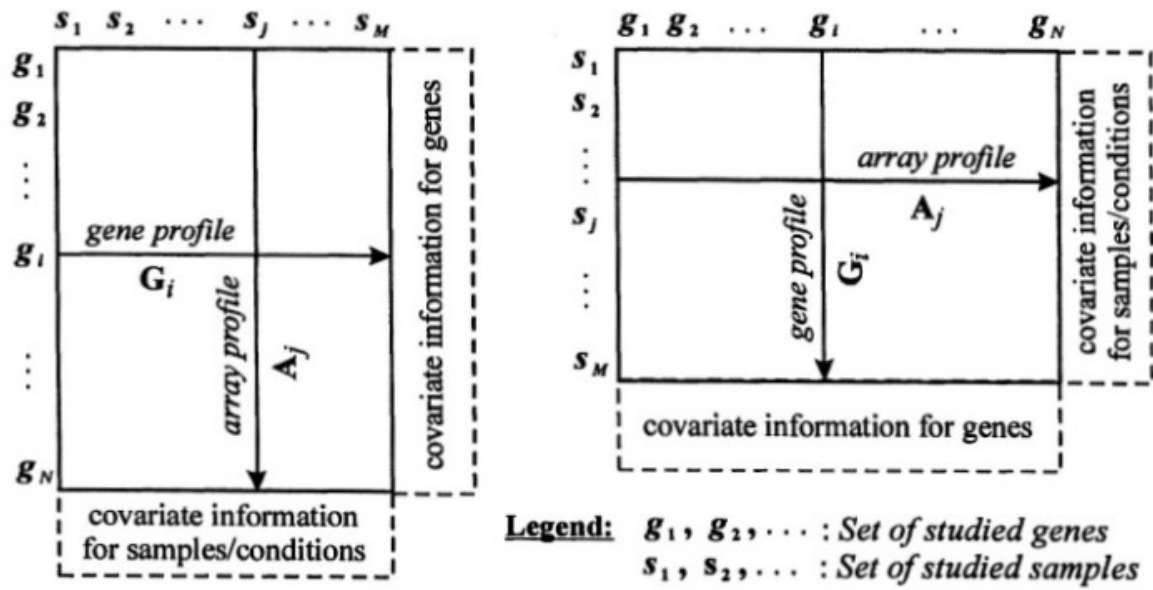


Figure 2.2: The most common gene expression data matrix formats obtained by [Berrar et al., 2009]. The actual values are included inside the bold boxes. The covariate information can vary from tissue and condition type or other clinical information.

Table 2.1: A simple example of a Gene Expression Matrix

samples\genes	gene ₁	...	gene _N
sample ₁	$e_{1,1}$...	$e_{1,N}$
...
sample _M	$e_{M,1}$...	$e_{M,N}$

2.1.2 Comprehending Microarray Data

As mentioned priorly, analyzing a microarray experiment can lead to numerous types of conclusions and results. In this part, we will describe which types of analysis we can conduct on a GEM, or simply put what questions we need to answer [Berrar et al., 2009]. Later in this chapter, we introduce the methods that are used in this work to answer these questions.

Differential Gene expression studies

These studies involve the investigation of genes that exhibit different expression levels under various experimental conditions, such as various stages of the development of an organism, or normal in contrast to diseased tissues. These studies examine a single *gene profile*.

Gene Co-regulation Studies

In this case, the objective is to compare gene profiles with each other, rather than studying a single profile. The aim is to discover whether there exist genes with either a *positive* or *negative* pattern of co-regulation. That is, if the increment of the expression level of a gene, occurs at the same time with the increment of another gene, or the expression level of a given gene increases when the other decreases, respectively.

Clinical Diagnosis

In this case, we wish to discover characteristic gene expression patterns for a particular disease. In addition to that, microarrays can be a powerful tool to reveal unknown subtypes of a disease.

Gene Function Identification Studies

Microarray experiments can serve to discover a novel gene's functionality. That is, a previously unknown gene is exposed to different conditions and its' expression profile is compared to the expression profile of others. Profiles with high similarity to the novel gene serve as candidates for inducing its function.

2.2 Dimensionality Reduction

In the modern days, the amount of data that researchers have at hand is massive. Data can be easily obtained and stored, due to the increasing capabilities of databases. The number of samples and features in a dataset can be very large. Although a large number of samples can only increase the performance and accuracy of a machine learning algorithm, the high dimensionality of feature space can lead to the opposite result [Verleysen and François, 2005]. The high dimensionality of the feature space can lead to datasets that contain features that are irrelevant, redundant and can lower the performance in accuracy and time of an algorithm that analyses them. This can be thought of as the expression of the well known "curse of dimensionality" problem [Bellman, 2015]. Particularly, in the microarray analysis setting, it is common that the number of samples is very smaller proportional to the number of genes, which can be thought of as the "curse of dimensionality" [Aziz et al., 2017]. For this reason, before performing any data analytical task, a well-established process is to reduce the dimensionality of the feature space. Dimensionality reduction is a technique used to solve the "curse of dimensionality". Selecting a subset of the initial features, a methodology known as feature selection is one solution to this problem. Another way to resolve this is to perform feature extraction, which projects a large number of features to fewer dimensions of the original data. We describe these techniques in term below.

2.2.1 Feature Selection

In machine learning and data mining, feature selection or variable selection is the process of selecting a subset of the initial features of the data set, in order to achieve better accuracy and speeding up the execution time of a classifier. The incentive is to retain the relevant and discard the redundant features based on a criterion or methodology [Lazar et al., 2012]. When performing feature/gene selection on a microarray experiment, two categories of algorithms are usually employed, *Filter* and *Wrapper* [Guyon and Elisseeff, 2003]. Filter methods make use of variable ranking methods. These feature selection algorithms do not depend on the classifier that will be trained. However, they are supervised, as the selection of features is performed based on their scores in statistical tests in correlation with their output class. Each feature is examined individually, in order to decide how relevant or redundant it is, in correlation to the class variable. Various scores can measure the significance of a feature, for example Pearson Correlation, Mutual Information, chi-squared tests or the Analysis of Variance F-Test [Guyon and Elisseeff, 2003]. On the other hand, wrapper methods perform feature selection based on the performance of a classifier [Tarca et al., 2006].

2.2.2 Feature Extraction

Feature extraction aims at reducing the feature space. That is, we do not eliminate features but create a new projected space. Feature extraction creates new features as combinations of others to reduce the high dimensionality of the data to a significantly smaller space, resulting in better accuracies in learning a model [Hira and Gillies, 2015]. A well-established method for extracting features is Principal Component Analysis (PCA).

2.2.2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a widely established algorithm for performing feature extraction, as it is very simplistic efficient and non-parametric. Dimensionality reduction is achieved by projecting the input data into a lower-dimensional linear space, also referred to as principal subspace, while it maximizes the variance of the projected data. Supposing that the data consist of $K \times D$ observations, where K is the number of observations and D corresponds to the dimensionality of each observation. As noted earlier, the incentive of PCA is to project the data into a principal subset of dimensionality $N \leq D$. In what follows, we provide a brief explanation of how PCA calculates the subspace [Bishop, 2006]. We define as $X = [x_1, \dots, x_K]^T$ the observation matrix. Let the projection of x_k , where $k \in 1 \dots K$, to a one-dimensional space, be described by a direction vector $u_1 \in \mathbb{R}^D$. The projection of x_k is derived by :

$$x_{k \perp u_1} = \frac{u_1^T x_k}{||u_1||} \quad (2.1)$$

Where the mean and variance of the projected data is given in Equations 2.2 and 2.3

$$\bar{x}_{\perp u_1} = \frac{u_1^T \bar{x}}{||u_1||} \quad (2.2)$$

$$\sigma_{\perp u_1}^2 = \frac{1}{||u_1||} u_1^T S u_1 \quad (2.3)$$

$$S = \left[\frac{1}{K} \sum_{k=1}^K (x_k - \bar{x})(x_k - \bar{x})^T \right] \quad (2.4)$$

By solving an optimization problem using Lagrangian multipliers and decomposition into eigenvectors and eigenvalues, PCA can find the principal subspace. A deeper insight into the algorithm can be found in [Bishop, 2006]. In a nutshell, principal component analysis implicates evaluating the mean \bar{x} and the covariance matrix S of the data X and then finding the N eigenvectors of S corresponding to the N largest eigenvalues.

2.3 Solving a Data Analysis Task

To this point, we have described the foundations of Microarray analysis, what is a gene expression matrix, and what knowledge we wish to extract. We have also outlined some basic dimensionality reduction algorithms and their functionality. As mentioned, a gene expression matrix contains high-dimensional data that are difficult to be interpreted by hand. Hence, it is crucial to employ data science in order to analyze the data and comprehend the results of a microarray experiment. When data need to be analyzed to extract information two fundamental classes of analysis tasks are used; *Supervised* and *Unsupervised Learning* [Mitchell et al., 1997]. Supervised Learning includes models that learn a classification or an estimation function. On the other hand, Unsupervised methods, examine the data in order to find previously unperceived consistencies and relationships.

In this section, we will present the necessary background knowledge of the algorithms used in this work. These methods will be used in order to comprehend and extract information from microarray experiments.

2.3.1 Classification

Classification problems are applications whose training data includes examples of the input vector along with their corresponding *target* or *class* [Bishop, 2006]. In particular, *classification* problems are cases where the target class is a single variable from a finite set of discrete variables. *Regression* problems, concern cases whose output variable is at least one continuous variable. In this work, we will be dealing with classification tasks. Formally:

Definition 2. *Let an observation be a vector $\mathbf{x}_k \in \mathbb{R}^n$, where each dimension $x_{k,i}$ represents the i^{th} stimuli. The corresponding target value $t \in \mathbb{R}$ represents the single-dimension outcome of the particular observation \mathbf{x}_k .*

The input data are pairs of (observation, target value) $(\langle \mathbf{x}_k, t_k \rangle)$. The output of a supervised learning model (SLM) is a function of the form $y = \text{SLM}(\mathbf{x}_k)$, where $\mathbf{x}_k \in \mathbb{R}^n$ is an observation, and $y \in \mathbb{R}$ is a prediction of the target value that corresponds to observation \mathbf{x}_k .

Intuitively, supervised learning “targets” a specific feature, which is determined via the target values provided, and attempts to predict which given an input vector of an observation, i.e., a set of stimuli, will come with a legitimate outcome, i.e., an accurate value of the class variable. An important task that emerges from gene expression analysis, is classifying samples to distinguish different diseases and their relation to different expression levels of genes. In this section, we will be discussing two classifiers, namely Support Vector Machines and k-Nearest Neighbors which are involved in this work.

2.3.1.1 Support Vector Machines

Support Vector Machines(SVMs) can be exploited both as classifiers and regressors. The model of Support Vector Machine is suitable for classifying high dimensional data in the case of a relatively small number of observations. Intuitively, SVMs are based on creating a single hyperplane (in a binary classification problem), to divide the data based on their outcome variable. The best solution is achieved by finding the hyperplane with the maximum *margin*, i.e the smallest distance between a data-point and the hyperplane [Bishop, 2006]. As it occurs, SVMs are decision machines and thus, do not yield posterior probabilities. Suppose that we have in hand a binary classification problem, and linearly separable data. That is $\mathbf{x}_1, \dots, \mathbf{x}_k$, stimuli and t_1, \dots, t_k corresponding class variables, where $t_k \in \{-1, +1\}$. Assuming the hyperplane is following $y(x) = 0$ and follows the form $y(x) = \mathbf{w}^T \phi(x) + b$, where $\phi(x)$ is a fixed feature space transformation, \mathbf{w} is the normal vector to the hyperplane and b is a bias parameter. The goal of this machine is to find the maximum margin solution by solving:

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min [t_k (\mathbf{w}^T \phi(\mathbf{x}_k) + b)] \right\} \quad (2.5)$$

Solving Equation’s 2.5 optimization problem is rather complex, and hence it is transformed to an equivalent and easiest to solve problem. We observe that all data points will satisfy :

$$t_k (\mathbf{w}^T \phi(\mathbf{x})_k + b) \geq 1, k = 1, \dots, K \quad (2.6)$$

which forms the canonical representation of the hyperplane. We can transform the optimization problem of Equation 2.5 as:

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.7)$$

$$\text{Subject to: } t_k (\mathbf{w}^T \phi(\mathbf{x}_k) + b) \geq 1 \quad (2.8)$$

By using Lagrangian Multipliers [Bishop, 2006], we introduce in this step the multipliers $a_k \geq 0$ for all constraints in 2.6 to tackle the optimization problem, resulting to the following Lagrangian function: The *dual representation* of the maximum marginal problem is given by maximizing:

$$\tilde{L}(a) = \sum_{k=1}^K a_k - \frac{1}{2} \sum_{k=1}^K \sum_{m=1}^K a_k a_m t_k y_m K(x_k, x_m) \quad (2.9)$$

$$\text{Subject to: } a_k \geq 0 \quad k = 1, \dots, K \quad (2.10)$$

$$\sum_{k=1}^K a_k t_k = 0 \quad (2.11)$$

where $K(x, x') = \phi(x)^T \phi(x')$ is the kernel function. Following the analysis in [Bishop, 2006], it is proven that three properties hold:

$$a_k \geq 0 \quad (2.12)$$

$$t_k y(x_k) - 1 \geq 0 \quad (2.13)$$

$$a_k \{t_k y(x_k) - 1\} = 0 \quad (2.14)$$

It is proved that in order to classify new points using SVMs using the multipliers and a kernel function K as :

$$y(x) = \sum_{k=1}^K a_k t_k K(x, x_k) + b \quad (2.15)$$

All data points that satisfy $a_k \neq 0$ and thus, $t_k y(x_k) = 1$ are called *support vectors* and are the points which are retained after training the model. This occurs, due to disregarding all point for which $a_k = 0$, since they are not included in the summation in Equation 2.15. As a result, the model is trained using only the support vectors. It should be pointed out that the kernel function K can be linear, non-linear or Radial Basis Functions [Haykin et al., 2009]. Moreover, SVMs can be easily transformed from a binary to multiclass machines by training multiple binary classifiers.

2.3.1.2 k Nearest Neighbors

In machine learning, k -nearest neighbors (k NN) is a non-parametric algorithm developed for employing classification and regression tasks [Altman, 1992]. The concept underlying this method is that given a training set, the data is projected to the D -dimensional space of the training features. Each unclassified instance is assigned to the majority membership class of the k nearest points. The metric that defines the distance between points can be the Euclidean Distance, Minkowski Distance, or Manhattan Distance. Formally, supposing a dataset comprised of N_k instances in C_k target class, with a total number of N instances in total such that $\sum_k N_k = N$. Assuming that a new instance x needs to be classified, a sphere is drawn, which has its center on x and contains exactly K points, disregarding their classes. An estimate of the density conditioned on each class is provided by :

$$p(x|C_k) = \frac{K_k}{N_k V} \quad (2.16)$$

in which K_k is the number of instances in class C_k , and V corresponds to the volume of the sphere. Likewise, the unconditioned density is given by:

$$p(x) = \frac{K}{NV} \quad (2.17)$$

while class priors are given by :

$$p(C_k) = \frac{N_k}{N}. \quad (2.18)$$

The Posterior probability of class membership is obtained by combining the above using Bayes' theorem :

$$p(C_k|x) = \frac{p(x|C_k) * p(C_k)}{p(x)} = \frac{K_k}{K} \quad (2.19)$$

The probability of misclassifying a new data point x is minimized when the point is assigned to the class which has the largest posterior.

2.3.2 Clustering

It often occurs that when we need to detect patterns in a dataset, the training data are not predefined with a particular class or category. These types of data are usually analyzed using *Unsupervised Learning*. The incentive of these problems is to infer patterns by learning from observations which are similar within the data, a process called *clustering*, or identify the distribution of the data within the input space, which is referred as *density estimation*, or perform *visualization* of high dimensional data to two or three dimensions [Bishop, 2006].

2.3.2.1 K-Means

The K-Means algorithm clusters data by attempting to distinguish samples in groups of equal variance, while minimizing the criterion of inertia or within-cluster sum-of-squares. This method comprises one of the oldest and most significant problems in computational geometry. Given an integer k and an input set of n data points X in \mathbb{R}^d , the objective is to choose k centers that minimize ϕ , the sum-of-squares distance that lies between each point and the center that is nearest to it. This is an NP-Hard problem, but a local search solution proposed by [Lloyd, 1982] established this algorithm as one of the most popular approaches for unsupervised learning[Berkhin, 2006]. K-means is initialized with k arbitrary “centers”, which are decided uniformly at random from the data points. Each point is then allocated to the nearest center. The next step is to recalculate the center by computing the center of mass for each assigned in the cluster point. This process is repeated until it converges, which is guaranteed.

Although k-means will always converge and is relatively fast, it produces bad clusters, in terms of accuracy. To untangle this, [Arthur and Vassilvitskii, 2007] proposed a variation regarding the initialization of the cluster centers. The process begins with choosing the first center uniformly at random from X ; then take each new center c_i , choosing $x \in X$, with a probability $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$, where D is the “ D^2 weighting”. This step is repeated until all centers k have been chosen. The procedure continues by applying the original k-means algorithm.

2.3.2.2 Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering (HAC) are unsupervised learning methods, that create hierarchies between groups of data. In particular, these methods perform “bottom-up” clustering, that is each observation initializes its individual cluster, and then pairs of clusters are merged when “leveling up” [Rokach and Maimon, 2005].

In order to perform the bottom-up merging, a similarity measure must be introduced. The HAC methods can be distinguished according to the computation of similarity to the following categories:

- **Single Linkage** : In this category, the similarity is considered as the shortest distance between any member of one cluster to any member of the other cluster.

- Complete Linkage : In this case, the similarity is defined as the maximum distance between any member of two distinct clusters.
- Average-Linkage : Similarity is computed by finding the average distance between any two members of two different clusters.
- Ward-Linkage : Minimizes the sum of squared differences within all clusters.

The distance can be calculated by choosing any appropriate distance metric, such as Euclidean, Manhattan and other well-established distance metrics. What derives from the above characteristics of a HAC algorithm is that choosing the suitable linkage criterion and metric, we can achieve the clustering results we wish.

2.4 Probabilistic Topic Modeling

In this section, we will provide the basic notion of Probabilistic Topic Models. The motivation of this work lies in using these models. Probabilistic topic models are a category of algorithms introduced in text mining, which can discover latent themes that pervade a collection of documents without exploiting prior knowledge. Hence, they fall into the unsupervised learning category.

In text analytics, a corpus is an assemblage of documents, which in turn is a collection of distinct words. The intuition behind these models is that a document, as a collection of words, discusses a certain topic (or topics). What is cutting edge about topic models, is that they can capture the multiplicity of topics in a document. For example in Figure 2.3 the article is entitled as “Seeking Life’s Bare (Genetic) Necessities” speaks about the use of data analysis to determine the number of genes needed by an organism to survive. Topic models eventually would infer the topics of this article are with a proportion of $k_1\%$ about genetics and $k_2\%$ about data analysis and so forth. In what follows, we will survey the basic assumption behind topic models, namely the “Bag of Words” paradigm and the topic models used in this work, Latent Dirichlet Allocation, Hierarchical Dirichlet Process and Latent Process Decomposition.

In the text mining field given a collection of *documents*, the goal is to extract knowledge that pervades the corpus. To do so, topic models require that the data follow a certain representation, that of the “Bag of Words” (BoW) model. In this representation each document d_i consists of a set of *words* whose exact ordering does not matter [Manning et al., 2008]. This interprets that given several instances of a problem, each instance is referred to as *bag*, whereas its variables are the *words*. By making the above assumption, this model examines only *multiplicity*, meaning it retains the number of occurrences of each term. In a more general formula, we use this model to describe a problem where a set of discrete data compose a cohesive structure. That is depicted as an object via a count vector of its elements. This representation is based on a fundamental statistical assumption, *exchangeability* [Pitman, 1995], which allows the order of random variables to be neglected by the specific model.

Definition 3 (Exchangeability). *A finite set of random variables $\{z_1, \dots, z_n\}$ is said to be exchangeable if the joint distribution is invariant to permutation. If π is a permutation of the integers from 1 to N :*

$$p(z_1, \dots, z_n) = p(z_{\pi(1)}, \dots, z_{\pi(N)}) \quad (2.20)$$

The BoW paradigm, considers both words and documents exchangeable. This assumption is rather unrealistic, but in the context of discovering semantic structure, it is rational. For instance, consider the example illustrated in Figure 2.3: by simply performing any shuffling in the words of this article, the article would be assigned to the same topic, in this case, genetics. Similarly, we assume exchangeability in documents, as Equation

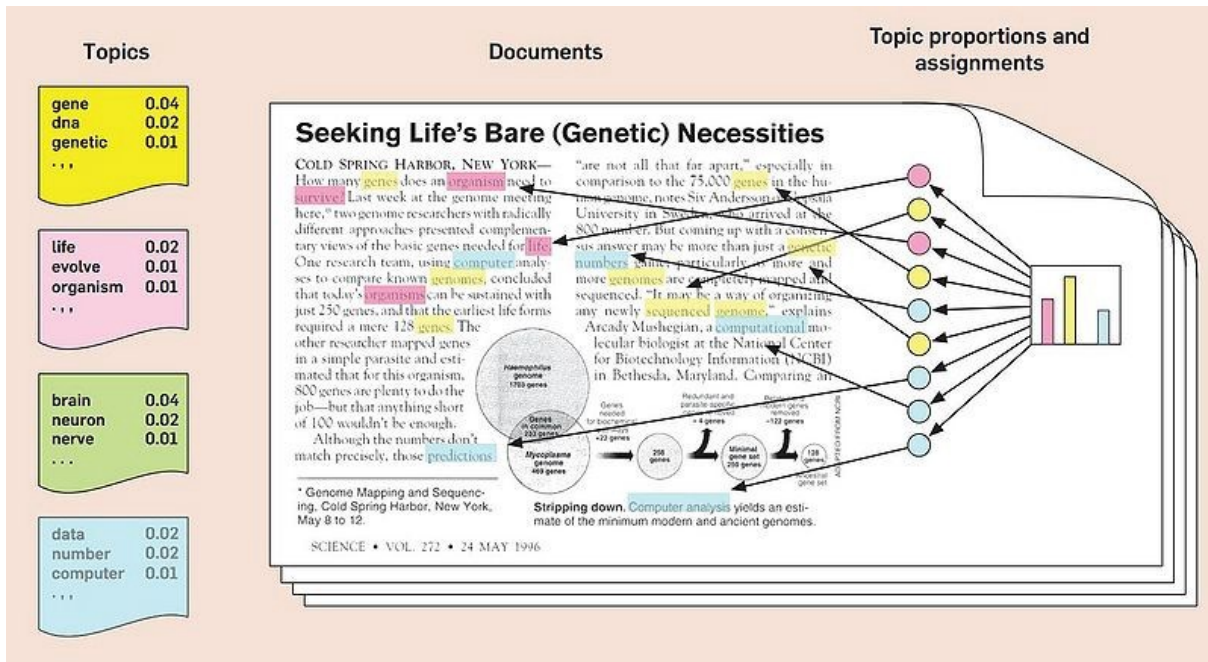


Figure 2.3: Example of the intuition of Latent Dirichlet Allocation. obtained by [Blei, 2012a].

2.24 indicates. This assumption is inconvenient, in case we would like to analyze how topics evolve over time, where time is captured by the arriving sequence of documents.

2.4.1 Latent Dirichlet Allocation

In this section we describe Latent Dirichlet Allocation(LDA)[Blei et al., 2003]. The LDA model has served as a springboard for many other topic models. The fundamental work of Latent Semantic Indexing (LSI) [Deerwester et al., 1990] and the later Probabilistic Latent Semantic Indexing (pLSI) [Hofmann, 1999] inspired [Blei et al., 2003] with the simplest topic model, LDA.

We first describe the probabilistic method, by introducing the basic ideas behind this generative model. To begin with, a *generative model* is a model that makes predictions by computing the joint probability of the inputs and the label by exploiting the Bayes rule [Ng and Jordan, 2002].

The intuition behind LDA is to model the documents such that they exhibit multiple topics. A topic is defined as a distribution over a fixed collection of terms; however, a topic's intuition requires humans' interpretation to be conceivable by others. Assuming that K topics are related to a corpus, each document is represented from a mixture of topics in different proportions. That is, a specific document d can be described by $k^{\text{th}} \in \{1, \dots, K\}$ topic in a proportion $p_k\%$. This assumption is vital and natural, as it often occurs that collections of documents are highly heterogeneous, combining ideas or themes that pervade the collection as a whole. In LDA the conception of topics is captured by introducing a hidden-variable model in documents.

Supposing we employ LDA on the article in Figure 2.3. If we follow the generative process of LDA and highlight the different words in the article according to a different topic, we would result in inferring the topics shown on the left side of the figure. A human annotator can decide that the first topic is likely to refer to genetics, the second topic to evolutionary biology and so forth. On the right side of this figure, the different proportions of the topics are shown. As a result of this process, this article blends data analysis, genetics evolutionary biology in different proportions. Moreover, the fact that this article exhibits these topics, would help the annotator

situate this as a scientific article.

Notation

Firstly, we will introduce the formal notation and terminology, to describe the generative process of LDA. What needs to be highlighted, is that LDA is not exclusively tied to text applications, but for the sake of consistency we will use the following notation as introduced in [Blei et al., 2003]:

- A *word* is the basic unit of discrete data, which is defined as an item from a vocabulary that is indexed by $\{1, \dots, V\}$.
- A *document* is an order of N words indicated by $w = (w_1, \dots, w_N)$, in which w_n denotes the n^{th} word in order.
- A *corpus* is a collection of M documents indicated by $D = \{w_1, \dots, w_M\}$

More particularly, in the article of Figure 2.3 the topic about *genetics* includes words about genetics with high probability, and the topic regarding data analysis contains words like data number computer with high probability.

2.4.1.1 The Generative Process

Having presented the basic notation and intuition, we will now describe the generative process, i.e. an “*imaginary random process*” which the model mimics the generation of the observed data. Specifically, the model attempts to “replicate” the way the data was generated. In fact, it will never know how the documents were actually generated, but attempt to model this procedure. By following the process described below, we model the data in order to discover the latent topics. Let K , as mentioned, denote the multitude of topics; in LDA variable K is known and fixed.

LDA assumes the following generative process:

- For each topic z , where $z \in \{1, \dots, K\}^a$:
 1. Draw a distribution over the words of the vocabulary, $\phi_z \sim \text{Dir}(\beta)$
- For each document w_i where $i \in \{1, \dots, M\}$ in a corpus D :
 1. Choose $\vartheta_i \sim \text{Dir}(\alpha)$.
 2. For each w_n the n^{th} word in i^{th} document, where $n \in \{1, \dots, N\}$
 - a) Choose a topic $z_n \sim \text{Multinomial}(\vartheta_i)$.
 - b) Choose a word $w_n \sim \text{Multinomial}(\phi_{z_n})$, a multinomial probability conditioned on the topic z_n .

^aThe topics are generated before the documents.

The distribution over the vocabulary (V) for each topic z is indicated a Dirichlet distribution with parameter β , namely ϕ_z . This Dirichlet parameter is a $K \times V$ matrix β , where $\beta_{ij} = p(w^j = 1 | z^i = 1)$, which is to be estimated. The K -dimensional Dirichlet random variable ϑ_{d_i} represents the topic distribution for document d_i . The topic mixture variable ϑ is drawn from a Dirichlet distribution parameterized by a variable α , a K -vector

with positive, non-zero elements ($\alpha_j > 0$, for $j = 1, \dots, K$). The variable z_n denotes the topic assignment for the n^{th} word, while z is a N (the size of the vocabulary) dimensional vector which defines which topic each of the words of the vocabulary belongs to. For convenience, we consider that the corpus contains N documents, that contain the same number of words N .

Following the notation in [Blei et al., 2003], equation 2.21 the joint distribution of a topic mixture \mathcal{Z} is provided.

$$p(\mathcal{Z}, z, w | \alpha, \beta) = p(\mathcal{Z} | \alpha) \prod_{n=1}^N p(z_n | \mathcal{Z}) p(w_n | z_n, \beta), \quad (2.21)$$

where α, β are the model parameters for a set of N topic z and a set of N words w . By integrating over \mathcal{Z} and summing over z :

$$p(w | \alpha, \beta) = \int p(\mathcal{Z} | \alpha) \left(\prod_{n=1}^N \sum_{z_n} p(z_n | \mathcal{Z}) p(w_n | z_n, \beta) \right) d\mathcal{Z} \quad (2.22)$$

while for all documents, the probability of a corpus occurring given the model parameters is:

$$p(\mathcal{D} | \alpha, \beta) = \prod_{d=1}^M \int p(\mathcal{Z}_d | \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} | \mathcal{Z}_d) p(w_{dn} | z_{dn}, \beta) \right) d\mathcal{Z}_d. \quad (2.23)$$

We illustrate using probabilistic graphical models, the three level model of LDA in Figure 2.4. Colored nodes indicate observed variables, while the rest are variables that need to be inferred. The plates denote replicate structure. As presented in the graphical model, variables α and β are corpus related variables which are sampled once, when generating the corpus. The variables which are generated once per document, are the variable \mathcal{Z}_d , whereas z_{dn} and w_{dn} are variables that are sampled for each of the words in a document.

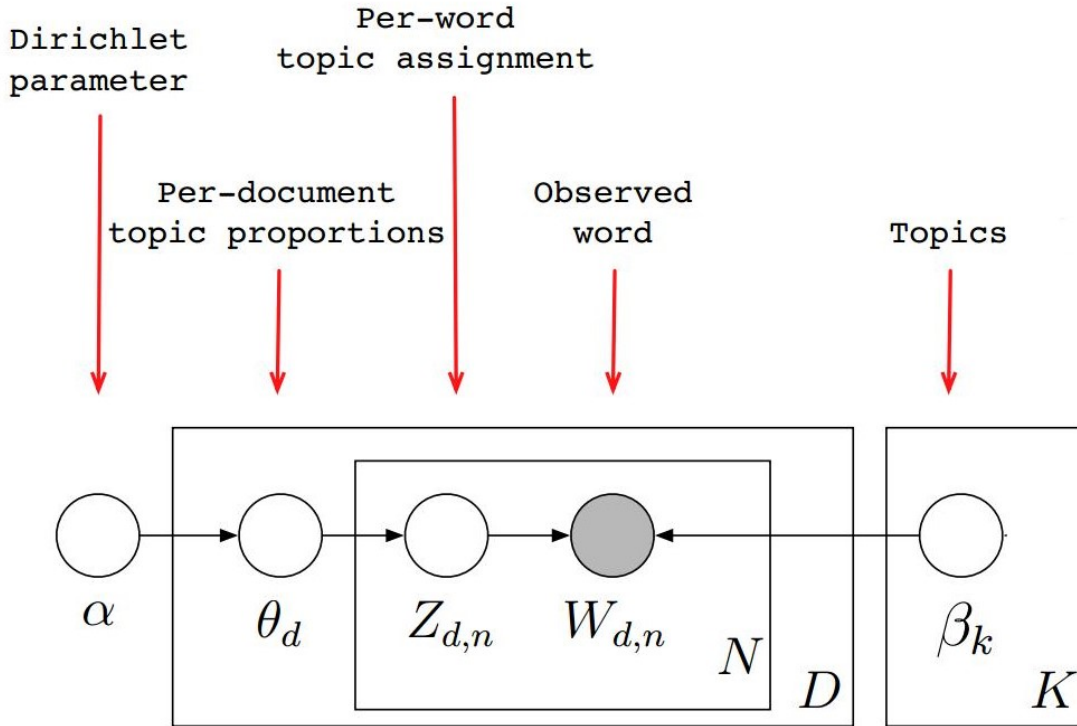


Figure 2.4: Probabilistic Graphical Model of Latent Dirichlet Allocation.

2.4.1.2 Posterior Inference

As mentioned earlier, the model's goal is to automatically discover the topics from a corpus. The documents are the observed data, while the topic structure—the topics, per-document topic distributions, and the topic-term distribution assignments—composes the hidden structure. A computational problem regarding this model, is how we are going to infer the so called, hidden structure which occurs along with the specific observed data. The general idea is to reverse the generative process described above.

Learning the various distributions ((a) the set of topics, (b) their associated word probabilities, (c) the topic of each word, and (d) the particular topic mixture of each document) is a problem of Bayesian inference. Explicitly, we define the posterior distribution of the hidden variables given a document:

$$p(\mathcal{Z}, z | w, \alpha, \beta) = \frac{p(\mathcal{Z}, z, w | \alpha, \beta)}{p(z | \alpha, \beta)} \quad (2.24)$$

which is proved in [Blei et al., 2003] to be intractable. By marginalizing over the latent variables of Equation 2.22 we obtain the normalized distribution over the hidden structure in terms of the parameters of the model:

$$p(w | \alpha, \beta) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^k \mathcal{Z}_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^N \sum_{i=1}^k \prod_{j=1}^V (\mathcal{Z}_i \beta_{ij})^{w_n^j} \right) d\mathcal{Z} \quad (2.25)$$

which is proven to be intractable as a result of the coupling between \mathcal{Z} and β . It is a standard approach to use approximation inference methods to untie this knot. Authors in [Blei et al., 2003] use variational Bayes approximation of the posterior distribution; alternative inference techniques use Gibbs sampling [Griffiths and Steyvers, 2004] a Markov Chain Monte Carlo method, and expectation propagation. For instance supposing we use variation inference to learn the model parameters, we utilize a simpler distribution q , which depends on the variables $\phi_{1:D}$, $\gamma_{1:D}$ and $\lambda_{1:D}$ defined as :

$$\begin{aligned} \phi_{dwk} &\propto \exp\{E_q[\log \mathcal{Z}_{dk}] + E[\log \beta_{kw}]\}, \\ \gamma_{dk} &= \alpha + \sum_w n_{dw} \phi_{dwk} \lambda_{kw} = \eta + \sum_d n_{dw} \phi_{dwk} \end{aligned}$$

We denote as ϕ_{dwk} , the probability of the word w in document d to be assigned to document k . The number of occurrences of word w in document d is represented by n_{dw} . Variational parameters $\gamma_{1:D}$ and $\lambda_{1:K}$ are related to the variable n_{dw} . In Algorithm 1 we present the Variational Inference algorithm for LDA, whose intuition is to minimize the *Kullback-Leibler divergence* between the variation distribution and the true posterior.

2.4.2 Hierarchical Dirichlet Process

Hierarchical Dirichlet Process(HDP) introduced in [Teh et al., 2005], is a non-parametric version of LDA. This algorithm exploits the Chinese Restaurant Process along with Dirichlet Processes to extract latent topics. In contrast to LDA, it does not require a fixed number of documents K as input, but rather infers its multitude during the posterior inference [Blei, 2012b]. This process is a stochastic process, which can define a non-parametric distribution on a mixture of mixture models. The top mixture is related to the global set of topics, which are shared amongst the documents of a corpus, while the secondary is the mixture of topics regarding a specific document.

Algorithm 1: Variational Inference for LDA [Blei et al., 2003]

```

1 Randomly initialize  $\lambda$ ;
2 repeat
3   Expectation step:
4   for ( $d = 1 \rightarrow D$ ):
5      $\gamma_{dk} = 1$ ;
6     repeat
7       Set  $\phi_{dwk} \propto \exp\{E_q[\log \vartheta_{dk}] + E_q[\log \beta_{kw}]\}$ ;
8       Set  $\gamma_{dk} = \alpha + \sum_w n_{dw} \cdot \phi_{dwk}$ ;
9     until ( $\frac{1}{K} \cdot \sum_{k=1}^K \text{change in } \gamma_{dk} \parallel < \epsilon$ );
10  Maximization step:
11  Set  $\lambda_{kw} = \eta + \sum_{d=1}^D n_{dw} \cdot \phi_{dwk}$ ;
12 until (relative KL divergence has not significantly decreased);

```

An HDP is formally defined from equations 2.26. Given a concentration parameter γ and a base measure H , we draw from a Dirichlet process $DP(\gamma, H)$ the base

$$G_0 | \gamma, H \sim DP(\gamma, H) \quad (2.26)$$

$$G_j | \alpha_0, G_0 \sim DP(\alpha_0, G_0) \quad (2.27)$$

Given the above definition, we can describe an HDP *mixture model* to fall in the topic modelling scenario. Supposing we have $j = 1 \dots J$ to index the J groups/documents., and $x_j = (x_{ji}), i = 1 \dots n_j$ correspond to the n_j observations in group j . The observations x_j , denote the words in document j . As in LDA, the words are exchangeable random variables, which are drawn from a mixture model, whose configuration is drawn once per each group j .

$$\phi_{ji} | G_j \sim G_j \quad (2.28)$$

$$x_{ij} | \phi_j \sim F(\phi_j) \quad (2.29)$$

where $F(\phi_{ji})$ is the distribution of x_{ji} given ϕ_{ji} and G_j is the prior of ϕ_j (Equation 2.27). To describe this model in a less bewildering matter suppose we have a corpus of $J = 2$ documents, $K = 3$ underlying topics, and each document contains $n_j = 4$ words that comprise a vocabulary $|V| = 8$. Each topic $k \in K$ specifies parameters ϑ_k of a multinomial distribution F over the distinct words. Supposing $\phi_1 = (2, 3, 3, 1)$ and $\phi_2 = (2, 2, 3, 1)$. The variable x_{11} will be a draw following $x_{11} \sim F(\vartheta_2)$ and similarly $x_{12} \sim F(\vartheta_3)$ and so on. Again, as in LDA, each document can be composed from different proportions of each latent topic.

In order to describe a representation of HDP using CRP we will provide an insight to the CRP and CRF sampling schemes. The Chinese restaurant process (CRP) is a discrete-time stochastic process, metaphorically is a procedure of seating people in a Chinese restaurant. In particular, assume that a restaurant has an infinite number of tables and incoming customers. Each customer can sit at a table which has already people, or choose to sit in an empty one. The customers form a set S , ξ is a partition of the set S , and the set containing all possible partitions is P_S . Each customer arrives at the restaurant and sits at table c with following probabilities:

$$P(\text{sit at table } c) = \frac{n_c}{\alpha + \sum_{c \in \xi} n_c} \quad (2.30)$$

$$P(\text{sit at empty table}) = \frac{\alpha}{\sum_{c \in \xi} n_c} \quad (2.31)$$

Parameter α governs the number of clusters in a significant manner. It can be observed that what increases the probability of sitting at a table is the size of the clusters and not the ordering. The probability mass function is given by :

$$P(g|\alpha) = \frac{\alpha^{|g|} \Gamma(\alpha)}{\Gamma(n + \alpha)} \prod_{c \in g} \Gamma(|c|) \quad (2.32)$$

The single restaurant CRP metaphor can be extended to a multiple restaurant setting known as the Chinese Restaurant Franchise (CRF). In this scenario, there are sampling schemes employed simultaneously : one for the tables and one for the dishes served at the tables. In the single CRP, we assume that each table contains a different dish, and all clients that sit on the same table share the same dish. However in the CRF sampling scheme, we assume that there exists a global menu of dishes, and a new table is allocated to a new dish with a probability that is proportional to m_k . This number indicates the number of tables allocated to dish k in all restaurants J . In a similar manner as in CRP, a previously unseen dish is created and assigned to a new table with a probability proportional to concentration parameter γ . To give the representation of HDP using CRF, we use random variables ϕ_{ji} to correlate to observations(i.e the customers), and indicate at which of the T_j tables found in restaurant j , a newly arrived customer x_{ji} will be seated. For simplification, a multitude of T_j random variables ψ_{jt} are introduced to indicate the tables of restaurant j . We demand that ψ_{jt} are i.i.d. distributed on G_0 , and each of the ψ_{jt} specifies the mixture component(i.e. topic) for table jt . Finally, random variable of K multitude, specified by \mathfrak{S}_k are used to correspond to dishes, and indicate the parameters of mixture components k , while they are i.i.d. and distributed according to H . It should be noted here that, this model exhibits two many-to-one relationships. The first corresponds for at least one ϕ_{jt} to one ψ_{jt} , and the second to relates at least one ϕ_{jt} to one \mathfrak{S}_k .

Considering the above, the assignment of the i^{th} customer in restaurant j , that is ϕ_{ji} to table ψ_{jt} , is given by :

$$\phi_{ji} | \phi_{j1}, \dots, \phi_{ji-1}, \alpha_0, G_0 \sim \sum_{t=1}^{T_j} \frac{n_{jt}}{i-1 + \alpha_0} \delta_{\psi_{jt}} + \frac{\alpha_0}{i-1 + \alpha_0} G_0 \quad (2.33)$$

where n_{jt} is the number of customers currently sitting, ϕ_{ji} is associated with table ψ_{jt} , and number T_j indicates the number of tables overall in restaurant j .

Moving on, to sample the ψ_{jt} variable the following process is followed:

$$\psi_{jt} | \psi_{11}, \psi_{12}, \dots, \psi_{21}, \dots, \psi_{jt-1}, \gamma, H \sim \sum_{k=1}^K \frac{m_k}{\sum_k m_k + \gamma} \delta_{\mathfrak{S}_k} + \frac{\gamma}{\sum_k m_k + \gamma} H \quad (2.34)$$

in which $m_k = \sum_j m_{jk}$ and m_{jk} indicates the tables that ψ_{jt} is associated with \mathfrak{S}_k . Samples of observations $x_{ji} \sim F(ji)$ are obtained by first sampling a value of ϕ_{ji} to the proportions set out in Equation 2.33. Given that a new table is needed, it is drawn according to the proportions in Equation 2.34. Equivalently, a new mixture component \mathfrak{S}_k values are drawn according to H .

The next step, is to perform posterior inference given a set of observations in an HDP model. A very simple method for inferring variables t_{ji} , k , \mathfrak{S} is MCMC method (Gibbs Routine). This inference algorithm can adapt well to our problem, as it can update the mixture components for multiple observations simultaneously. It provides also easy re-calculation of the values of each component at each iteration, which results to better mixing and guarantees convergence.

2.4.3 Latent Process Decomposition

Latent Process Decomposition (LPD) can be considered as the continuous version of LDA. It was first introduced in the microarray setting [Rogers et al., 2005]. Topic multiplicity, namely the attribute of allowing each document to be modeled as a mixture of multiple topics, is a shared feature of the both algorithms. However, microarray data are comprised of continuous variables opposed to the discrete variables of text corpora. In order to untie this knot, authors in [Rogers et al., 2005] propose the usage of Gaussian distributions in place of word-multinomial in LDA. We will provide a short generative description :

- Draw a Gaussian distribution $N(x; \mu_{gk}, \lambda_{gk})$ for each pair of gene g and process(i.e topic) k from prior distributions. Precisely, a mean parameter μ_{gk} is drawn from a Gaussian prior $N(\mu, \mu_0, \lambda_0)$, and a precision parameter λ_{gk} is drawn from a Gamma prior $Gam(\lambda; \alpha_0, b_0)$. The equivalent step in LDA is determining the probability of a word given a specific topic.
- Draw a multinomial distribution $Multi(z; \mathfrak{Y})$ for each sample d from a symmetric Dirichlet prior Distribution $Dir(\mathfrak{Y}, \alpha)$. In LDA, documents instead of samples are used, and topics instead of processes, however this part is exactly the same in both algorithms.
- If gene g occurs in sample d , draw a process z_{dg} from $Multi(z, \mathfrak{Y}_d)$, and then draw an observed real value x_{dg} from $N(x; \mu_{gz_{dg}}, \lambda_{gz_{dg}})$.

Based on the notion presented, the joint distribution of a topic mixture \mathfrak{Y} , a set of N topics z_n and N genes g_n that are expressed in a sample is given by :

$$P(\mathfrak{Y}, z, g | \alpha, \beta) = p(\mathfrak{Y} | \alpha) \prod_{n=1}^N p(g_n | z_n, \beta) p(z_n | \mathfrak{Y}). \quad (2.35)$$

2.5 Related Work

Although topic models have been mainly used in analyzing text data, research has been conducted in applying them in bioinformatics. The intrinsic challenge regarding the application of these models in this field is the mapping between documents and words to the specific problem. For example in [Chen et al., 2012], [Chen et al., 2011], authors relate the words to K-Mers of DNA sequences (nucleobases) and documents as DNA sequences, whereas the topics inferred are taxonomic components of the whole genome. In the work of [Coelho et al., 2010], authors employed PTMs on fluorescence images of a biological experiment, corresponding images to documents and object classes as words. The topic model discovered latent topics of fundamental patterns. Gene expression data are fed to Latent Dirichlet allocation in order to uncover functional groups of genes. The words that are recounted as the genes names and the documents as tissue samples.

Exploring relevant studies, someone can conclude that there exist three main tasks that involve probabilistic topic models for biological data; biological cluster analysis, biological data classification and biological data feature extraction [Liu et al., 2016]. These tasks are illustrated in Figure 2.5. The different shapes and colors correspond to biological tissue samples that have been processed by a topic model. The tissue samples are distinguished by colors to indicate that these samples have higher probabilities for different topics. That is, a sample belongs to each topic with a different proportion. We will exhibit the prior work on these three different topics individually.

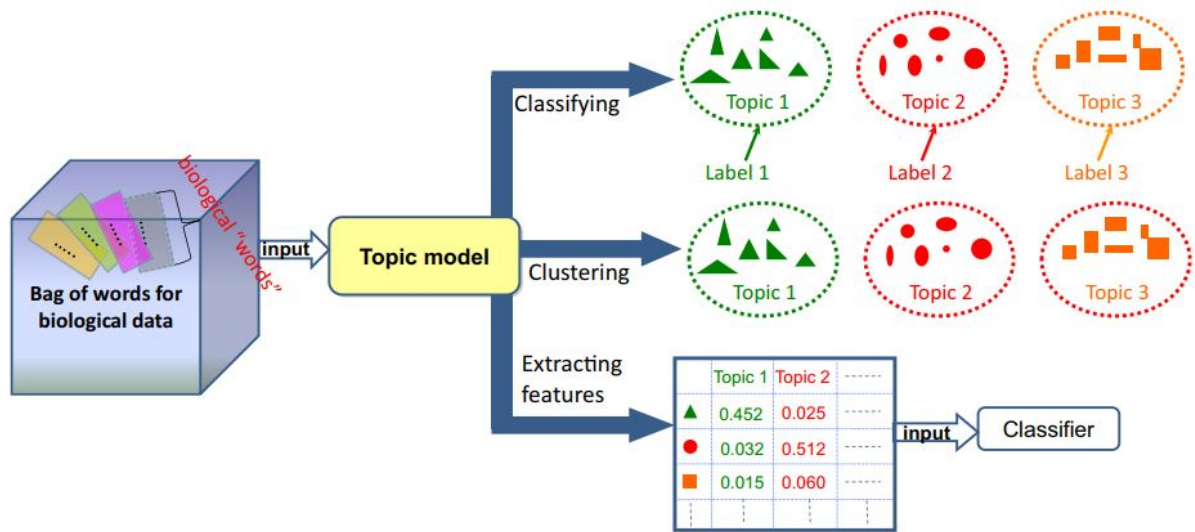


Figure 2.5: The tasks of a topic model in Bioinformatics. [Liu et al., 2016]

2.5.1 Dimensionality Reduction

As noted before, microarray data suffer from the “curse of dimensionality”. To untie this knot, two tasks can be performed; feature extraction and feature selection. In this approach, studies have been conducted in order to use topic models as a feature extraction method, in a similar way as PCA. After employing, for example, LDA, the output document-term matrix serves as a projection of the high dimensional data into a smaller space, i.e. the subspace created by the topics. Given a sample x , whose initial dimension corresponds to the number of genes G is projected to the K dimensional space of the topics. This approach was followed in [Zhao et al., 2014], and for the evaluation of the approach, they employed the k-means algorithm. In more recent work, authors

[Kho et al., 2017] also developed this methodology; the derived document topic distribution was used as an input to a hierarchical clustering algorithm and was compared with applying PCA and clustered according to a hierarchical clustering algorithm. In a similar manner, in the paper of [Bicego et al., 2010], authors employ PLSA and LDA variation LPD in order to extract the feature vector. For their experiments, in order to classify the samples according to the document-term matrix, K-Nearest Neighbor was employed. This approach was also followed in [Bicego et al., 2012], in which the capabilities of Topic models were examined by using a hybrid generative classification scheme.

Moreover, LDA has been also used to perform feature selection. As mentioned earlier, the intention is to eliminate features that either produce noise or are not relevant to the class to be predicted. To this end, the topic-term distribution is exploited, where selecting the h terms that are most probable in each topic k can result in the subset of features that are most relevant. More precisely, [Zhao et al., 2014] selected a prefixed number of h most probable genes inside a topic suggesting that these genes express the topic. After selecting the subset of genes, k-Means was applied for cluster analysis. This approach was also evaluated by comparing to the entire feature space, and by applying PCA and projecting to $k \times h$ subspace.

2.5.2 Cluster Analysis

As discussed earlier, topic models and particularly LDA produces two outputs. The sample-topic matrix depicts the probability distribution of the topics in the samples/documents where each row represents the proportion of the topic in a sample. The second output is the topic-word matrix, which exhibits the probability distribution of the words in each topic. In the work of [Zhao et al., 2014], authors exploited these outputs for clustering according to the highest probable assignment. This method uses the sample-topic matrix in order to cluster the samples according to the cluster(topic) in which they exhibit the highest probability. The authors evaluated the performance of this method by applying as a next step hierarchical agglomerative clustering. Another work that involves clustering analysis is the work of [Bicego et al., 2010], who perform biclustering, which is simultaneous clustering of the samples and the genes of gene expression data using PLSA topic model.

2.5.3 Classification

Beyond clustering analysis for unlabeled biological data, a topic model can be exploited in the supervised learning setting. In other words, a topic model can endeavor to match the topics to true biological labels. Due to the nature of topic models, a supervised set is not easy to incorporate into these models. Albeit, these models need adaptation in order to work in that manner. To adapt topic models to this scenario, the work of [Perina et al., 2010] introduced a model called biologically aware LDA (BaLDA). This method combines Latent Process Decomposition, Latent Dirichlet Allocation and integrates document dependencies to perform classification tasks. This novel topic model introduces the categorization of genes, modeled by a random variable that is later inferred by exploiting a priori knowledge on genes of the analyzed problem. Moving on, authors in [Kho et al., 2017] and [Yalamanchili et al., 2017] develop an LDA-based classification approach that performs supervised learning on unseen samples by comparing the similarity of co-expression patterns.

A framework on employing PTMs on Gene Expression Data

In this chapter, we will exhibit the proposed framework for employing Probabilistic Topic Models on Gene Expression Data. The framework necessitates an adaptation of the notation to gene expression data, exploiting and studying their results. In Figure 3.1 we present the general framework through a pipe-lined process. The first step is to transform our input data, namely the gene expression matrix, into the “Bag of Words” paradigm. This step is required if we intend to employ a topic model such as LDA, HDP, Pachinko Allocation Model, etc., that need input in such formats. However, if we decide to employ Latent Process Decomposition, this step is omitted. This process is depicted by the “Bag of Words Transformation” node in Figure 3.1. Then, the topic model is implemented and fitted on the data, as presented in the node “Topic Model”. These models, principally produce two outputs; : (a) distributions of topics over the vocabulary, and (b) the distributions of the documents over the inferred topics. After having obtained the inferred distributions, we exploit them in order to perform Dimensionality Reduction and what we call Cluster Analysis. In particular, we will thoroughly describe our methodology for dimensionality reduction using the topic term distribution for Feature Selection, and the document term distribution for Feature Extraction. With respect to Cluster Analysis we visualize the distributions on the 2-dimensional space and provide to biologists an easily readable mean to interpret. Concluding this framework, we evaluate the dimensionality reduction techniques using classification and clustering techniques. In the rest of the chapter, we will describe thoroughly the processes shown in the workflow of Figure 3.1.

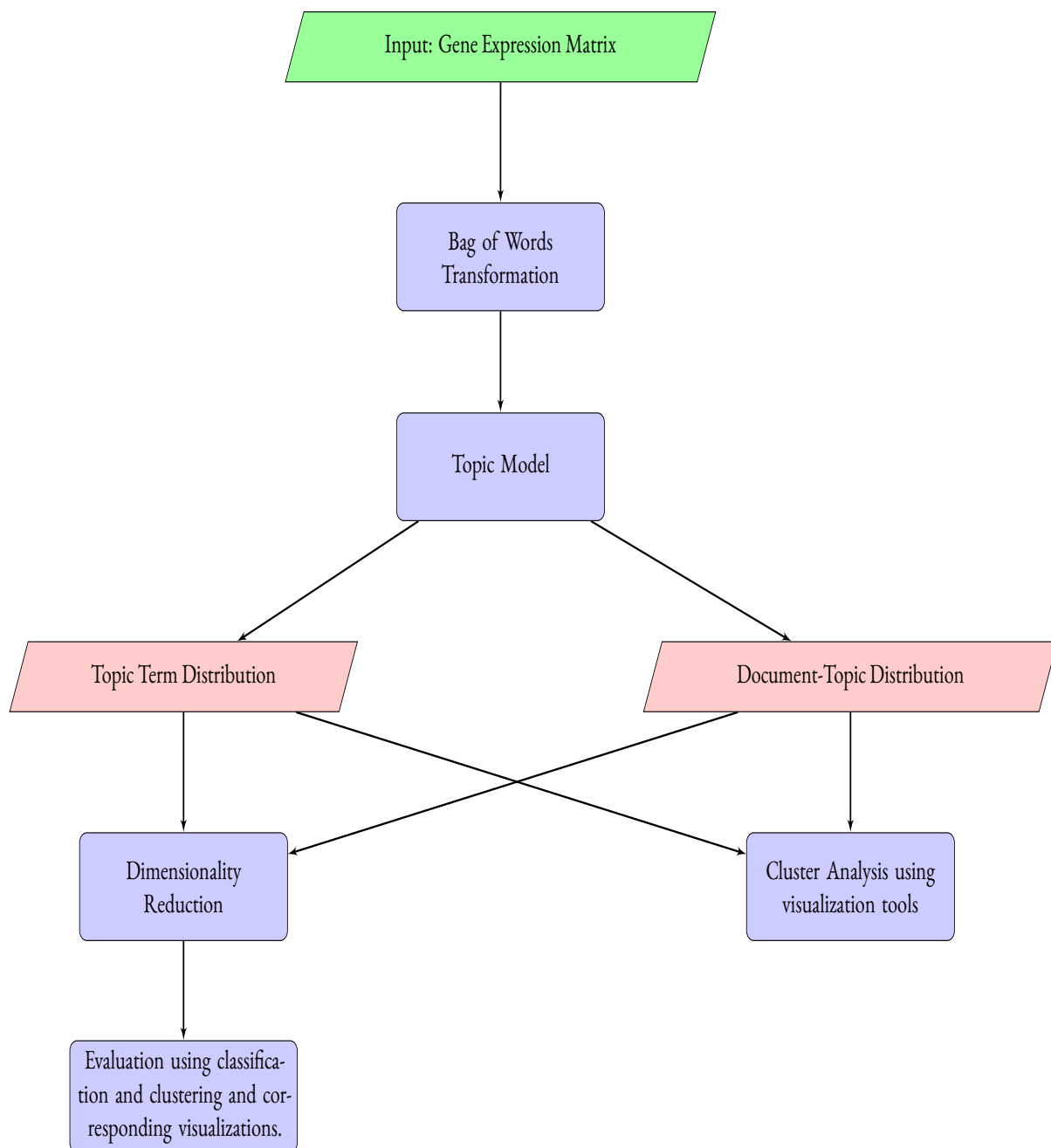


Figure 3.1: A framework on employing PTMs on Gene expression data.

3.1 Transforming the Data into the Bag of Words Paradigm

As already mentioned, Probabilistic Topic Models require discrete data as input and more accurately, the data should follow the “Bag of Words” paradigm. Firstly, given the problem under investigation, we need to define which entities will be treated as documents, and which as words/terms. For example, if we were to perform probabilistic topic modeling on videos, in which we wished to extract salient behaviour, the words would correspond to visual events [Hospedales et al., 2012]. In our case, the data are composed of a gene expression matrix in which each observation is a tissue or cell sample obtained by a patient, and the variables are the genes and their corresponding expression values. This leads to interpreting the gene expression matrix as multiple gene profiles. Hence, representing a sample in the BoW form, we consider a sample as a document and the genes (i.e. their names) as the words. The next question we need to address is *how to count*, more precisely how we will extract the *count vector* of each sample and its included genes, since the values at hand are continuous. More accurately, our goal in this step is to provide the PTMs with a representation that is translated as *how many times will a gene name appear in each document-sample*? This process is associated with the *BoW Transformation* node in the pipeline of Figure 3.1, while in this work we propose three alternative approaches to handle this issue. The Transformation procedure is visualized in the workflow of Figure 3.2. We have implemented three alternative approaches to generate the count vector, and thus the corpus, which namely are *Median*, *Repetition* and our novel proposed method, *Bibin*.

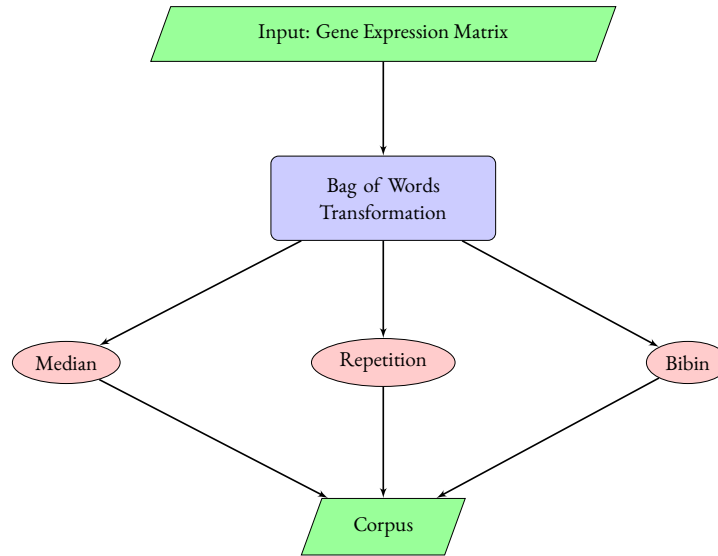


Figure 3.2: Workflow for Transforming the data into BoW.

3.1.1 Median

This approach was first used in [Zhao et al., 2014] in order to discretize gene expression data. First, for each gene $g_i \in G$, the median value is computed across all samples in the dataset D . Then, each gene expression value was set to 0, if it was less than the median or 1 if it was greater or equal to the median value. This discretization leads to the desirable count vector of each sample. More precisely, each sample is associated with a document, with the genes' names as words; while each gene occurrence in the document depends on the samples' count vector. That is, a gene occurs either once or does not occur at all. Hence, the resulting count vector includes binary variables (either 0 or 1). After performing the above process, we obtain the BoW representation: a corpus of documents (one for each sample) containing words from a fixed vocabulary V , which is formed by the names of the genes. In this discretization technique, the genes that are included in the corpus are those whose expression value is either close to the "usual" value or genes that are over-expressed in a sample. It is also important to comprehend, that this method is influenced by how much balanced the data are, regarding the samples and their conditions. As stated in [Yalamanchili et al., 2017], one downside of this method is that given a new sample, the preprocessing should be re-calculated all over again, leading to a whole new LDA experiment. The workflow of this process is shown in Figure 3.3.

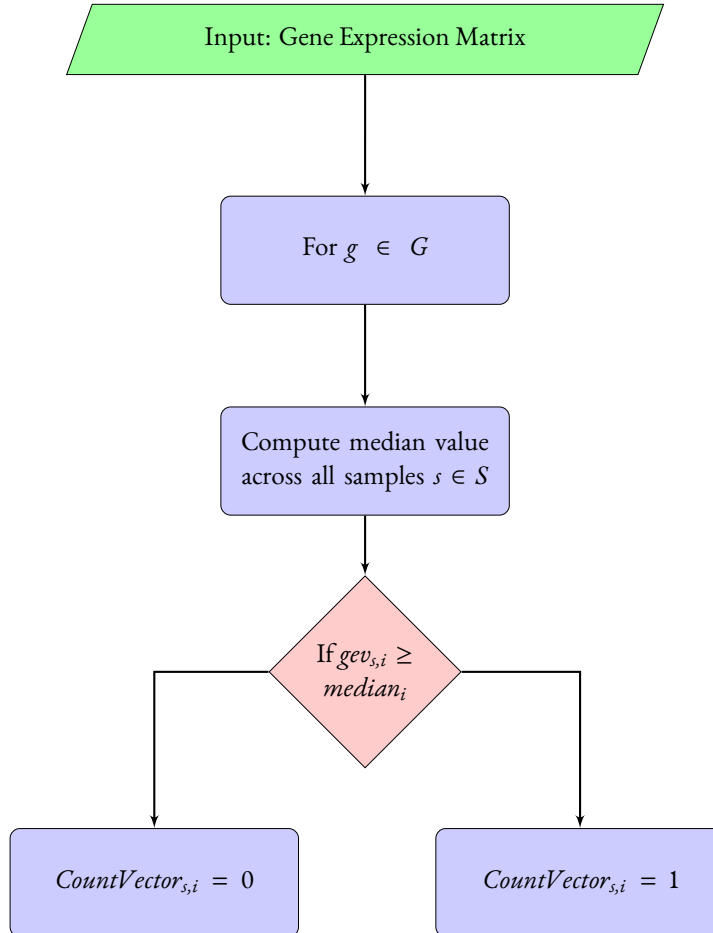


Figure 3.3: Workflow for implementing the Median Method.

3.1.2 Repetition

Another well established BoW transformation approached used in the Microarray Data setting is the *Repetition method* [Bicego et al., 2010]. This approach exploits segmentation of data into bins. The intuition behind this approach is that for each sample, the gene expression values are sorted and then divided into b bins of equal size, i.e. bins whose range is of same length. To create the count vector, we interpret the ascending order of the bin in which a gene is assigned to, to the corresponding number of occurrences in the count vector. To exhibit this behaviour think of the following example: suppose we have a sample s_i , for which we need to obtain its document d_i , and the sample contains gene expression values from a multitude of $V = 5$ genes $G = g_1, \dots, g_5$. Assume that we desire a total number of bins $b = 5$. The graphical representation of this procedure is shown in Figure 3.4. On the left side of Figure 3.4, we present the representation of 5 genes into 5 bins, and on right the produced/resulting document.

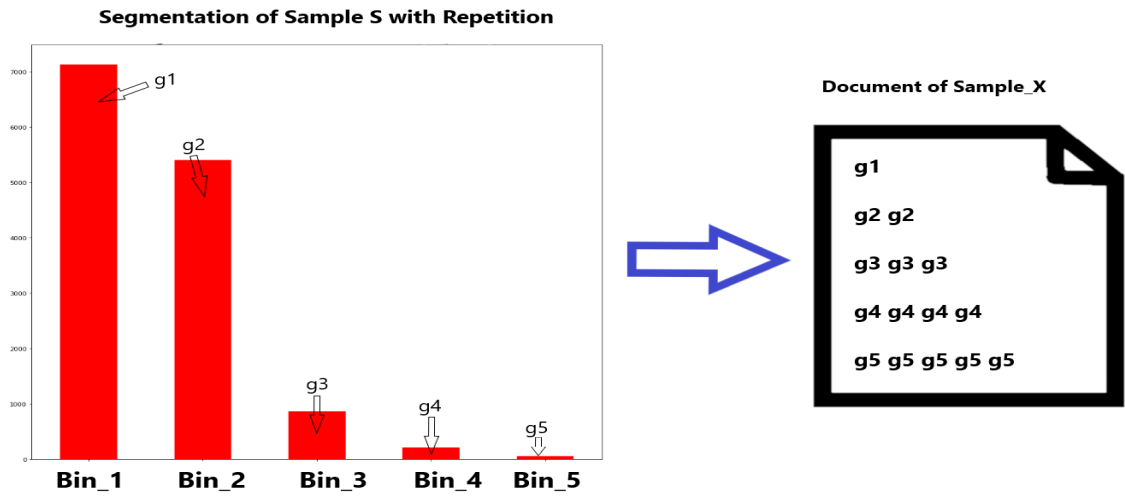


Figure 3.4: Example of transforming a gene profile to a Bag of Words with Repetition Method.

Having the explained the fundamentals of this approach, we introduce a novel pipe-lined process based on this method in order to remove existing noise from the data. The pipeline is illustrated in Figure 3.5. Following all the possible steps in this pipeline we can create several different corpora. We will examine the significant

nodes of this pipeline.

- If t_1 : In the first step, we introduce a *pre-threshold* t_1 , which we adopt in order to remove gene expression values below this threshold. We refer to the threshold as pre-threshold since it is applied prior to performing data binning. It is distinguishable, that we can decide whether if we wish to apply t_1 .
- Remove $gev < t_1$: In this process, considering we have chosen a value of t_1 , expression values in that sample that fall below t_1 are removed and replaced with *nan* values.
- Segment data to b bins. This process corresponds to the methodology described earlier. The objective is to create b bins, whose range is of same length and assign each gene to its equivalent bin.
- Remove Bin 1? At this step we can choose to remove the bin which is found first in the ascending order. This interprets to removing the genes (and their gene expression values) included in the first bin, and hence remove the genes with the lowest expression.
- If t_2 : In this step we introduce *post-threshold* t_2 . The notion of this threshold is to remove genes below t_2 , and not an entire bin.
- Remove $gev < t_2$: Using the same methodology as in removing t_1 , all the expression values larger or equal to t_2 are retain, while the rest are replaced by nan values.
- Create the count vector. Following the above, the last step is to create the count vector of the sample. To do so, each genes' number of occurrences in the generated document for sample i , is set to the corresponding bin number. That is, given g_j belonging to bin k , the count variable of g_j is k .
- Generate *Document_i*. In order to transform the sample to a BoW depiction, the count vector is exploited. That is, we generate a document which contains words representing the expressed genes, while the times each word is repeated derives from the count vector.

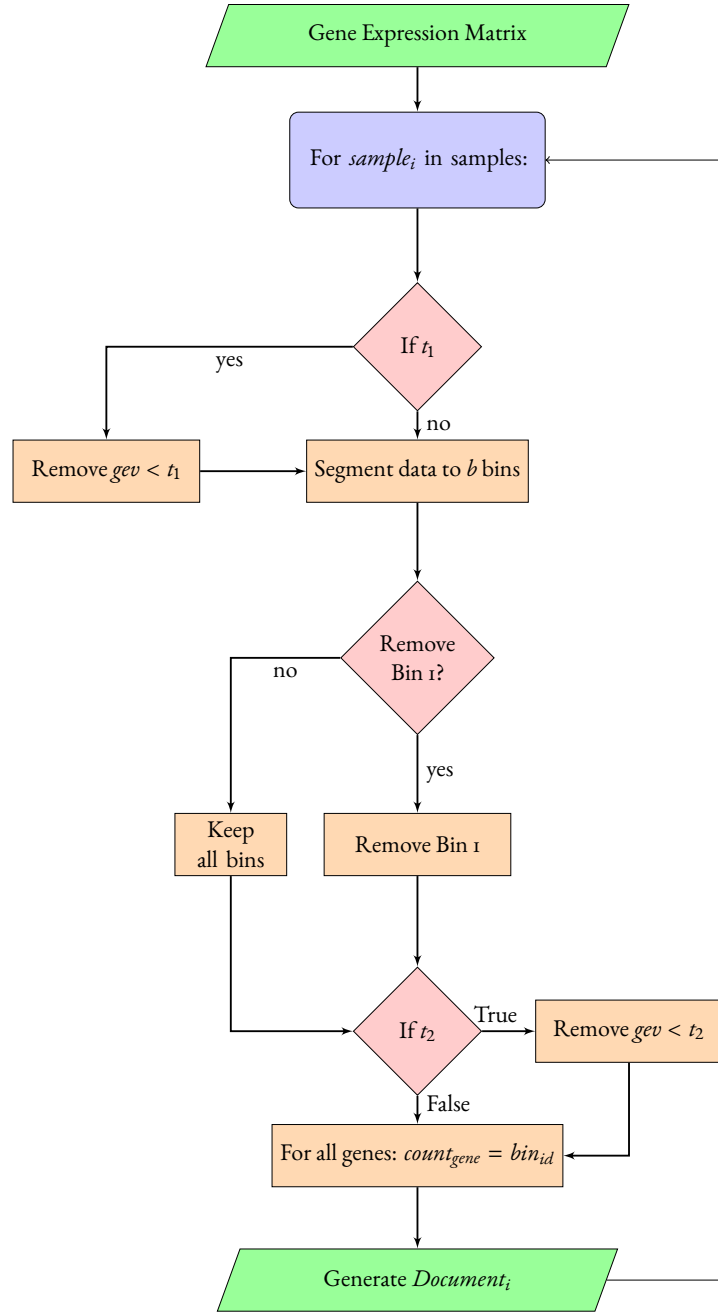


Figure 3.5: Workflow for implementing the Repetition Method

3.1.3 Bibin: A Novel Transformation Method

Bibin is our novel transformation method, which is an *(a) unsupervised*, and *(b) direct* method. Bibin is unsupervised, as no information of the class associated with the data is used, and direct as it is a method that divides the data into n sub intervals (bins) of equal size simultaneously. In most existing methods, as in Repetition, the data is typically discretized by assigning to the continuous value the corresponding bin number; this method is configured in a slightly different way. Likewise Repetition method, we begin by segmenting the data (i.e., the gene expression values of a tissue sample), into n bins simultaneously. The range of each bin is of equal size. Then, we detect the median bin. This bin is set to be Bin_1 . The immediate sub intervals, both on the left and on the right of the median bin, are assigned respectively to number two, and so forth.

More concretely, let n be the total number of bins used, and $h = (\max - \min)/n$ be the width of each bin. The median bin is defined as:

$$Bin_1 = [\min + h * (b + 1)/2 - h, \min + h * (n + 1)/2) \quad (3.1)$$

as such, we let Bin_2 be

$$Bin_2 = [\min + h * (n + 1)/2 - 2 * h, \min + h * (n + 1)/2 - h) \cup [\min + h * (n + 1)/2, \min + h * (n + 1)/2 + h) \quad (3.2)$$

Equivalently, we build the remaining bins. The gene expression (real numbered) values that belong in a certain bin are converted to discrete by setting those values to their assigned bin number. The resulted discrete gene expression matrix comprises the count vector of each sample, which is then transformed to a document. Within a produced document a given gene appears as many times as the discrete value indicated by the bin the gene is placed in. For example, if for a specific tissue sample s_1 , the specific gene g_1 is placed in the median bin, then the term g_1 will appear in the produced document exactly once. By following the described procedure we allocate to both the bin containing the smallest and the bin containing the highest values, the same discrete variable. Using this technique, the same significance is granted to both the under-expressed and over-expressed genes. This idea is essential, since a disease can be caused by groups of genes that can be either expressed in a smaller or in a larger amount than the intended. In Figure 3.6 we present a simple example of this transformation method. Suppose that we examine tissue Sample_X; set the number of bins to $n = 5$. Next, we segment the data into $n = 5$ bins and assign each bin its corresponding number as stated earlier on. On the left part of this figure, five different genes are illustrated, which belong in the five different bins, while on the right, an illustration of the transformed document is shown.

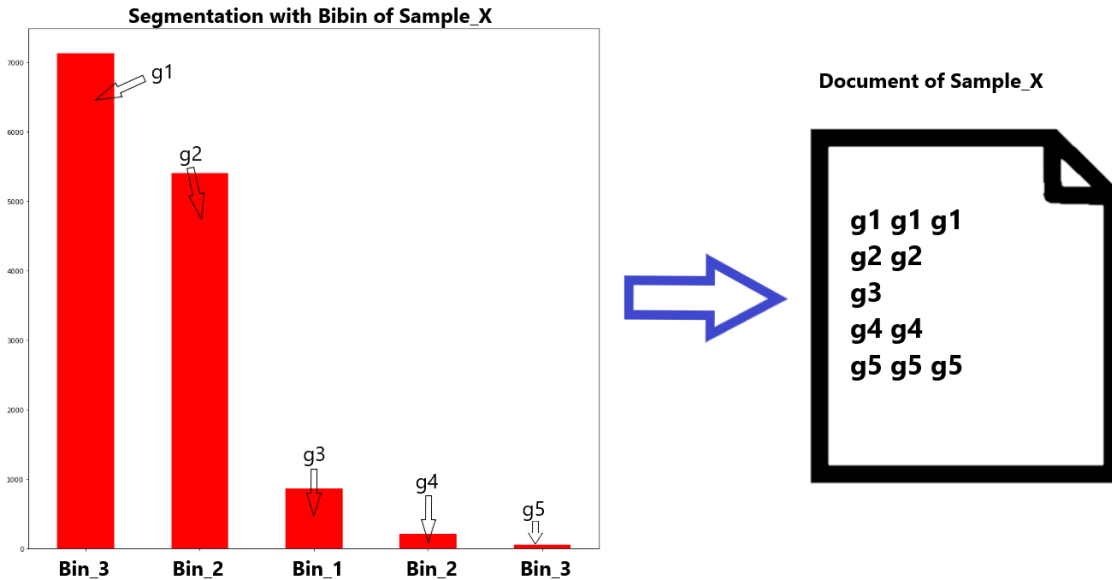


Figure 3.6: Example of transforming a gene profile to a Bag of Words with Bibin Method.

3.2 Probabilistic Topic Models on Gene Expression Data

Having employed any of the above transformation methods, the BoW representation is obtained. The next part of our proposed pipeline illustrated in Figure 3.1 relates to decide the probabilistic topic model algorithm alongside with its corresponding parameters. As discussed in Chapter 2, there exists multiple topic models such as PLSA, LDA, HDP, LPD and others. If one decides to employ Latent Dirichlet Allocation, the number of topics to be inferred must be provided as input. In the Hierarchical Dirichlet Process, this parameter is inferred by the model. The number of topics, is a non-trivial decision. Supposing that our data is labeled according to a condition, we could bias the topic model, by setting the number of topics K to that of the number of different diseases. In another case, we could wish to infer biological processes that were active during the microarray experiment, and thus set the number of topics to an equivalent number.

In any case, provided that a topic model has been employed in a corpus C comprised by a set of documents D , a vocabulary of V terms/genes, we obtain the following:

- a set of K topics
- the distribution ϑ_D , a K -dimensional vector which exhibit the distribution of topics over all documents in D
- and ϕ_K , is the distribution over the vocabulary V for each topic in K .

Both distributions are visualized in Figure 3.7, in which ϑ is illustrated as the document term distribution matrix on the right, and similarly, distribution ϕ is depicted on the left as topic-term distribution matrix. Given these outputs, we will study/examine how we can exploit them for Dimensionality Reduction and direct visual interpretation of the PTMs produced clusters. For short we will refer to this as PTM-C.A . Precisely, we aim at using the outputs of PTMs for two purposes; the first for additional ML-tasks (i.e. perform dimensionality reduction), and the second for interpreting the derived clusters of genes and samples using visualization algorithms and tools.

3.3 Dimensionality Reduction

Following the above, the next step is to analyze the output of topic models. As described in the previous chapter, a significant task relative to microarray data is to reduce the number of dimensions of the dataset. By doing so, we are provided with the capability of extracting more meaningful knowledge, and training more accurate and faster learning models. In this part we will present approaches for exploiting the results of probabilistic topic modeling algorithms in order to perform the “Dimensionality Reduction” node of Figure 3.1. We visualize this process in a workflow diagram in Figure 3.7. Specifically, the topic-term distribution ϕ_K is exploited in order to perform, feature selection, while the document-topic distribution ϑ_D is used for feature extraction. It should be noted at this part, that in our instance of the problem, i.e. gene expression matrices, a feature is equivalent to a gene, as the variables of the observations are genes. Both dimensionality reduction approaches are evaluated using classification, where the class to be predicted is the condition of the tissue or cell sample. In addition, we evaluate the feature selection method by using clustering, as well.

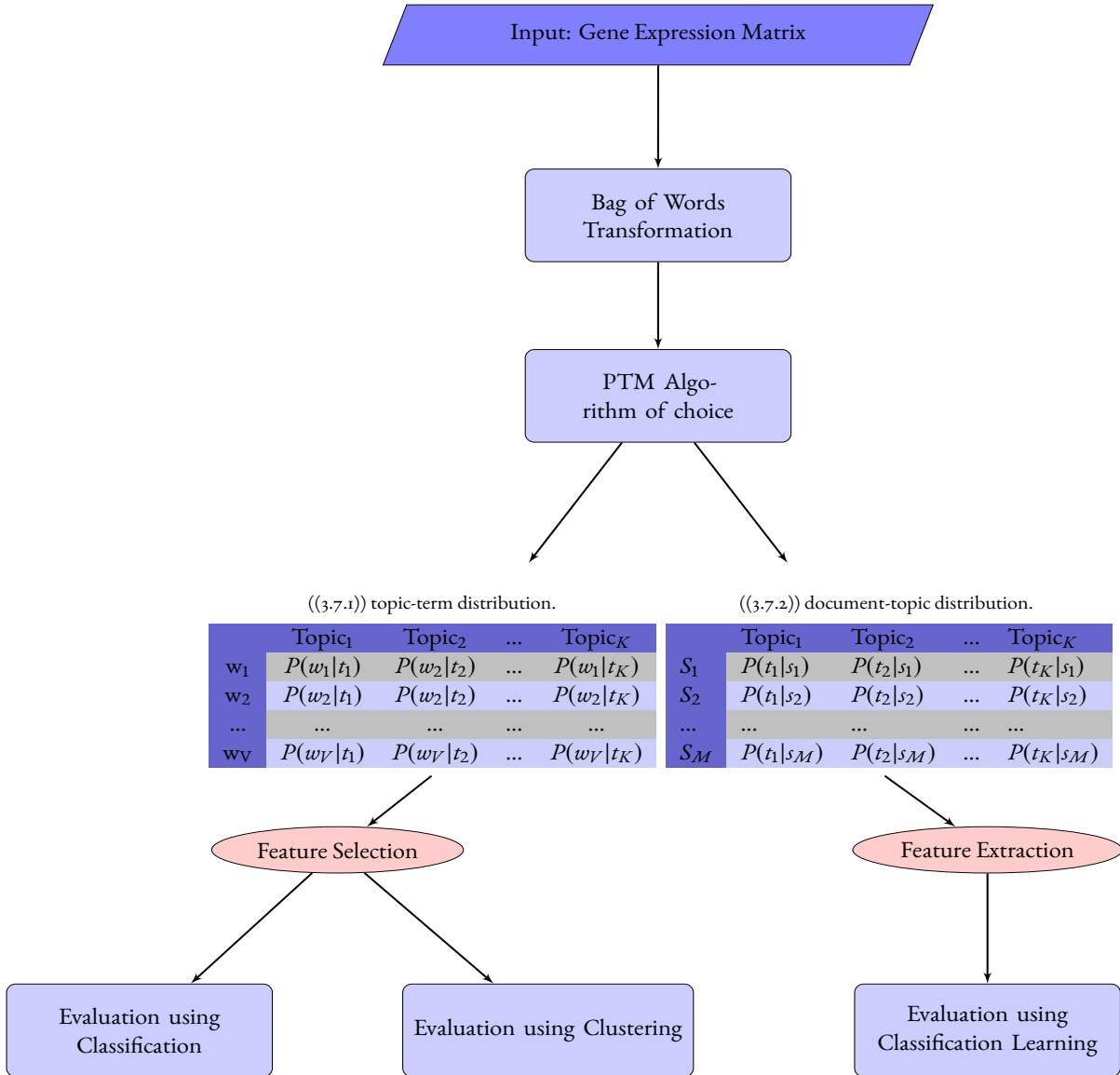


Figure 3.7: Workflow for Dimensionality Reduction

3.3.1 Feature Selection Using Extracted Topic Models

When performing feature selection, our goal is to select a subset of features S , that characterizes the dataset, while removing features that are irrelevant and redundant. We put forward an unsupervised feature selection method to select a primitive subset of features/genes based on the results of a probabilistic topic modeling algorithm, shown on Figure 3.8.

In order to perform feature selection, we exploit directly the Topic Term distribution, obtained by the PTM algorithm, to select the genes that are most “important” (i.e., those most probable to belong to a specific topic). Moreover, we have utilized two metrics, namely the *Relevance Score* [Sievert and Shirley, 2014] and *KL-divergence* [Dey et al., 2017], which we applied on the Topic Term distribution, as alternative ways to select different gene subsets S . For convenience, we will refer to these three “gene ranking” methods—i.e the (simple) “Topic Term distribution”, the “Relevance Score” (over the Topic-Term distribution), and “KL-Divergence” (over the Topic-Term distribution)—methods as “metrics” or “scoring approaches”. Again, the intuition is to select subsets S of the initial set of genes V , depending on the corresponding score. These subsets emerge by choosing the b^1 most significant terms from each topic, given the selection metric used.

After having selected a subset S of the initial features V , one should measure the quality of this subset and how well it represents the initial dataset. To do so, we will perform clustering and classification for evaluation. We feed the algorithms with the gene expression matrix that includes only the genes selected in S . We correlate the performance of the ML-Algorithms to how well S describes the dataset. To evaluate classification techniques we measure their performance by predicting the ground truth, which in our case is the disease. In clustering, we can measure the performance by taking into account that we have labelled data, or visualize the results in order to extract information by specialists.

Hence, through this process we eliminate genes from the expression matrix, which we assume that are redundant using the Topic-Gene Distribution. By doing so, not only are we able to perform gene selection, but also we are *ranking* the genes according to their score using the Topic-Gene Distributions, the Relevance Score, and KL-Divergence. In what follows, we describe the policy of selecting a subset of genes for each of the scoring approaches.

¹Not to be confused with b in the definition of Bibin Method.

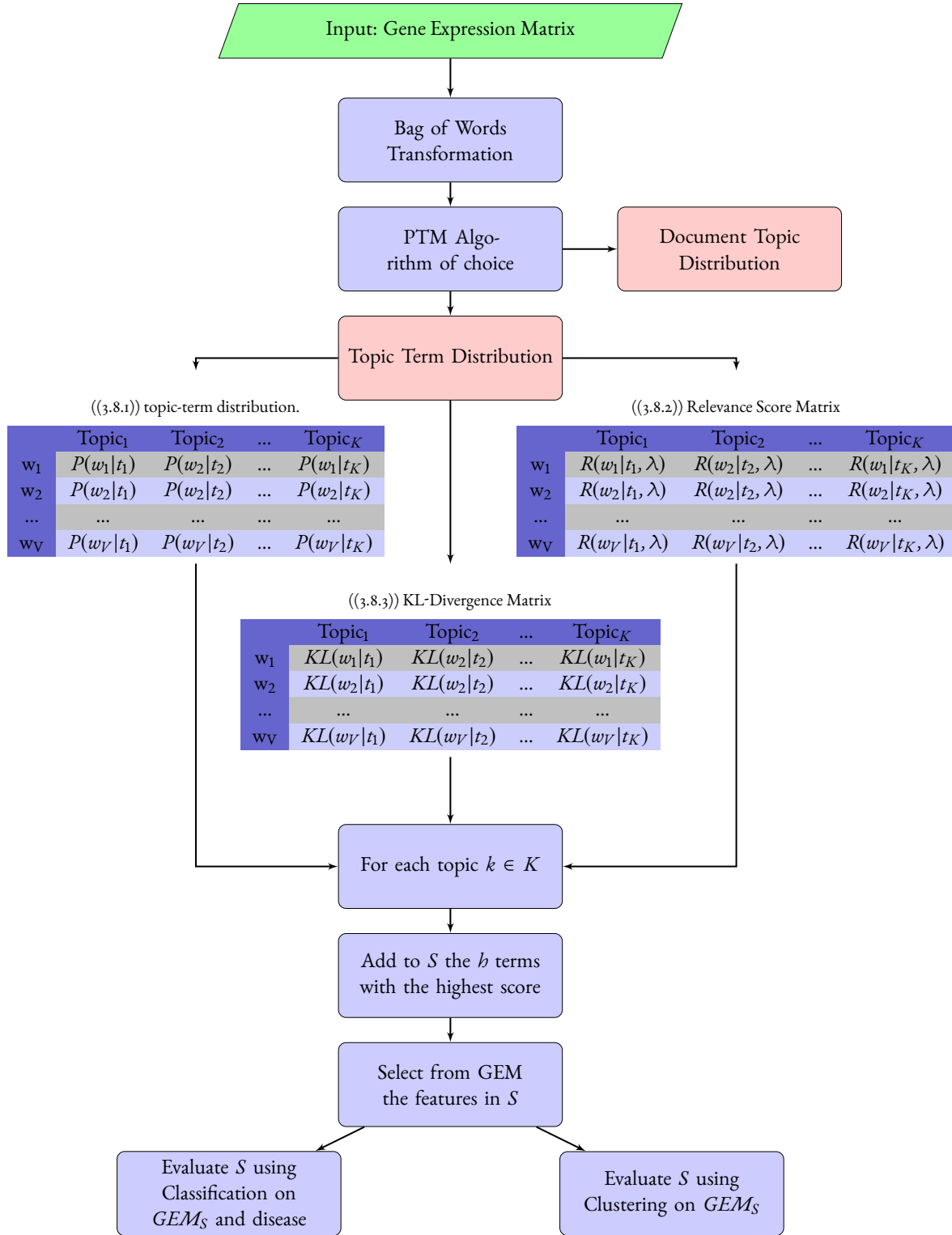


Figure 3.8: Feature Selection using Extracted Topic models.

3.3.1.1 Topic-Term Distribution

Given a corpus C of D documents, a vocabulary V of discrete terms, and the number K of topics the topic-term distribution ϕ is produced as seen in Figure 3.8. The amount $P(w_1|t_1)$ indicates the probability of term (word) w_1 appearing in topic t_1 . This value's interpretation is considered as the significance of term w_1 in topic t_1 . The intention is to select a subset S of the original number of features V , where $S \subset V$. This is achieved by selecting

the h genes which have the highest probability for each of the topics $k \in K$. Hence, $|S| = h * |K|$.

3.3.1.2 Relevance Score

The second technique we introduce is utilizing in a innovative way the *Relevance Score* which was proposed in [Sievert and Shirley, 2014] in order to visualize topic models. By contrast, we exploit this score in order to perform feature selection. This metric is applied on the yielded topic term distribution ϕ by ranking the terms within each topic $k \in K$ according to their relevance in the corpus. More precisely, using the probability of a term w within a topic k and the overall probability of term w , p_w from the empirical distribution of the corpus.² The relevance score is defined as:

$$r(w, k|\lambda) = \lambda \log(\phi_{kw}) + (1 - \lambda) \log \frac{\phi_{kw}}{p_w} \quad (3.3)$$

where variable λ specifies the weight assigned to the probability of term w in topic k relative to its ratio to p_w and $0 \leq \lambda \leq 1$. Apparently, assigning $\lambda = 1$ will result to rank the terms only on the probability of the term occurring in the specified topic. In contrast, if λ is set to zero, the terms will be ranked only by their lift. In this context, we choose S , as a subset of the h most relevant genes in each $k \in K$ given a value for λ . Apparently, the Relevance score can be used only in topic models that follow the BoW paradigm. For example, in LPD, we cannot measure the probability of a term appearing in the document.

3.3.1.3 Differentially Expressed Genes using Kullback-Leibler Divergence

In the work of [Dey et al., 2017], in an attempt to clarify the topics, the authors develop a method that identifies the most “*distinctively differential expressed genes*” in each topic. Particularly, considering that LDA generates the topic-term distribution ϕ , the intention is to identify the genes that determine a biological process in each topic using the Kullback-Leibler Divergence [Kullback and Leibler, 1951]. However, we exploit this metric in a different, novel way, in order to select a subset S of the initial genes, which contains the distinctively differential expressed genes. Specifically, for each topic $k \in K$, the distinctiveness of each gene g with respect to any other topic $l \in K$ is measured using:

$$KL^g[k, l] := \phi_{kg} \log \frac{\phi_{kg}}{\phi_{lg}} + \phi_{lg} - \phi_{kg} \quad (3.4)$$

which measures the Kullback-Leibler divergence with parameter ϕ_{kg} to parameter ϕ_{lg} . The variable ϕ_{kg} indicates the probability of gene g occurring in topic k , inferred by a probabilistic topic model. Then the *distinctiveness* of gene g for each cluster $k \in K$ is defined as:

$$D^g[k] = \min_{l \neq k} KL^g[k, l]. \quad (3.5)$$

The higher the value of D^g , the more indicative it is that the role of gene g in topic k is vital. In order to select the subset of features/genes S , we select for each topic $k \in K$ the h features/genes, which have the highest D^g value.

3.3.2 Feature Extraction Using Extracted Topic Models

Another dimensionality reduction technique, is that of feature extraction. In this approach, we do not select a subset of features and their corresponding values on the dataset, but project the initial feature space to a smaller

²The probability p_w can be smoothed by including prior weights as pseudocounts [Sievert and Shirley, 2014]

one. Our approach regarding feature extraction, is exploiting the document topic distribution \mathcal{D} as the projected space of the initial data as seen in Figure 3.9. Specifically, supposing that D is the number of samples, V is the number of features/genes and K the number of topics, the initial dimension of the GEM is $N \times M$. We project these dimensions to a smaller space by depicting the GEM as the document-term distribution, and thus the space is reduced to $D \times K$. That is, we interpret the topics as the reduced feature space, and their corresponding probabilities in each document as the values of the new feature space. As in the feature selection setting we evaluate our approach by employing Supervised Learning. In a similar manner, we associate the performance of the classifier on predicting the tissue samples condition, to the performance of the feature extraction approach.

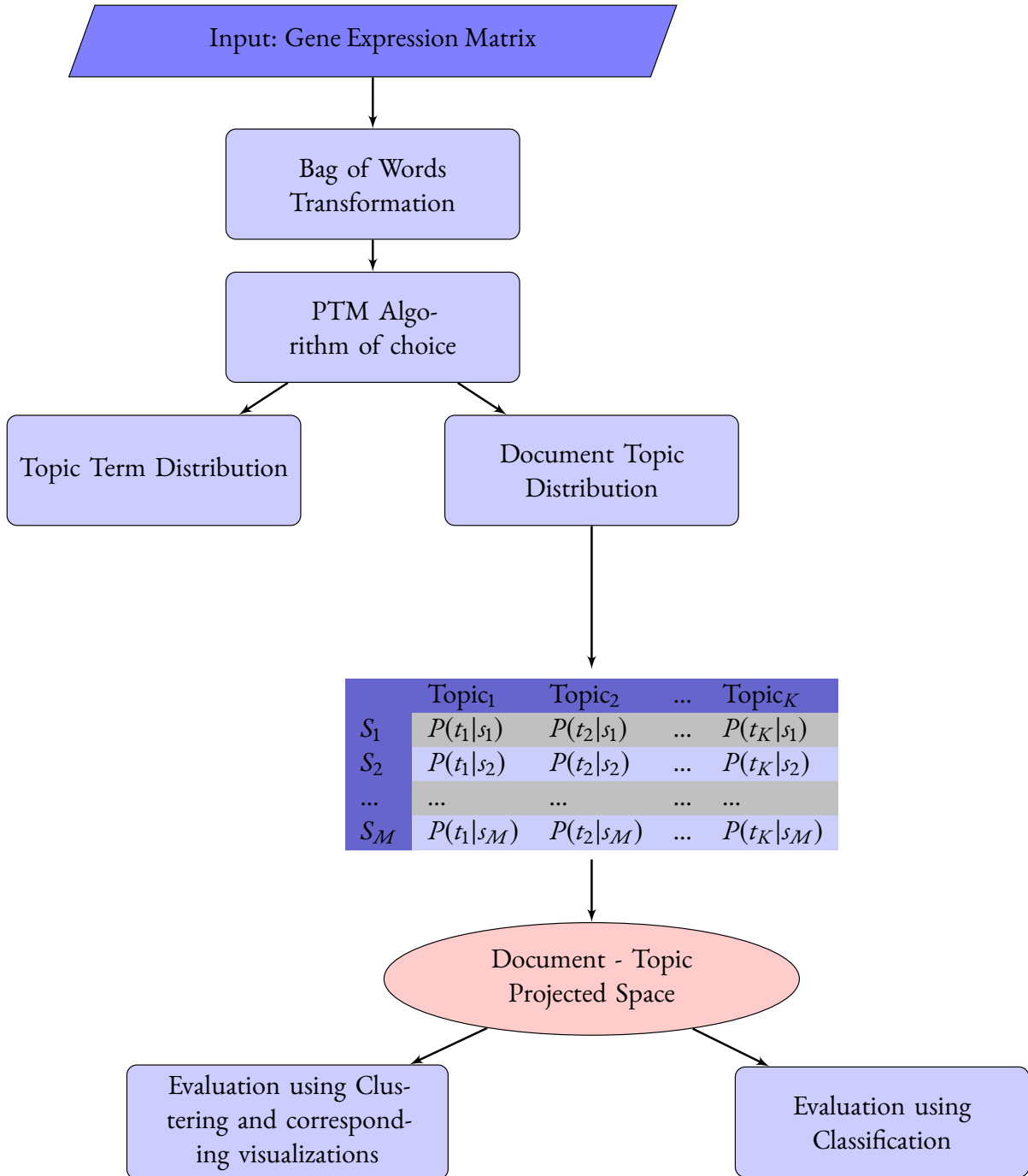


Figure 3.9: Workflow for Feature Extraction

3.3.3 PTM-Clustering Analysis (Ca) Visualizations

Probabilistic Topic Models are unsupervised models, that allow us to simultaneously create overlapping clusters of words and documents. In the microarray data setting, we are able to cluster genes and tissue samples at the same time. Also, the documents/tissue samples and the words/genes can belong with a different proportion to various clusters. However, unsupervised learning is difficult to evaluate. For this reason, we exploit visualization tools to give the ability to specialists to analyze the results of topic models, and extract knowledge. That is, we utilized pyLDAviz [Sievert and Shirley, 2014] to visualize the topic term distributions on the 2D space, and t-distributed Stochastic Neighbor Embedding [Maaten and Hinton, 2008] for the document-term distributions. The results drawn from these models are presented in the following chapter.

Distribution	Clustering Visualization Algorithm
Topic Term	pyLDAviz
Document Topic	t-SNE

Table 3.1: Cluster Analysis Visualization.

Experimental Setting

In this chapter, we exhibit the experiments we conducted on the scope of this thesis. We first present in detail the datasets used, namely a well-established dataset of gene expression data in normal and cancerous breast tissues, and a microarray experiment comprised of healthy and diseased muscle tissues. Recall that in the workflow presented in Figure 3.1 the first step is to create a corpus by implementing a Bag of Words Transformation Method. Here we depict the different variants of corpora produced with the three transformation methods (i.e. Median, Repetition, Bibin) with respect to the different thresholds we employed in each method. Furthermore, we provide the topic models implemented alongside with their settings. We then present the results we drew by performing dimensionality reduction, cluster analysis using visualizations of the extracted topic models.

4.1 Data Preprocessing

First and foremost, we will be examining the data included in the microarray experiments we utilized, and the various corpora we produced using the “Bag of Words” transformation methods.

4.1.1 The Datasets

We will briefly recount the datasets used in this thesis to evaluate our methods. Both datasets are gene expression matrices obtained from microarray experiments. For the sake of consistency, we assume that:

- D is the set of documents/samples of size M .
- V denotes the set of the distinct words/terms/genes/features, which contains N elements.

Muscle Tissue Diseases

The first dataset used to evaluate the work and methodologies proposed, is a microarray experiment of Muscle Tissues. For this Dataset no analysis similar to the conducted has been performed. This data is comprised by a gene expression matrix(GEM) that consists of $M = 114$ samples and $N = 13650$ genes. Each entry in the matrix characterizes the expression level, (i.e a real number) of a particular gene in the particulate sample. Thus, the GEM at hand is a table of $114 \text{ samples} \times 13650 \text{ gene names}$.

Table 4.1: An visualized example of the gene expression matrix

samples\genes	Gene Name 1	Gene Name 2	...	Gene Name 13649	Gene Name 13650
Sample 1	expression_value _{1,1}	expression_value _{1,2}		expression_value _{1,13649}	expression_value _{1,13650}
Sample 2	expression_value _{2,1}	expression_value _{2,2}		expression_value _{2,13649}	expression_value _{2,13650}
⋮					
Sample 114	expression_value _{114,1}	expression_value _{114,2}		expression_value _{114,13649}	expression_value _{114,13650}

Accompanied by the gene expression data, we have at our disposal a collection of metadata that distinguish each sample. The included metadata are:

- The date of the experiment.
- Sex, containing categorical variables Mixed, Female and Male.
- Age $\in [1, 90]$
- Condition. Shown in Table 4.2
- Anatomic Part. Shown in Table 4.2

Conditions	Anatomic Parts
Normal	Biceps
Myotonic Dystrophy type 1 (DM1)	Deltoid
Myotonic Dystrophy type 2 (DM2)	Quadriceps
Tibial Muscular Dystrophy (TMD)	Tibialis Anterior
Facioscapulohumeral Muscular Dystrophy (FSHD)	Tibialis Posterior
Inclusion Body Myositis IBM	Soleus
Necrotizing Myopathy NM	Extensor Digitorum Longus
Dermatomyositis DM (with and without perifascicular atrophy (PFA))	Gastrocnemius
Polymyositis PM	wna
Duchenne Muscular Dystrophy DMD	

Table 4.2: Conditions and Anatomic Parts found in the Muscle Disease Dataset.

TCGA Breast Cancer

The Breast Cancer dataset contains mRNA-Seq data of 229 breast invasive carcinoma samples obtained from The Cancer Genome Atlas (TCGA) portal [NCI, 2019]. Within this dataset 117 samples represent primary solid tumour samples and 112 solid normal tissues. Following the filtering authors performed in [Kho et al., 2017], genes that have expression value equal to zero in more than 10% of the genes are removed. After performing this filtering step, a total of 23424 genes are present in the Gene Expression Matrix. This resulted to a GEM of $M = 229$ samples and $N = 23424$ genes, hence to a dimensionality of 229×23424 .

4.1.2 Corpora Variants

Both of the datasets were transformed into the BoW paradigm using the Median, the Repetition and the Bibin approaches. Each transformation produces several corpora depending on its methodology. Due to the thresholds we introduce for each of the transformation methods, we extract different corpora, with different *Vocabularies* and *Total number of Words*. A *corpus*, in our setting is the collection of the datasets samples and their corresponding representations as documents. These collection can be perceived as different versions of the initial

dataset (distinguished by the different transformation methods and thresholds). We provide this information on these corpora and the settings in which they were generated.

We incorporated different thresholds, under the guidance of the provides of the Muscle Dataset [Malatras et al., 2019]. In particular we were suggested that expression values below 0.2 were noise. However we test various threshold values (and no removal of data at all), in order to evaluate the results produced by th PTMs. That is, we wish to examine the impact of the noise, and if no removal of data can lead to extracting different knowledge.

4.1.2.1 Median

On the Muscle dataset, regarding the Median transformation, we performed experiments in which we applied a threshold $t_M \in \{0.0, 0.2\}$ and an approach in which we conducted no filtering. That is, we remove expression values that do not exceed a certain value. These expression values will not appear in the produced document of the corresponding sample. In addition their values are not taken into account when finding the median value. However we do not remove these genes from the GEM, only if they are removed from all the samples. Simply put, we their word-count in the document-sample will be zero. As a result, we obtained three different variations of corpora using this discretization method. In Table 4.3 we present these variations for the Muscle Diseases Microarray Experiment Data alongside their corresponding corpus details.

When transforming the TCGA dataset using the Median method, we experimented on using both no pre-processing and a threshold $t_M = 0.2$. We present in Table 4.4 the corpora produced by the Median transformation method in this dataset. In Tables 4.3, 4.4 the Vocabulary(V) indicates the number of distinct words that occur in the corpus, and the Total Word Positions specify the total number of words in all the documents included in the corpus. The different values of the thresholds on the two datasets occur due to the range of data in each dataset. The Muscle Disease Dataset includes $gev \in \mathbb{R}$, whereas the TCGA dataset includes $gev \in \mathbb{R}^+$.

Method	t_M	Vocabulary (V)	Total word positions
Median	\times	13650	778050
Median	0.0	13608	698922
Median	0.2	13228	549734

Table 4.3: Transforming the Muscle Disease Data with Median Method. Variable t_M indicates the threshold, V is the size of the vocabulary, to keep consistency with LDA notations. Total word positions indicate the entire corpus size.

Method	t_M	Vocabulary (V)	Total Word Positions
Median	\times	23424	2693760
Median	0.2	23369	2149766

Table 4.4: Transforming the TCGA Breast Cancer Data with Median Method. Variable t_M indicates the threshold, V is the size of the vocabulary, to keep consistency with LDA notations. Total word positions indicate the entire corpus size.

4.1.2.2 Repetition

The Repetition Transformation method was applied to both datasets and produced several corpora. After removing the variants-corpora that were identical we concluded to the resulting corpora shown in Table 4.5 for the Muscle Disease Dataset. For this dataset, when applied, the pre-threshold t_1 was set to $\{0.0, 0.2\}$ and the post threshold t_2 to $\{0.2, 0.3\}$. Equivalently, for the TCGA dataset we produced the corpora shown in Table 4.6. In this case the pre-threshold t_1 was set to 0.2 and the post-threshold $t_3 = 0.3$. In Tables 4.5, 4.6, column *Remove Bin 1* indicates if the first bin was removed as our proposed workflow suggests.

An interesting observation, is that if decide to remove the first bin in the TCGA dataset, the resulting corpus has very few total word positions. This occurs since the concentration of the values are very close to 0. Later on we will remark on how this affected our experiments.

METHOD	t_1	Remove Bin 1	t_2	Vocabulary (V)	Total Word Positions
Repetition	\times	\times	\times	13650	8387854
Repetition	\times	\times	0.2	13228	6830832
Repetition	\times	\checkmark	0	13608	7904311
Repetition	\times	\checkmark	\times	13650	8386233
Repetition	0.0	\times	\times	13608	4553131
Repetition	0.0	\times	0.2	13228	4253367
Repetition	0.0	\checkmark	\times	12782	4131419
Repetition	0.2	\times	\times	13228	3624149
Repetition	0.2	\times	0.3	12446	3483033
Repetition	0.2	\checkmark	\times	10940	3280584

Table 4.5: Datasets that occur after applying Repetition transformation method on the Muscle Disease Dataset. Variables t_1, t_2 denote the pre and post threshold respectively. Vocabulary(N) is the size of the vocabulary, keeping consistency with LDA's vocabulary variable. Total word positions indicate the total size of each corpus considering how many words the corpus contains.

Method	t_1	Remove Bin 1	t_2	Vocabulary(V)	Total Word Positions
Repetition	\times	\times	\times	23424	5420316
Repetition	\times	\times	0.2	23424	4338994
Repetition	\times	\checkmark	\times	23424	72391
Repetition	0.2	\times	\times	23369	4338977
Repetition	0.2	\times	0.3	23369	4057598
Repetition	0.2	\checkmark	\times	23369	72363

Table 4.6: Datasets that occur after applying Repetition transformation method on the TCGA Dataset. Variables t_1, t_2 denote the pre and post threshold respectively. Vocabulary(V) is the size of the vocabulary, keeping consistency with LDA's vocabulary variable. Total word positions indicate the total size of each corpus considering how many words the corpus contains

4.1.2.3 Bibin

The Bibin Transformation method was conducted on both datasets also. For the MD microarray experiment the value of the threshold t_B (equivalent to the Median threshold) was assigned to $t_B = \{0.0, 0.2\}$. In Table 4.7 we present the different corpora produced in each case, depending on the value of the threshold. Similarly, for the TCGA dataset we produced the corpora shown in Table 4.8.

Method	t_B	Vocabulary (V)	Total word positions
Bibin	χ	13650	6110907
Bibin	0.0	13608	7007144
Bibin	0.2	13228	5482386

Table 4.7: Transforming the Muscle Disease Data with Bibin Method. Variable t_B indicates the threshold, V is the size of the vocabulary. Total word positions indicate the entire corpus size.

Method	t_B	Vocabulary (V)	Total Word Positions
Bibin	χ	23424	58962959
Bibin	0.2	23369	47068438

Table 4.8: Transforming the TCGA Data with Bibin Method. Variable t_B indicates the threshold, N is the size of the vocabulary. Total word positions indicate the entire corpus size.

4.2 Experimental Setup

In this section we outline the experimental setup, regarding on which topic models we utilized and their settings, the microarray analysis results, and specifically results concerning feature selection and extraction, whose performance is evaluated using both supervised and unsupervised learning. We outline the algorithms used in this section in Table 4.9. Furthermore, we provide visualizations of the results of topic models to allow biologists to extract the uttermost amount of information regarding the interpretation of the topics.

Model	Usage	Catergory
PTM	Extracting topic-gene, sample-topic distributions	Unsupervised
SVM	Evaluate Feature Selection, Feature Extraction	Supervised
k-NN	Evaluate Feature Selection, Feature Extraction	Supervised
kmeans	Evaluate & Visualize Feature Extraction	Unsupervised
HAC	Evaluate & Visualize Feature Selection, Feature Extraction	Unsupervised
t-SNE	Visualize Sample-Topic Distributions	Unsupervised

Table 4.9: Machine Learning Algorithms implemented for in this work.

4.2.1 Topic Modeling Algorithms

A significant part of our proposed framework is the implementation of topic modeling algorithms (or “topic models” for short) of choice. To compare the performance of topic models we exploited Latent Dirichlet Allocation (LDA) and Latent Process Decomposition (LPD). Furthermore we utilized the Hierarchical Dirichlet Process in order to extract the number of topics when needed.

PTM	Number of Topics K	Data Input
LDA	Required as Input	BoW
HDP	Inferred by the Model	BoW
LPD	Required as Input	Continuous

Table 4.10: Topic Models implemented in this work.

Latent Dirichlet Allocation

Regarding the implementation of LDA on the Muscle Disease Dataset we employed the following process for each transformation method and its generated corpora:

- Firstly, we use the HDP algorithm from [Wang, 2010] in order to retrieve an “optimal” number of topics, K , for each of the produced corpus. We ran HDP several times and set the number of topics as the average inferred number.¹
- Next, we performed the *LDA with Gibbs Sampling* inference method with the Python wrapper for Latent Dirichlet Allocation (LDA) from MALLET [McCallum, 2002] implemented in the Gensim Library [Řehůřek and Sojka, 2010]; using as topics parameter the number K obtained from the HDP algorithm. The online version of LDA was utilized, and the number of iterations was set to 1000.²

¹Note that we performed the HDP algorithm only to obtain a good enough K , to be used as a later input to LDA.

²We could simply run HDP, but we ran into technical problems regarding the output of the software.

The number of topics inferred for each of the variants of the Muscle dataset are presented in Tables 4.11, 4.12, 4.13 for the Median, Repetition and Bibin Method respectively.

Method	t_M	Vocabulary (V)	Total word positions	HDP- K
Median	\times	13650	778050	10
Median	0.0	13608	698922	11
Median	0.2	13228	549734	25

Table 4.11: Corpus Information and number of topics K inferred by HDP the Median Method and its Variants in the Muscle Disease Dataset.

METHOD	t_1	Remove Bin 1	t_2	Vocabulary (V)	Total Word Positions	HDP- K
Repetition	\times	\times	\times	13650	8387854	14
Repetition	\times	\times	0.2	13228	6830832	12
Repetition	\times	\checkmark	0	13608	7904311	7
Repetition	\times	\checkmark	\times	13650	8386233	16
Repetition	0.0	\times	\times	13608	4553131	12
Repetition	0.0	\times	0.2	13228	4253367	16
Repetition	0.0	\checkmark	\times	12782	4131419	15
Repetition	0.2	\times	\times	13228	3624149	15
Repetition	0.2	\times	0.3	12446	3483033	18
Repetition	0.2	\checkmark	\times	10940	3280584	18

Table 4.12: Corpus Information and number of topics K inferred by HDP for the Repetition Method and its Variants in the Muscle Disease Dataset.

Method	t_B	Vocabulary (N)	Total word positions	HDP- K
Bibin	\times	13650	6110907	12
Bibin	0.0	13608	7007144	17
Bibin	0.2	13228	5482386	19

Table 4.13: Corpus Information and number of topics K inferred by HDP for the Bibin Method and its Variants in Muscle Disease Dataset.

In Figure 4.1 we present an indicative visualization of the distribution over the samples with respect to each topic for the Bibin Transformation method where $t_B = 0.2$ for the MD dataset. In these figures we have labeled the disease of each sample. Each bar represents a sample and the value on axis y is the probability of this sample belonging to each topic respectively. The various diseases are expressed in an ambiguous manner, however the results are not that disappointing as we will present later. We can observe for example that similar diseases such DM1 and DM2, co-occur in the topics. In the Appendix of this thesis, we include visualizations of the topic term distributions for all the variants.

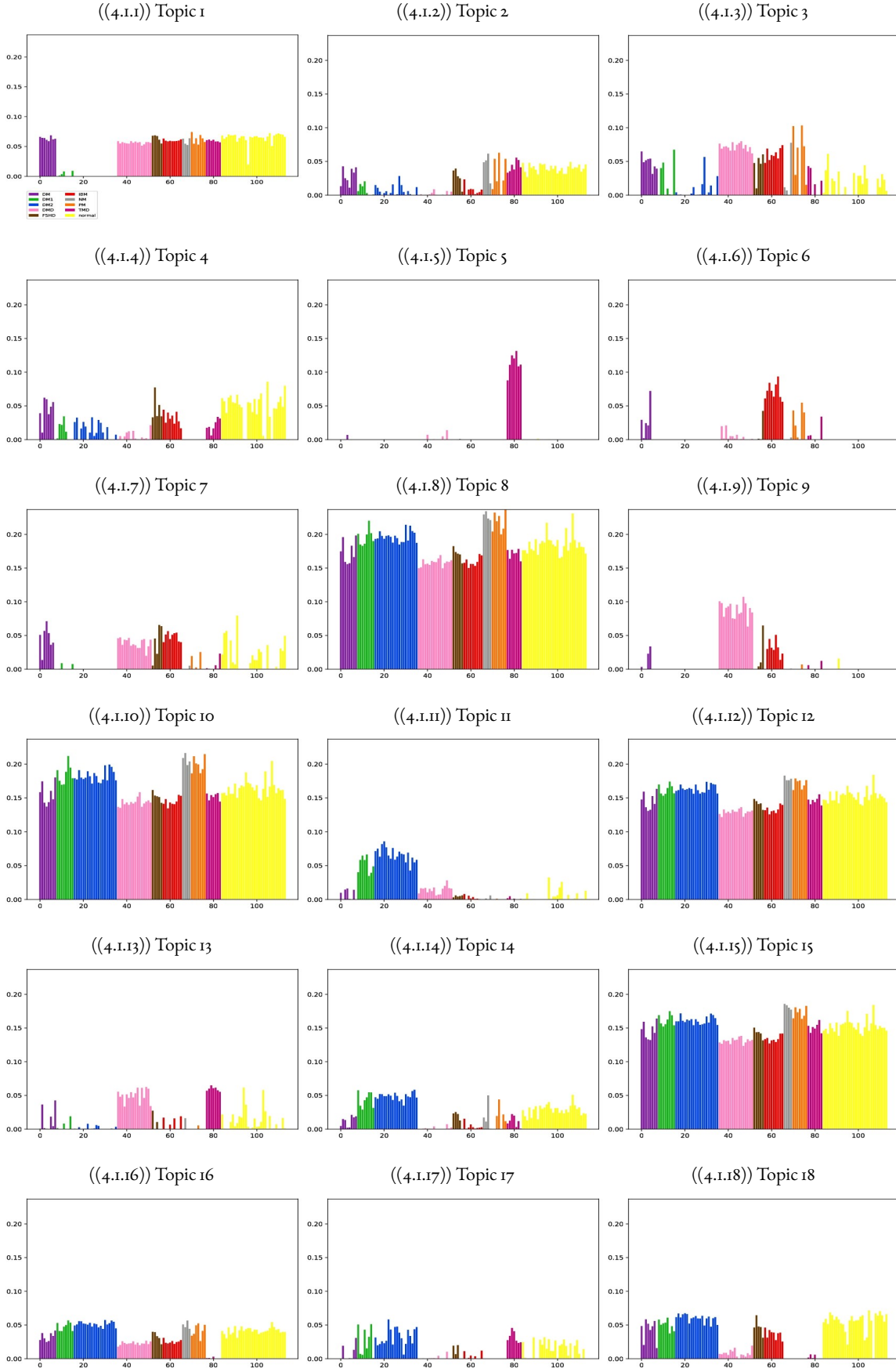


Figure 4.1: The distribution over the samples for each topic on the Muscle Dataset using the Bibin method with $t_B = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

If we observe topic 11 in Figure 4.2, it can be seen that this topic exhibits with higher probability the topics labeled with “Myotonic Dystrophies of type 1,2” and “Dermatomyositis”. Moreover, topic 4 in Figure 4.3 depicts a topic which is characterized by samples with “Tibial Muscular Dystrophy”.

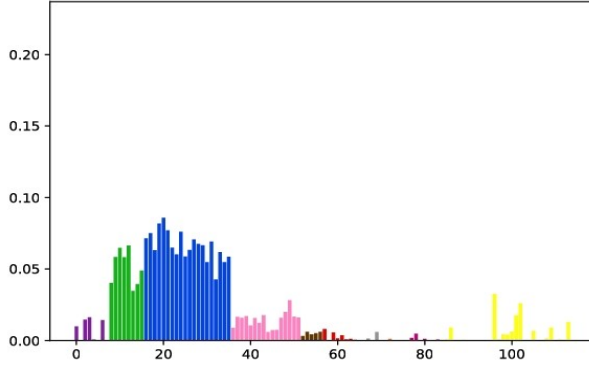


Figure 4.2: The distribution over the documents for Topic 11 in the MD Dataset using Bibin with $t_B = 0.2$.

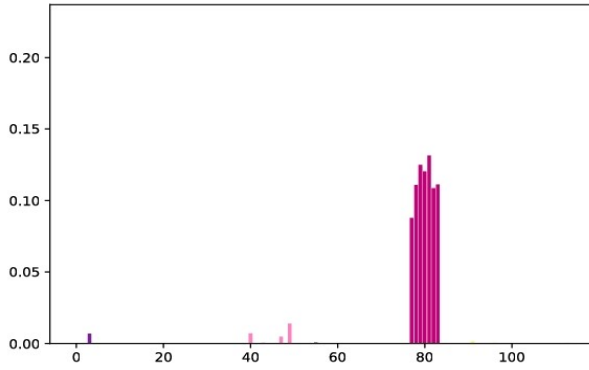


Figure 4.3: The distribution over the documents for Topic 4 in the MD Dataset using Bibin with $t_B = 0.2$.

Moving on or the TCGA Breast Cancer Dataset, we set the number of topics in the same setting of [Kho et al., 2017] to compare our results, which is $K = 3$. In Figure 4.4 we present the distribution of the topics over the samples. Each sample is represented in the graphs by the bar and the height of the bar exhibits the probability of the topic. The samples are also colored according to their condition, which is either cancerous (yellow) or normal (black).

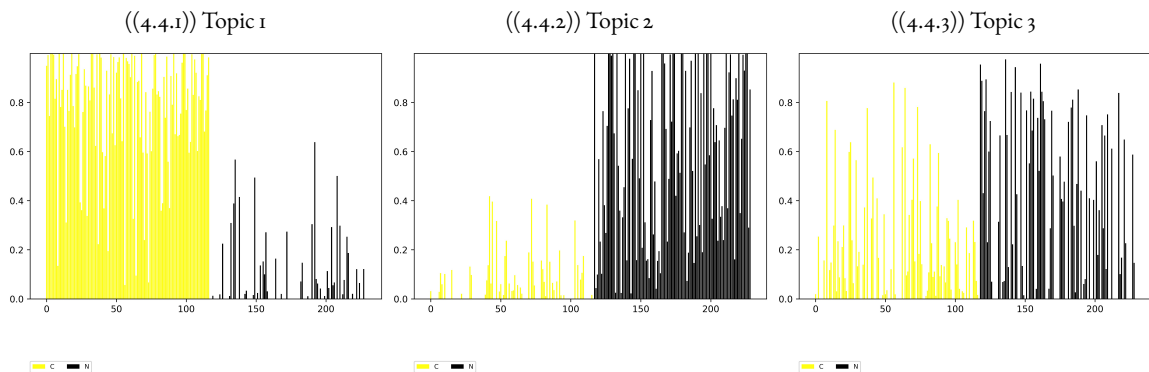


Figure 4.4: The distribution of the topics over the samples for the TCGA dataset using the Median method with $t_M = 0.2$

Latent Process Decomposition

One of the incentives of this thesis is to compare the performance of a topic model that requires data that follow the “discrete” bag of words paradigm, to a topic model that does not, and hence is adjusted to the microarray setting, namely Latent Process Decomposition. For this reason we employed the implementation of LPD obtained from [Rogers et al., 2005].

In what concerns the Muscle Disease Dataset, LPD was carried out on the gene expression matrix, while the number of processes was set to $K = \{5, 10, 15\}$ and the number of iterations was set to 1000. In this part we also applied a preprocessing methodology in which we removed data below a certain threshold $t_{LPD} = \{0.0, 0.2\}$. Genes whose values were below this threshold across all samples were removed. The remaining gene expression values lower than t_{LPD} were replaced with 0. In Figure 4.5 we exhibit the distribution over the samples for each topic produced by LPD, applied on the original gene expression matrix (i.e. no value was set to t_{LPD}) where the number of topics was set to $K = 15$. We observe that each topic is distinguished by the disease more accurately than the topics distinguished by LDA in the Figure 4.1. Moreover, diseases that are similar in a biological manner, are more probable in each topic. For example in topic 10 and topic 14 the most probable samples are samples that have been label with Myotonic Dystrophy of Type 1 and Type 2. In the Appendix we provide all the document topic distributions obtained by LPD on the muscle Dataset.

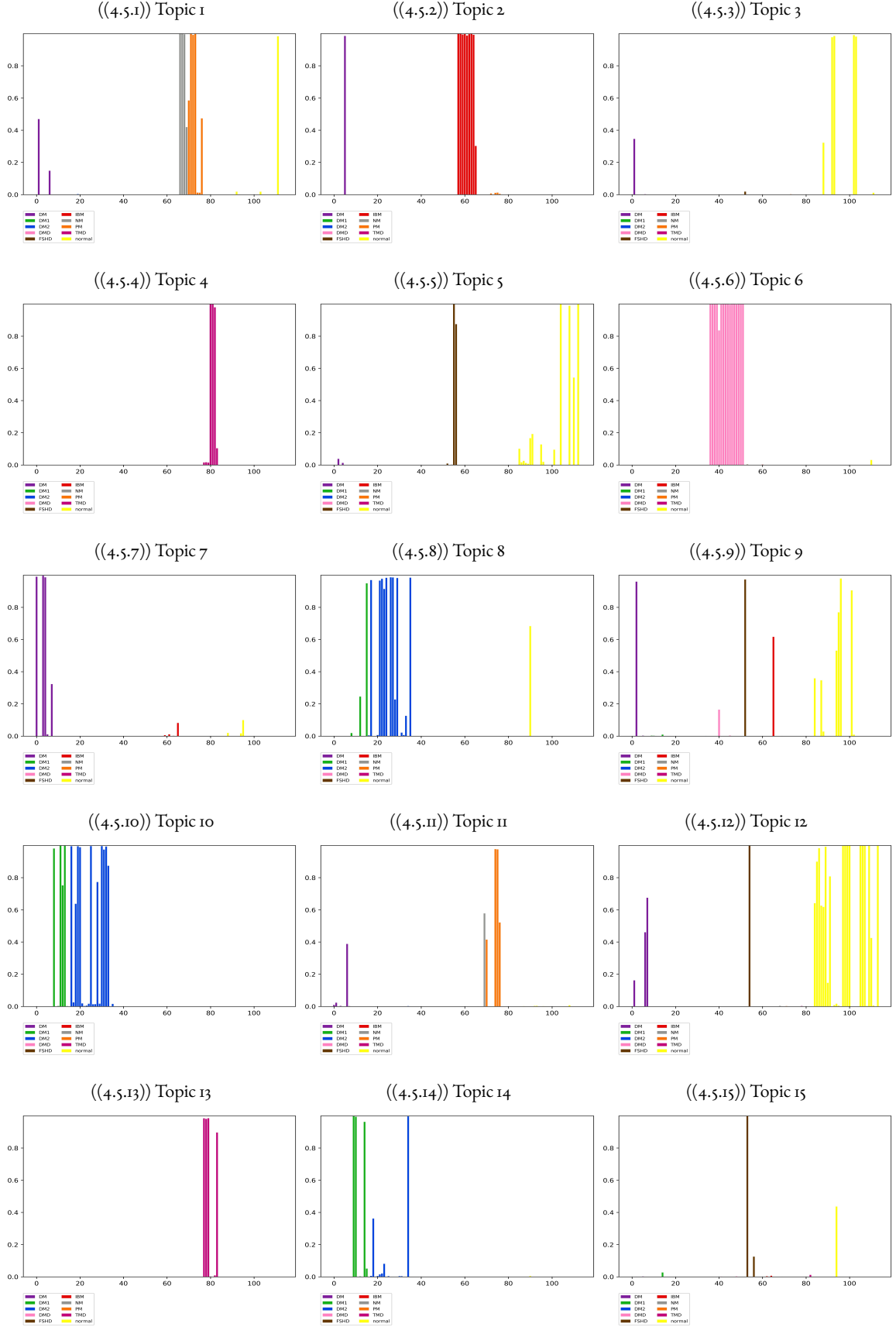


Figure 4.5: The distribution over the samples for each topic on the Muscle Dataset on the original GEM obtained by LPD. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are coloured according to the disease of the sample.

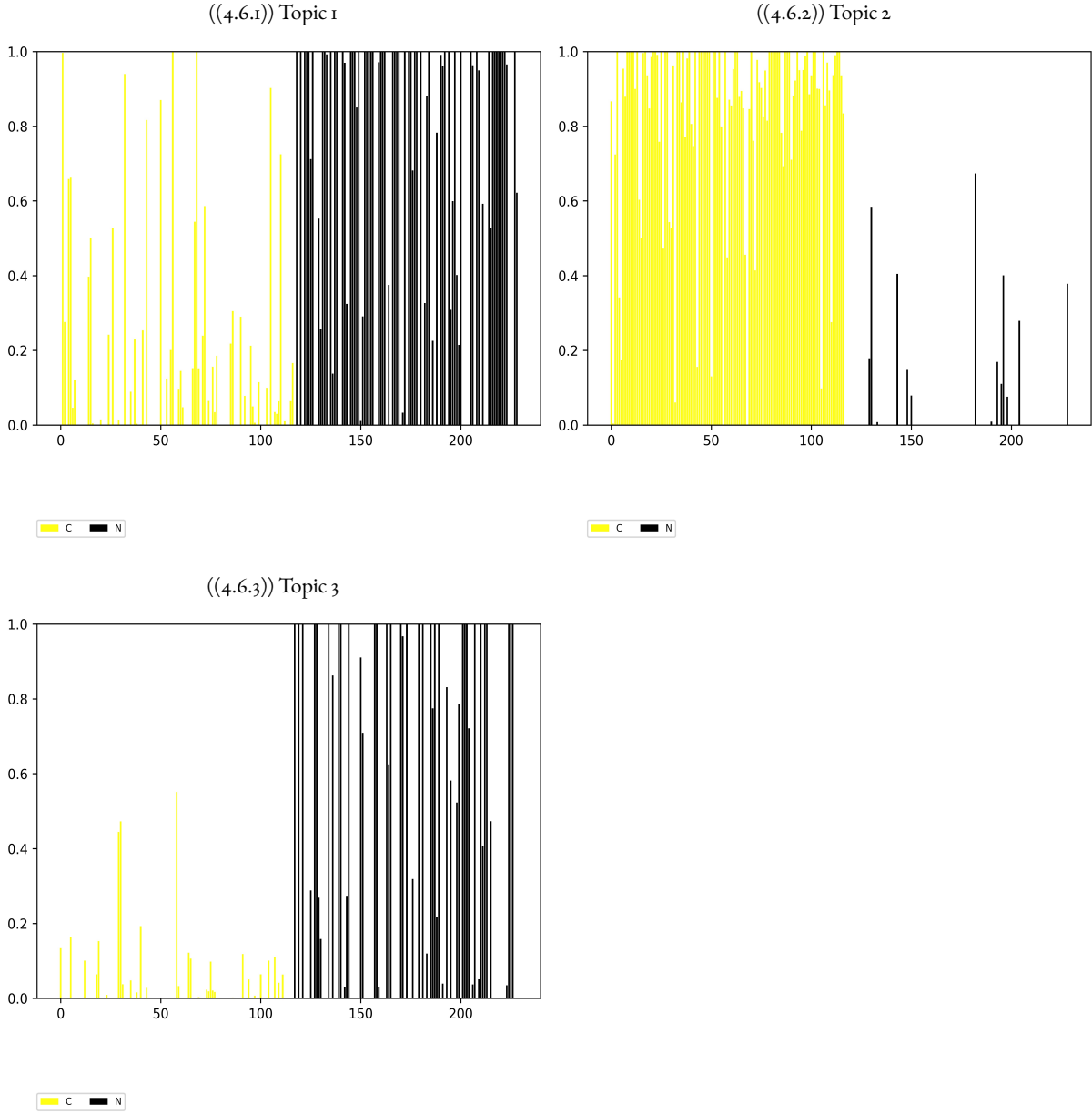


Figure 4.6: The distribution over the samples for each topic on the TCGA Dataset on the original GEM obtained by LPD. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

Regarding the TCGA Breast Cancer dataset, we employed LPD on the original gene expression matrix, while setting the number of topics to $K = \{3, 5\}$. In Figure 4.6, we present the regarding distribution over the documents with respect to each topic. Here the topics are relatively distinguishable regarding the condition of the sample.

4.2.2 Feature Selection using the extracted Topic Models

In this part we will present the methodology and results regarding the feature selection approaches, which are the *Unsupervised Feature Selection using LDA & LPD*. As such, the methodology we provide is one that serves as ranking gene collections by importance. As we discussed in the presentation of the framework, we describe an approach for selecting the most important genes in each topic according to various metrics. These extracted genes, can be thought as different lists of genes (one for each topic), and then used for enrichment analysis for characterizing the topics in a biological manner.

4.2.2.1 Unsupervised Feature Selection using LDA

Following, we show the results of our novel approach to solve the curse of dimensionality with feature selection. As described in the previous chapter, we have developed a methodology on how to select a subset of the initial feature space, which is able to efficiently represent the dataset by utilizing the *Topic-Genes matrix*, the *Relevance Score matrix*, and the *Differentially Expressed Genes using Kullback-Leibler matrix*. These components can be extracted from most of the topic models that use the bag of words paradigm. In this scenario, these outputs were obtained by Latent Dirichlet Allocation. From each of these outputs, we obtain a subset of genes S by selecting from each topic $k \in K$ the h genes with the highest score. To evaluate the performance of our approach we ran systematic experiments with a variety of values for the parameter h . For the assessment of our proposed method, we conduct both classification and clustering and compared our proposition to Univariate Feature Selection.

It should be noted also that in the Relevance Score instance, an assignment to variable λ is required. Recall that the Relevance Score ranks the terms within a topic, by weighting the importance of a term in a topic relevant to its significance in the whole corpus. We intend to evaluate the influence of the weight λ in this scenario, and thus assign $\lambda = \{0.3, 0.5, 0.8\}$.

In the classification scenario we assigned variable h values from the range of $\{10, 300\}$, and in the clustering evaluation we set h to $[50, 100, 150, 200]$. To train the classifiers and our baseline feature selection algorithms, we utilized both the initial gene expression matrix of continuous data, and the discretized matrix (i.e. the count-vector) which we generated using the Bag of Words transformation methods. On the other hand, the clustering models were trained only on the Gene Expression Data.

At this point, we should consider the fact that during the transformation methods, we “cleaned” the data. Contemplate that for example in the Median Method, we generated a corpus where we removed gene expression values below a threshold $t_M = 0.2$. In the corresponding generated count vector, entries that are not included in a sample have *count* = 0, which results to no missing or *nan* values and thus creates no conflict for our baseline or classification algorithms. However, when utilizing the continuous expression data (i.e. the Gene Expression Matrix) for training our classifiers and baseline methods, the data will contain *nan* entries, which would produce implications on the employment and the performance of the feature selection baseline algorithm and the classifier (or clustering algorithm). Of course, for consistency reasons, we wish to evaluate the results of the PTM algorithms with respect to those of the baseline feature selection algorithm when operating on the same data. To do so, we generate new Gene Expression Matrices, that contain the same information used by the PTMs. Thus, to tackle the problem of the “missing” values, we replace the gene expression values that were removed with $e_{ij} = 0$.

Particularly, by performing this methodical comparison, we are also able to evaluate the performance of the transformation methods and their preprocessing steps.

To perform a systematic and methodical comparison, we employ the Univariate Feature Selection algorithm (SelectKBest) implemented using scikit-learn [Pedregosa et al., 2011], and examine its behavior using the same number of features as in our approach. In the case where the data is the continuous gene expression matrix, we remove all but the $|k * b|$ features with the highest ANOVA F-value, while on the discrete data we choose the $|k * b|$ genes with the highest chi-squared statistics. As mentioned earlier, for the Unsupervised F.S, we assign variable b a multitude of different values. The total number of features selected in any case is $|k * b|$. For consistency, we provide as input to the Univariate feature selection algorithm the same number of features.

Evaluation using Classification

As a first step we employed supervised learning. In this evaluation approach we assigned variable b values from the range of $\{10, 300\}$. In order to assess the optimality of each of the subset S we extracted using the three different scores, (Topic-Term Matrix, KL-Divergence, Relevance Score) we employed two classifiers. The first classifier utilized is a Support Vector Machine with Linear Kernel. The second classifier employed is the K-Nearest Neighbor algorithm, where the value of the number of neighbors K_n is optimized and selected from values $\{3, 5, 7\}$.

Moreover, we use K-fold cross Validation with number of folds $K_f = 3$ for the MD Dataset, and $K_f = 8$ for TCGA, to prevent overfitting in the learning models. In general K-Fold Cross Validations is a technique retained in supervised learning, that partitions the dataset into K_f different folds. In particular, the $K_f - 1$ partitions are used as the training set and the held out as test set. This process is repeated until the all the partitions are used as a test set.

The MD dataset is a multi-class problem, and in specific classes the amount of instances is very small, leading to a small K_f . This occurs, due to the fact that we do not wish to create train or test folds in which some classes are not included. On the contrary, the TCGA data is a binary-classification problem, allowing us to use a larger K_f , that of 8. Moreover, we utilized a version of K-fold cross validation, that allows to preserve a relative percentage of samples for the different classes.

In both classifiers, in order to evaluate their performance we use as accuracy score we use the mean cross-validated score. Supposing that \hat{y}_i is the predicted values of the i -th sample and y_i is the corresponding true value, the accuracy of a classifier is defined as:

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

where $1(x)$ is the indicator function.

To begin with, we will exhibit the results of Feature Selection using LDA (henceforth “LDA FS” for short) on the Muscle Dataset. Overall, we observed that the optimal results of our proposed method regarding the classifier were achieved using the Support Vector Machine. We trained both classifiers for all the transformation methods and their variants, on both discrete and continuous expression matrices, and for all the scores we mentioned in our approach (Topic-Term distribution, KL-Divergence and the Relevance score with $\lambda = \{0.3, 0.5, 0.8\}$). In addition, we trained the models with various values regarding the variable b ranging from $[10, 300]$. In the Median Transformation method and its produced variants, we trained the classifiers only on the continuous gene expression values and not on the count vector, because the nature of the method (binomial variables) resulted to learning models which achieved very low scores.

In Tables 4.14, 4.15 and 4.16 we present for each of the transformation methods respectively, the optimal average results which our approach achieved.³ We consider the optimal as the score that attained the average accuracy for all the different values of variable h . In the Matrix field we depict which scoring method achieved the best results. It should be noted also that the average is computed by finding the mean accuracy for all the different values of h that were used. Variable K indicates the number of topics, whereas the Data Type denotes which of the continuous (GEM) or the discrete (Count Vector) attained the best average accuracy. The colored cells indicate which algorithm attained the best accuracy. We also performed classification on the corresponding gene expression matrix without performing any feature selection and the results lie under the Classification Accuracy columns. due to the computational cost of the classifier including all the genes, this experiment was performed only once. For our approach on applying Feature Selection with LDA on the Muscle Disease Dataset we achieved the best results using the Repetition Transformation.

The Median method, behaves comparable, in contrast to the Univariate algorithm and simple Classification. This was not unexpected, due to the nature of the Media Transformation. The fact that a word appears in a document or not produced a topic term distribution, in which we observed that the majority of terms occurred with the same probability. In a way, the UFS Algorithm was choosing randomly genes. Classification (via SVM) without performing feature selection achieves better accuracy in this case, although its computational time was demanding. Due to its running time, we only ran one experiment of simple classification for each of the datasets. The Bibin method also produces comparable results to the SelectKBest method and the simple classifier. As one can notice, Repetition transformation method outperforms the baseline algorithm.

Method	t_m	Metric	Classifier	K	Data Type	LDA FS	SelectKBest	Simple SVM ("no feature selection")
Median	X	KL-Divergence	SVM	10	GEM	0.812	0.841	0.859
Median	0.0	0.8-Relevance	SVM	11	GEM	0.819	0.845	0.860
Median	0.2	KL-Divergence	SVM	25	GEM	0.828	0.842	0.842

Table 4.14: Classification results (average over all h values) using SVM for all threshold-defined "Median" transformation method variants in the MD dataset. Each row presents the best LDA results (across all genes selection/scoring metrics tried). Though Simple SVM (without feature selection) achieves best results for $t_m < 0.2$, SelectKBest does best for $t_m = 0.2$, which is the t_m value indicated by the biologist specialists providing the dataset. LDA FS has a performance that is comparable to that of SelectKBest.

Method	t_m	Metric	Classifier	K	Data Type	LDA FS	SelectKBest	Simple SVM ("no feature selection")
Bibin	X	KL-Divergence	SVM	12	GEM	0.842	0.845	0.859
Bibin	0.0	KL-Divergence	SVM	17	GEM	0.837	0.842	0.860
Bibin	0.2	0.3-Relevance	SVM	19	GEM	0.843	0.833	0.842

Table 4.15: Classification results (average over all h values) using SVM for all threshold-defined "Bibin" transformation method variants in the MD dataset. Each row presents the best LDA results (across all genes selection/scoring metrics tried). Though Simple SVM (without feature selection) achieves best results for $t_m < 0.2$, LDA FS does best for $t_m = 0.2$, which is the t_m value indicated by the biologist specialists providing the dataset.

³The average is computed on the $K_f = 3$ folds.

Method	t_1	Remove Bin 1	t_2	Metric	Classifier	K	Data Type	LDA FS	SelectKBest	Simple SVM ("no feature selection")
Repetition	✗	✗	✗	Topic-Term	SVM	14	GEM	0.852	0.835	0.838
Repetition	✗	✗	0.2	0.8-Relevance	SVM	12	GEM	0.863	0.847	0.835
Repetition	✗	✓	✗	0.8-Relevance	SVM	16	GEM	0.853	0.834	0.860
Repetition	✗	✓	0.0	0.8-Relevance	SVM	7	GEM	0.855	0.839	0.860
Repetition	0.0	✗	✗	0.5-Relevance	SVM	12	GEM	0.864	0.839	0.825
Repetition	0.0	✗	0.2	0.3-Relevance	SVM	16	GEM	0.864	0.841	0.842
Repetition	0.0	✓	✗	KL-Divergence	SVM	15	GEM	0.856	0.840	0.801
Repetition	0.2	✗	✗	0.3-Relevance	SVM	15	GEM	0.868	0.843	0.842
Repetition	0.2	✗	0.3	0.3-Relevance	SVM	18	GEM	0.863	0.835	0.842
Repetition	0.2	✓	✗	KL-Divergence	SVM	18	GEM	0.861	0.841	0.825

Table 4.16: Classification results (average over all h values) using SVM for all "Repetition" transformation method variants in the MD dataset. Each row presents the best LDA results (across all genes selection/scoring metrics tried). LDA FS achieves best results for $t_1 = 0$ and $t_1 = 0.2$. In the variant where no data was removed LDA FS again outperforms both SelectKBest and simple classification with SVM. In two particular variants (at which Bin 1 was removed) simple SVM achieves an accuracy that is only slightly better than that of LDA FS.

In Tables 4.17, 4.18, 4.19 we show the best result achieved by the proposed feature selection approach as per a particular value of h and the corresponding score that attained the best accuracy. We also present for the Bibin and Repetition methods, the best results for both the GEM and the Count Vector. Although the average classification accuracy for our proposed method show that sometimes the baseline performs better, consistently better results are obtained for distinct combination of k and h specific values. Visualizing this data we show that our approach is comparable and can reach rather high accuracy scores in a rather indistinguishable dataset. That occurs due to the fact that the MD dataset has very few samples for a classifier, and a large amount of classes relative to its size.

Method	t_M	Metric	Classifier	Data Type	K	h	k*h	LDA FS	SelectKBest	Simple SVM ("no feature selection")
Median	✗	0.3 Relevance	SVM	GEM	10	25	250	0.824	0.856	0.859
Median	0	0.3 Relevance	SVM	GEM	11	65	715	0.833	0.867	0.860
Median	0.2	KL-Divergence	SVM	GEM	25	230	5209	0.855	0.877	0.842

Table 4.17: Classification accuracy results using SVM for all threshold-defined "Median" transformation method variants in the MD dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. Though Simple SVM (without feature selection) achieves best results for no t_M , SelectKBest does best for $t_m \geq 0.0$. LDA FS has a performance that is comparable to that of SelectKBest.

Method	t_B	Metric	Classifier	Data Type	K	h	K*h	LDA FS	SelectKBest	Simple SVM ("no feature selection")
Bibin	✗	0.5-Relevance	SVM	Count-Vector	12	25	300	0.842	0.839	0.808
Bibin	✗	KL-Divergence	SVM	GEM	12	230	2760	0.868	0.857	0.860
Bibin	0.0	Topic-Term	SVM	Count-Vector	17	215	3557	0.825	0.813	0.779
Bibin	0.0	KL-Divergence	SVM	GEM	17	205	3261	0.868	0.876	0.798
Bibin	0.2	0.3-Relevance	SVM	Count-Vector	19	35	665	0.825	0.782	0.807
Bibin	0.2	0.3-Relevance	SVM	GEM	19	15	285	0.860	0.807	0.842

Table 4.18: Classification accuracy results using SVM for all threshold-defined "Bibin" transformation method variants in the MD dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. LDA FS attains in the majority of certain combinations of h and K better accuracies than SelectKBest and simple SVM. The Data Type column indicates on which type of data (continuous or discrete variables) the SVM was employed.

Method	t_1	Remove Bin 1	t_2	Metric	Classifier	Data Type	K	h	K*h	LDA FS	SelectKBest	Simple SVM ("no feature selection")
Repetition	\times	\times	\times	0.3-Relevance	SVM	Count-Vector	14	130	1820	0.842	0.811	0.841
Repetition	\times	\times	\times	Topic-Term	SVM	GEM	14	70	975	0.886	0.845	0.838
Repetition	\times	\times	0.2	0.8-Relevance	SVM	Count-Vector	12	20	239	0.851	0.805	0.808
Repetition	\times	\times	0.2	0.8-Relevance	SVM	GEM	12	45	537	0.895	0.828	0.835
Repetition	\times	\checkmark	\times	0.8-Relevance	SVM	Count-Vector	16	75	1200	0.860	0.825	0.818
Repetition	\times	\checkmark	\times	0.8-Relevance	SVM	GEM	16	50	800	0.877	0.900	0.872
Repetition	\times	\checkmark	0.0	0.5-Relevance	SVM	Count-Vector	7	140	980	0.842	0.861	0.809
Repetition	\times	\checkmark	0.0	0.5-Relevance	SVM	GEM	7	90	630	0.886	0.829	0.86
Repetition	0.0	\times	\times	0.3-Relevance	SVM	Count-Vector	12	115	1380	0.868	0.843	0.805
Repetition	0.0	\times	\times	0.3-Relevance	SVM	GEM	12	105	1260	0.886	0.854	0.825
Repetition	0.0	\times	0.2	0.3-Relevance	SVM	Count-Vector	16	35	560	0.886	0.823	0.774
Repetition	0.0	\times	0.2	0.3-Relevance	SVM	GEM	16	10	160	0.886	0.831	0.853
Repetition	0.0	\checkmark	\times	0.8-Relevance	SVM	Count-Vector	15	115	1656	0.851	0.817	0.807
Repetition	0.0	\checkmark	\times	KL-Divergence	SVM	GEM	15	65	967	0.886	0.817	0.801
Repetition	0.2	\times	\times	0.3-Relevance	SVM	Count-Vector	15	65	975	0.877	0.801	0.788
Repetition	0.2	\times	\times	0.3-Relevance	SVM	GEM	15	50	750	0.877	0.804	0.842
Repetition	0.2	\times	0.3	0.3-Relevance	SVM	Count-Vector	18	55	984	0.886	0.799	0.781
Repetition	0.2	\times	0.3	0.3-Relevance	SVM	GEM	18	40	715	0.886	0.826	0.842
Repetition	0.2	\checkmark	\times	0.3-Relevance	SVM	Count-Vector	18	35	629	0.868	0.792	0.828
Repetition	0.2	\checkmark	\times	KL-Divergence	SVM	GEM	18	30	534	0.886	0.855	0.825

Table 4.19: Classification accuracy results using SVM for all “Repetition” transformation method variants in the MD dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. LDA FS attains in the majority of certain combinations of h and K better accuracies than SelectKBest and simple SVM. The Data Type (continuous or discrete variables) on which SVM was employed is also indicated.

Furthermore, we visualize in Figure 4.7 the performance of the Unsupervised Feature Selection using LDA approach for the Repetition variant($t_1 = 0$, $RemoveBin = False$, $t_2 = False$) for each of the metrics we utilized using SVM as a classifier. In each of these graphs we visualize the number of features $k * h$ and their corresponding accuracy for each of the algorithms (Unsupervised and Univariate FS). The data type used is the GEM. We observe that our approach, in contrast to the Univariate Feature Selection algorithm, overall performs better, although the baseline algorithm may achieve better accuracies in some values of h . In the we provide the equivalent graphs for all variants, for both discrete and continuous data and for both classifiers.

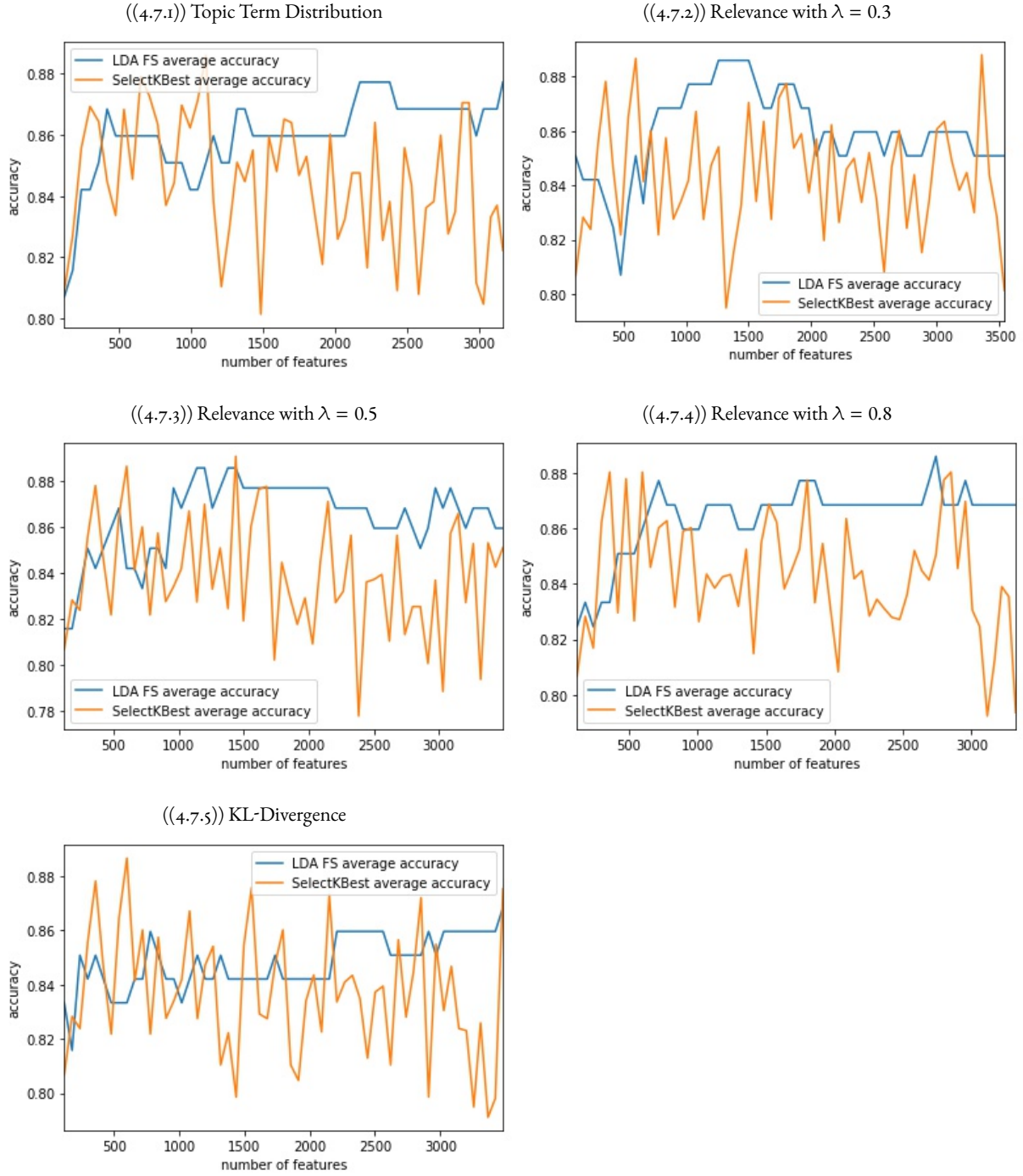


Figure 4.7: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset. These experiments were performed on the Repetition variant ($t_1 = 0$, $RemoveBin = False$, $t_2 = False$) on the GEM using SVM

In what follows we will exhibit the results of classification evaluation of our feature selection approach in the TCGA dataset. We utilized both the count vector and the gene expression data, but in this case, the count vector did not produce competitive results, so we will not be exhibiting these results. The TCGA dataset includes a larger number of samples and genes. Furthermore, the ground truth is either “normal” or “cancer”, which makes this problem a binary classification problem, leading to better accuracies than the Muscle Disease Dataset. In a similar manner, Tables 4.20, 4.21, 4.22 show the average accuracy and the matrix/score which achieved it. The average accuracies are very close in most cases, proving that our approach is solid.⁴ Another remark, is that in general the Topic-Term Matrix does not achieve the best accuracies (either for a given h , or in average). Topic-Term Matrix seems to perform well in certain instances of h in the TCGA dataset, for the Repetition method.

In Tables 4.20, 4.21, 4.22, we show the best accuracies for each of the Transformation method variants, given a certain values of h . In other words, we exhibit the particular h and the score that performs best. Moreover, in Figure 4.8 we show how the two algorithms perform for each of the classifiers for a certain variant of Repetition. When employing KNN the baseline algorithm, achieves accuracies very close to 1, whereas in SVM all of the scores of our approach achieve better accuracies.

At this point, we should outline how the transformation methods affect the results. The median transformation method achieves both optimal and average effective results. On the other hand, the Bibin method, is not that effective although the scores are comparable. In this dataset, the Bibin transformation method overall does not behave so well, due to the distribution that the data follow. A large proportion of the data is rather close to 0, and thus almost all of the genes in each tissue-sample appear in the document with the same frequency. We see also, that the Relevance score achieves good results for both Median and Bibin. Repetition transformation is capturing the continuous data and effectively discretizing them. Another observation, similar to how Bibin affects the results, is that in the case where we remove the first bin (i.e the bin with the most genes) we obtain both good average and optimal results.

Method	t_m	Metric	Classifier	Topics K	LDA FS	SelectKBest	Simple Classification (“no feature selection”)
Median	\times	KL-Divergence	KNN	3	0.983	0.987	0.996
Median	\times	0.5-Relevance	SVM	3	0.995	0.989	0.987
Median	0.2	0.5-Relevance	KNN	3	0.996	0.988	0.952
Median	0.2	0.3-Relevance	SVM	3	0.990	0.988	0.987

Table 4.20: Classification results (average over all h values) using SVM and KNN for all threshold-defined “Median” transformation method variants in the TCGA dataset. Each row presents the best LDA FS results (across all genes selection/scoring metrics tried). LDA FS accomplishes the best results using SVM as a classifier. When removing expression values below $t_m = 0.2$, LDA FS always achieves the highest classification accuracy.

Method	t_b	Metric	Classifier	Topics K	LDA FS	SelectKBest	Simple Classification (“no feature selection”)
Bibin	\times	0.8-Relevance	SVM	3	0.977	0.990	0.987
Bibin	\times	0.8-Relevance	KNN	3	0.973	0.994	0.996
Bibin	0.2	0.3-Relevance	SVM	3	0.980	0.990	0.987
Bibin	0.2	0.5-Relevance	KNN	3	0.977	0.994	0.952

Table 4.21: Classification results (average over all h values) using SVM and KNN for all threshold-defined “Bibin” transformation method variants in the TCGA dataset. Each row presents the best LDA FS results (across all genes selection/scoring metrics tried). SelectKBest feature selection accomplishes the best results. LDA FS performance is entirely comparable to that of SelectKBest.

⁴The average is computed on the $K_f = 8$ folds.

Method	t_1	Remove Bin 1	t_2	Metric	Classifier	Topics K	LDA FS	SelectKBest	Simple Classification (“no feature selection”)
Repetition	\times	\times	\times	KL-Divergence	SVM	3	0.994	0.992	0.965
Repetition	\times	\times	\times	KL-Divergence	KNN	3	0.980	0.994	0.979
Repetition	\times	\times	0.2	KL-Divergence	SVM	3	0.995	0.992	0.994
Repetition	\times	\times	0.2	KL-Divergence	KNN	3	0.975	0.994	0.954
Repetition	\times	\checkmark	\times	KL-Divergence	SVM	3	0.999	0.993	0.997
Repetition	\times	\checkmark	\times	0.3-Relevance	KNN	3	0.970	0.995	0.992
Repetition	0.2	\times	\times	KL-Divergence	SVM	3	0.994	0.992	0.988
Repetition	0.2	\times	\times	KL-Divergence	KNN	3	0.970	0.995	0.961
Repetition	0.2	\times	0.3	KL-Divergence	SVM	3	0.998	0.992	0.998
Repetition	0.2	\times	0.3	KL-Divergence	KNN	3	0.989	0.995	0.994
Repetition	0.2	\checkmark	\times	KL-Divergence	SVM	3	0.997	0.993	0.954
Repetition	0.2	\checkmark	\times	KL-Divergence	KNN	3	0.966	0.994	0.961

Table 4.22: Classification results (average over all h values) using SVM and KNN for all “Repetition” transformation method variants in the TCGA dataset. Each row presents the best LDA FS results (across all genes selection/scoring metrics tried). When SVM is used for classification, LDA FS is consistently the best method, while when KNN is employed SelectKBest is usually the better method.

Method	t_M	Metric	Classifier	Topics K	h	$k \cdot h$	LDA FS	SelectKBest
Median	\times	0.5-Relevance	SVM	3	135	405	1.000	0.996
Median	\times	0.3-Relevance	KNN	3	235	705	0.996	0.991
Median	0.2	0.5-Relevance	SVM	3	135	405	1.000	0.996
Median	0.2	0.5-Relevance	KNN	3	110	330	1.000	0.987

Table 4.23: Classification accuracy results using SVM for all threshold-defined “Median” transformation method variants in the TCGA dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. LDA FS is consistently the best method.

Method	t_B	Matrix	Classifier	Topics K	h	$k \cdot h$	LDA FS	SelectKBest
Bibin	\times	0.5-Relevance	SVM	3	225	675	0.996	0.991
Bibin	\times	0.3-Relevance	KNN	3	255	765	0.991	0.991
Bibin	0.2	0.3-Relevance	SVM	3	115	345	0.991	0.991
Bibin	0.2	0.5-Relevance	KNN	3	80	240	0.991	0.991

Table 4.24: Classification accuracy results using SVM for all threshold-defined “Bibin” transformation method variants in the TCGA dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. Classification results are almost the same so no conclusion can be made in this case.

Method	t_1	RemoveBin	t_2	Matrix	Classifier	Topics K	h	$k \cdot h$	LDA FS	SelectKBest
Repetition	✗	✗	✗	KL-Divergence	SVM	3	75	223	0.996	0.991
Repetition	✗	✗	✗	KL-Divergence	KNN	3	215	641	0.991	0.996
Repetition	✗	✗	0.2	Topic-Term	SVM	3	70	94	1.000	0.996
Repetition	✗	✗	0.2	KL-Divergence	KNN	3	115	341	0.987	0.996
Repetition	✗	✓	✗	Topic-Term	SVM	3	115	178	1.000	0.991
Repetition	✗	✓	✗	Topic-Term	KNN	3	270	443	0.983	0.996
Repetition	0.2	✗	✗	Topic-Term	SVM	3	230	527	1.000	0.991
Repetition	0.2	✗	✗	Topic-Term	KNN	3	225	512	0.978	0.996
Repetition	0.2	✗	0.3	Topic-Term	SVM	3	95	150	1.000	0.991
Repetition	0.2	✗	0.3	KL-Divergence	KNN	3	25	75	0.991	0.996
Repetition	0.2	✓	✗	Topic-Term	SVM	3	205	315	1.000	0.996
Repetition	0.2	✓	✗	KL-Divergence	KNN	3	255	468	0.983	0.991

Table 4.25: Classification results (average over all h values) using SVM and KNN for all pipelined process-defined “Repetition” transformation method variants in the TCGA dataset, for specific values of h and corresponding metrics in which LDA FS achieved the best accuracy score. SVM classification accuracy scores are higher for feature selection using LDA while KNN accuracies are higher for feature selection using SelectKBest algorithm.

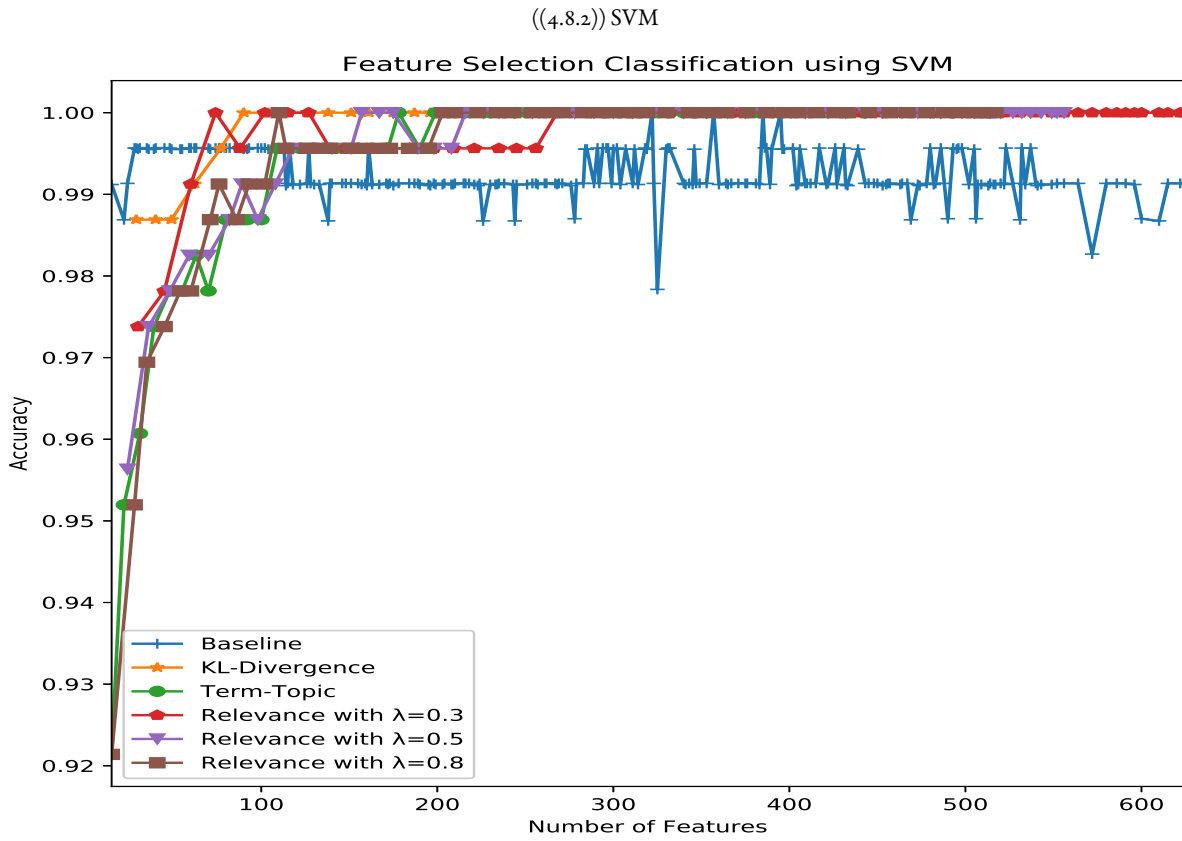
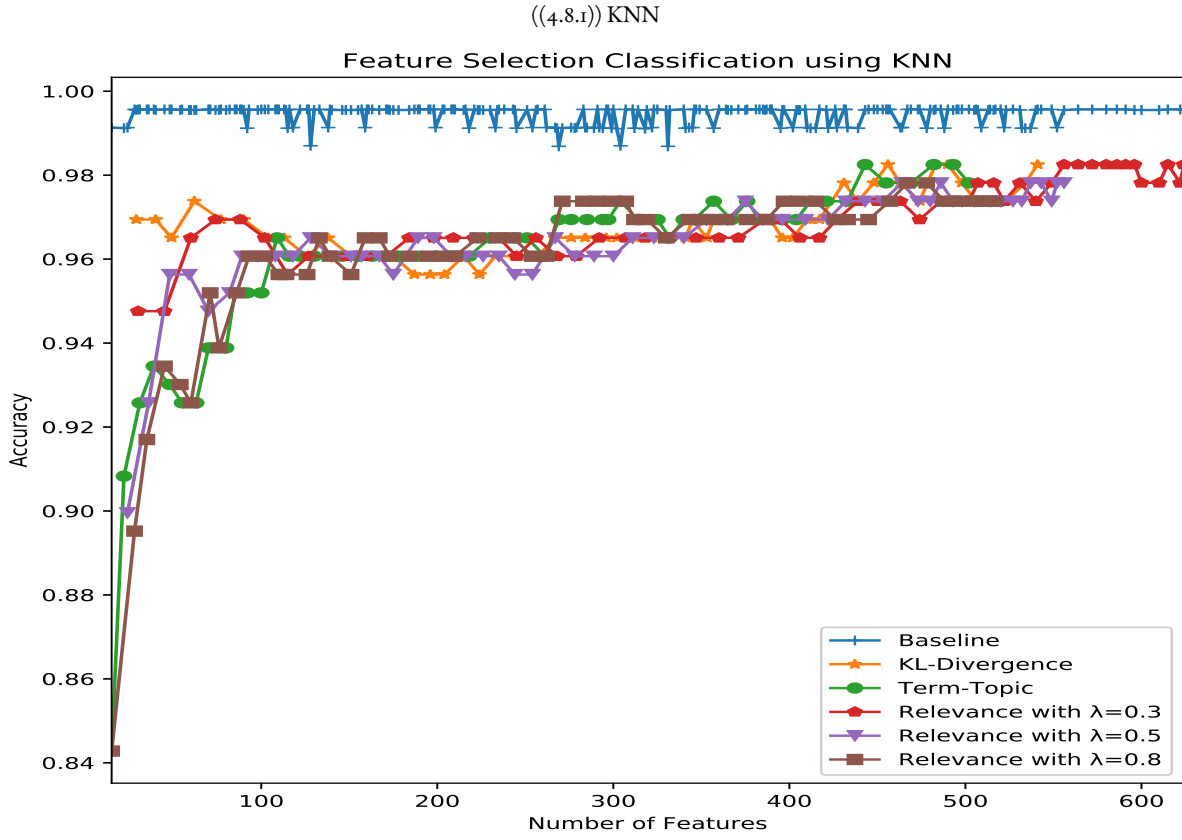


Figure 4.8: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Repetition variant ($t_1 = \text{X}$, RemoveBin= \checkmark , $t_2 = \text{X}$) on the GEM.

Evaluation using Clustering

Following the methodology described above, we evaluate the Unsupervised Feature Selection using LDA approach by employing clustering algorithms. In a similar manner as in the classification task, we compare our proposed method with Univariate Feature Selection. The clustering algorithm employed is the Hierarchical Agglomerative Clustering algorithm. Unsupervised learning is intricate in evaluating the performance and analyzing the clusters, since we do not provide a class to predict. To untie this knot, in each algorithm we utilize clustering evaluation scores that consider the ground truth labels, and provide visualizations of the results for further analysis. The metrics we employed are:

- Normalized Mutual Information (NMI):

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{H(Y) + H(C)}$$

Where Y are the class labels, C are the cluster labels, $H(\cdot)$ is the entropy, and $I(Y; C)$ the mutual information. More precisely, the Entropy of a random variable X is defined as :

$$H(X) = - \sum_{i=1}^n P(x_i) \log P(x_i)$$

and Mutual Information between discrete random variable X, Y is derived from:

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

- Adjusted Rand Score: If C is a ground truth class assignment and K the clustering, let us define and as:
 - a , the number of pairs of elements that are in the same set in C and in the same set in K .
 - b , the number of pairs of elements that are in different sets in C and in different sets in K .

The raw (unadjusted) Rand index is then given by:

$$RI = \frac{a + b}{C_2^{n_{samples}}} \quad (4.1)$$

Where is the total number of possible pairs in the dataset (without ordering).

$$C_2^{n_{samples}}$$

However the RI score does not guarantee that random label assignments will get a value close to zero (esp. if the number of clusters is in the same order of magnitude as the number of samples). To counter this effect we can discount the expected RI of random labellings by defining the adjusted Rand index as follows:

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \quad (4.2)$$

For the Muscle Disease Dataset, we set the number of clusters to $n = [7, 14]$. In Tables 4.26, 4.27, 4.28 we present the best results obtained by our approach for the Median, Bibin and Repetition Method correspondingly. We exhibit for each variant and for each of the metrics, the clustering performance scores for Unsupervised Feature Selection and the Baseline Algorithm. In this part, we observe that in clustering the data, the

scores do not change whether we use any type of the different metrics. See for example in Table 4.26, that given $n_{clusters} = 11$ and $K * h = 1500$, we obtain the same NMI, Rand. In the classification evaluation this occurred after a certain point (see Figure 4.8)

Method	t_M	Metric	n-clusters	K	h	$K * h$	LDA FS-NMI	SelectKbest-NMI	LDA FSFS-Rand	SelectKbest-Rand
Median	o	Topic-Term	11	11	100	1100	0.775	0.769	0.614	0.595
Median	o	o.8-Relevance	11	11	100	1100	0.775	0.769	0.614	0.595
Median	o	KL-Divergence	7	11	100	1100	0.773	0.804	0.616	0.678
Median	o	o.5-Relevance	11	11	150	1650	0.777	0.792	0.611	0.678
Median	o	o.3-Relevance	11	11	200	2200	0.777	0.772	0.611	0.580
Median	0.2	Topic-Term	7	25	50	1250	0.772	0.804	0.617	0.678
Median	0.2	o.5-Relevance	7	25	50	1250	0.768	0.804	0.630	0.678
Median	0.2	o.8-Relevance	7	25	50	1250	0.772	0.804	0.617	0.678
Median	0.2	KL-Divergence	8	25	100	2493	0.771	0.837	0.621	0.739
Median	0.2	o.3-Relevance	9	25	150	3750	0.780	0.800	0.607	0.632
Median	X	Topic-Term	11	10	150	1500	0.758	0.773	0.590	0.659
Median	X	o.3-Relevance	11	10	150	1500	0.758	0.773	0.590	0.659
Median	X	o.5-Relevance	11	10	150	1500	0.758	0.773	0.590	0.659
Median	X	o.8-Relevance	11	10	150	1500	0.758	0.773	0.590	0.659
Median	X	KL-Divergence	11	10	150	1500	0.758	0.773	0.590	0.659

Table 4.26: Clustering scores using HAC for all threshold-defined “Median” transformation method variants in the MD dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_M and for each metric. SelectKBest achieves the highest NMI and RAND scores.

Method	t_B	Metric	n-clusters	K	h	$k * h$	LDA FS-NMI	SelectKbest-NMI	LDA FS-Rand	SelectKbest-Rand
Bibin	X	o.3-Relevance	7	12	50	600	0.816	0.780	0.651	0.653
Bibin	X	o.5-Relevance	7	12	50	600	0.817	0.780	0.653	0.653
Bibin	X	o.8-Relevance	7	12	50	600	0.817	0.780	0.653	0.653
Bibin	X	Topic-Term	7	12	50	600	0.817	0.780	0.653	0.653
Bibin	o	o.3-Relevance	8	17	100	1700	0.789	0.792	0.646	0.698
Bibin	o	o.5-Relevance	8	17	100	1699	0.789	0.792	0.646	0.698
Bibin	o	o.8-Relevance	7	17	50	850	0.789	0.804	0.652	0.678
Bibin	o	KL-Divergence	9	17	50	849	0.790	0.789	0.649	0.621
Bibin	o	Topic-Term	8	17	150	2523	0.788	0.781	0.634	0.669
Bibin	0.2	o.3-Relevance	7	19	50	950	0.799	0.801	0.629	0.701
Bibin	0.2	o.5-Relevance	7	19	100	1890	0.799	0.788	0.629	0.683
Bibin	0.2	o.8-Relevance	12	19	100	1867	0.783	0.759	0.620	0.642
Bibin	0.2	KL-Divergence	7	19	200	3479	0.807	0.826	0.664	0.729
Bibin	0.2	Topic-Term	11	19	200	3365	0.775	0.783	0.614	0.625

Table 4.27: Clustering scores using HAC for all threshold-defined “Bibin” transformation method variants in the MD dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_B and for each metric. SelectKBest achieves the highest RAND scores. LDA FS achieves higher clustering scores where no data was removed.

Method	t_1	Remove Bin 1	t_2	Metric	n-clusters	K	h	k*h	LDA FS-NMI	SelectKbest-NMI	LDA FS-Rand	SelectKbest-rand
Repetition	X	X	X	Topic-Term	8	14	150	2061	0.781	0.802	0.655	0.678
Repetition	X	X	X	0.3-Relevance	8	14	100	1400	0.767	0.808	0.611	0.714
Repetition	X	X	X	0.5-Relevance	7	14	200	2797	0.781	0.789	0.637	0.658
Repetition	X	X	X	0.8-Relevance	8	14	200	2765	0.792	0.788	0.660	0.658
Repetition	X	X	X	KL-Divergence	8	14	150	2096	0.789	0.802	0.641	0.678
Repetition	X	X	0.2	Topic-Term	9	12	100	1168	0.786	0.811	0.609	0.717
Repetition	X	X	0.2	0.3-Relevance	7	12	100	1200	0.793	0.792	0.649	0.667
Repetition	X	X	0.2	0.5-Relevance	10	12	100	1200	0.787	0.776	0.609	0.644
Repetition	X	X	0.2	0.8-Relevance	7	12	200	2350	0.784	0.789	0.649	0.658
Repetition	X	X	0.2	KL-Divergence	7	12	200	2348	0.793	0.789	0.636	0.658
Repetition	X	✓	X	Topic-Term	7	16	100	1598	0.774	0.802	0.646	0.693
Repetition	X	✓	X	0.3-Relevance	9	16	200	3200	0.783	0.812	0.617	0.640
Repetition	X	✓	X	0.5-Relevance	7	16	100	1600	0.774	0.794	0.646	0.686
Repetition	X	✓	X	0.8-Relevance	7	16	200	3184	0.774	0.803	0.646	0.640
Repetition	X	✓	X	KL-Divergence	7	16	100	1600	0.782	0.794	0.639	0.686
Repetition	X	✓	o	Topic-Term	7	7	50	345	0.774	0.802	0.646	0.669
Repetition	X	✓	o	0.3-Relevance	7	7	150	1050	0.781	0.796	0.642	0.692
Repetition	X	✓	o	0.5-Relevance	7	7	50	350	0.796	0.802	0.660	0.669
Repetition	X	✓	o	0.8-Relevance	7	7	50	350	0.773	0.802	0.627	0.669
Repetition	X	✓	o	KL-Divergence	8	7	150	1050	0.760	0.802	0.627	0.702
Repetition	o	X	X	Topic-Term	8	12	200	2223	0.767	0.794	0.598	0.687
Repetition	o	X	X	0.3-Relevance	7	12	100	1200	0.793	0.804	0.649	0.678
Repetition	o	X	X	0.5-Relevance	7	12	100	1199	0.781	0.804	0.637	0.678
Repetition	o	X	X	0.8-Relevance	7	12	200	2309	0.781	0.788	0.637	0.684
Repetition	o	X	X	KL-Divergence	8	12	100	1200	0.789	0.802	0.646	0.678
Repetition	o	X	0.2	Topic-Term	8	16	200	2798	0.802	0.801	0.662	0.705
Repetition	o	X	0.2	0.3-Relevance	7	16	200	3195	0.807	0.782	0.664	0.601
Repetition	o	X	0.2	0.5-Relevance	7	16	100	1585	0.795	0.804	0.652	0.678
Repetition	o	X	0.2	0.8-Relevance	7	16	50	772	0.795	0.825	0.652	0.720
Repetition	o	X	0.2	KL-Divergence	7	16	150	2367	0.807	0.811	0.664	0.713
Repetition	o	✓	X	Topic-Term	7	15	150	2045	0.774	0.805	0.633	0.681
Repetition	o	✓	X	0.3-Relevance	8	15	100	1500	0.790	0.808	0.648	0.714
Repetition	o	✓	X	0.5-Relevance	8	15	50	746	0.783	0.808	0.615	0.714
Repetition	o	✓	X	0.8-Relevance	7	15	150	2137	0.795	0.805	0.652	0.681
Repetition	o	✓	X	KL-Divergence	7	15	150	2196	0.797	0.805	0.641	0.681
Repetition	0.2	X	X	Topic-Term	8	15	150	1979	0.802	0.799	0.662	0.692
Repetition	0.2	X	X	0.3-Relevance	7	15	150	2250	0.807	0.798	0.664	0.689
Repetition	0.2	X	X	0.5-Relevance	8	15	100	1488	0.802	0.802	0.662	0.678
Repetition	0.2	X	X	0.8-Relevance	8	15	150	2094	0.802	0.807	0.662	0.700
Repetition	0.2	X	X	KL-Divergence	7	15	200	2906	0.807	0.802	0.664	0.702
Repetition	0.2	X	0.3	Topic-Term	7	18	100	1578	0.795	0.808	0.652	0.660
Repetition	0.2	X	0.3	0.3-Relevance	7	18	150	2673	0.815	0.822	0.669	0.729
Repetition	0.2	X	0.3	0.5-Relevance	7	18	200	3420	0.807	0.826	0.664	0.729
Repetition	0.2	X	0.3	0.8-Relevance	7	18	50	847	0.795	0.804	0.652	0.678
Repetition	0.2	X	0.3	KL-Divergence	7	18	50	892	0.807	0.804	0.664	0.678
Repetition	0.2	✓	X	Topic-Term	7	18	200	3022	0.795	0.798	0.652	0.689
Repetition	0.2	✓	X	0.3-Relevance	7	18	150	2673	0.807	0.802	0.664	0.702
Repetition	0.2	✓	X	0.5-Relevance	7	18	50	882	0.807	0.805	0.664	0.705
Repetition	0.2	✓	X	0.8-Relevance	7	18	150	2446	0.807	0.788	0.664	0.684
Repetition	0.2	✓	X	KL-Divergence	8	18	50	880	0.809	0.817	0.668	0.720

Table 4.28: Clustering scores using HAC for all pipelined process-defined “Repetition” transformation method variants in the MD dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_B and for each metric. SelectKBest achieves higher scores in most variants.

In General, on the Muscle disease datasets, the clustering evaluation scores depict a comparable performance to the baseline algorithm. However we would like to focus on observing the visualizations of the clustering algorithm. We visualize HCA Dendrograms which are labeled according to the disease and the metadata for each sample. In Figure 4.9 we show a dendrogram produced by the Median Method, in Figure 4.10 produced by the Bibin Method and in Figure 4.10 by the Repetition Method. All methods manage to create hierarchical clusters that are very close to how the data is labeled. In Figure 4.9, samples labeled as “normal” are in general in the same clusters, while “DM1” and “DM2” are in the same clusters also.

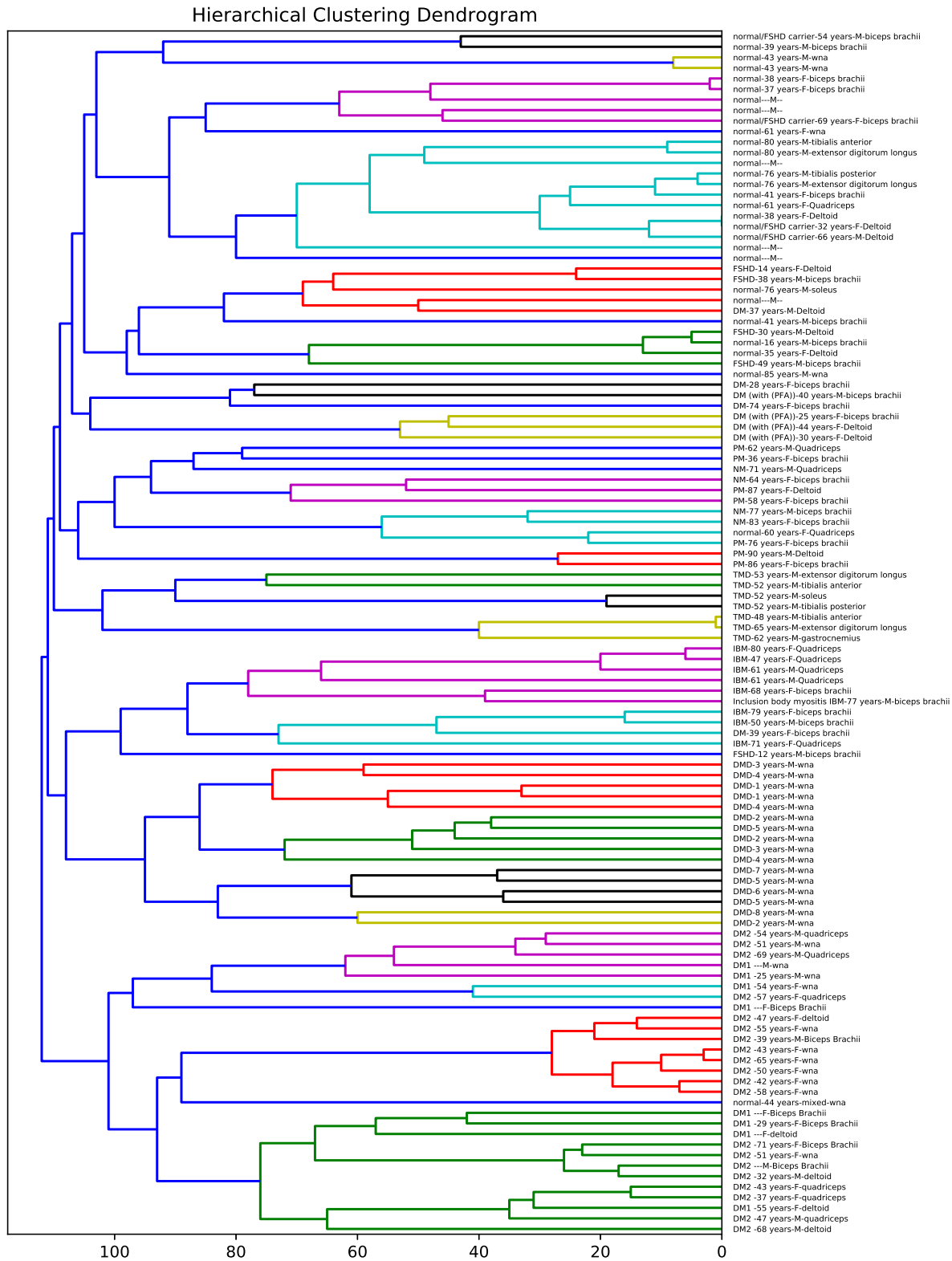


Figure 4.9: HAC Dendrogram for UFS on Median variant of MD Dataset and Relevance Score with $\lambda = 0.3$ and number of clusters 7.

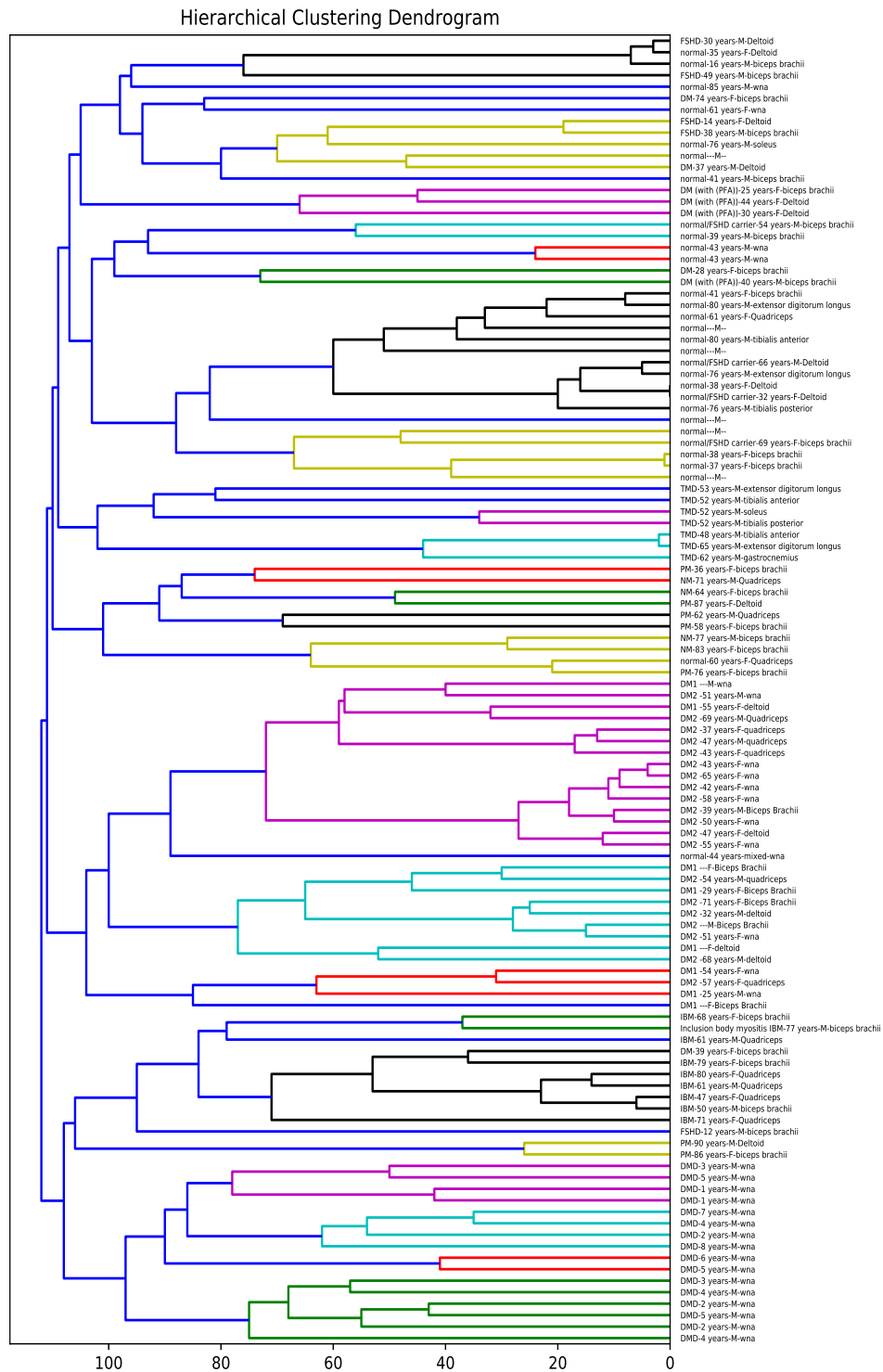


Figure 4.10: HAC Dendrogram for UFS on Bibin variant of MD Dataset and Relevance Score with $\lambda = 0.3$ and number of clusters 9.

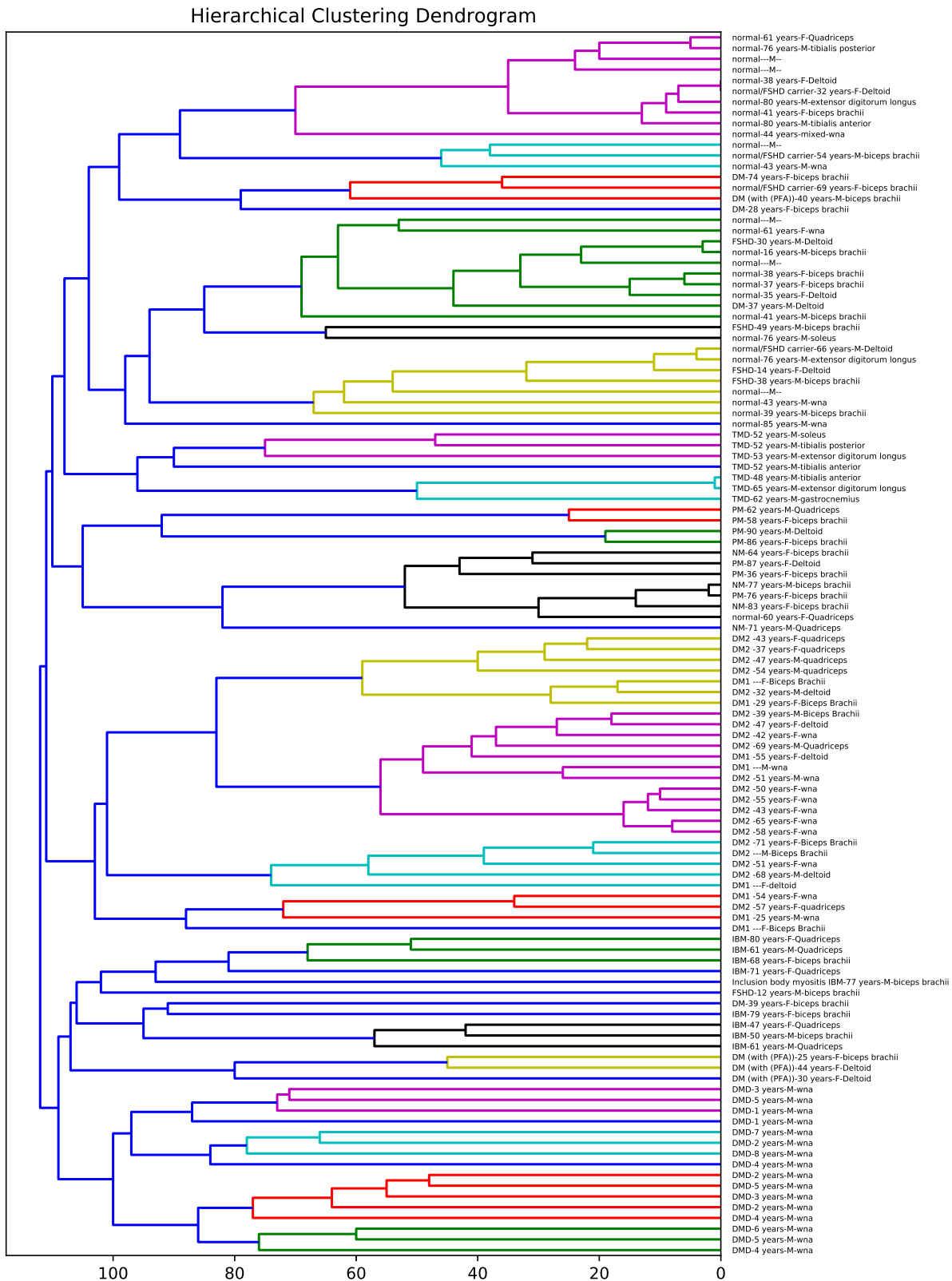


Figure 4.ii: HAC Dendrogram for UFS on Repetition variant of MD Dataset on the Topic-Term matrix and number of clusters 9.

In Figure 4.12 we observe how “DM1”, “DM2” are distinguished (obtained from Figure 4.10).

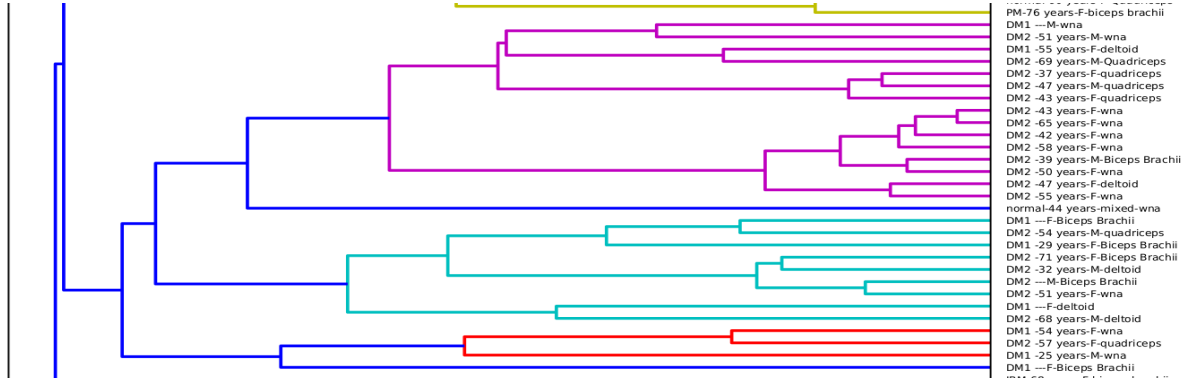


Figure 4.12: HAC part of the Dendrogram for UFS on Repetition variant of MD Dataset on the Topic-Term matrix and number of clusters 9.

Regarding the TCGA data we also followed the same procedure.

Model	Method	t_M	Metric	n-clusters	topics	h	k*h	LDA FS-NMI	SelectKbest-NMI	LDA FS-RAND	SelectKbest-RAND
HCA	Median	\times	Topic-Term	4	3	150	450	0.527	0.587	0.458	0.734
HCA	Median	\times	0.3 Relevance	4	3	150	450	0.527	0.587	0.458	0.734
HCA	Median	\times	0.5 Relevance	4	3	150	450	0.527	0.587	0.458	0.734
HCA	Median	\times	0.8 Relevance	4	3	150	450	0.527	0.587	0.458	0.734
HCA	Median	\times	KL-Divergence	3	3	50	150	0.603	0.748	0.622	0.734
HCA	Median	0.2	Topic-Term	3	3	200	600	0.947	0.726	0.974	0.753
HCA	Median	0.2	0.3 Relevance	2	3	150	450	0.851	0.770	0.897	0.848
HCA	Median	0.2	0.5 Relevance	2	3	150	450	0.900	0.770	0.948	0.848
HCA	Median	0.2	0.8 Relevance	3	3	200	600	0.947	0.726	0.974	0.753
HCA	Median	0.2	KL-Divergence	4	3	200	584	0.369	0.726	0.354	0.753

Table 4.29: Clustering scores using HAC for all threshold-defined “Median” transformation method variants in the TCGA dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_M and for each metric. SelectKBest achieves the highest NMI and RAND scores.

Model	Method	t_B	Metric	n-clusters	Topics K	h	k*h	LDA FS-NMI	SelectKbest-NMI	LDA FS-RAND	SelectKbest-RAND
HCA	Bibin	0.2	Topic-Term	4	3	100	300	0.511	0.643	0.406	0.662
HCA	Bibin	0.2	0.3 Relevance	4	3	50	150	0.528	0.565	0.466	0.516
HCA	Bibin	0.2	0.5 Relevance	3	3	200	600	0.593	0.768	0.702	0.783
HCA	Bibin	0.2	0.8 Relevance	4	3	100	300	0.520	0.643	0.468	0.662
HCA	Bibin	0.2	KL-Divergence	3	3	100	300	0.573	0.666	0.638	0.681
HCA	Bibin	\times	Topic-Term	3	3	200	600	0.300	0.768	0.192	0.783
HCA	Bibin	\times	0.3 Relevance	4	3	200	600	0.371	0.663	0.344	0.541
HCA	Bibin	\times	0.5 Relevance	4	3	200	600	0.379	0.663	0.325	0.541
HCA	Bibin	\times	0.8 Relevance	3	3	200	600	0.300	0.768	0.192	0.783
HCA	Bibin	\times	KL-Divergence	4	3	150	450	0.319	0.626	0.225	0.595

Table 4.30: Clustering scores using HAC for all threshold-defined “Bibin” transformation method variants in the TCGA dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_B and for each metric. SelectKBest achieves the highest NMI and RAND scores.

Model	Method	t_1	Remove Bin 1	t_2	Matrix	n-clusters	Topics k	h	$k \cdot h$	LDA FS-NMI	SelectKbest-NMI	LDA FS-RAND	SelectKbest-RAND
HCA	Repetition	0.2	✗	✗	Topic-Term	3	3	50	66	0.666	0.820	0.427	0.813
HCA	Repetition	0.2	✗	✗	0.8-Relevance	3	3	50	69	0.666	0.820	0.427	0.813
HCA	Repetition	0.2	✗	✗	0.3-Relevance	4	3	150	403	0.103	0.643	0.800	0.662
HCA	Repetition	0.2	✗	✗	0.5-Relevance	4	3	100	227	0.808	0.643	0.411	0.662
HCA	Repetition	0.2	✗	✗	KL-Divergence	4	3	150	446	0.890	0.643	0.948	0.662
HCA	Repetition	0.2	✗	0.3	Topic-Term	4	3	50	62	0.808	0.728	0.411	0.673
HCA	Repetition	0.2	✗	0.3	0.3-Relevance	2	3	50	112	0.552	0.964	-0.003	0.983
HCA	Repetition	0.2	✗	0.3	0.5-Relevance	2	3	50	82	0.552	0.964	-0.003	0.983
HCA	Repetition	0.2	✗	0.3	0.8-Relevance	4	3	50	69	0.808	0.728	0.411	0.673
HCA	Repetition	0.2	✗	0.3	KL-Divergence	2	3	50	150	0.552	0.964	-0.003	0.983
HCA	Repetition	0.2	✓	✗	Topic-Term	4	3	50	72	0.808	0.728	0.411	0.673
HCA	Repetition	0.2	✓	✗	0.3-Relevance	2	3	50	130	0.552	0.964	-0.003	0.983
HCA	Repetition	0.2	✓	✗	0.5-Relevance	2	3	50	106	0.552	0.964	-0.003	0.983
HCA	Repetition	0.2	✓	✗	0.8-Relevance	2	3	50	81	0.552	0.964	-0.003	0.983
HCA	Repetition	0.2	✓	✗	KL-Divergence	2	3	50	128	0.552	0.964	-0.003	0.983
HCA	Repetition	✗	✗	✗	Topic-Term	4	3	50	61	0.808	0.735	0.411	0.691
HCA	Repetition	✗	✗	✗	0.3-Relevance	3	3	50	115	0.666	0.820	0.427	0.813
HCA	Repetition	✗	✗	✗	0.5-Relevance	4	3	150	372	0.720	0.528	0.374	0.527
HCA	Repetition	✗	✗	✗	0.8-Relevance	4	3	50	65	0.808	0.735	0.411	0.691
HCA	Repetition	✗	✗	✗	KL-Divergence	4	3	100	298	0.612	0.668	0.850	0.643
HCA	Repetition	✗	✗	0.2	Topic-Term	3	3	50	62	0.703	0.820	0.465	0.813
HCA	Repetition	✗	✗	0.2	0.3-Relevance	4	3	200	558	0.720	0.627	0.374	0.597
HCA	Repetition	✗	✗	0.2	0.5-Relevance	3	3	50	88	0.703	0.820	0.465	0.813
HCA	Repetition	✗	✗	0.2	0.8-Relevance	4	3	50	65	0.720	0.634	0.374	0.598
HCA	Repetition	✗	✗	0.2	KL-Divergence	2	3	50	148	0.798	0.964	0.979	0.983
HCA	Repetition	✗	✓	✗	Topic-Term	2	3	50	80	0.552	0.753	-0.003	0.801
HCA	Repetition	✗	✓	✗	0.3-Relevance	2	3	50	139	0.552	0.753	-0.003	0.801
HCA	Repetition	✗	✓	✗	0.5-Relevance	2	3	50	108	0.552	0.753	-0.003	0.801
HCA	Repetition	✗	✓	✗	0.8-Relevance	2	3	50	86	0.552	0.753	-0.003	0.801
HCA	Repetition	✗	✓	✗	KL-Divergence	2	3	50	126	0.552	0.753	-0.003	0.801

Table 4.31: Clustering scores using HAC for all pipelined process-defined “Repetition” transformation method variants in the TCGA dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_B and for each metric-ranking. SelectKBest achieves higher scores in most of the variants.

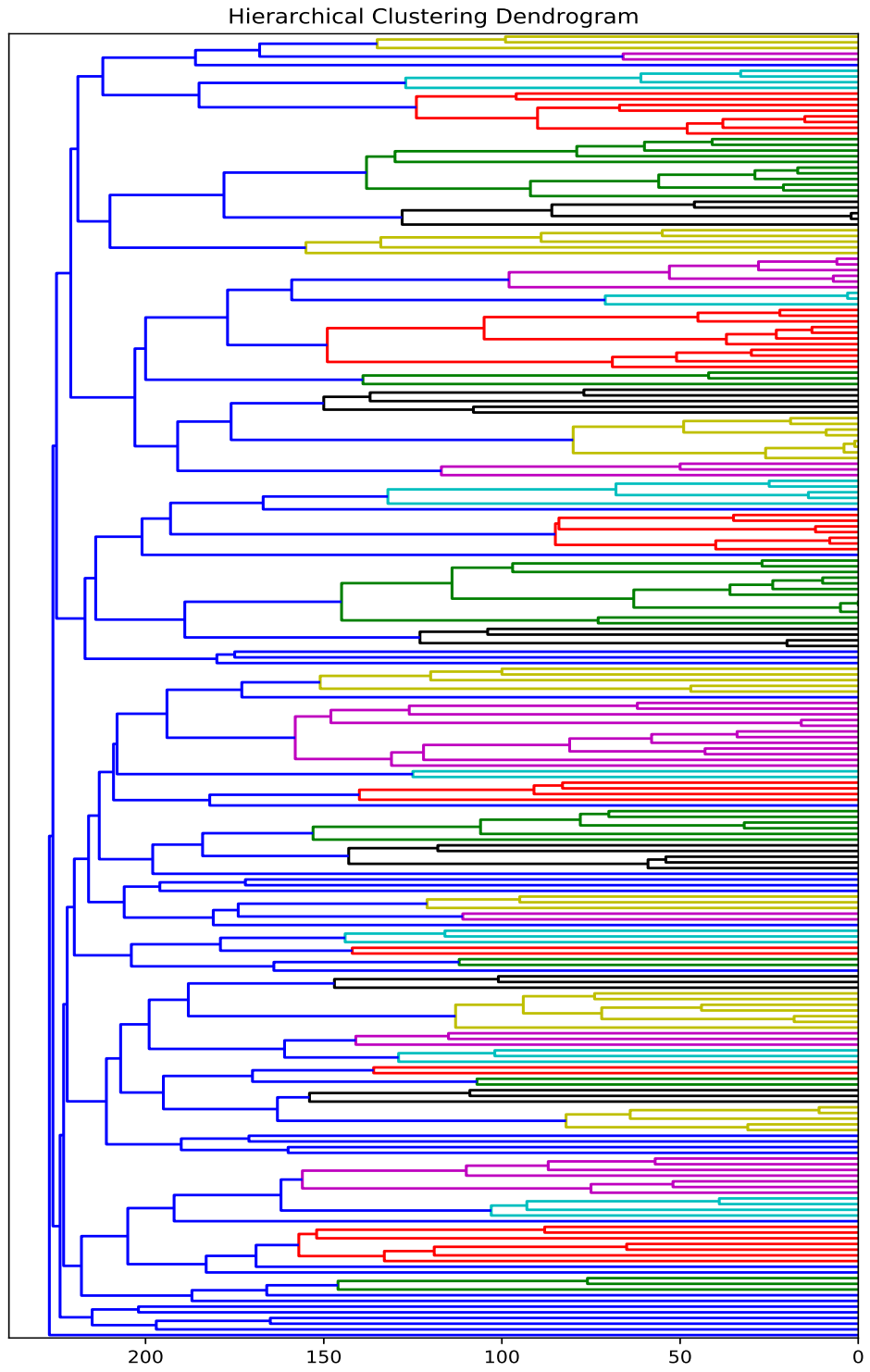


Figure 4.13: Dendrogram of HAC using Unsupervised Feature Selection with LDA on the TCGA with Median Variant with $(t_M = 0.2, h = 200, n_c = 3)$ on the Topic-Term Matrix.

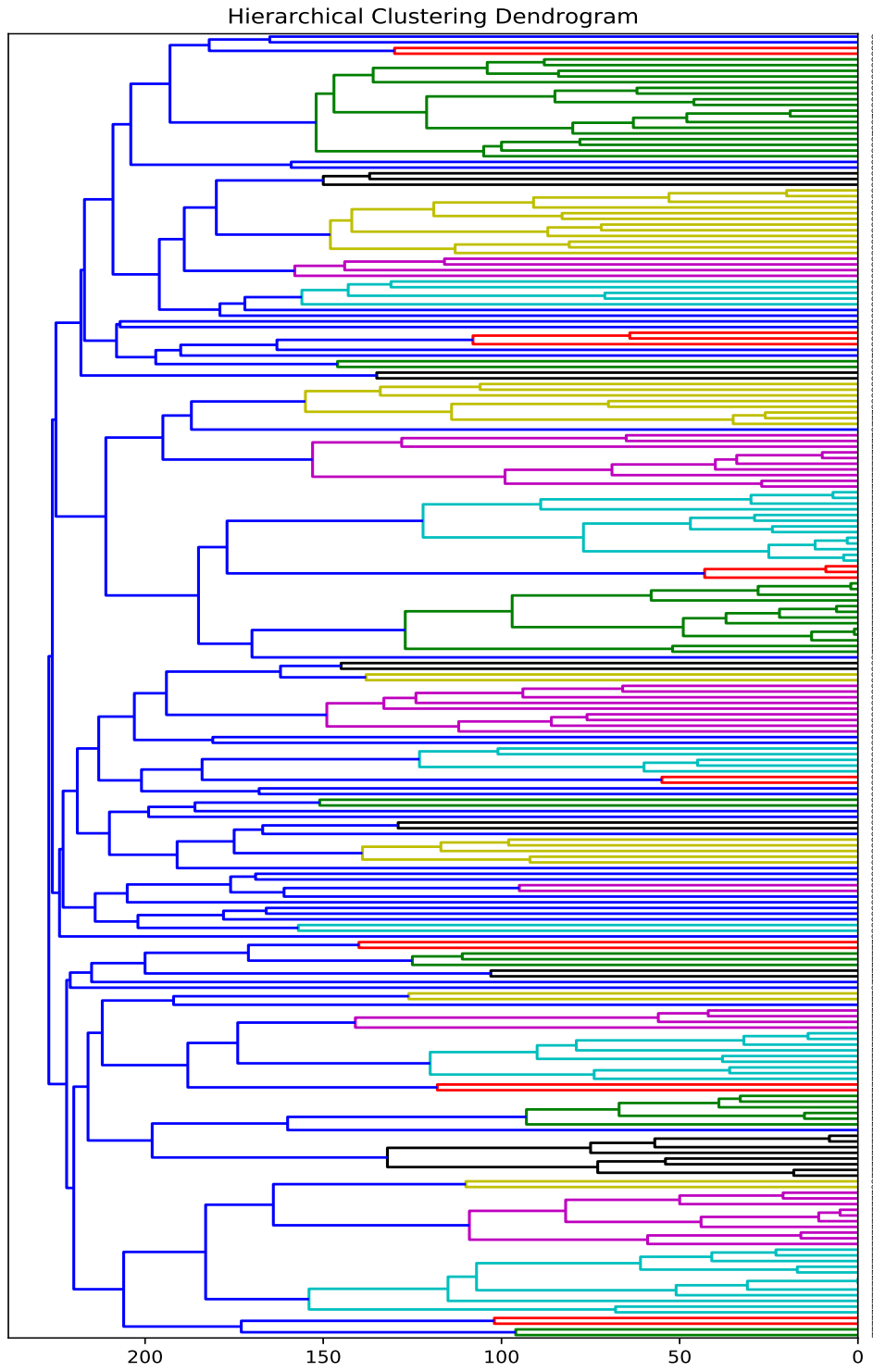


Figure 4.14: Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the TCGA with Bibin Variant with $(t_B = 0.2, b = 200, n_c = 3)$ on the Topic-Term Matrix.

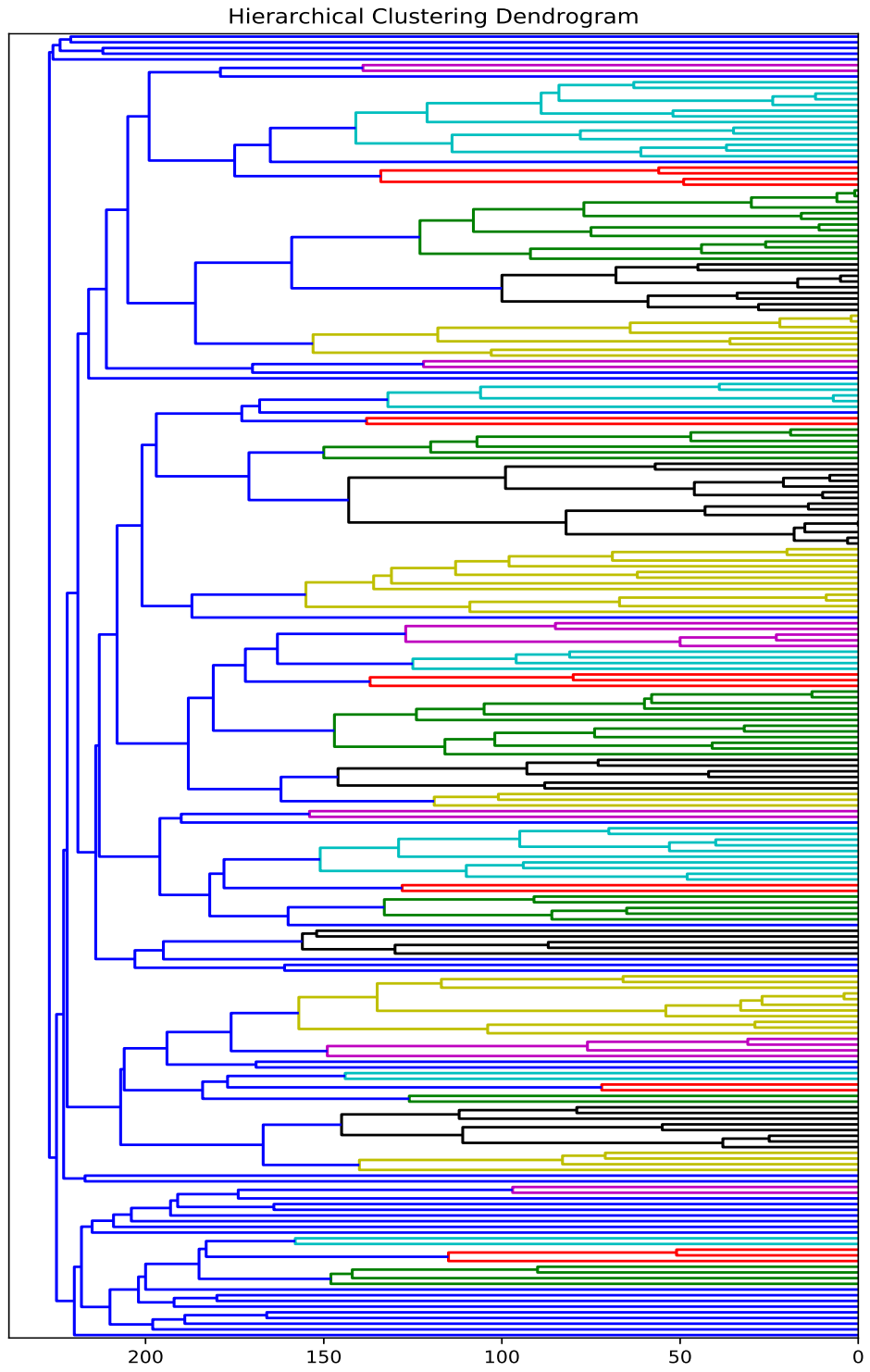


Figure 4.15: Dendrogram of HAC using Unsupervised Feature Selection with LDA on the TCGA with Repetition Variant with $(t_1=\mathbf{X}, \text{Remove Bin } \mathbf{I} = \mathbf{X}t_2 = \mathbf{X}, b = 50, n_c = 4)$ on the Relevance Matrix with $\lambda = 0.8$.

4.2.2.2 Unsupervised Feature Selection Using LPD

As a topic model, Latent Process Decomposition also produces the distributions of topics over the genes. Our intention is to exploit these distributions in the same manner as described previously in order to perform feature selection.

Due to the nature of this topic model, we used the topic-gene distributions and the KL-Divergence score, in contrast with the methodology employed in the LDA approach. That is mainly because we cannot compute the Relevance score of the genes, since its intuition cannot be captured in the continuous variable setting. The problem of feature selection is solved and evaluated in a similar manner to performing Unsupervised Feature Selection using LDA. We conducted for each dataset using the corresponding number of topics K , and exploited the equivalent topic term distribution and KL-Divergence metrics. Again the evaluation of our approach was performed by using supervised and unsupervised learning.

The procedure followed is similar; we select for each topic $k \in K$ the h most significant genes, which are defined by the equivalent score. That is the h most probable genes in each topic in case of the distribution of topics over the genes, and the h differentially expressed genes in each topic according to the Kullback-Leibler Divergence metric. After having selected the subset of the initial set of genes, we then employ supervised and unsupervised learning, to evaluate our approach. The comparison of our approach is again performed against Univariate Feature Selection Algorithms, and specifically SelectKBest from [Pedregosa et al., 2011] with score the ANOVA F-Value. At this point we will present the results of each dataset and for each evaluation algorithm.

Evaluation using Classification

We performed classification on the Muscle Disease Dataset using SVM and kNearest Neighbors algorithms. The value of variable h is assigned from the range [10, 300]. We use the same metrics as in the previous approach. As seen in Tables 4.32, 4.33, 4.34, on average of the values of variable h , our approach on the MD Dataset outperforms the baseline algorithm. In what concerns the particular values of h , for the different assignments of t_{LPD} and the different number of topics K , our approach has reached better accuracy scores. Moreover, we observed that the KL-Divergence Matrix, achieves better results than the Topic-Term matrix. It can also be seen in Figure 4.16 that overall our approach performs better regardless of the classifier. In this Figure we visualize the accuracies with respect to the values of h , for $t_{LPD} = \mathbf{X}$ and $K = 5$.

Classifier	t_{LPD}	Metric	Topics K	LPD FS	SelectKbest
SVM	o	KL-Divergence	5	0.868	0.846
SVM	o	Topic-Term	5	0.855	0.845
KNN	o	KL-Divergence	5	0.822	0.783
KNN	o	Topic-Term	5	0.788	0.775
SVM	o	KL-Divergence	10	0.860	0.843
SVM	o	Topic-Term	10	0.864	0.850
KNN	o	KL-Divergence	10	0.825	0.789
KNN	o	Topic-Term	10	0.796	0.783
SVM	o	KL-Divergence	15	0.862	0.839
SVM	o	Topic-Term	15	0.868	0.839
KNN	o	KL-Divergence	15	0.806	0.783
KNN	o	Topic-Term	15	0.798	0.784

Table 4.32: Classification accuracy results (average over all h values) with SVM and KNN using LPD for feature selection (LPD FS) on the MD Dataset. Each row presents the classification accuracies for each metric and number of topics while having removed expression values below $t_{LPD} = 0$. LPD FS outperforms SelectKBest for each combination of classification-metric-number of topics.

Classifier	t_{LPD}	Metric	K	LPD FS	SelectKbest
SVM	0.2	KL-Divergence	5	0.882	0.845
SVM	0.2	Topic-Term	5	0.864	0.841
KNN	0.2	KL-Divergence	5	0.822	0.781
KNN	0.2	Topic-Term	5	0.800	0.777
SVM	0.2	KL-Divergence	10	0.876	0.842
SVM	0.2	Topic-Term	10	0.859	0.849
KNN	0.2	KL-Divergence	10	0.808	0.787
KNN	0.2	Topic-Term	10	0.793	0.787
SVM	0.2	KL-Divergence	15	0.882	0.841
SVM	0.2	Topic-Term	15	0.865	0.843
KNN	0.2	KL-Divergence	15	0.812	0.787
KNN	0.2	Topic-Term	15	0.797	0.784

Table 4.33: Classification accuracy results (average over all b values) with SVM and KNN using LPD for feature selection (LPD FS) on the MD Dataset. Each row presents the classification accuracies for each metric and number of topics while having removed expression values below $t_{LPD} = 0.2$. LPD FS outperforms SelectKBest for each combination of classification-metric-number of topics.

Classifier	t_{LPD}	Metric	K	LPD FS	SelectKbest
SVM	\times	KL-Divergence	5	0.868	0.837
SVM	\times	Topic-Term	5	0.860	0.839
KNN	\times	KL-Divergence	5	0.818	0.783
KNN	\times	Topic-Term	5	0.798	0.777
SVM	\times	KL-Divergence	10	0.881	0.835
SVM	\times	Topic-Term	10	0.862	0.836
KNN	\times	KL-Divergence	10	0.817	0.787
KNN	\times	Topic-Term	10	0.796	0.780
SVM	\times	KL-Divergence	15	0.883	0.837
SVM	\times	Topic-Term	15	0.871	0.831
KNN	\times	KL-Divergence	15	0.813	0.785
KNN	\times	Topic-Term	15	0.799	0.780

Table 4.34: Classification accuracy results (average over all h values) with SVM and KNN using LPD for feature selection (LPD FS) on the MD Dataset. Each row presents the classification accuracies for each metric and number of topics while not having removed any data ($t_{LPD} = \times$). LPD FS outperforms SelectKBest for each combination of classification-metric-number of topics.

Classifier	t_{LPD}	Metric	K	h	$k * h$	LPD FS	SelectKBest
KNN	\times	KL-Divergence	5	70	303	0.833	0.793
SVM	\times	KL-Divergence	5	225	891	0.886	0.818
KNN	\times	Topic-Term	5	60	163	0.816	0.774
SVM	\times	Topic-Term	5	10	31	0.807	0.789
KNN	\times	KL-Divergence	10	10	90	0.833	0.752
SVM	\times	KL-Divergence	10	10	90	0.895	0.807
KNN	\times	Topic-Term	10	110	451	0.807	0.785
SVM	\times	Topic-Term	10	255	942	0.886	0.846
KNN	\times	KL-Divergence	15	170	1785	0.825	0.787
SVM	\times	KL-Divergence	15	50	635	0.851	0.873
KNN	\times	Topic-Term	15	240	1323	0.833	0.818
SVM	\times	Topic-Term	15	10	83	0.833	0.764

Table 4.35: Classification results with SVM and KNN using LPD for feature selection (LPD FS) on the MD dataset, for specific values of h and for each metric in which LDA FS achieved the best accuracy score. This table depicts classification accuracies for certain values of h in which LPD FS achieved the best accuracy score for each metric and number of topics, while $t_{LPD} = \times$. LPD FS outperforms SelectKBest for almost every combination of classifier-metric-number of topics.

Classifier	t_{LPD}	Metric	K	h	$k * h$	LPD FS	SelectKBest
KNN	0	KL-Divergence	5	145	636	0.833	0.804
SVM	0	KL-Divergence	5	290	1220	0.895	0.835
KNN	0	Topic-Term	5	290	732	0.816	0.785
SVM	0	Topic-Term	5	190	502	0.868	0.838
KNN	0	KL-Divergence	10	10	99	0.842	0.729
SVM	0	KL-Divergence	10	55	514	0.886	0.842
KNN	0	Topic-Term	10	225	1067	0.816	0.783
SVM	0	Topic-Term	10	75	412	0.877	0.844
KNN	0	KL-Divergence	15	10	146	0.816	0.775
SVM	0	KL-Divergence	15	155	1927	0.877	0.871
KNN	0	Topic-Term	15	260	1702	0.816	0.772
SVM	0	Topic-Term	15	285	1840	0.886	0.847

Table 4.36: Classification results with SVM and KNN using LPD for feature selection (LPD FS) on the MD dataset. This table depicts classification accuracies for certain values of h in which LPD FS achieved the best accuracy score for each metric and number of topics, while $t_{LPD} = 0$. LPD FS outperforms SelectKBest for each combination of classification-metric-number of topics.

Classifier	t_{LPD}	Metric	K	h	$k * h$	LPD FS	SelectKBest
KNN	0.2	KL-Divergence	5	90	407	0.833	0.791
SVM	0.2	KL-Divergence	5	20	97	0.895	0.822
KNN	0.2	Topic-Term	5	75	233	0.816	0.774
SVM	0.2	Topic-Term	5	75	233	0.877	0.866
KNN	0.2	KL-Divergence	10	20	182	0.825	0.778
SVM	0.2	KL-Divergence	10	30	266	0.886	0.855
KNN	0.2	Topic-Term	10	270	1039	0.816	0.804
SVM	0.2	Topic-Term	10	290	1116	0.877	0.831
KNN	0.2	KL-Divergence	15	25	334	0.842	0.781
SVM	0.2	KL-Divergence	15	10	138	0.904	0.801
KNN	0.2	Topic-Term	15	15	118	0.825	0.770
SVM	0.2	Topic-Term	15	120	710	0.877	0.880

Table 4.37: Classification results using SVM and KNN for feature Selection using LPD in the MD dataset. This table depicts classification accuracies for certain values of h in which LPD FS achieved the best accuracy score for each metric and number of topics, while $t_{LPD} = 0.2$. LPD FS outperforms SelectKBest for each combination of classification-metric-number of topics.

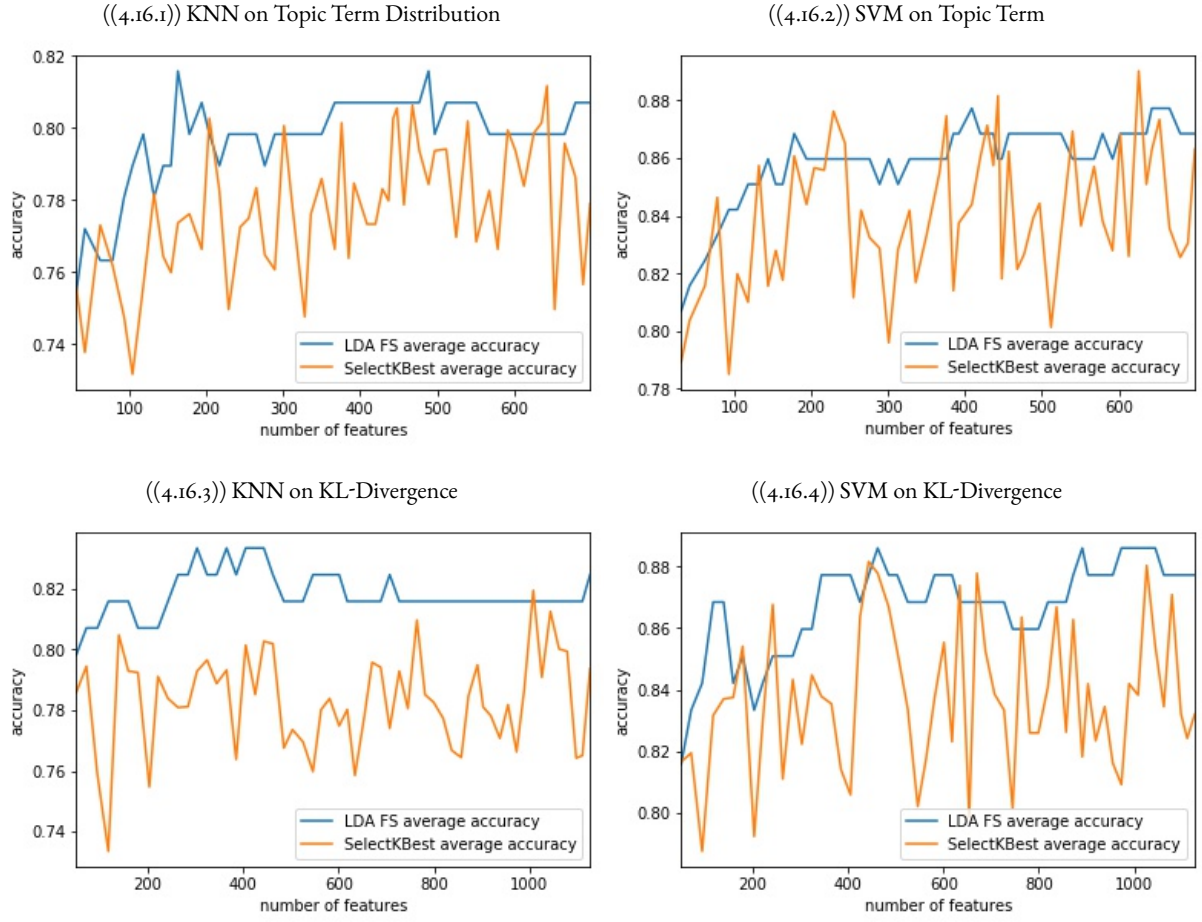


Figure 4.16: Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset. These experiments were performed on original GEM without applying any preprocessing. The number of topics is set to $K = 5$

When employing LPD on the TCGA data we assigned no threshold and thus kept the original gene expression matrix. Here, our approach is comparable to the baseline, but the later performs on average better, attaining very good accuracies. Again we observe that the accuracies in this dataset are very close to 1, due to the nature of the dataset. However in Figure 4.17, we see that when $K = 5$ and training an SVM classifier, both KL-Divergence and the Term-Topic matrix achieve really good results.

Method	Metric	Classifier	Topics K	LPD FS	SelectKBest
LPD	Topic-Term	KNN	5	0.843	0.997
LPD	KL-Divergence	KNN	5	0.894	0.997
LPD	Topic-Term	SVM	5	0.993	0.998
LPD	KL-Divergence	SVM	5	0.995	0.998
LPD	Topic-Term	KNN	3	0.827	0.996
LPD	KL-Divergence	KNN	3	0.831	0.996
LPD	Topic-Term	SVM	3	0.988	0.997
LPD	KL-Divergence	SVM	3	0.987	0.997

Table 4.38: Classification accuracy results (average over all h values) with SVM and KNN using LPD for feature selection (LPD FS) on the TCGA Dataset. Each row presents the classification accuracies for each metric and number of topics. SelectKBest outperforms LPD FS for each combination of classifier-metric-number of topics.

Method	Classifier	Metric	Topics K	h	$k \cdot h$	LPD FS	SelectKBest
LPD	KNN	Topic-Term	3	290	864	0.777	0.996
LPD	KNN	KL-Divergence	3	265	781	0.808	0.991
LPD	SVM	Topic-Term	3	170	509	1.000	1.000
LPD	SVM	KL-Divergence	3	160	477	1.000	1.000
LPD	KNN	Topic-Term	5	235	1158	0.908	1.000
LPD	KNN	KL-Divergence	5	285	1409	0.934	1.000
LPD	SVM	Topic-Term	5	190	937	1.000	1.000
LPD	SVM	KL-Divergence	5	170	845	1.000	0.996

Table 4.39: Classification results with SVM and KNN using LPD for feature selection (LPD FS) on the TCGA dataset. This table depicts classification accuracies for certain values of h in which LPD FS achieved the best accuracy score for each metric and number of topics. LPD FS and SelectKBest exhibit comparable performance in most cases.

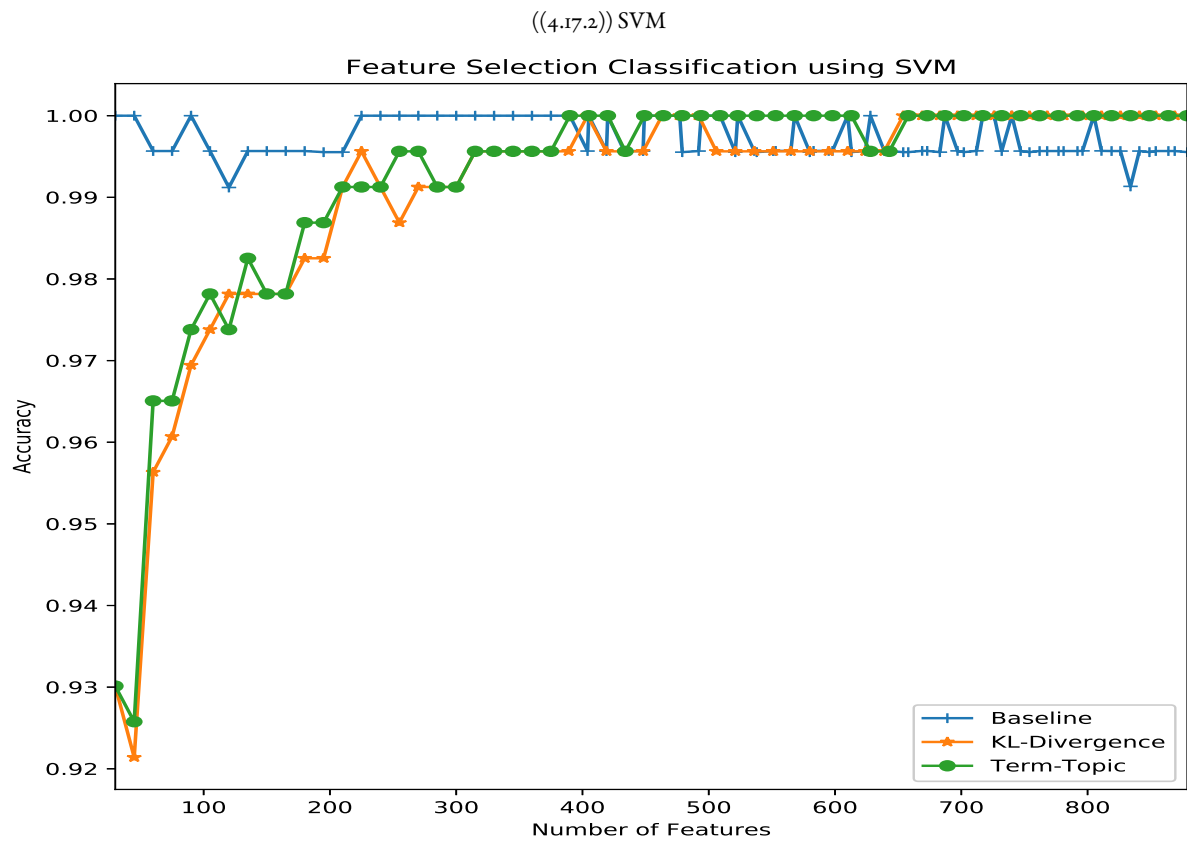
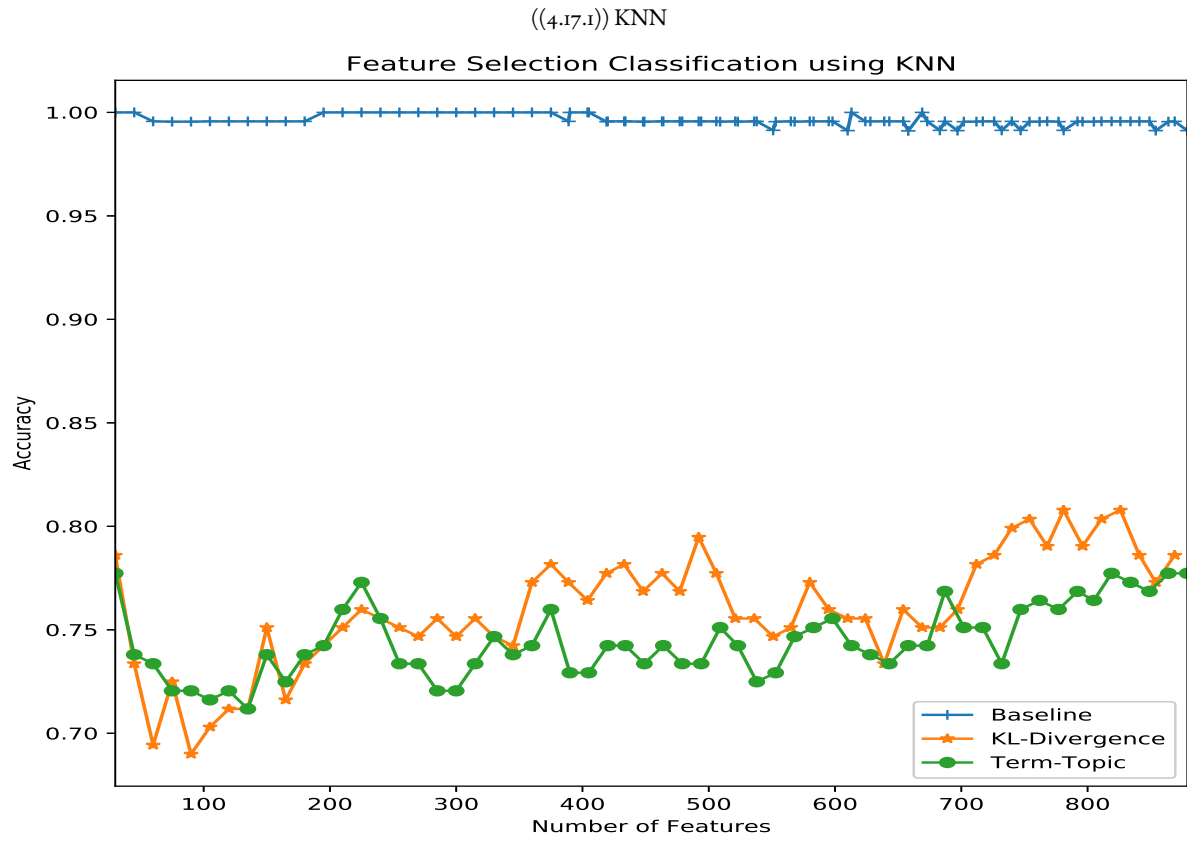


Figure 4.17: Unsupervised Feature Selection using LPD and Univariate Feature Selection on TCGA Dataset where $K = 3$ on the GEM.

Evaluation using Clustering

To evaluate feature selection using LPD, we also employed a clustering algorithm. In the Muscle Dataset the number of clusters was set to $[7, 14]$ and variable $h = \{50, 100, 150, 200\}$. In Table 4.40 the optimal results are shown for all values of $t_{LPD} = 0$, and the different metrics/scores. The results shown are the settings in which our method achieved the highest NMI given a certain Matrix and value of h .

t_{LPD}	Metric	n-clusters	K	h	$k * h$	LPD FS-NMI	SelectKBest-NMI	LPD FS-Rand	SelectKBest-Rand
o	Topic Term	7	5	200	527	0.803	0.804	0.698	0.678
o	KL-Divergence	7	5	100	444	0.815	0.804	0.728	0.678
o	Topic Term	7	10	100	524	0.803	0.804	0.698	0.678
o	KL-Divergence	7	10	100	902	0.812	0.804	0.715	0.678
o	Topic Term	7	15	50	414	0.827	0.804	0.734	0.678
o	KL-Divergence	8	15	50	705	0.822	0.796	0.723	0.692
o.2	Topic Term	7	5	150	430	0.815	0.801	0.728	0.701
o.2	KL-Divergence	7	5	50	232	0.827	0.801	0.734	0.701
o.2	Topic Term	7	10	200	795	0.827	0.801	0.734	0.701
o.2	KL-Divergence	7	10	100	815	0.827	0.801	0.734	0.701
o.2	Topic Term	7	15	100	609	0.827	0.807	0.734	0.691
o.2	KL-Divergence	7	15	150	1720	0.827	0.794	0.734	0.684
✗	Topic Term	7	5	150	385	0.827	0.780	0.734	0.653
✗	KL-Divergence	7	5	100	425	0.827	0.811	0.734	0.711
✗	Topic Term	7	10	100	420	0.809	0.804	0.701	0.678
✗	KL-Divergence	7	10	50	398	0.827	0.794	0.734	0.673
✗	Topic Term	7	15	100	624	0.803	0.808	0.698	0.692
✗	KL-Divergence	7	15	50	635	0.812	0.788	0.715	0.701

Table 4.40: Clustering scores using HAC for all threshold-defined variants of LPD in the MD dataset. Each row shows the supervised clustering metrics to evaluate HAC for all values of t_{LPD} and for each metric. LPD FS achieves the highest NMI and RAND scores.

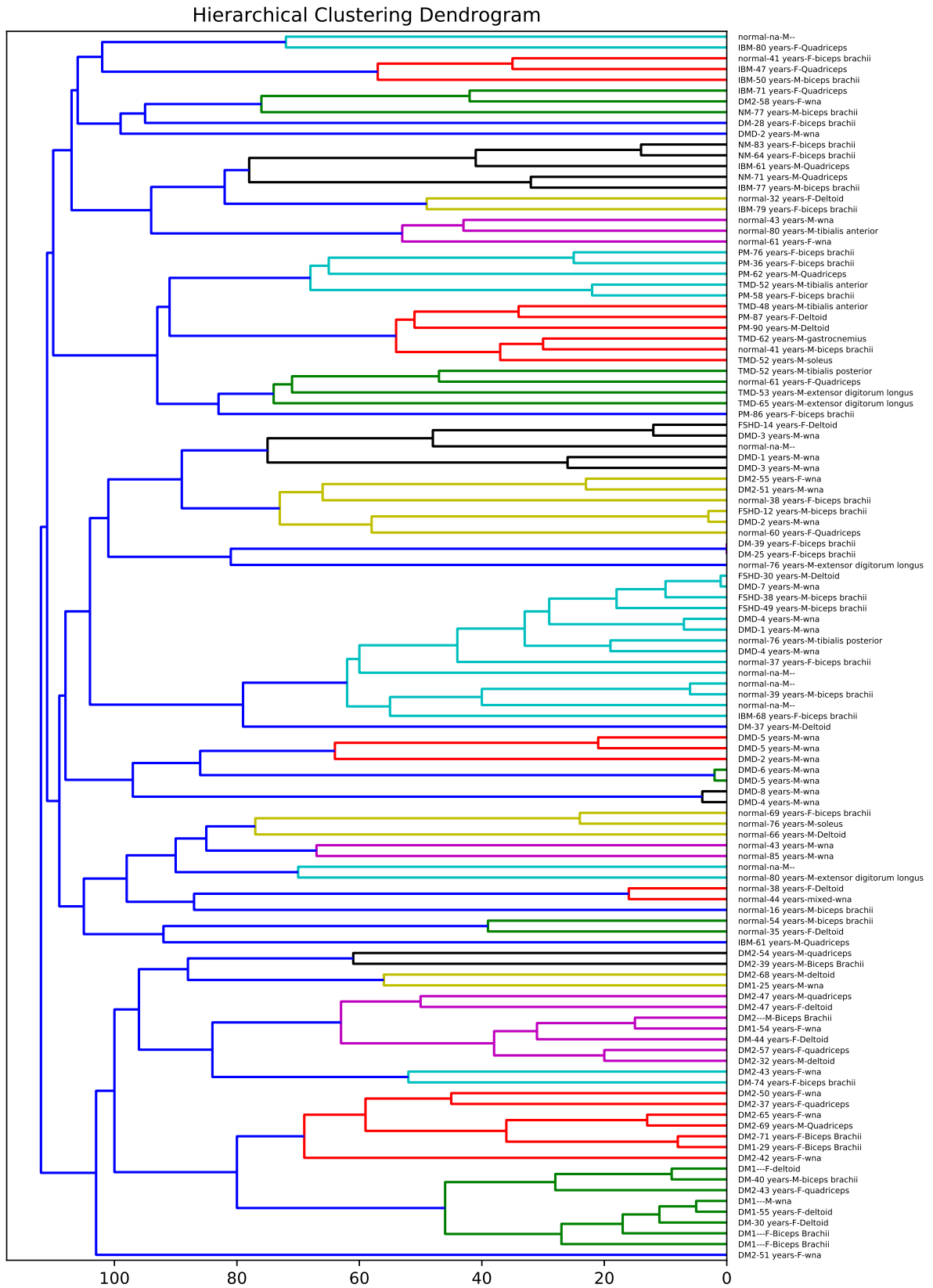


Figure 4.18: HAC Dendrogram for UFS-LPD on the TCGA Dataset KL-Divergence Matrix where $t_{LPD} = 0$, $K = 15$, $N_{clusters} = 7$

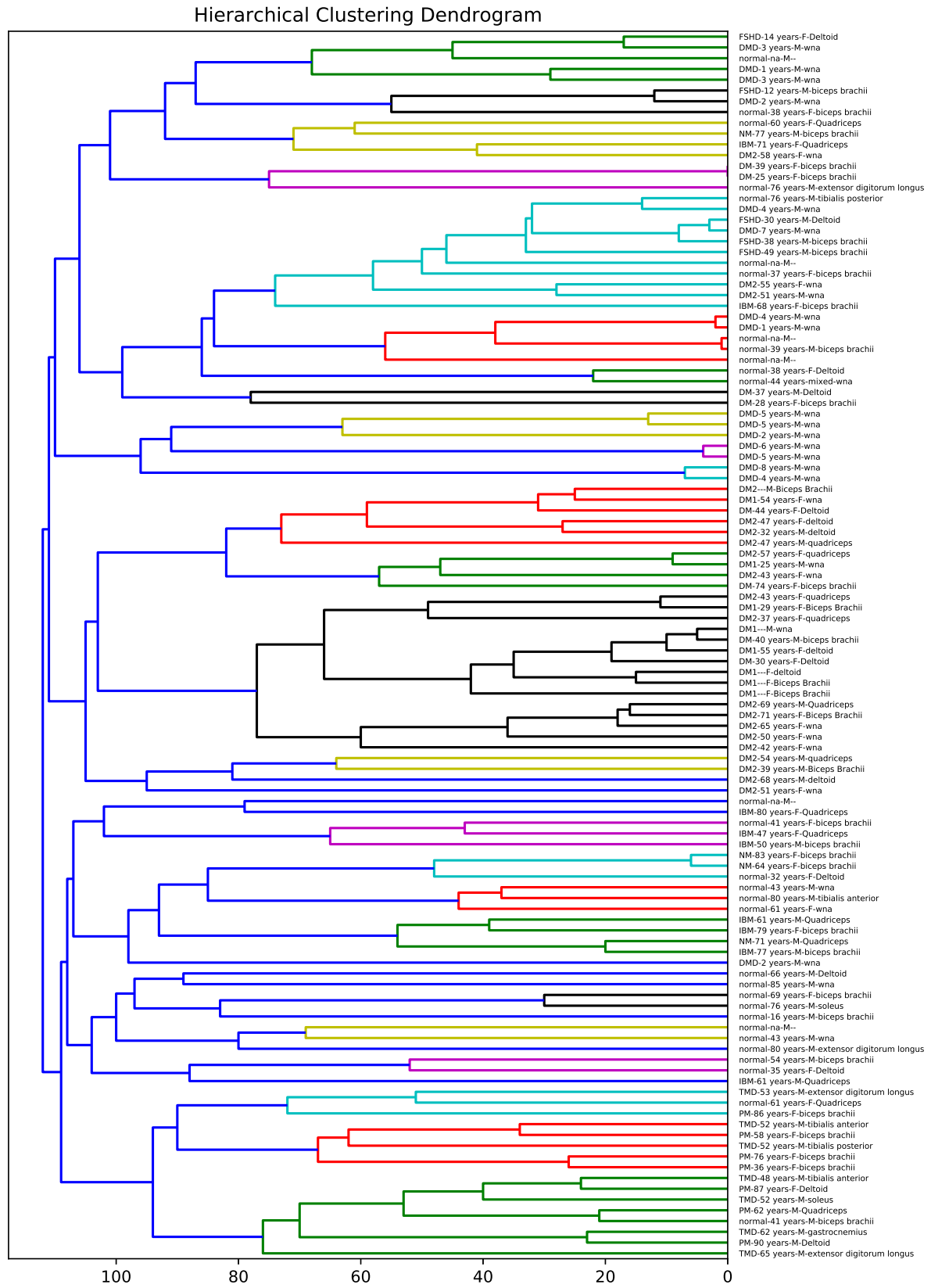


Figure 4.19: HAC Dendrogram for UFS-LPD on the MD Dataset KL-Divergence Matrix where $t_{LPD} = 0.2$, $K = 5$, $N_{clusters} = 7$

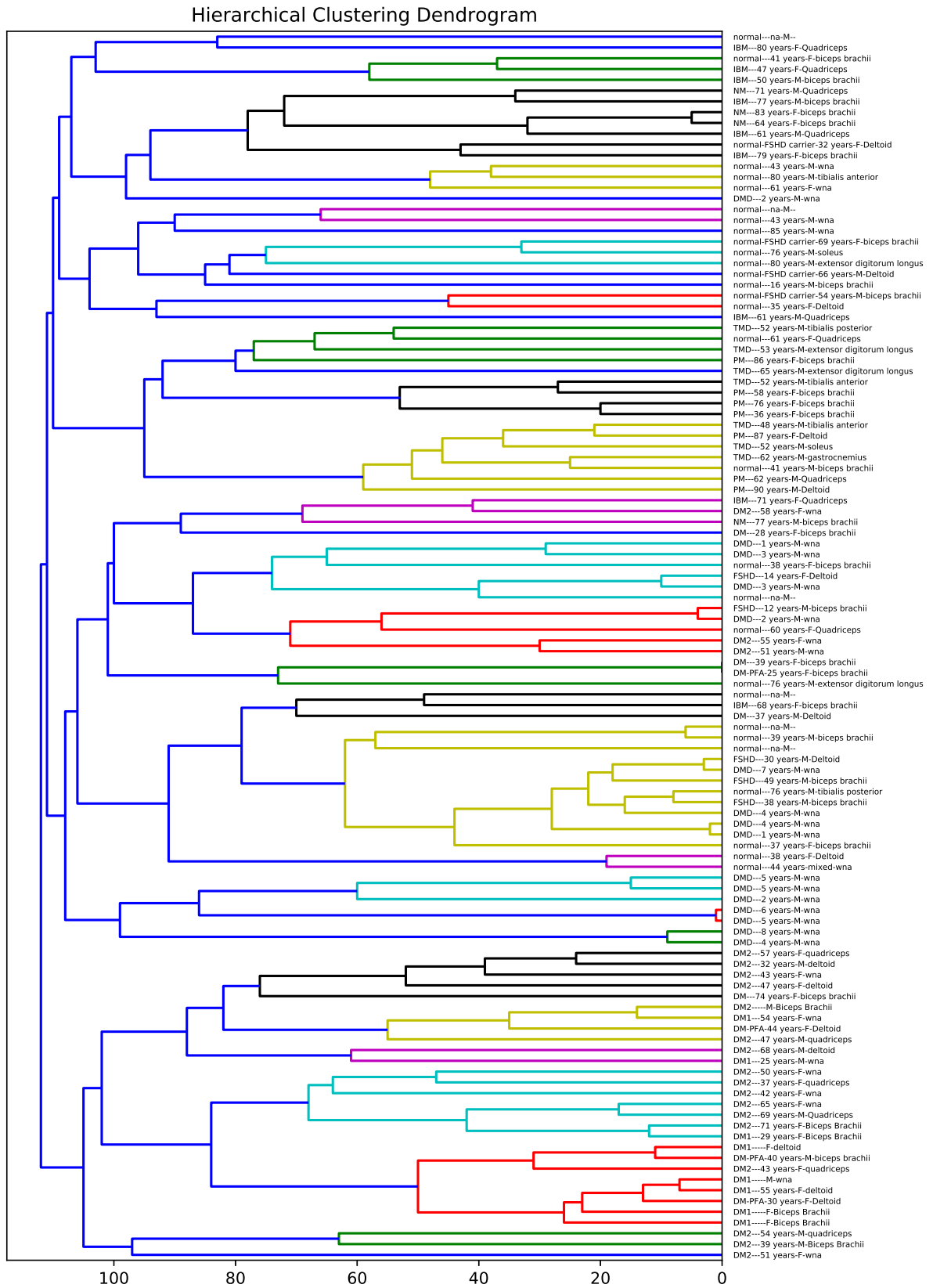


Figure 4.20: HAC Dendrogram for UFS-LPD on the MD Dataset KL-Divergence Matrix where $t_{LPD} = \mathbf{X}$, $K = 10$, $N_{clusters} = 7$

For the TCGA Dataset, clustering evaluation for our approach with respect to the evaluation metrics we have utilized, did not work well. Instead of visualizing the evaluation metrics we will be exhibiting a rather well distinguished Dendrogram produced with HAC in Figure 4.21.

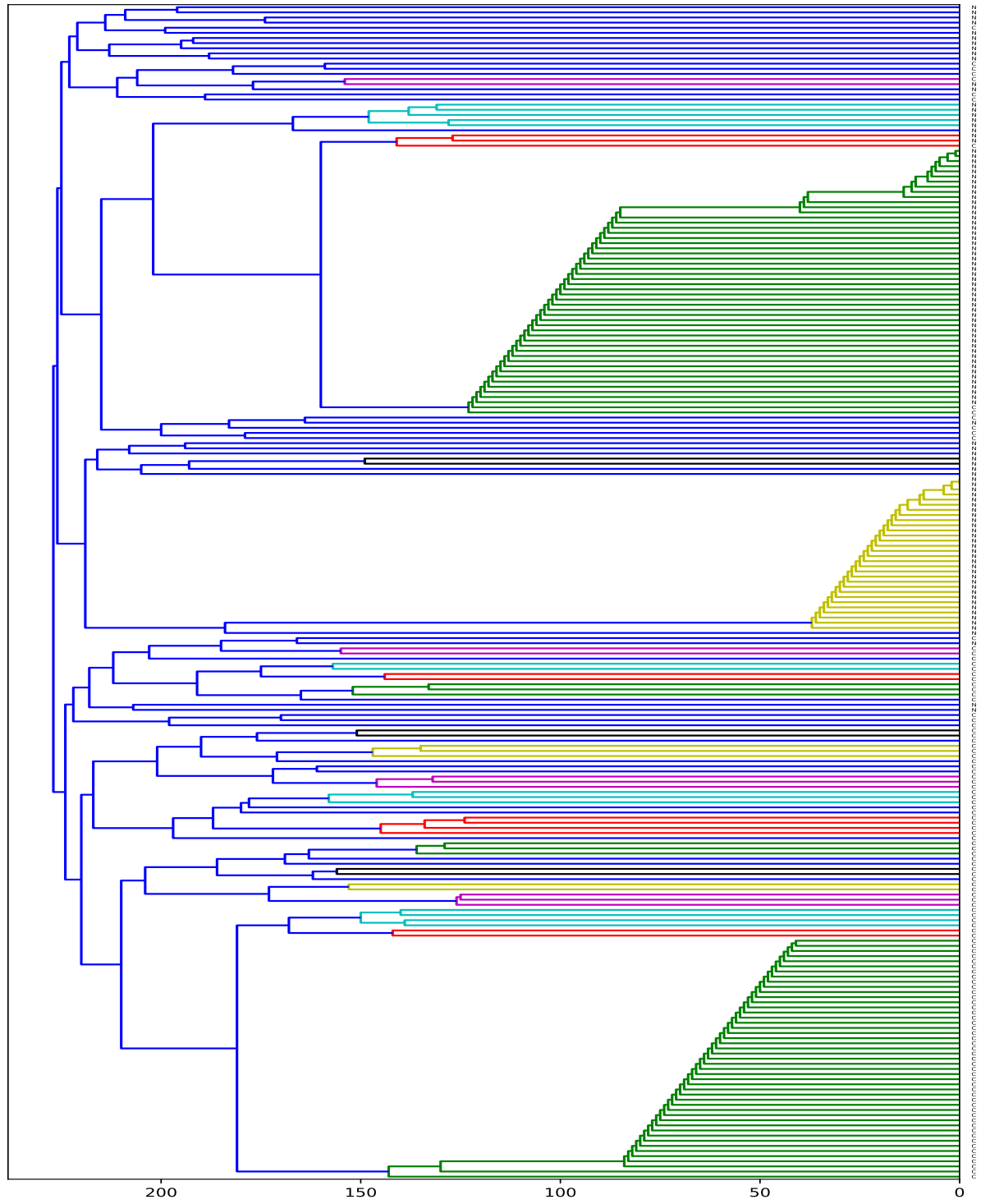


Figure 4.21: HAC Dendrogram for UFS-LPD on the TCGA Dataset KL-Divergence Matrix where $t_{LPD} = \mathbf{X}$, $K = 3$, $N_{clusters} = 2$

4.2.3 Feature Extraction

In this part we will present the results of using the extracted topic models as a feature extraction method. As already mentioned, LDAs and LPDs document-term distribution (9) has been utilized as a dimensionality reduction technique. Specifically, supposing that M is the number of samples, N is the number of features/genes and K the number of topics, the initial dimension of the GEM is $N \times M$. We project these dimensions to a smaller space by depicting the GEM as the document-term distribution, and thus the space is reduced to $M \times K$. That is, the intuition is to interpret the topics as the reduced feature space. To evaluate this approach, we employed both classification algorithms and clustering. When training a classifier, we measure the performance of the approach by computing the mean accuracy of K-fold cross validation. For the clustering algorithms we visualized the results. In what follows we will exhibit the feature extraction algorithm using LDA and afterwards LPD.

4.2.3.1 Feature Extraction using LDA

We compare systematically feature extraction using LDA (henceforth “LDA FE” for short) to a well established feature extraction method, namely Principal Component Analysis (PCA). The number of components that we assign to PCA to extract, is set to the same number of topics HDP has inferred for the particular variant in the Muscle Disease Dataset, and $K = 3$ for the TCGA Dataset.

Evaluation using Classification

In this part, we examine the performance of using the document-topic distribution as a feature extraction algorithm via employing two classifiers, SVM and k-Nearest Neighbor. Similarly we utilize both classifiers using K_f -fold cross validation, where $K_f = 3$ for the MD Dataset and $K_f = 8$ for the TCGA Dataset and present the optimal results.

For the Muscle Disease Dataset, we observed that the results of the extracted topic models behaved very poorly (≈ 0.4). In order to resolve this, and figure out whether the extracted topics could not “describe” the data, we altered the classification problem. Particularly, we performed binary classification in order to predict if the samples are “normal” or “diseased”. The number of classes in this case is 2. In another approach, we eliminated samples suffering from diseases *Necrotizing Myopathy NM* and *Facioscapulohumeral Muscular Dystrophy (FSHD)*, which appeared in a very small amount of samples. In Tables 4.43, 4.42, 4.43, we present the Feature Extraction Results for the Muscle Dataset. Naturally, when we decrease the number of classes (i.e. diseases), better classification accuracy is obtained for our approach, and PCA. However the results are not very promising.

Method	t_M	Classifier	#Classes	n Features	LDA FE	PCA
Median	0	SVM	2	11	0.903	0.929
Median	0	KNN	2	11	0.947	0.939
Median	0	SVM	8	11	0.771	0.846
Median	0	KNN	8	11	0.846	0.833
Median	0	SVM	10	11	0.703	0.796
Median	0	KNN	10	11	0.756	0.815
Median	0.2	SVM	2	25	0.913	0.913
Median	0.2	KNN	2	25	0.912	0.921
Median	0.2	SVM	8	25	0.828	0.829
Median	0.2	KNN	8	25	0.829	0.895
Median	0.2	SVM	10	25	0.745	0.826
Median	0.2	KNN	10	25	0.770	0.822
Median	X	SVM	2	10	0.912	0.948
Median	X	KNN	2	10	0.920	0.956
Median	X	SVM	8	10	0.733	0.867
Median	X	KNN	8	10	0.819	0.858
Median	X	SVM	10	10	0.660	0.843
Median	X	KNN	10	10	0.761	0.807

Table 4.41: Classification results using SVM and KNN for all “Median” transformation method variants in the MD dataset. Each row presents the classification accuracies obtained after feature extraction is performed with LDA and PCA, for each of the different values of t_M and the corresponding number of conditions (classes) the classifier should predict. Feature Extraction using PCA outperforms LDA in most cases.

Method	t_B	Classifier	#Classes	n Features	LDA FE	PCA
Bibin	0	KNN	2	17	0.903	0.903
Bibin	0	SVM	2	17	0.737	0.921
Bibin	0	KNN	8	17	0.826	0.870
Bibin	0	SVM	8	17	0.468	0.875
Bibin	0	KNN	10	17	0.746	0.808
Bibin	0	SVM	10	17	0.431	0.816
Bibin	0.2	KNN	2	19	0.912	0.921
Bibin	0.2	SVM	2	19	0.895	0.921
Bibin	0.2	KNN	8	19	0.839	0.857
Bibin	0.2	SVM	8	19	0.763	0.875
Bibin	0.2	KNN	10	19	0.773	0.784
Bibin	0.2	SVM	10	19	0.712	0.816
Bibin	X	KNN	2	12	0.737	0.903
Bibin	X	SVM	2	12	0.737	0.912
Bibin	X	KNN	8	12	0.554	0.798
Bibin	X	SVM	8	12	0.286	0.848
Bibin	X	KNN	10	12	0.554	0.789
Bibin	X	SVM	10	12	0.263	0.791

Table 4.42: Classification results using SVM and KNN for all “Bibin” transformation method variants in the MD dataset. Each row presents the classification accuracy obtained after feature extraction is performed with LDA and PCA, for each of the different values of t_B and the corresponding number of conditions (classes) the classifier should predict. Feature Extraction using PCA consistently outperforms LDA.

Method	t_1	Remove bin 1	t_2	Classifier	#Classes	LDA FE	PCA
Repetition	X	X	X	SVM	2	0.737	0.912
Repetition	X	X	X	KNN	2	0.868	0.938
Repetition	X	X	X	SVM	8	0.286	0.867
Repetition	X	X	X	KNN	8	0.724	0.883
Repetition	X	X	X	SVM	10	0.263	0.82
Repetition	X	X	X	KNN	10	0.657	0.765
Repetition	X	X	0.2	KNN	2	0.903	0.947
Repetition	X	X	0.2	SVM	2	0.737	0.913
Repetition	X	X	0.2	KNN	8	0.85	0.847
Repetition	X	X	0.2	SVM	8	0.667	0.856
Repetition	X	X	0.2	KNN	10	0.739	0.834
Repetition	X	X	0.2	SVM	10	0.607	0.805
Repetition	X	✓	X	KNN	2	0.868	0.93
Repetition	X	✓	X	SVM	2	0.737	0.93
Repetition	X	✓	X	KNN	8	0.678	0.893
Repetition	X	✓	X	SVM	8	0.286	0.879
Repetition	X	✓	X	KNN	10	0.649	0.795
Repetition	X	✓	X	SVM	10	0.263	0.819
Repetition	X	✓	0	SVM	2	0.737	0.956
Repetition	X	✓	0	KNN	2	0.894	0.912
Repetition	X	✓	0	SVM	8	0.619	0.867
Repetition	X	✓	0	KNN	8	0.774	0.867
Repetition	X	✓	0	SVM	10	0.571	0.79
Repetition	X	✓	0	KNN	10	0.684	0.787
Repetition	0.2	X	X	SVM	2	0.737	0.904
Repetition	0.2	X	X	KNN	2	0.912	0.798
Repetition	0.2	X	X	SVM	8	0.609	0.877
Repetition	0.2	X	X	KNN	8	0.83	0.929
Repetition	0.2	X	X	SVM	10	0.571	0.809
Repetition	0.2	X	X	KNN	10	0.717	0.866
Repetition	0.2	X	0.3	KNN	2	0.913	0.896
Repetition	0.2	X	0.3	SVM	2	0.737	0.903
Repetition	0.2	X	0.3	KNN	8	0.781	0.867
Repetition	0.2	X	0.3	SVM	8	0.632	0.85
Repetition	0.2	X	0.3	KNN	10	0.784	0.8
Repetition	0.2	X	0.3	SVM	10	0.583	0.856
Repetition	0.2	✓	X	SVM	2	0.763	0.93
Repetition	0.2	✓	X	KNN	2	0.894	
Repetition	0.2	✓	X	SVM	8	0.687	0.884
Repetition	0.2	✓	X	KNN	8	0.828	0.877
Repetition	0.2	✓	X	SVM	10	0.642	
Repetition	0.2	✓	X	KNN	10	0.715	0.921
Repetition	0	X	X	SVM	2	0.737	0.913
Repetition	0	X	X	KNN	2	0.921	0.921
Repetition	0	X	X	SVM	8	0.629	0.886
Repetition	0	X	X	KNN	8	0.819	0.876
Repetition	0	X	X	SVM	10	0.58	0.803
Repetition	0	X	X	KNN	10	0.774	0.788
Repetition	0	X	0.2	SVM	2	0.737	0.894
Repetition	0	X	0.2	KNN	2	0.886	0.939
Repetition	0	X	0.2	SVM	8	0.62	0.867
Repetition	0	X	0.2	KNN	8	0.784	0.886
Repetition	0	X	0.2	SVM	10	0.571	0.801
Repetition	0	X	0.2	KNN	10	0.746	0.825
Repetition	0	✓	X	SVM	2	0.737	0.921
Repetition	0	✓	X	KNN	2	0.895	0.938
Repetition	0	✓	X	SVM	8	0.619	0.857
Repetition	0	✓	X	KNN	8	0.771	0.866
Repetition	0	✓	X	SVM	10	0.571	0.794
Repetition	0	✓	X	KNN	10	0.718	0.795

Table 4.43: Classification results using SVM and KNN for all “Repetition” transformation method variants in the MD dataset. Each row presents the classification accuracies obtained after feature extraction is performed with LDA and PCA, for each of the different values of the thresholds (t_1 , RemoveBin1, t_2). Feature Extraction using PCA outperforms LDA in most cases.

In Tables 4.44,4.45,4.46 the equivalent results for the TCGA Dataset are exhibited. In general the performance is enhanced for this dataset, although when LDA is employed on this dataset using the Bibin variant, the results are disappointing. This proves the intuition we pointed out in the feature selection approach, that due to the variance of the data, this method is not suitable for discretizing this data.

Method	t_M	Classifier	Topics K	LDA FE	PCA
Median	0.2	SVM	3	0.983	0.811
Median	0.2	KNN	3	0.983	0.637
Median	X	SVM	3	0.952	0.860
Median	X	KNN	3	0.952	0.633

Table 4.44: Classification results using SVM and KNN for all “Median” transformation method variants in the TCGA dataset. Each row presents the classification accuracies obtained after feature extraction is performed with LDA and PCA, for each of the different values of t_M . Feature Extraction using LDA outperforms PCA.

Method	t_B	Classifier	Topics K	LDA FE	PCA
Bibin	0.2	SVM	3	0.511	0.811
Bibin	0.2	KNN	3	0.572	0.637
Bibin	X	SVM	3	0.533	0.860
Bibin	X	KNN	3	0.555	0.633

Table 4.45: Classification results using SVM and KNN for all “Bibin” transformation method variants in the TCGA dataset. Each row presents the classification accuracies obtained after feature extraction is performed with LDA and PCA, for each of the different values of t_B . PCA outperforms LDA.

Method	t_1	Remove Bin 1	t_2	Classifier	LDA FE	PCA
Repetition	0.2	✗	✗	SVM	0.799	0.633
Repetition	0.2	✗	✗	KNN	0.777	0.659
Repetition	0.2	✗	0.3	SVM	0.983	0.668
Repetition	0.2	✗	0.3	KNN	0.987	0.664
Repetition	0.2	✓	✗	SVM	0.808	0.642
Repetition	0.2	✓	✗	KNN	0.821	0.659
Repetition	✗	✗	✗	SVM	0.511	0.811
Repetition	✗	✗	✗	KNN	0.576	0.860
Repetition	✗	✗	0.2	SVM	0.900	0.790
Repetition	✗	✗	0.2	KNN	0.904	0.799
Repetition	✗	✓	✗	SVM	0.843	0.834
Repetition	✗	✓	✗	KNN	0.900	0.839

Table 4.46: Classification results using SVM and KNN for all “Repetition” transformation method variants in the MD dataset. Each row presents the classification accuracies obtained after feature extraction is performed with LDA and PCA, for each of the different values of the thresholds (t_1 , RemoveBin1, t_2). Feature Extraction using LDA outperforms PCA.

Evaluation using Clustering

In an equivalent matter, we implement two clustering techniques, Hierarchical Agglomerative Clustering (HAC) and k-Means for the MD Dataset and HAC for the TCGA Dataset. For the MD dataset the number of clusters was set to [7, 14], while for the TCGA data $n_{clusters} = [2, 4]$ We did not evaluate their results using evaluation metrics, but rather preferred to visualize the results for both clustering algorithms, using our approach comparing to the PCA method. We also visualize the results of k-means. After having employed this algorithm, we project the created clusters to the 3-Dimensional space by using PCA with the number of components equal to three. An example of this visualization is shown in figure . In case of the HAC we indicatively show one dendrogram for each dataset in Figures 4.23,4.24.

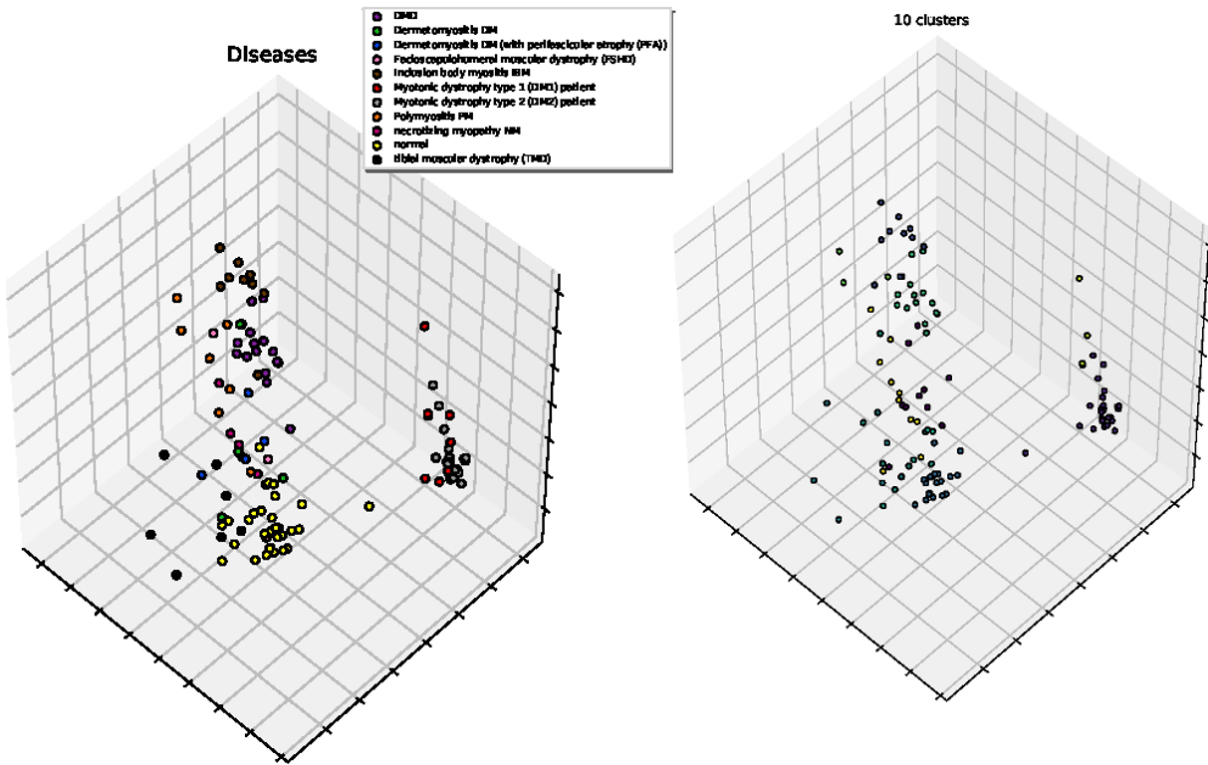


Figure 4.22: Feature Extraction using LDA and on Repetition variant with $(t_1 = 0, \text{Remove Bin } 1=\mathbf{X}, t_2 = 0)$ of MD Dataset. On the right we visualize the clusters inferred by k-means, and each of the $n_{clusters} = 10$ are colored according to the assigned cluster. On the left the same clusters are coloured according to the disease.

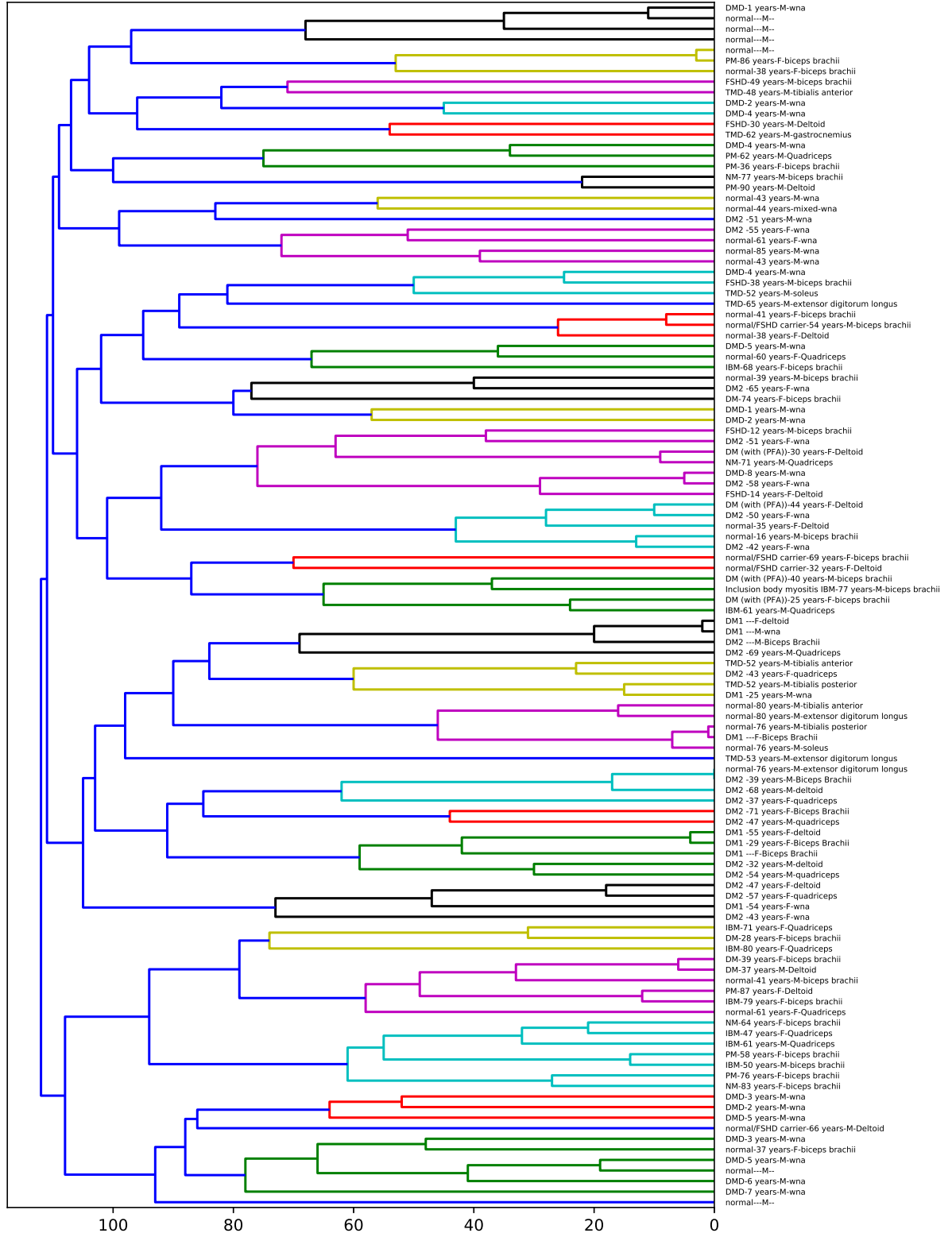


Figure 4.23: HAC Dendrogram for Feature Extraction using LDA and on Bibin variant with ($t_B = 0.2$, $n_{clusters} = 10$) of MD Dataset.

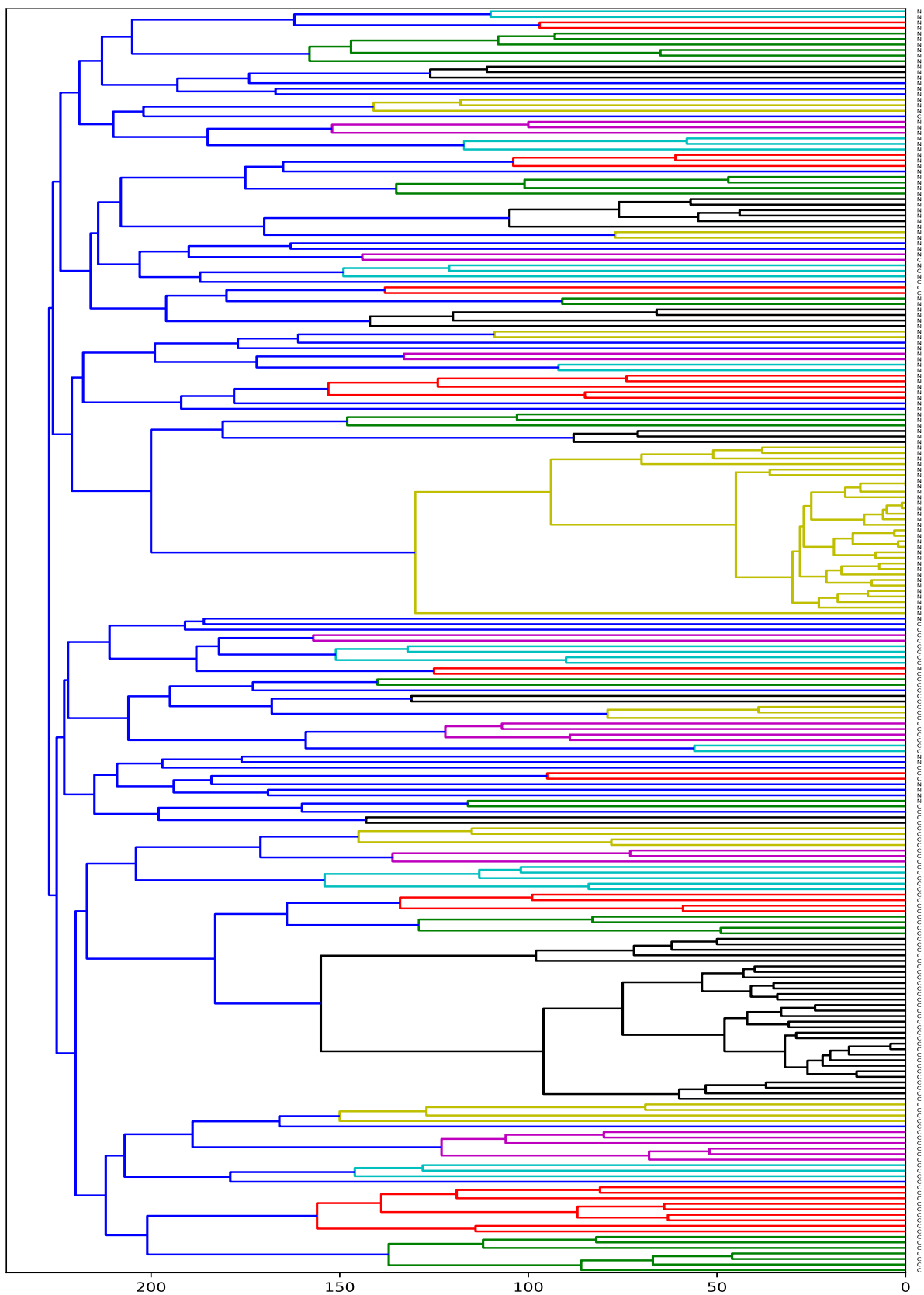


Figure 4.24: HAC Dendrogram for Feature Extraction using LDA and on Median variant with $(t_M = X, n_{clusters} = 3)$ of TCGA Dataset.

4.2.3.2 Feature Extraction using LPD

In this work, we also provide feature extraction classification results using LPD (henceforth “LPD FE” for short), for each of the Datasets. For the MD Dataset, we follow the same procedure regarding the number of classes. In Tables 4.47,4.48,4.49 we present the classification results of our approach compared to extracting features with PCA. Again we have divided the problem to three sub-problems by predicting the binary,8-class,and 10-class problems, as stated in extracting features with LDA. In general, PCA performs better, however we observe that when we predict all the classes, feature extraction with LPD achieves higher accuracies. It is interesting that when the data is removed, we attain significantly better accuracy. Moving on with the TCGA dataset, in Table 4.50, we present the equivalent results, which show the competitiveness of our approach to the baseline PCA. It seems that LPD can distinguish the identity of the topics more clearly, leading to better classification results.

Method	Classifier	Topics K	LPD FE	PCA
LPD	SVM	3	0.948	0.852
LPD	KNN	3	0.961	0.671
LPD	SVM	5	0.965	0.876
LPD	KNN	5	0.969	0.221

Table 4.50: Classification Results with SVM and KNN using LPD FE on the TCGA Dataset. Each row depicts the classification accuracy depending on the number of topics. LPD FE achieves the best results.

Classifier	t_{LPD}	Classes	Features	LPD	PCA
SVM	\times	2	5	0.912	0.895
KNN	\times	2	5	0.912	0.938
SVM	\times	2	10	0.930	0.921
KNN	\times	2	10	0.904	0.921
SVM	\times	2	15	0.877	0.911
KNN	\times	2	15	0.886	0.921
SVM	\times	8	5	0.781	0.837
KNN	\times	8	5	0.810	0.829
SVM	\times	8	10	0.867	0.875
KNN	\times	8	10	0.857	0.848
SVM	\times	8	15	0.876	0.849
KNN	\times	8	15	0.867	0.876
SVM	\times	10	5	0.719	0.764
KNN	\times	10	5	0.737	0.785
SVM	\times	10	10	0.798	0.796
KNN	\times	10	10	0.772	0.825
SVM	\times	10	15	0.781	0.764
KNN	\times	10	15	0.772	0.785

Table 4.47: Classification Results with SVM and KNN using LPD FE on the MD Dataset. Each row depicts the classification accuracy depending on the number of conditions (classes) to predict) and the number of topics, while no data has been removed ($t_{LPD} = \times$). PCA performs better than LPD FE, although LPD FE achieves comparable results.

Classifier	t_{LPD}	Classes	Features	LPD FE	PCA
SVM	0	2	5	0.886	0.912
KNN	0	2	5	0.868	0.921
SVM	0	2	10	0.868	0.912
KNN	0	2	10	0.816	0.930
SVM	0	2	15	0.895	0.930
KNN	0	2	15	0.851	0.921
SVM	0	8	5	0.771	0.826
KNN	0	8	5	0.810	0.828
SVM	0	8	10	0.848	0.857
KNN	0	8	10	0.829	0.857
SVM	0	8	15	0.848	0.868
KNN	0	8	15	0.838	0.886
SVM	0	10	5	0.702	0.565
KNN	0	10	5	0.711	0.580
SVM	0	10	10	0.763	0.612
KNN	0	10	10	0.763	0.809
SVM	0	10	15	0.763	0.641
KNN	0	10	15	0.763	0.614

Table 4.48: Classification Results with SVM and KNN using LPD FE on the MD Dataset. Each row depicts the classification accuracy depending on the number of conditions (classes) to predict) the number of topics, while $t_{LPD} = 0.0$. PCA performs better than LPD FE, although LPD FE achieves higher classification results when predicting all conditions.

Classifier	t_{LPD}	Classes	Features	LPD FE	PCA
SVM	0.2	2	5	0.912	0.921
KNN	0.2	2	5	0.912	0.912
SVM	0.2	2	10	0.886	0.930
KNN	0.2	2	10	0.886	0.921
SVM	0.2	2	15	0.877	0.912
KNN	0.2	2	15	0.868	0.921
SVM	0.2	8	5	0.848	0.857
KNN	0.2	8	5	0.819	0.782
SVM	0.2	8	10	0.819	0.771
KNN	0.2	8	10	0.819	0.838
SVM	0.2	8	15	0.838	0.856
KNN	0.2	8	15	0.848	0.888
SVM	0.2	10	5	0.763	0.521
KNN	0.2	10	5	0.746	0.599
SVM	0.2	10	10	0.737	0.693
KNN	0.2	10	10	0.754	0.566
SVM	0.2	10	15	0.763	0.732
KNN	0.2	10	15	0.772	0.637

Table 4.49: Classification Results with SVM and KNN using LPD FE on the MD Dataset. Each row depicts the classification accuracy depending on the number of conditions (classes) to predict) the number of topics, while $t_{LPD} = 0.2$. PCA performs better than LPD FE, although LPD FE achieves higher classification results when predicting all conditions regardless of the number of topics.

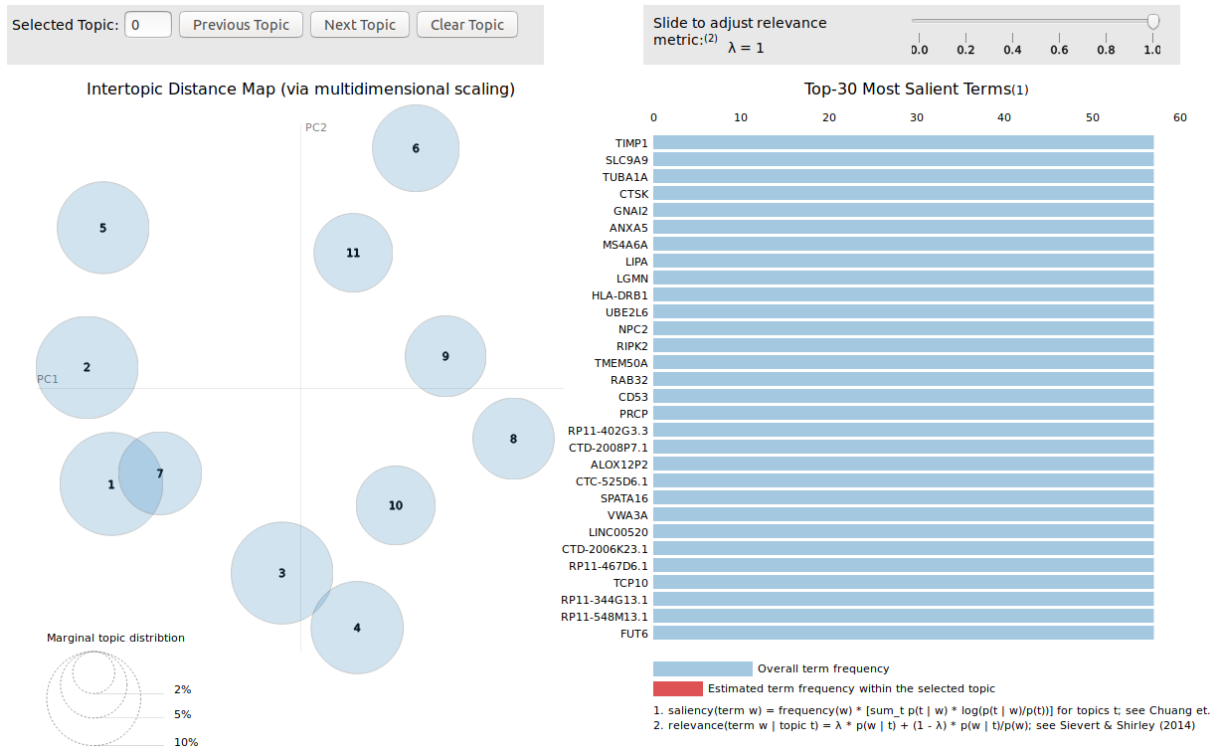


Figure 4.25: The LDA topics visualized with pyLDAvis for the MD dataset with Median variant $t_M = 0.0$

4.2.4 PTM-cluster Analysis. Visualizing the Topic Models

As already mentioned, topic models extract latent topics which are hidden inside a data structure. The interpretation of those topics, can be conducted using analysis tasks as the approaches we employed in this thesis. However, it can be useful to let a human interpret the topics and their significant components. For this reason, we utilized a Python library for interactive topic model visualization, *pyLDAvis* [Sievert and Shirley, 2014]. This tool provides a global inspection of the topics in the 2D-space which can allow the user “labelling” the topics to view the intra-topic similarities and relationships. The user can also observe the most significant terms in a selected topic according to their probability inside the topic, and according to their relevance score. An example of this tool is shown in Figure 4.25. The topics are placed in the 2-D space, and hovering over any topic we can see the most significant terms of that topic. This tool was only employed on the LDA extracted topics, because no compatibility with LPD is provided. The following links direct to the visualization for each of the variant and datasets.

Muscle Disease Dataset LDA Topic Term Distributions Using PyLDAviz

- [Topic Term Distribution of Bibin with \$t_B = \text{X}\$](#)
- [Topic Term Distribution of Bibin with \$t_B = 0\$](#)
- [Topic Term Distribution of Bibin with \$t_B = 0.2\$](#)
- [Topic Term Distribution of Median with \$t_M = \text{X}\$](#)
- [Topic Term Distribution of Median with \$t_M = 0\$](#)
- [Topic Term Distribution of Median with \$t_M = 0.2\$](#)

- Topic Term Distribution of Repetition with $t_1 = 0$, Remove Bin 1 = $\times t_2 = \times$
- Topic Term Distribution of Repetition with $t_1 = 0$, Remove Bin 1 = $\times t_2 = 0.2$
- Topic Term Distribution of Repetition with $t_1 = 0$, Remove Bin 1 = $\checkmark t_2 = \times$
- Topic Term Distribution of Repetition with $t_1 = 0.2$, Remove Bin 1 = $\times t_2 = \times$
- Topic Term Distribution of Repetition with $t_1 = 0.2$, Remove Bin 1 = $\times t_2 = 0.3$
- Topic Term Distribution of Repetition with $t_1 = 0.2$, Remove Bin 1 = $\checkmark t_2 = \times$
- Topic Term Distribution of Repetition with $t_1 = \times$, Remove Bin 1 = $\checkmark t_2 = \times$
- Topic Term Distribution of Repetition with $t_1 = \times$, Remove Bin 1 = $\checkmark t_2 = 0.0$
- Topic Term Distribution of Repetition with $t_1 = \times$, Remove Bin 1 = $\times t_2 = \times$
- Topic Term Distribution of Repetition with $t_1 = \times$, Remove Bin 1 = $\times t_2 = 0.2$

TCGA Dataset Topic Term Distributions Using PyLDAviz

- Topic Term Distribution of Bibin with $t_B = \times$
- Topic Term Distribution of Bibin with $t_B = 0.2$
- Topic Term Distribution of Median with $t_M = \times$
- Topic Term Distribution of Median with $t_M = 0.2$
- Topic Term Distribution of Repetition with $t_1 = 0.2$, Remove Bin 1 = $\times t_2 = \times$
- Topic Term Distribution of Repetition with $t_1 = 0.2$, Remove Bin 1 = $\times t_2 = 0.3$
- Topic Term Distribution of Repetition with $t_1 = 0.2$, Remove Bin 1 = $\checkmark t_2 = \times$
- Topic Term Distribution of Repetition with $t_1 = \times$, Remove Bin 1 = $\times t_2 = \times$
- Topic Term Distribution of Repetition with $t_1 = \times$, Remove Bin 1 = $\times t_2 = 0.2$
- Topic Term Distribution of Repetition with $t_1 = 0 \times$, Remove Bin 1 = $\checkmark t_2 = \times$

Moreover we utilized an interesting nonlinear dimensionality reduction algorithm that is used to visualize high dimensional data, t- Stochastic Neighbor Embedding. We exploited its ability to reduce the feature space to 2 dimensions, and visualize the clusters of the documents. Each document is colored by the topic with the highest probability, and by hovering over the documents, the disease is also revealed. By inspecting carefully these visualizations we observed that there are instances of the topic models we employed, where all the documents have the same most probable topic. On the contrary LPD led to more distinguished topics, an intuition we had when we visualized the distributions over the documents for each topic.

Muscle Data Visualize LDA and LPD Document Term Distributions using t-SNE

LDA

- Visualize LDA model of Bibin with $t_B = \times$
- Visualize LDA model of Bibin with $t_B = 0$
- Visualize LDA model of Bibin with $t_B = 0.2$

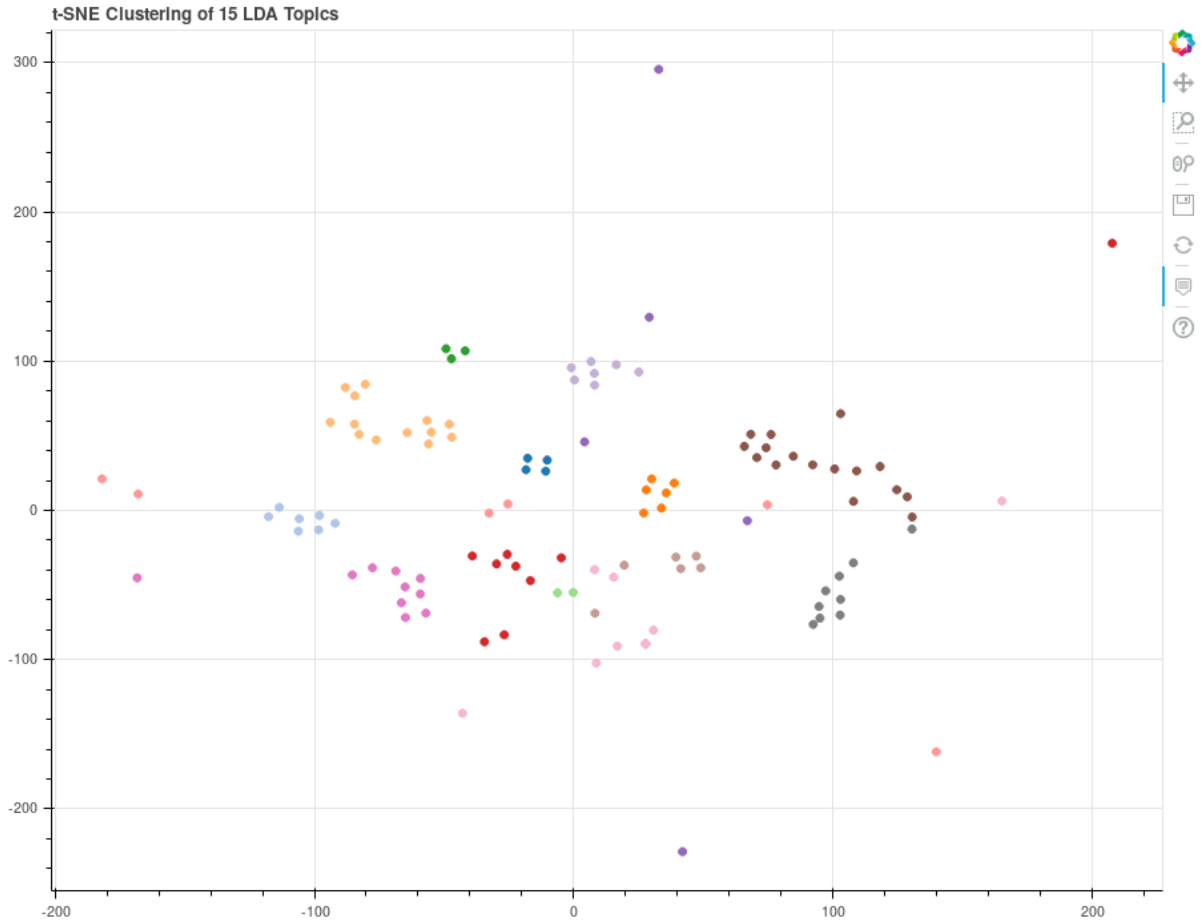


Figure 4.26: The LDA topics visualized with t-SNE for the MD dataset with Media variant $t_B = 0.2$

- Visualize LDA model of Median with $t_M = \mathbf{X}$
- Visualize LDA model of Median with $t_M = 0$
- Visualize LDA model of Median with $t_M = 0.2$
- Visualize LDA model of Repetition with $t_1 = 0, \text{Remove Bin 1} = \mathbf{X} t_2 = \mathbf{X}$
- Visualize LDA model of Repetition with $t_1 = 0, \text{Remove Bin 1} = \mathbf{X} t_2 = 0.2$
- Visualize LDA model of Repetition with $t_1 = 0, \text{Remove Bin 1} = \checkmark t_2 = \mathbf{X}$
- Visualize LDA model of Repetition with $t_1 = 0.2, \text{Remove Bin 1} = \mathbf{X} t_2 = \mathbf{X}$
- Visualize LDA model of Repetition with $t_1 = 0.2, \text{Remove Bin 1} = \mathbf{X} t_2 = 0.3$
- Visualize LDA model of Repetition with $t_1 = 0.2, \text{Remove Bin 1} = \checkmark t_2 = \mathbf{X}$
- Visualize LDA model of Repetition with $t_1 = \mathbf{X}, \text{Remove Bin 1} = \checkmark t_2 = \mathbf{X}$
- Visualize LDA model of Repetition with $t_1 = \mathbf{X}, \text{Remove Bin 1} = \checkmark t_2 = 0.0$
- Visualize LDA model of Repetition with $t_1 = \mathbf{X}, \text{Remove Bin 1} = \mathbf{X} t_2 = \mathbf{X}$
- Visualize LDA model of Repetition with $t_1 = \mathbf{X}, \text{Remove Bin 1} = \mathbf{X} t_2 = 0.2$

LPD

- Visualize LPD model with $t_{LPD} = 0$ and $K = 5$

- Visualize LPD model with $t_{LPD} = 0$ and $K = 10$
- Visualize LPD model with $t_{LPD} = 0$ and $K = 15$
- Visualize LPD model with $t_{LPD} = 0.2$ and $K = 5$
- Visualize LPD model with $t_{LPD} = 0.2$ and $K = 10$
- Visualize LPD model with $t_{LPD} = 0.2$ and $K = 15$
- Visualize LPD model with $t_{LPD} = \text{X}$ and $K = 5$
- Visualize LPD model with $t_{LPD} = \text{X}$ and $K = 10$
- Visualize LPD model with $t_{LPD} = \text{X}$ and $K = 15$

TCGA Data Visualize LDA and LPD Document Term Distributions using t-SNE

LDA

- Topic Term Distribution of Bibin with $t_B = \text{X}$
- Topic Term Distribution of Bibin with $t_B = 0.2$
- Topic Term Distribution of Median with $t_M = \text{X}$
- Topic Term Distribution of Median with $t_M = 0.2$
- Visualize LDA model of Repetition with $t_1 = 0.2$, Remove Bin 1 = X $t_2 = \text{X}$
- Visualize LDA model of Repetition with $t_1 = 0.2$, Remove Bin 1 = X $t_2 = 0.3$
- Visualize LDA model of Repetition with $t_1 = 0.2$, Remove Bin 1 = \checkmark $t_2 = \text{X}$
- Visualize LDA model of Repetition with $t_1 = \text{X}$, Remove Bin 1 = \checkmark $t_2 = \text{X}$
- Visualize LDA model of Repetition with $t_1 = \text{X}$, Remove Bin 1 = \checkmark $t_2 = \text{X}$
- Visualize LDA model of Repetition with $t_1 = \text{X}$, Remove Bin 1 = X $t_2 = \text{X}$
- Visualize LDA model of Repetition with $t_1 = \text{X}$, Remove Bin 1 = X $t_2 = 0.2$

LPD

- Visualize LPD model with $t_{LPD} = \text{X}$ and $K = 3$
- Visualize LPD model with $t_{LPD} = \text{X}$ and $K = 5$

Conclusions and Future Work

5.1 Conclusions

This thesis approached the problem of performing Microarray Analysis using the power of Probabilistic Topic Models (PTMs) to identify “latent” groups of genes (represented by extracted probabilistic topics) that can be associated with specific diseases. The use of PTMs in this domain readily allows the visualization of important information (i.e., disease-related topics), and enables the subsequent employment of their results for enrichment analysis purposes.

We created a generic framework for microarray analysis using PTM. We populated our framework with several data preprocessing and discretization methods, that allow the transformation of microarray data into a “bag of words” form that can be employed in state of the art PTMs. Moreover we examine the use of a PTM, LPD that can handle continuous data. We performed dimensionality reduction (feature extraction, feature selection) on two real world dataset, using our framework and methods, and performed a systematic and extensive evaluation of our approach against commonly used against well known techniques that reduce the feature space without employing PTMs.

The first thing that our results indicate is that the choice of the discretization method to be used matters. Our novel discretization technique, Bibin, achieved rather good results for the Muscle Disease Dataset; on the other hand, in the TCGA dataset, with genes whose expression data was largely close to zero, resulting to the creation of words with very similar frequency, did not perform well. Another conclusion is that, removing under-expressed genes data while employing the (extended to a “pipelined” process) “Repetition” discretization technique, can achieve better classification and clustering results. A third conclusion is that LDA results are in general comparable to those of LPD (which operates on continuous data). This is despite LPD being able to produce topics that are better distinguished and more clearly linked to specific diseases, a fact which naturally makes it more successful and thus appropriate for feature extraction tasks, as confirmed by our results.

Our PTMs-employing approach is quite effective in performing feature selection in general, producing very good classification and clustering results. It is worth noting that our proposed approach applies scoring techniques (“KL-divergence” and “Relevance Score”) on the Topic-Term distributions, to identify the most important genes in a topic. Its success for microarray analysis thus indicates that such scoring techniques can be used to assist feature selection in other application settings as well.

Regarding feature extraction, the LPD topic model appears to be more successful than LDA. Moreover, LPD manages to perform equivalently to the well-established Principal Component Analysis method, even

when operating with a small number of topics.

Overall, we believe that our generic framework, has proven its ability to incorporate and evaluate different PTM models and discretization techniques. Moreover, it comes complete with incorporating modern visualization tools for the presentation of the extracted topics and clusters. As such, our hope is that it will be adopted by biologists and assist their quest in uncovering hidden biological processes.

5.2 Future Work

There are several possible extensions of this work.

First, there is room for improving the performance of the PTMs used in this work. In particular, we believe that LPD's performance for feature extraction tasks can be improved with the fine-tuning of its parameters. A related issue is that in this thesis we used rather simple data discretization techniques, but one could try more complex techniques (such as [Biba et al., 2007]) and evaluate their impact and performance in this setting.

Moreover, one could attempt to "train" PTMs on datasets of progressively increasing sizes, something that was not possible in our work here due to the limited size of our available datasets. In addition, one can incorporate and study different or more advanced topic models than the ones used in this work.

Finally, in this thesis, we focus on employing these models on *array profiles*. A possible alteration would be to employ topic models on *gene profiles*. That is, to exploit meta-data and information of datasets, and implement topic models which assume that a document is the gene, and its corresponding words are attributes such as the condition, age, and so on. This approach can be then compared to this framework and its results.

Another work in progress, is that of combining the feature selection approach using the extracted Topic Models, with cooperative game theory, and the Shapley Value power index. In particular, we intend to use the subsets of features we select with Topic Models that occur by the proposed metrics (i.e. Topic Term Distribution, KL-Divergence, and Relevance Score). Then exploit these subsets by computing the impact of each individual feature by measuring its impact on the feature space using its Shapley Value. The intuition behind this approach is to combine the impact of a feature (gene) given a certain class (condition) measured by a probabilistic topic model, with the power of a feature on a coalition (subset of features) using the power index.

Appendix Title

A.1 Sys-myo Dataset

A.1.1 Data Collection

To collect muscle-specific microarray data and discard low quality samples, we followed a pipeline similar to that used for our Muscle Gene Sets resource [Malatras et al., 2019], as described below. We screened the online high-throughput repositories Gene Expression Omnibus (GEO) [Barrett et al., 2011] and ArrayExpress [Kolesnikov et al., 2014]. The most popular platform for muscle disorders is Affymetrix Human Genome U133 Plus 2.0 GeneChip (GEO platform GPL570 or ArrayExpress ID A-AFFY-44). Next, we developed a script to parse automatically their MIAME [Brazma et al., 2001] metadata and confirm them manually, selecting only those with muscle genetic disorders. We excluded all series that did not include the raw CEL files (Affymetrix fluorescence light intensity files) because we pre-processed them using a robust data analysis pipeline in order to homogenize the data as much as possible. We selected a set of normal muscle tissue and 10 different muscle genetic disorders including Myotonic dystrophy type 2, Myotonic dystrophy type 1, Tibial muscular dystrophy, Facioscapulohumeral muscular dystrophy (FSHD), Inclusion body myositis, Necrotizing myopathy, Dermatomyositis (with perifascicular atrophy), Dermatomyositis, Polymyositis, Duchenne Muscular Dystrophy.

A.1.2 Quality Control

The quality control pipeline was identical to that used previously for our Muscle Gene Sets resource [Malatras et al., 2019]. Arrays that had extreme values or were above our set thresholds on the combined quality controls, were excluded from any further analysis.

A.1.3 Data Normalization

The arrays that passed quality controls were pre-processed with the Single Channel Array Normalization (SCAN) algorithm [Piccolo et al., 2012] with default parameters except for the Chip Description file (CDF), which reorganizes probesets with up-to-date genomic, cDNA and single nucleotide polymorphism (SNP) information in order to create a more accurate and precise CDF. For this study we used the CDF from BrainArray Ensembl ENSG version 20.0.0 [Dai et al., 2005]. SCAN normalizes each array independently from its series, corrects GC bias and reduces probe and array variation from each individual sample while increasing signal-to-noise ra-

tio. Single array normalization is the preferred way for our dataset, since we combine microarray samples from different series and laboratories.

A.1.4 Batch Correction

For batch effect reduction we used the ComBat algorithm [Johnson et al., 2007] from the “SVA” Bioconductor package [Leek et al., 2012]. By default, we considered each data series (i.e. study) to be a different batch. Using principal component analysis (PCA) 3D plot, we identified if the samples correlate with batch surrogates and proceeded with batch correction as it was necessary.

A.1.5 Filtering expressed genes and annotating with gene symbols

In order to distinguish between expressed and unexpressed genes (such as genes with expression levels close to or lower than the background noise), we used the Universal exPression Code (UPC) algorithm [Piccolo et al., 2013]. We did that because pathological conditions have distinct genetic profiles. UPC corrects for background noise using linear statistical models and estimates the percentage of gene expression by calculating the active and inactive gene population. We kept the genes that had at least one sample with expression percentage value of 25 and above. To map Ensembl gene IDs to HGNC gene symbols [Braschi et al., 2018] we used Ensembl BioMart [Kinsella et al., 2011]. We extracted the required information from the human assembly GRCh38.p5.

A.2 Distributions of Documents for each Topic

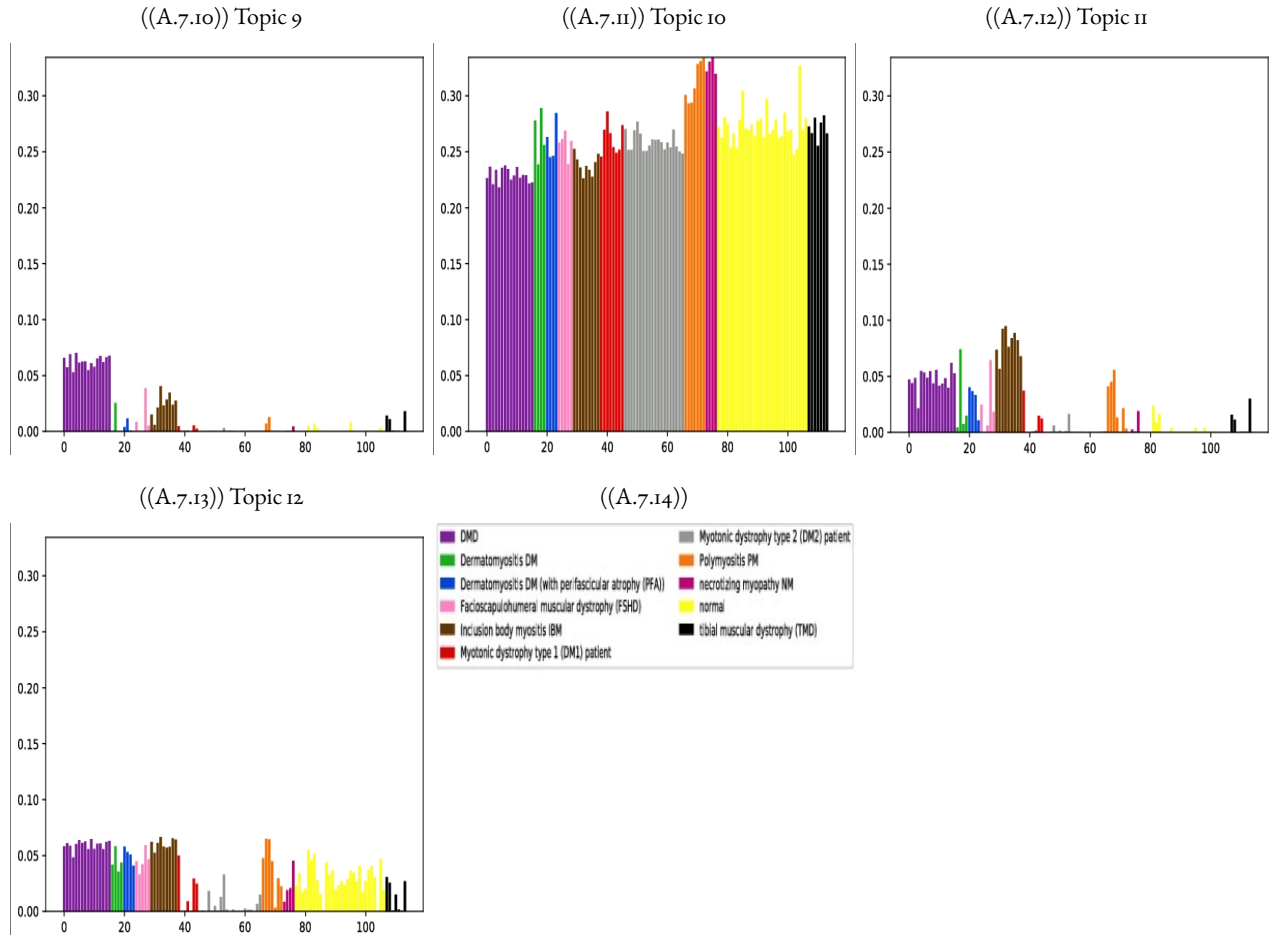


Figure A.7: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = \text{XRemove Bin 1} = \text{X}t_2 = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

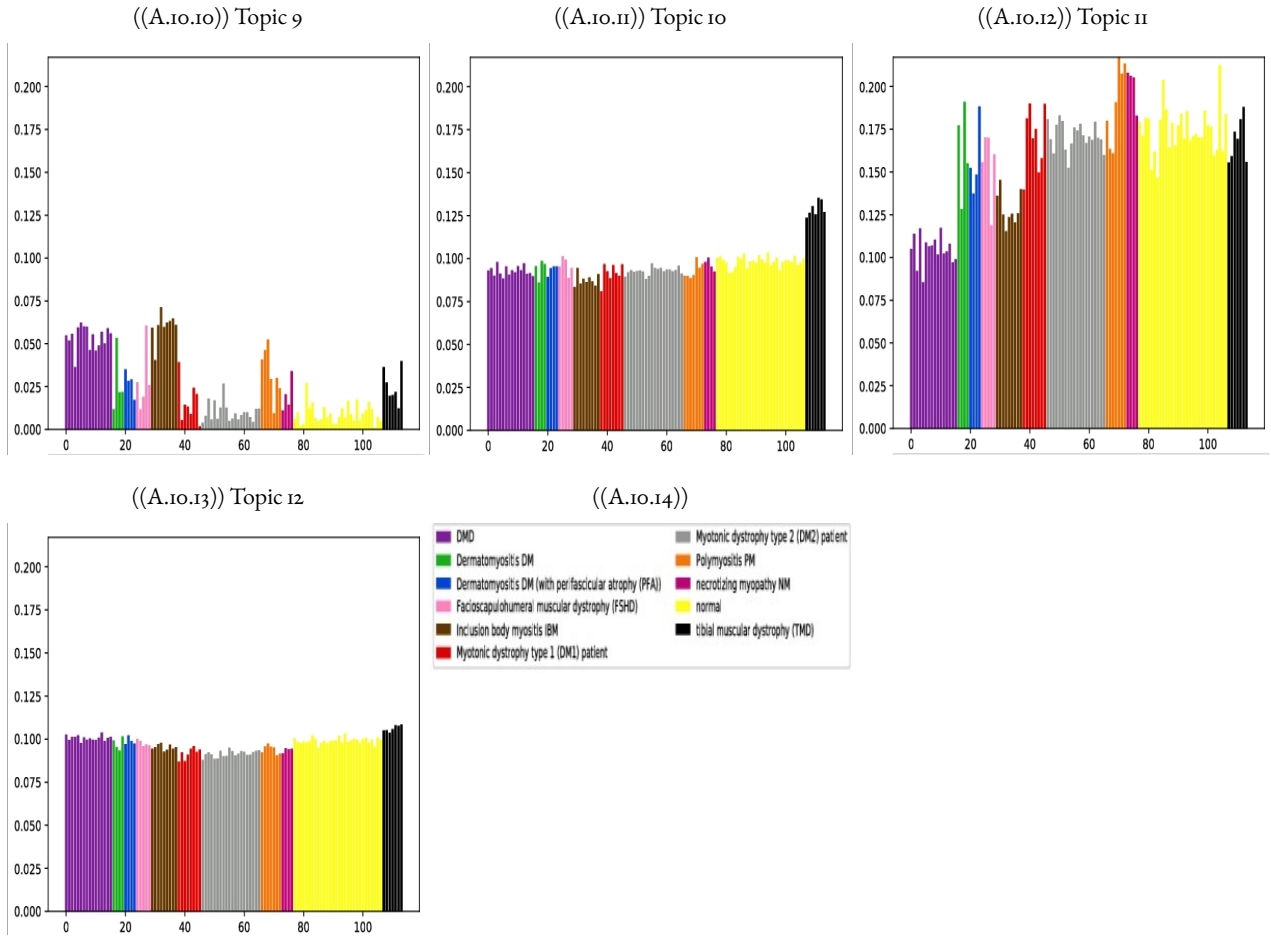


Figure A.10: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0$ Remove Bin $1 = \mathbf{x}_{t_2} = \mathbf{x}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

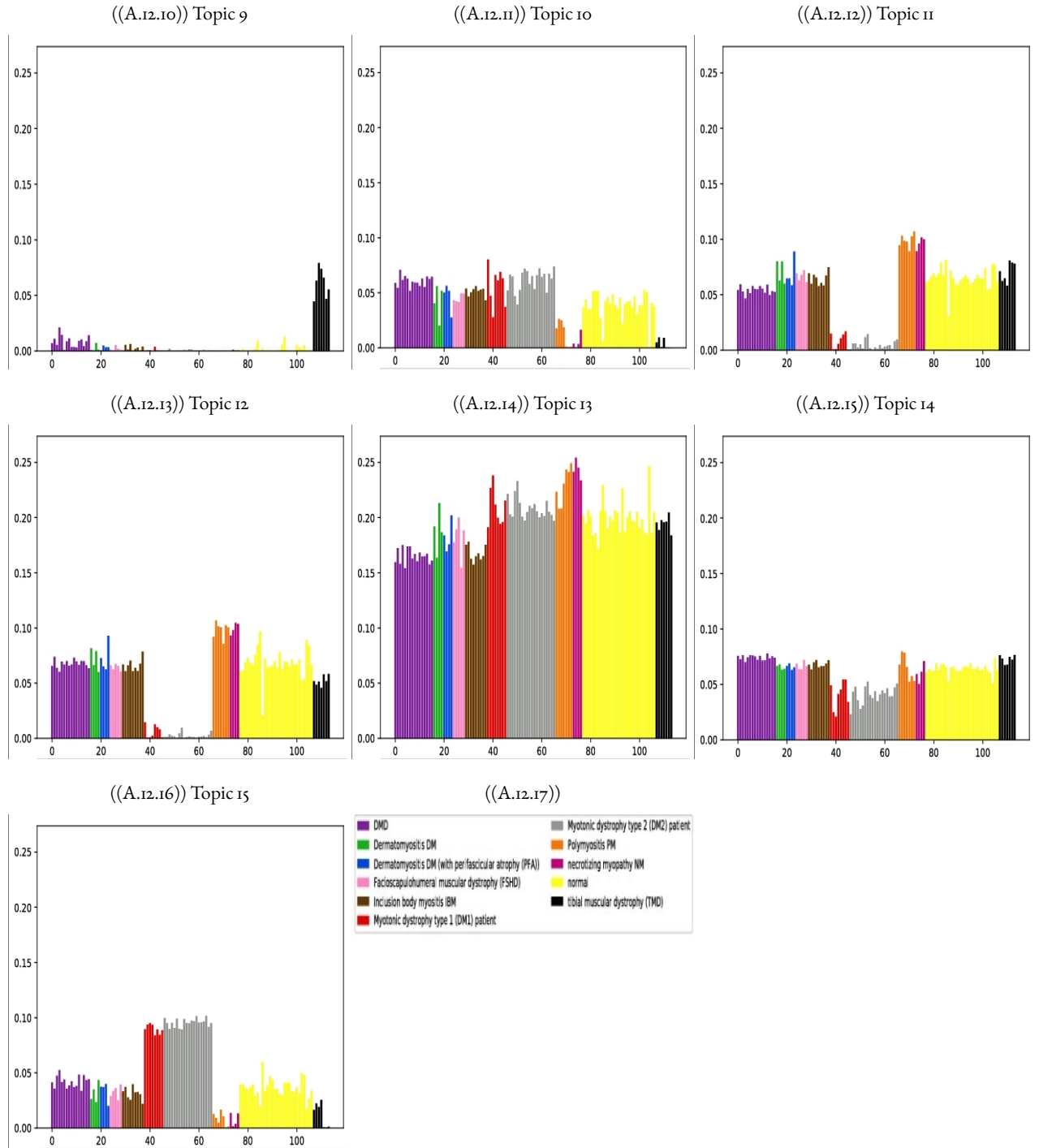


Figure A.12: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0$ Remove Bin $1 = \mathbf{x}_{t_2} = \mathbf{x}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

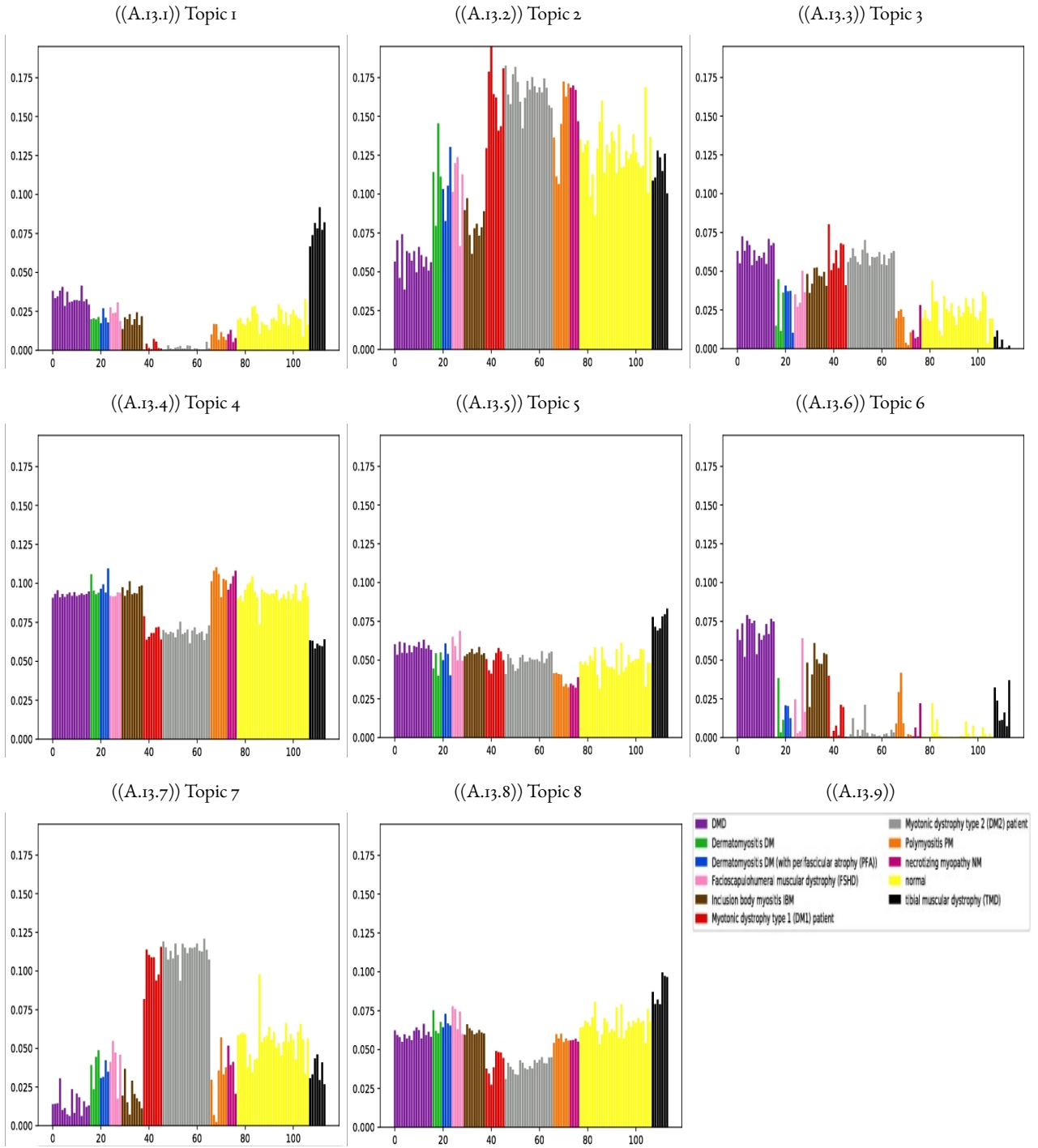


Figure A.13: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0.2$ Remove Bin $\mathbf{1} = \mathbf{X}_{t_2} = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

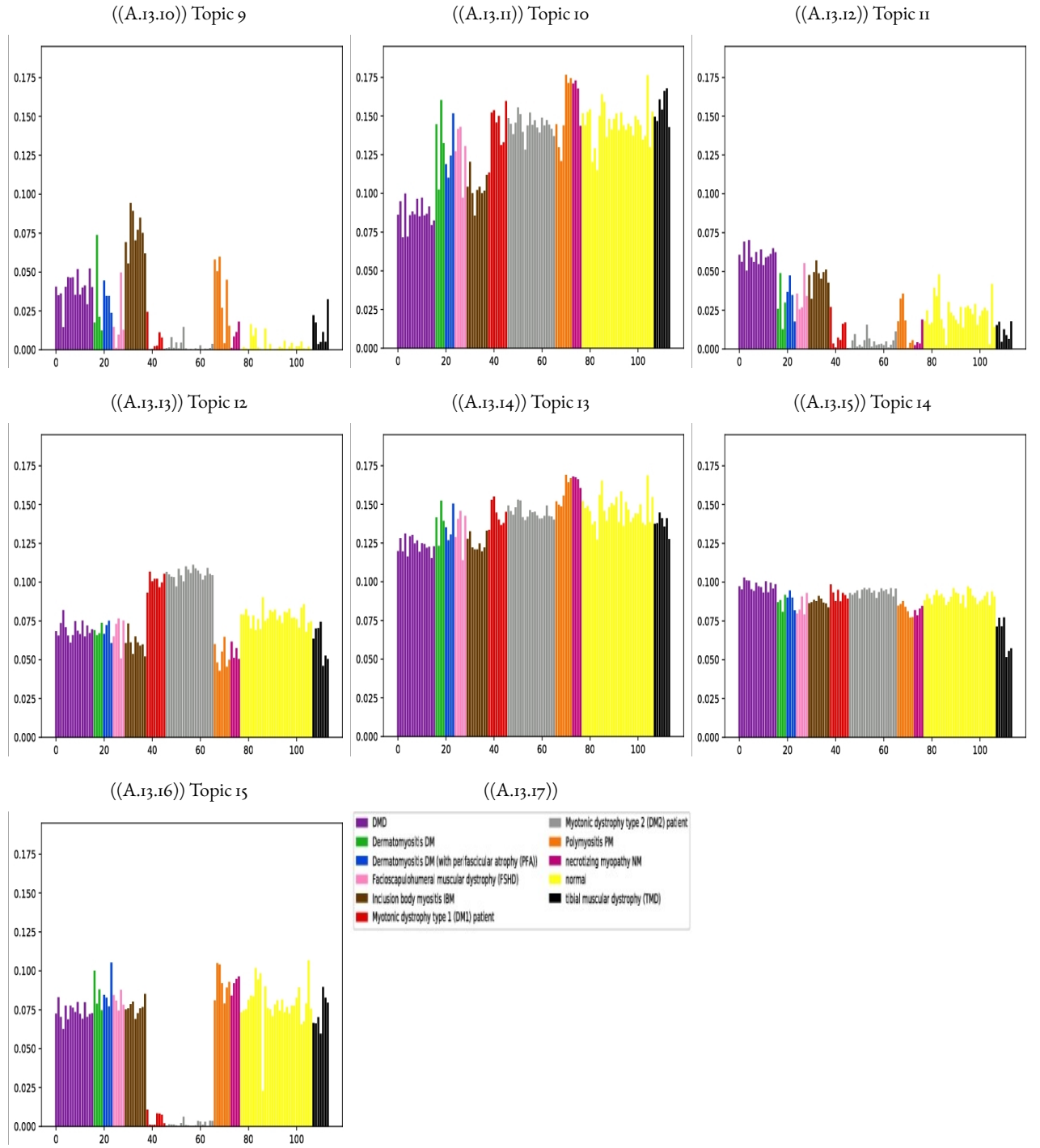


Figure A.13: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0.2$ Remove Bin $1 = \mathbf{x}_{t_2} = \mathbf{x}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

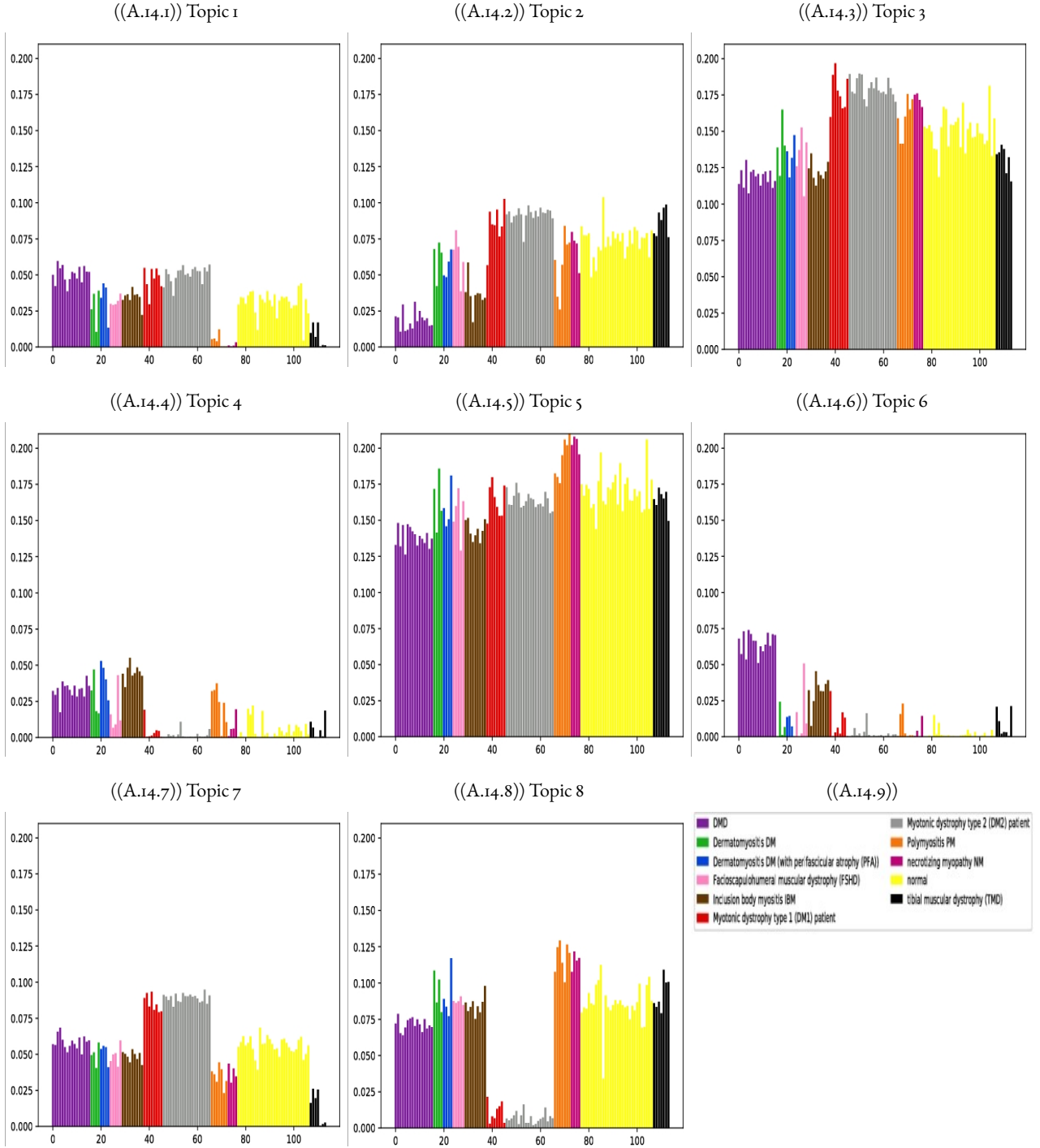


Figure A.14: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0.2$ Remove Bin $1 = \mathbf{x}_{t_2} = 0.3$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

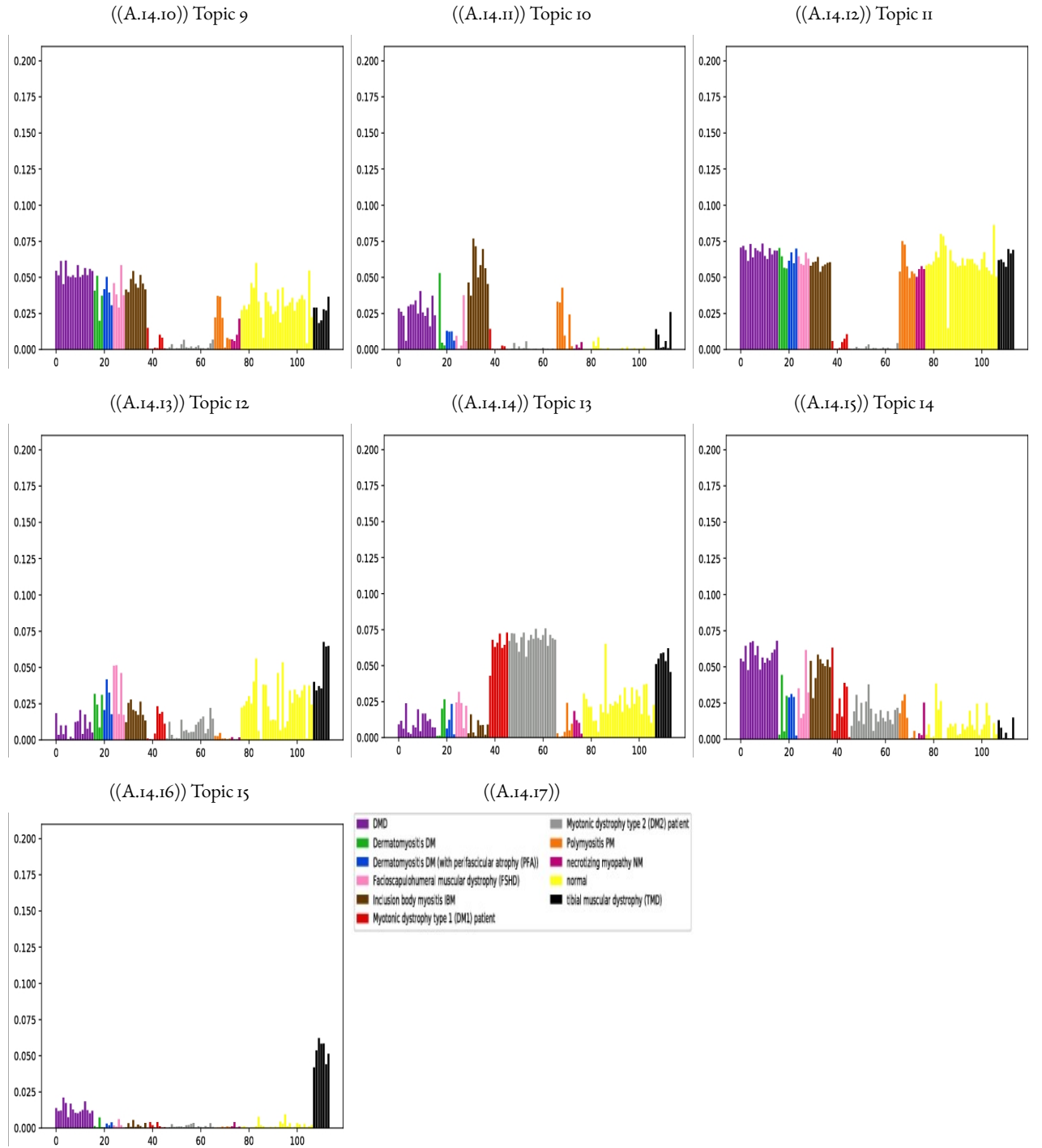


Figure A.14: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0.2$ Remove Bin 1 = $\times t_2 = 0.3$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

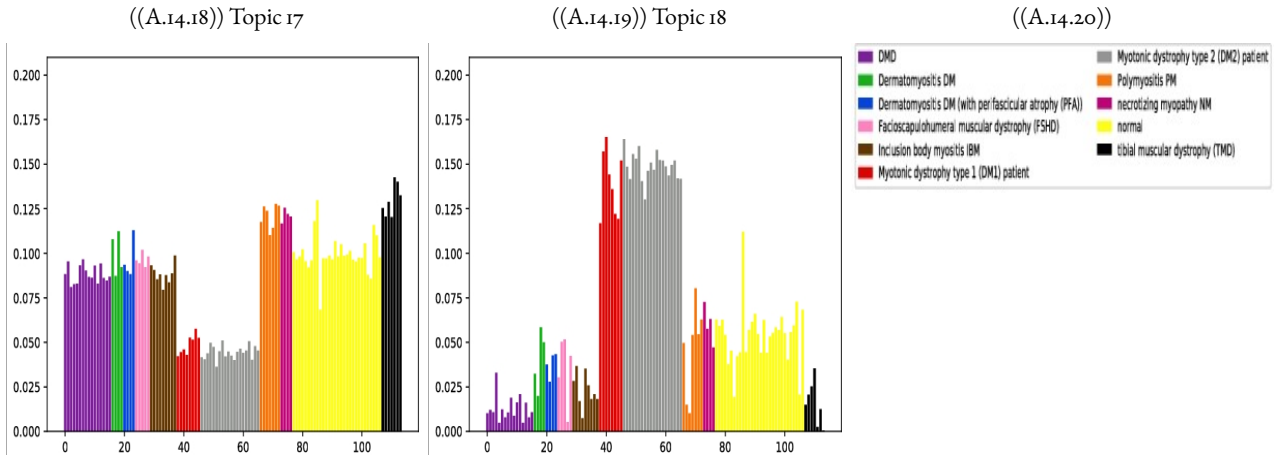


Figure A.14: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0.2$ Remove Bin $1 = \mathbf{X}_{t_2} = 0.3$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

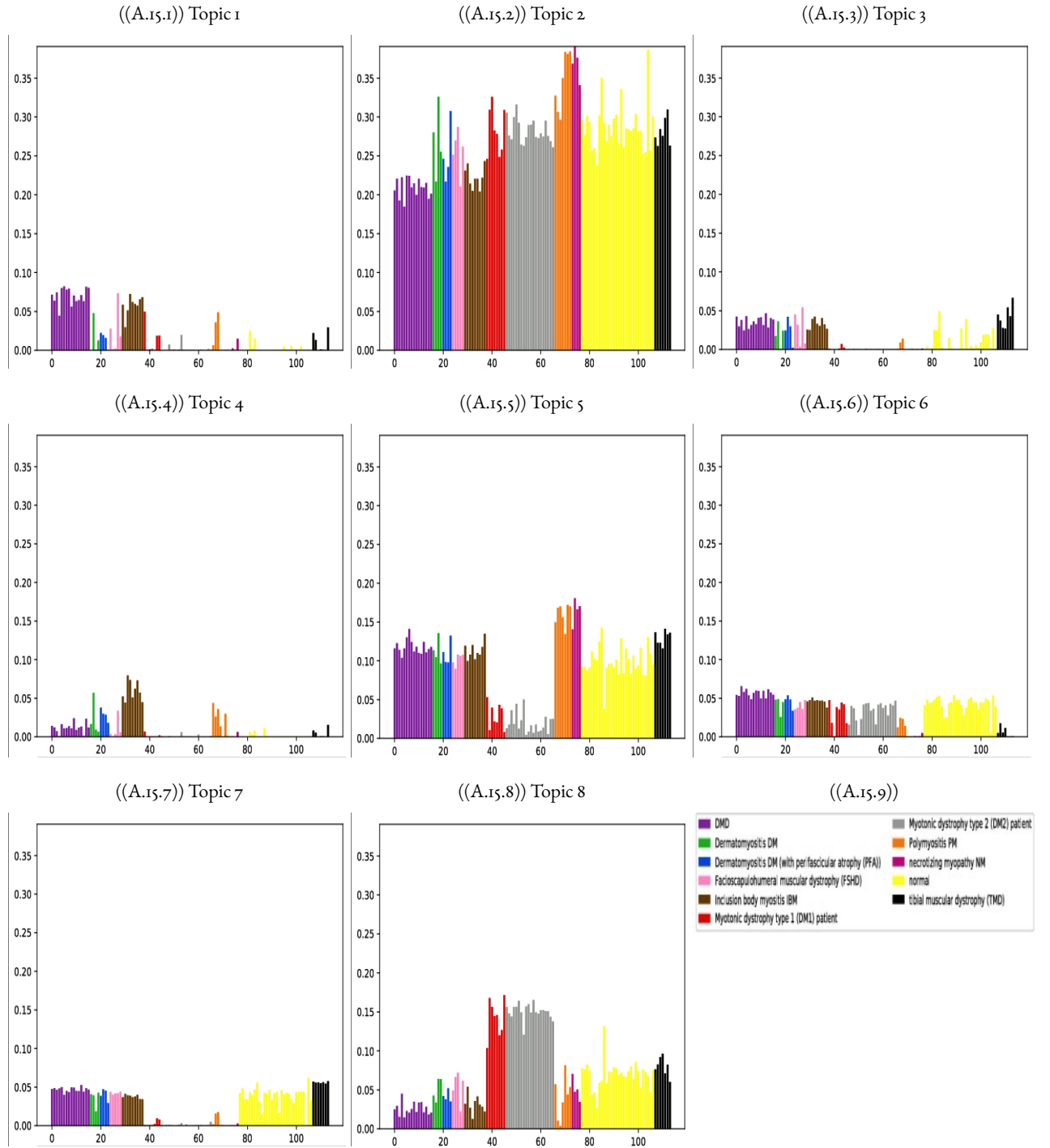


Figure A.15: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0.2$. Remove Bin 1 = $\sqrt{t_2} = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

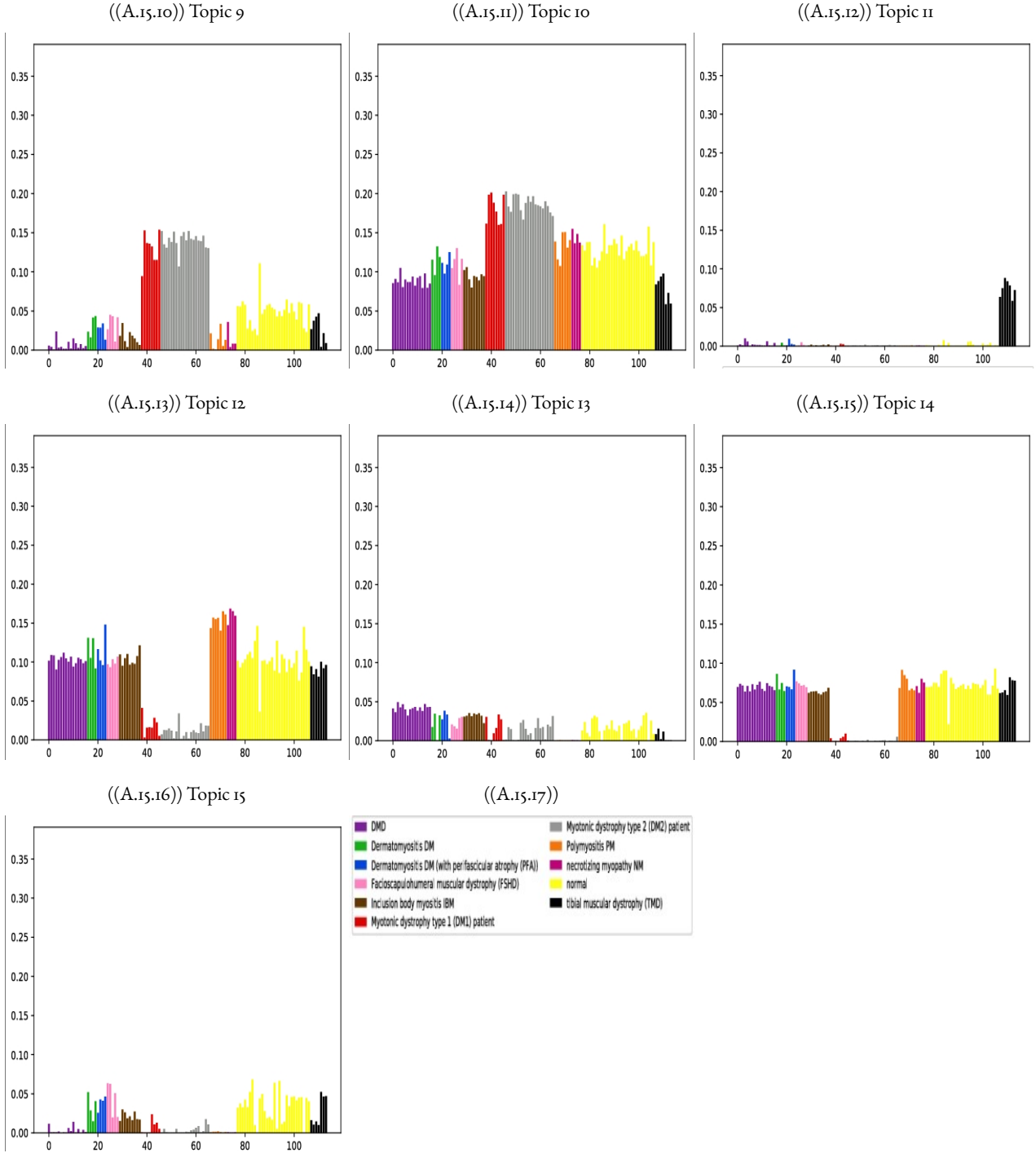


Figure A.15: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0.2$ Remove Bin $1 = \sqrt{t_2} = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

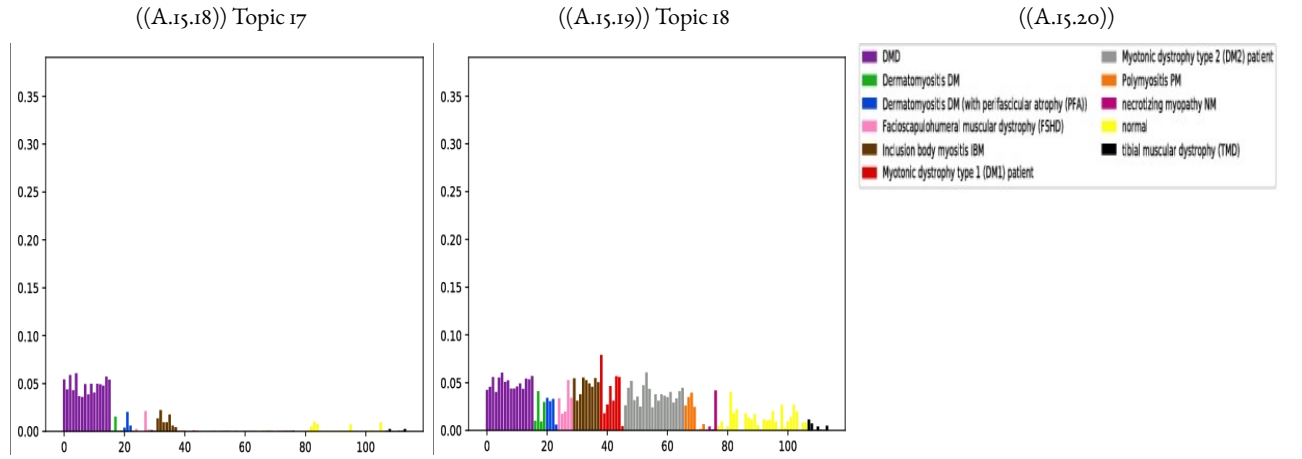


Figure A.15: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0.2$ Remove Bin $1 = \sqrt{t_2} = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

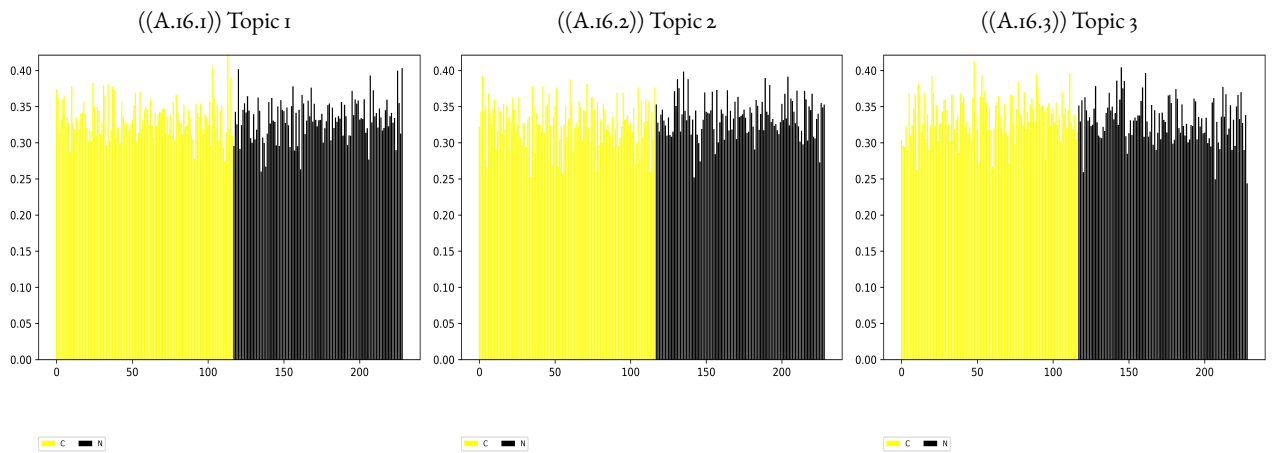


Figure A.16: The distribution over the samples for each topic on the TCGA Dataset obtained by LDA for Repetition with $t_1 = \mathbf{X}$ Remove Bin $1 = \mathbf{X}$ $t_2 = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

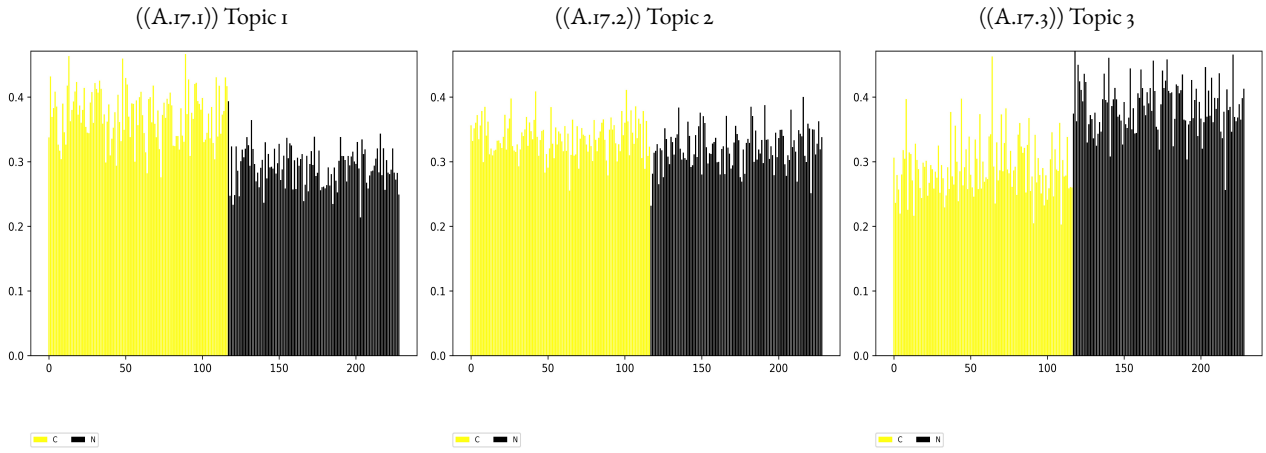


Figure A.17: The distribution over the samples for each topic on the TCGA Dataset obtained by LDA for Repetition with $t_1 = \text{Remove Bin 1} = t_2 = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

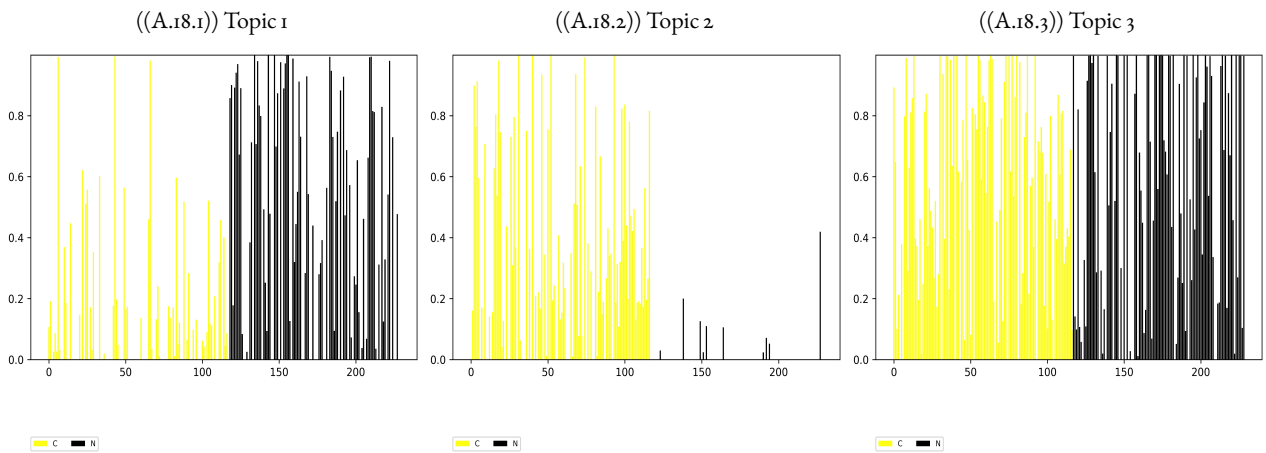


Figure A.18: The distribution over the samples for each topic on the TCGA Dataset obtained by LDA for Repetition with $t_1 = \text{Remove Bin 1} = t_2 = \text{}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

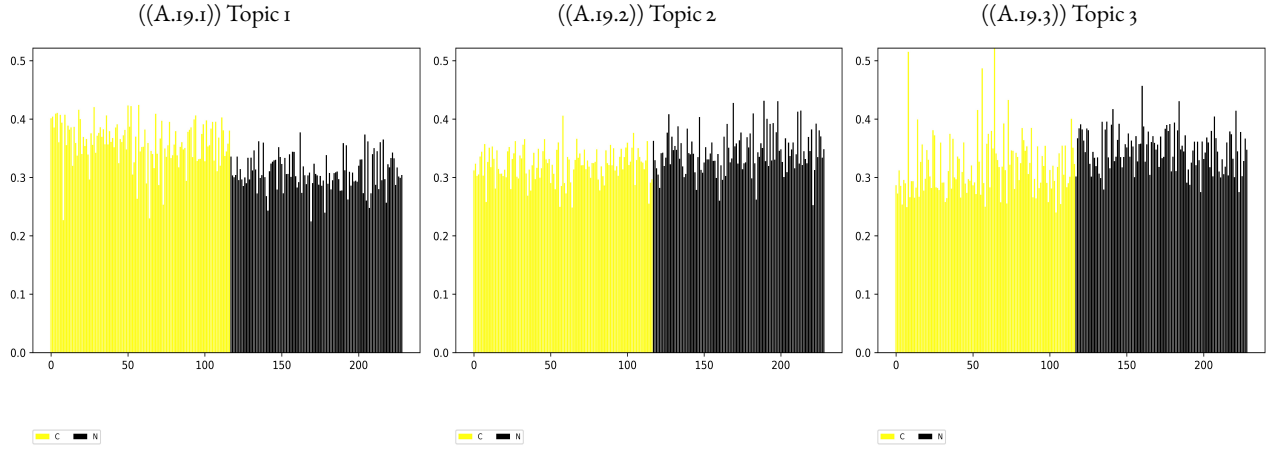


Figure A.19: The distribution over the samples for each topic on the TCGA Dataset obtained by LDA for Repetition with $t_1 = 0.2$ Remove Bin 1 = $\mathbf{X}t_2 = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

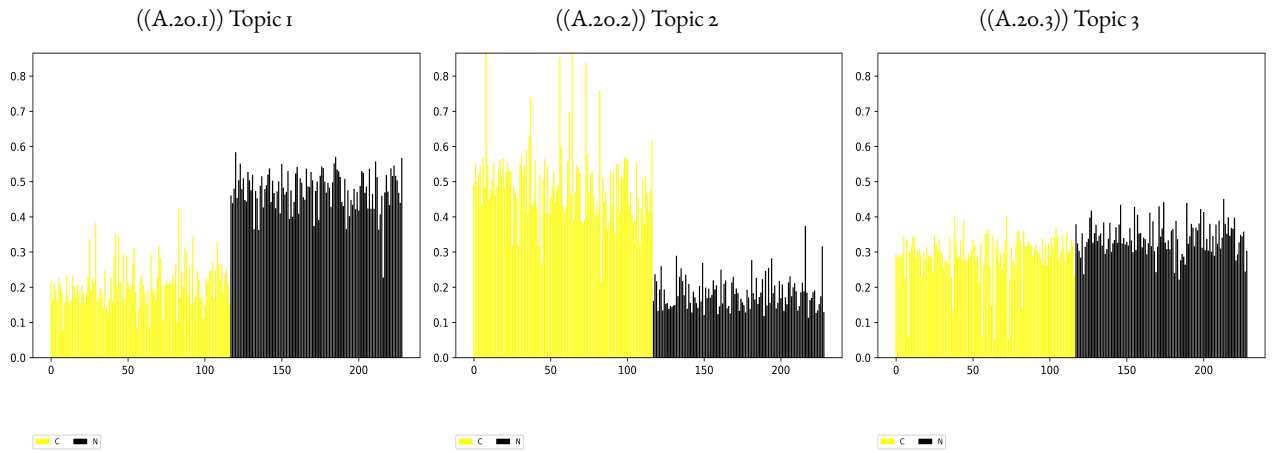


Figure A.20: The distribution over the samples for each topic on the TCGA Dataset obtained by LDA for Repetition with $t_1 = 0.2$ Remove Bin 1 = $\mathbf{X}t_2 = 0.3$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

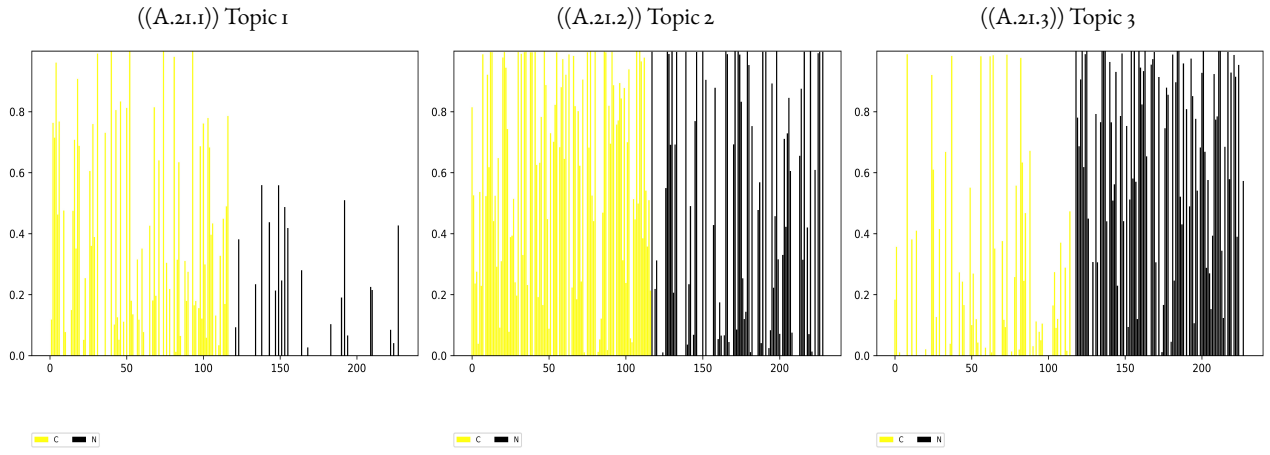


Figure A.21: The distribution over the samples for each topic on the TCGA Dataset obtained by LDA for Repetition with $t_1 = 0.2$ Remove Bin 1 $= \sqrt{t_2} = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

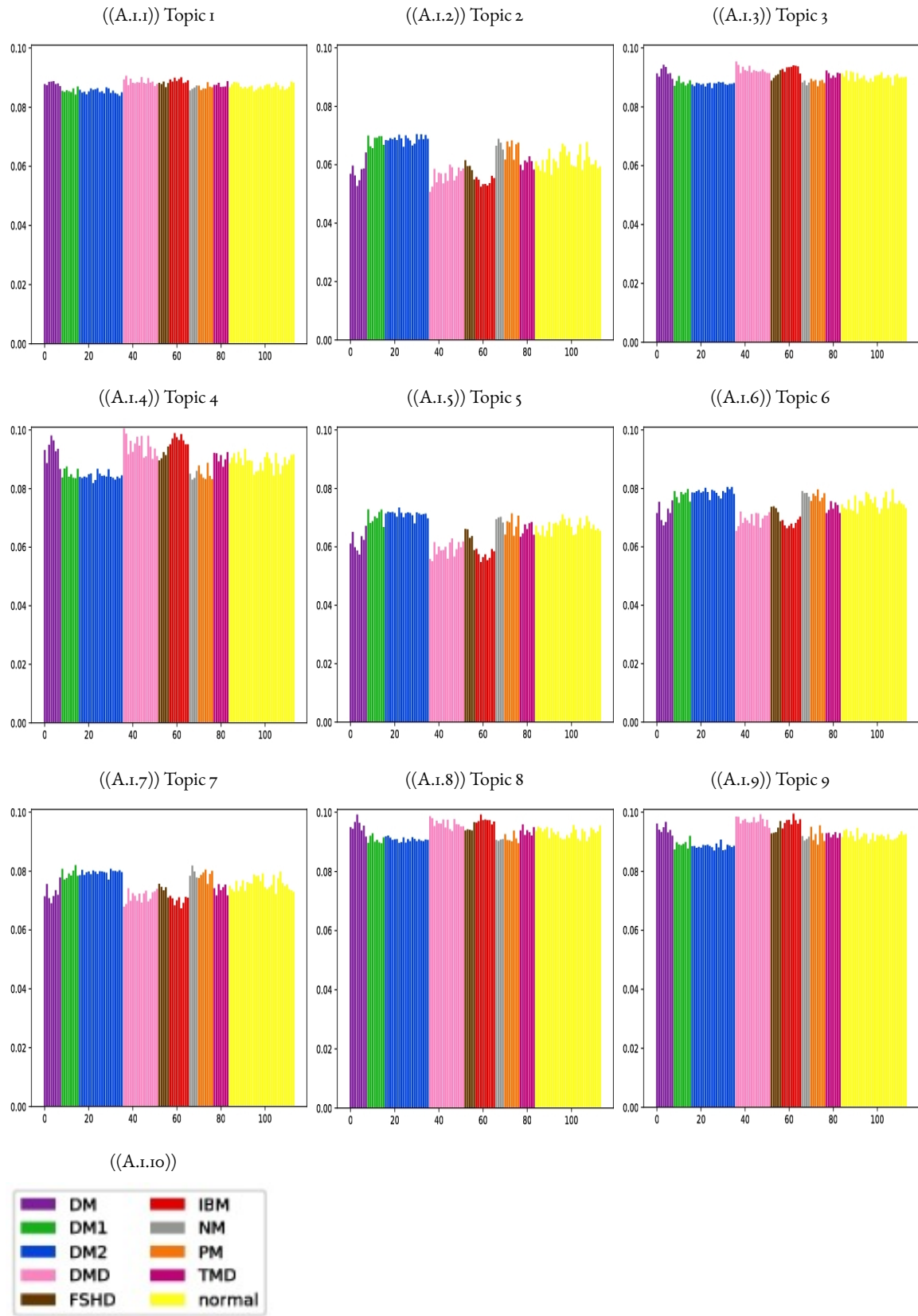


Figure A.1: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Bibin $t_B = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

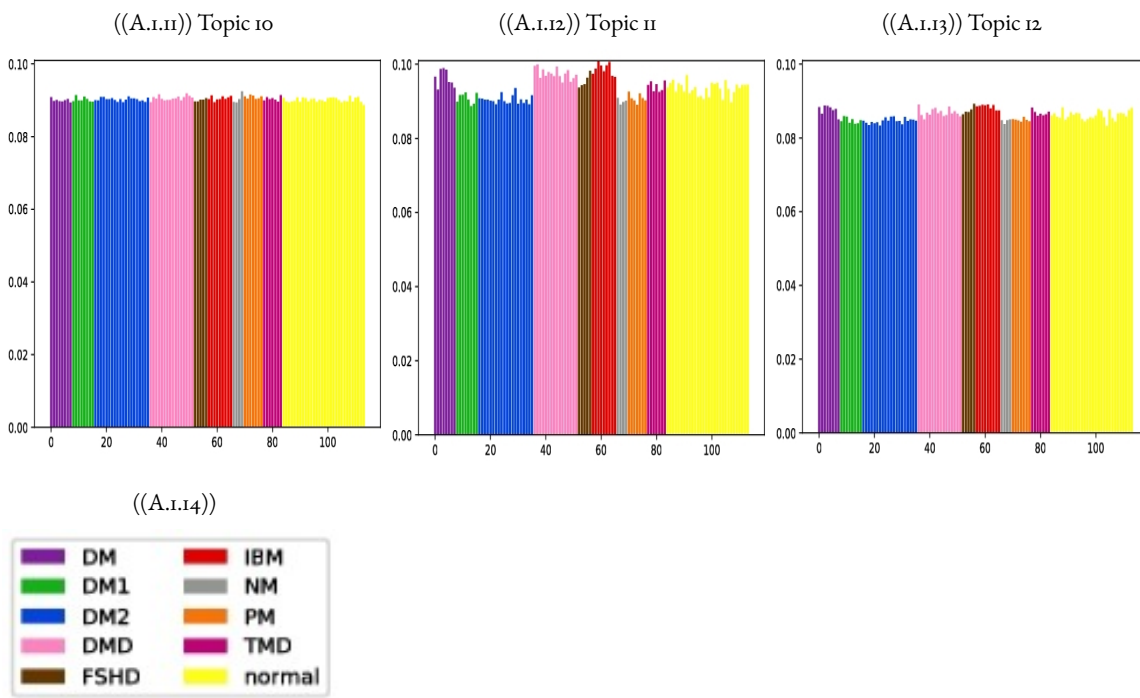


Figure A.1: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Bibin $t_B = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

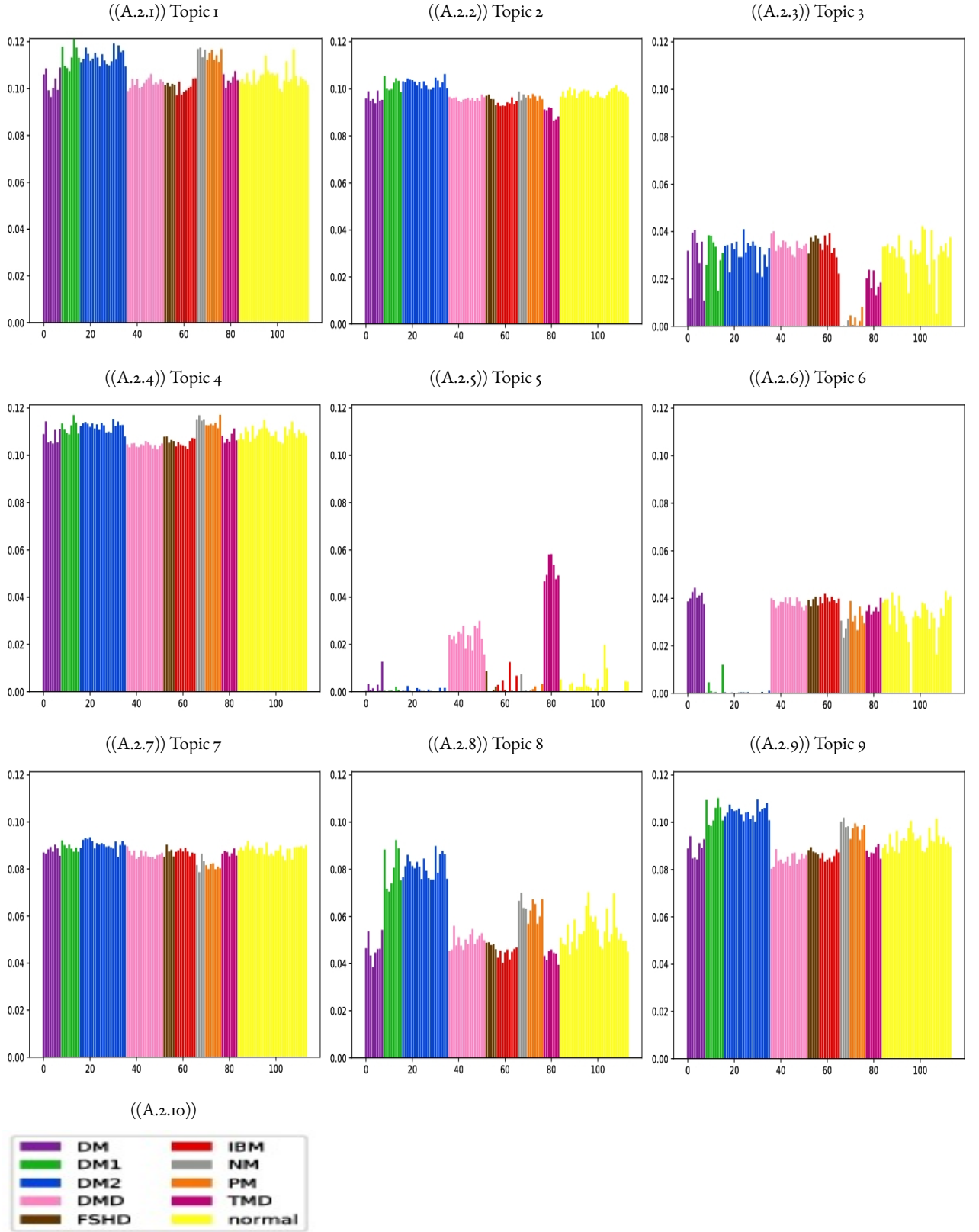


Figure A.2: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Bibin $t_B = 0.0$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

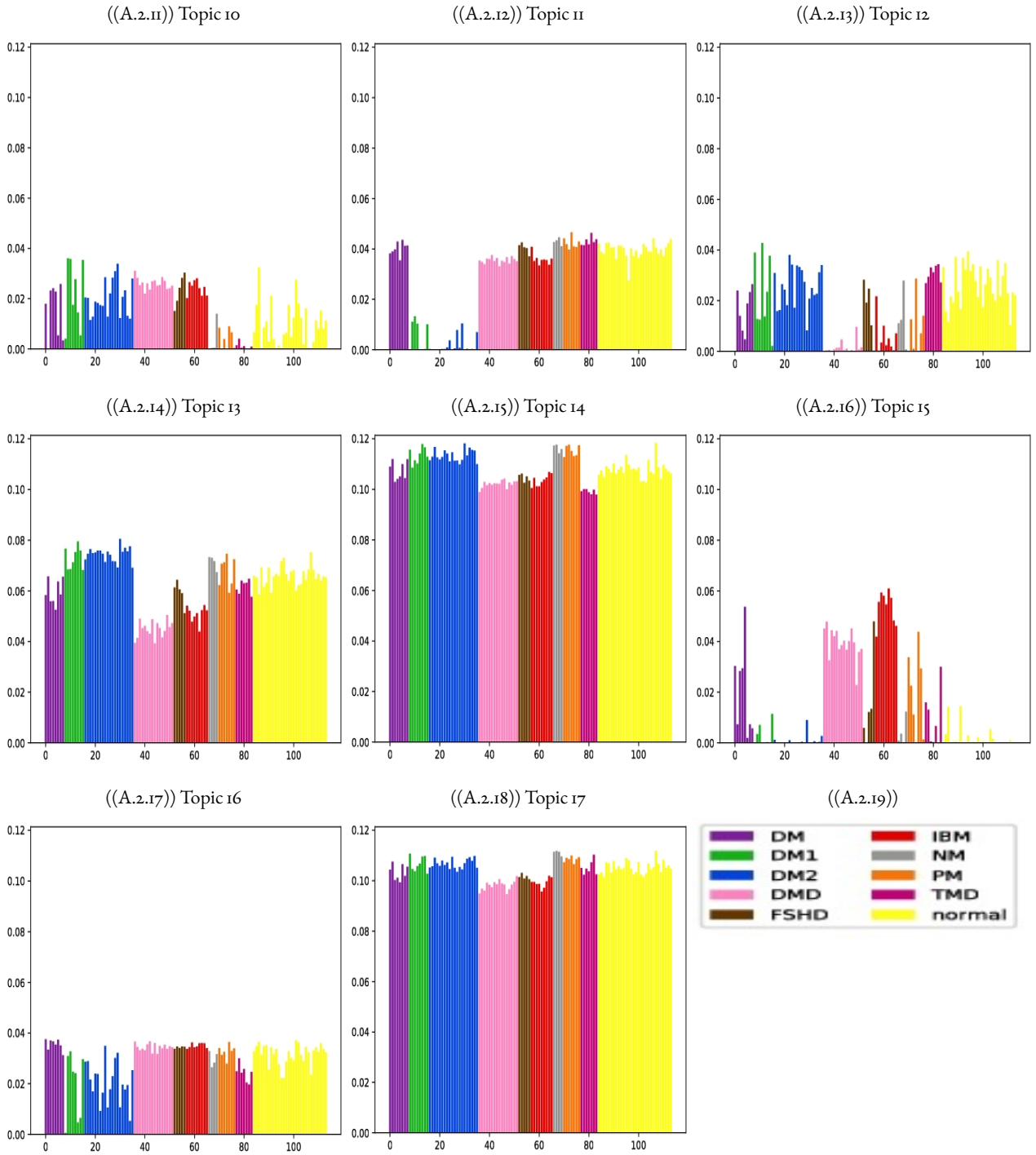


Figure A.2: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Bibin $t_B = 0.0$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

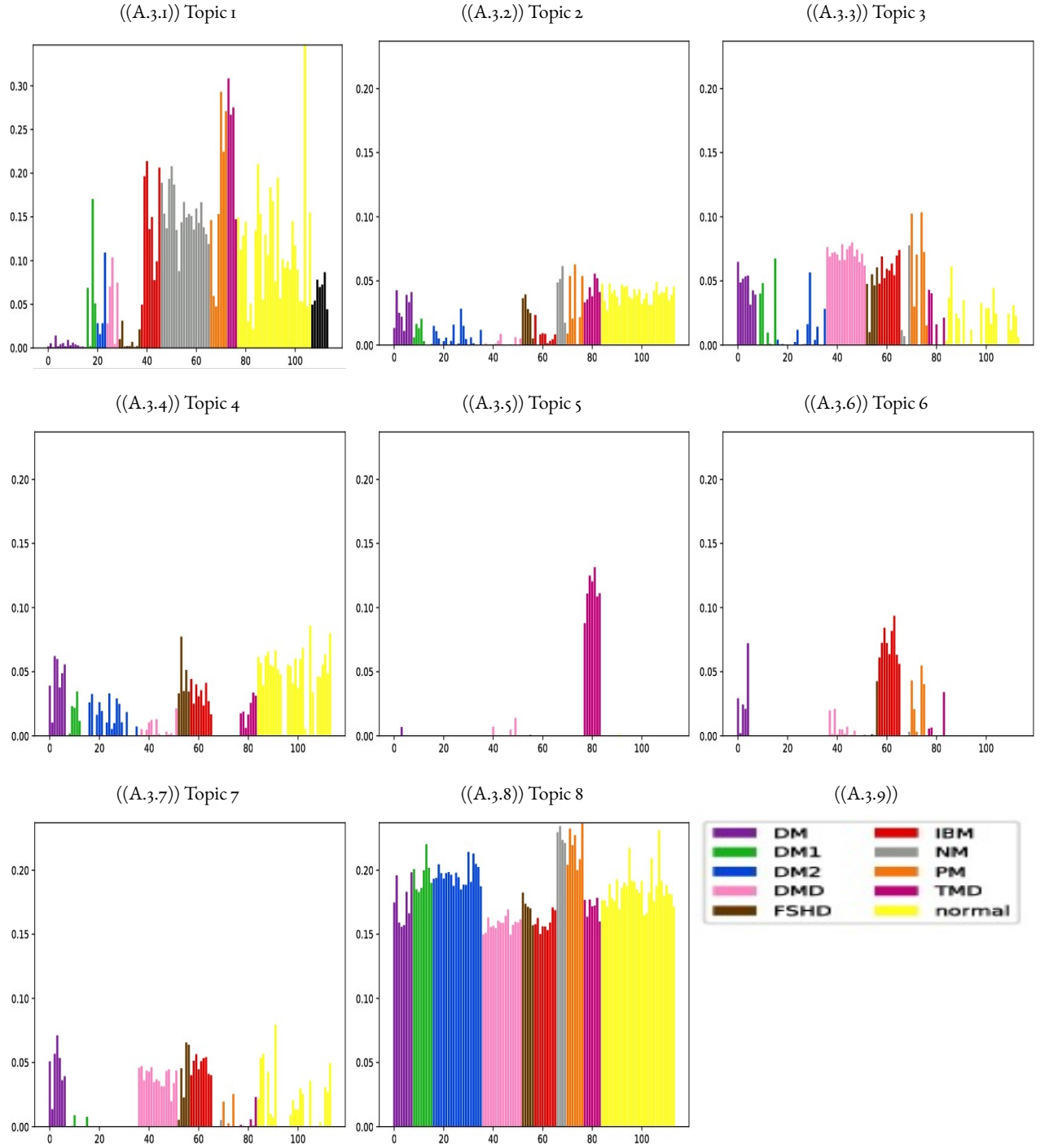


Figure A.3: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Bibin $t_B = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

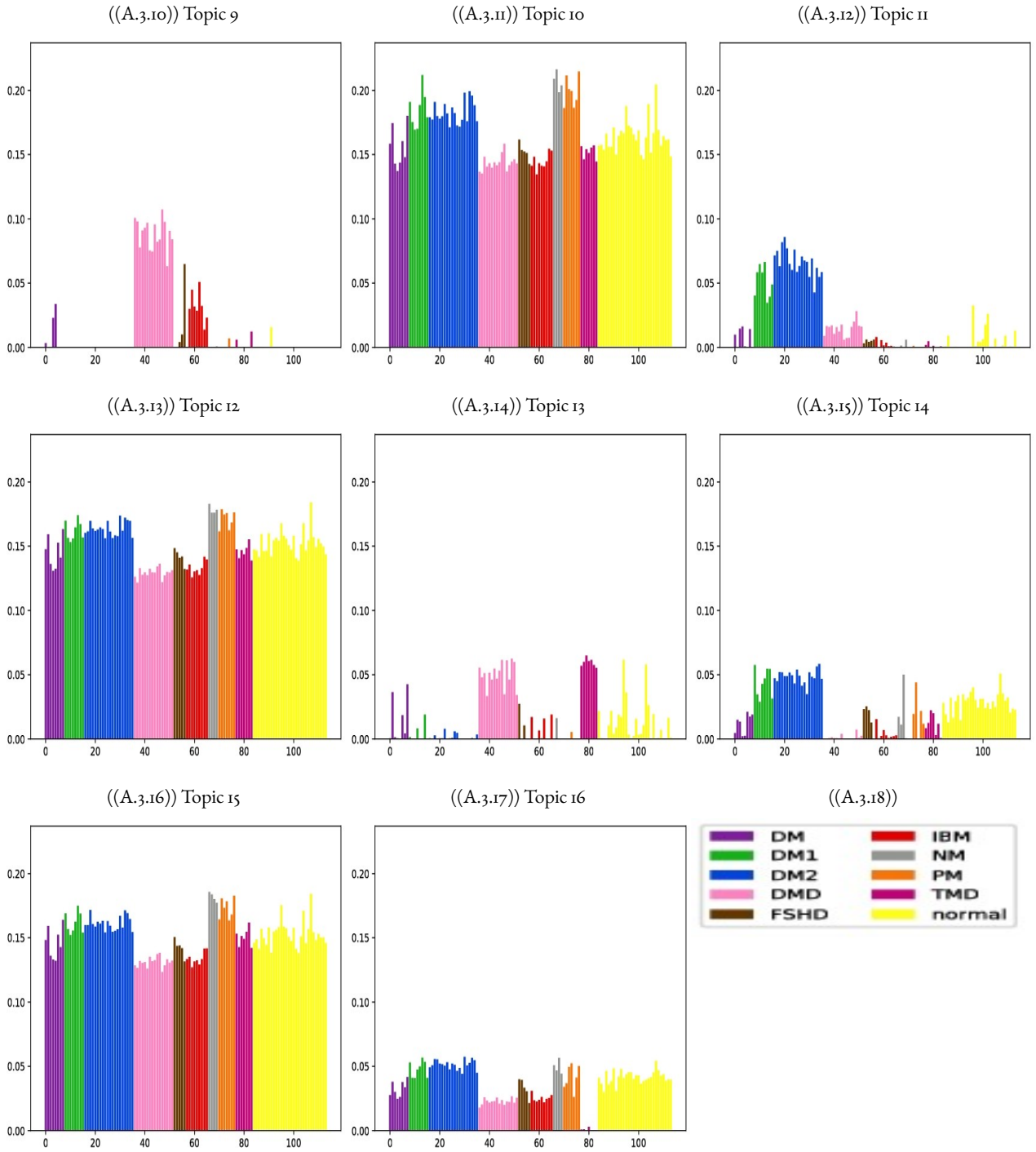


Figure A.3: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Bibin $t_B = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

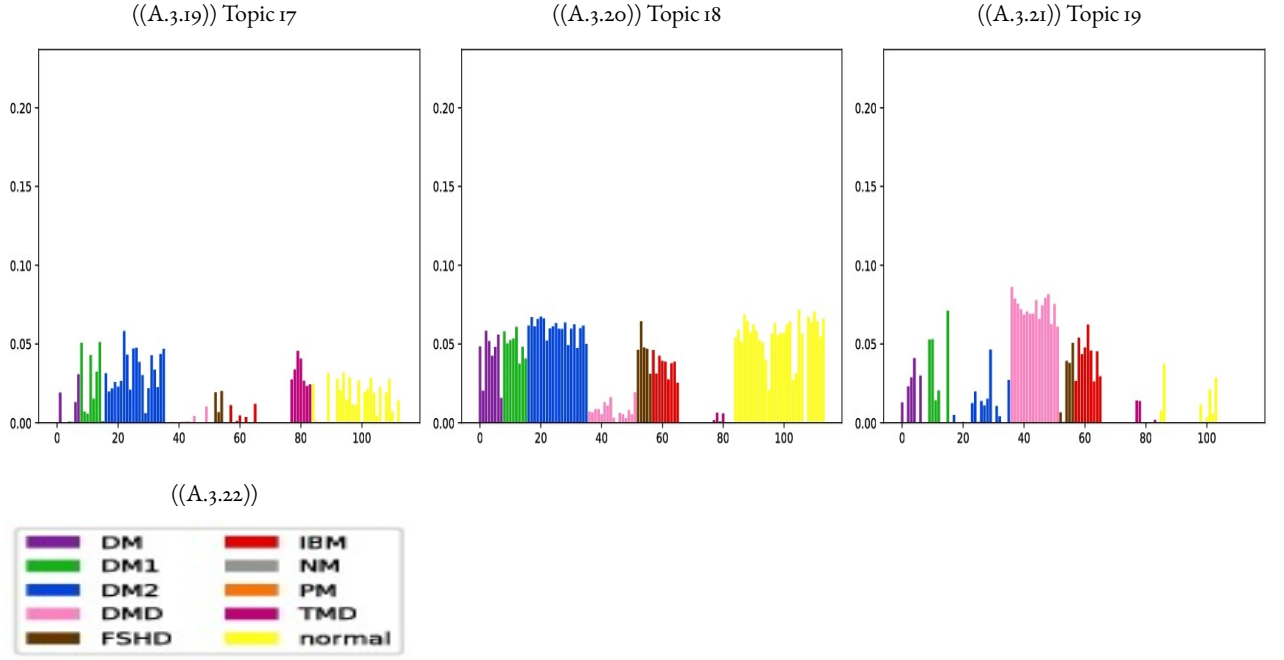


Figure A.3: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Bibin $t_B = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

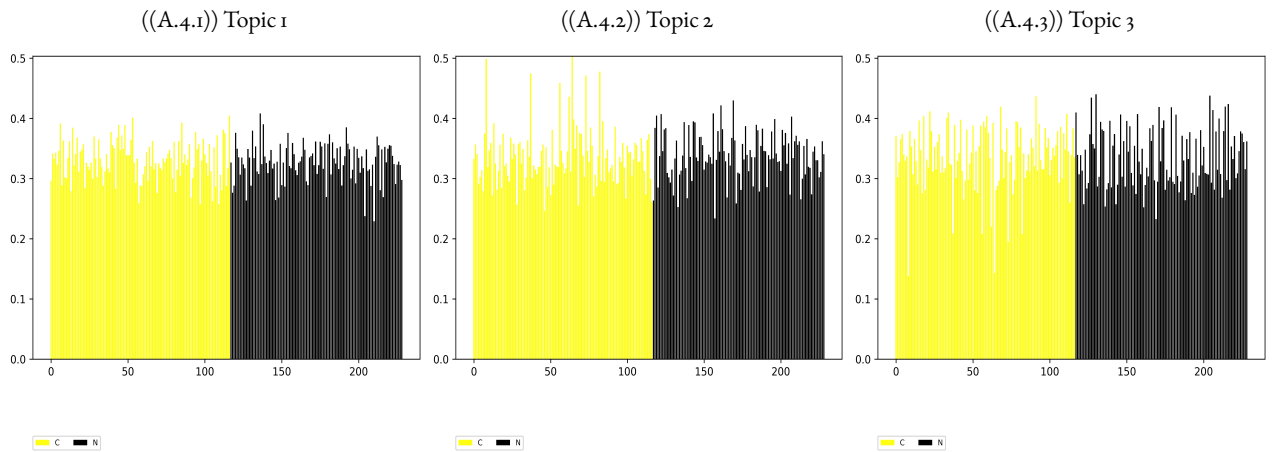


Figure A.4: The distribution over the samples for each topic on the TCGA Dataset obtained by LDA for Bibin with $t_B = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

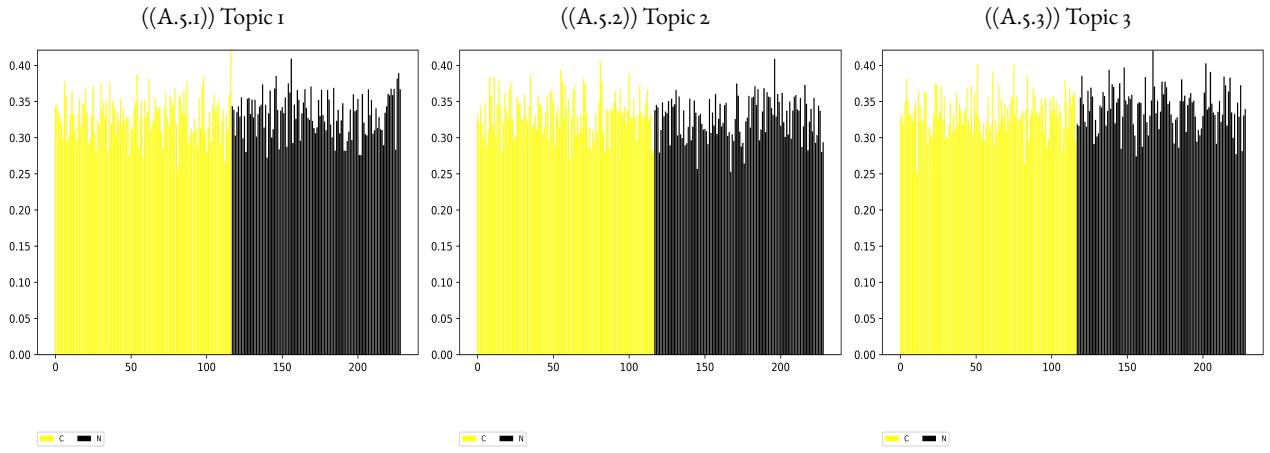


Figure A.5: The distribution over the samples for each topic on the TCGA Dataset obtained by LDA for Bibin with $t_B = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

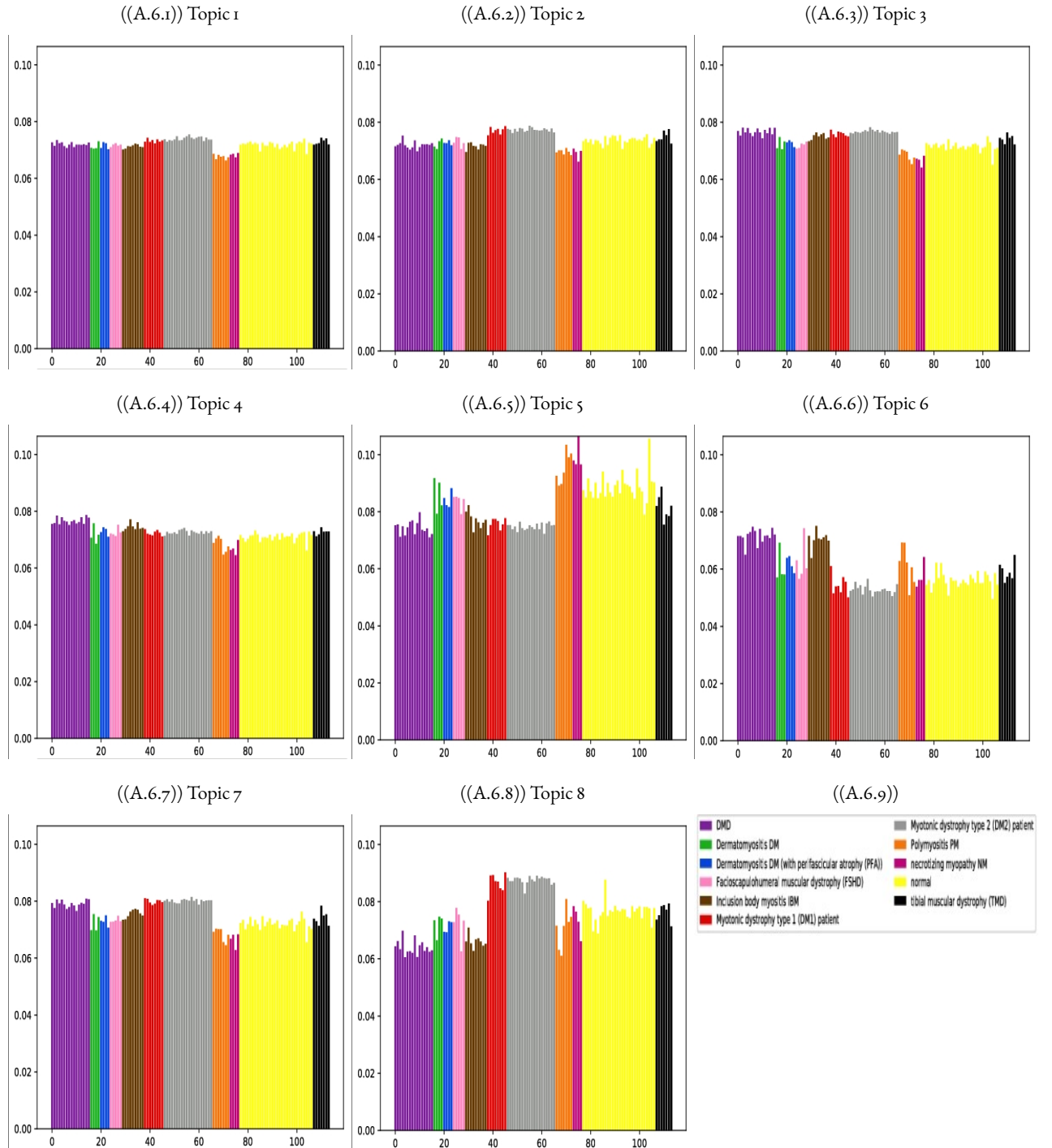


Figure A.6: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = \mathbf{X}$ Remove Bin 1 $= \mathbf{X}$ $t_2 = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

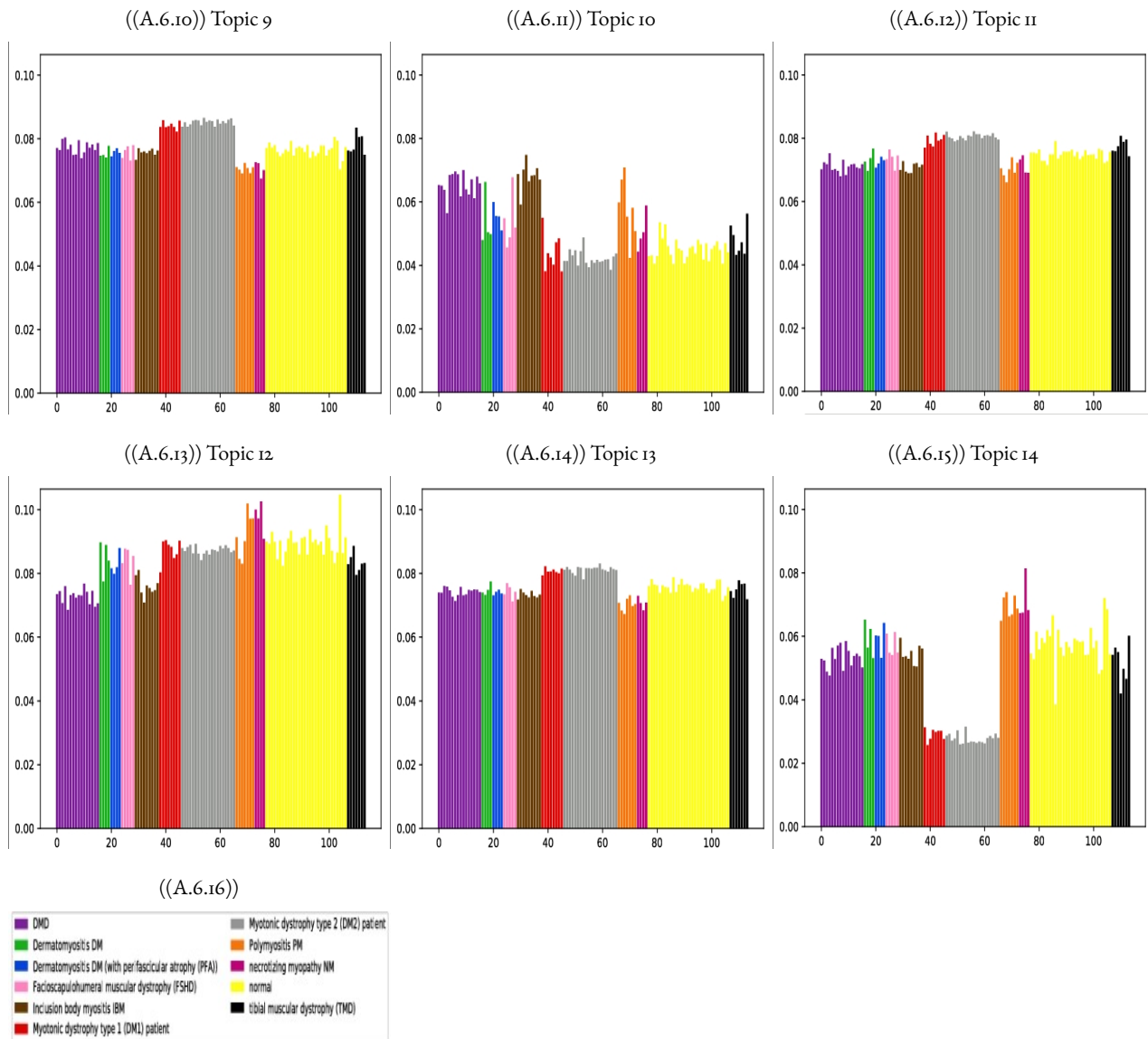


Figure A.6: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = \mathbf{X}$ Remove Bin 1 $= \mathbf{X} t_2 = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

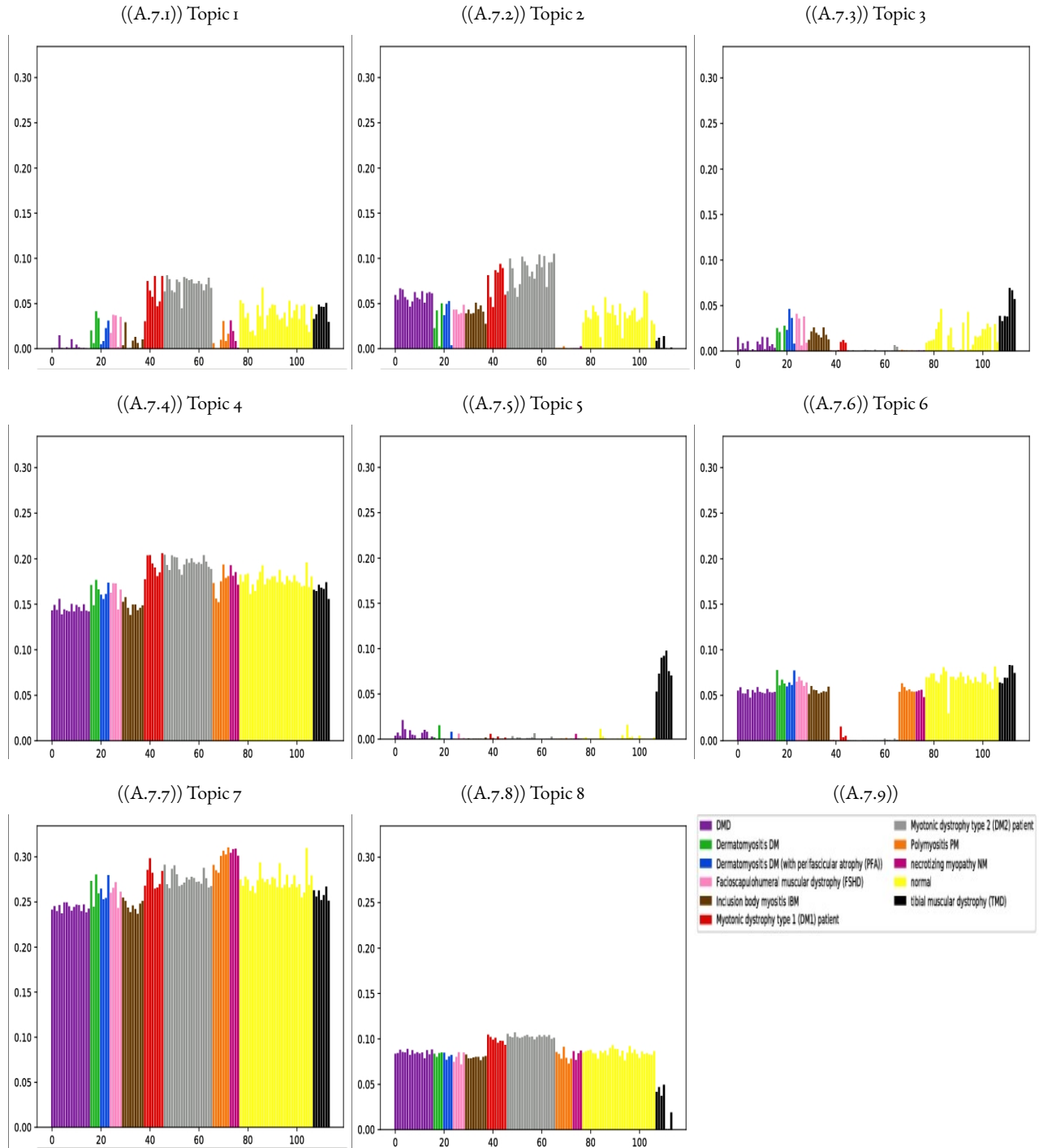


Figure A.7: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = \text{Remove Bin } 1 = \text{Remove Bin } 2 = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

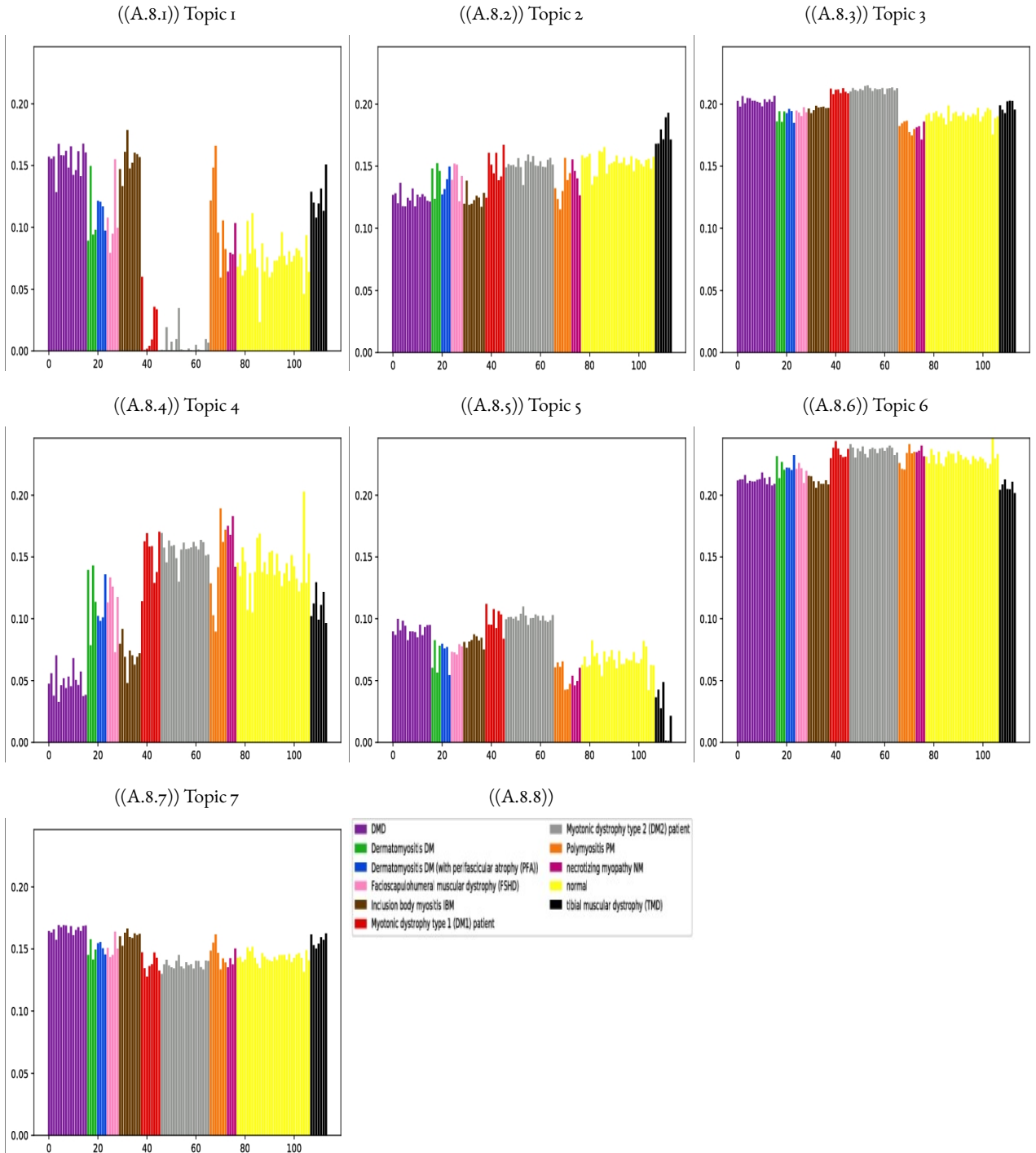


Figure A.8: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = \text{Remove Bin } 1 = \sqrt{t_2} = 0.0$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

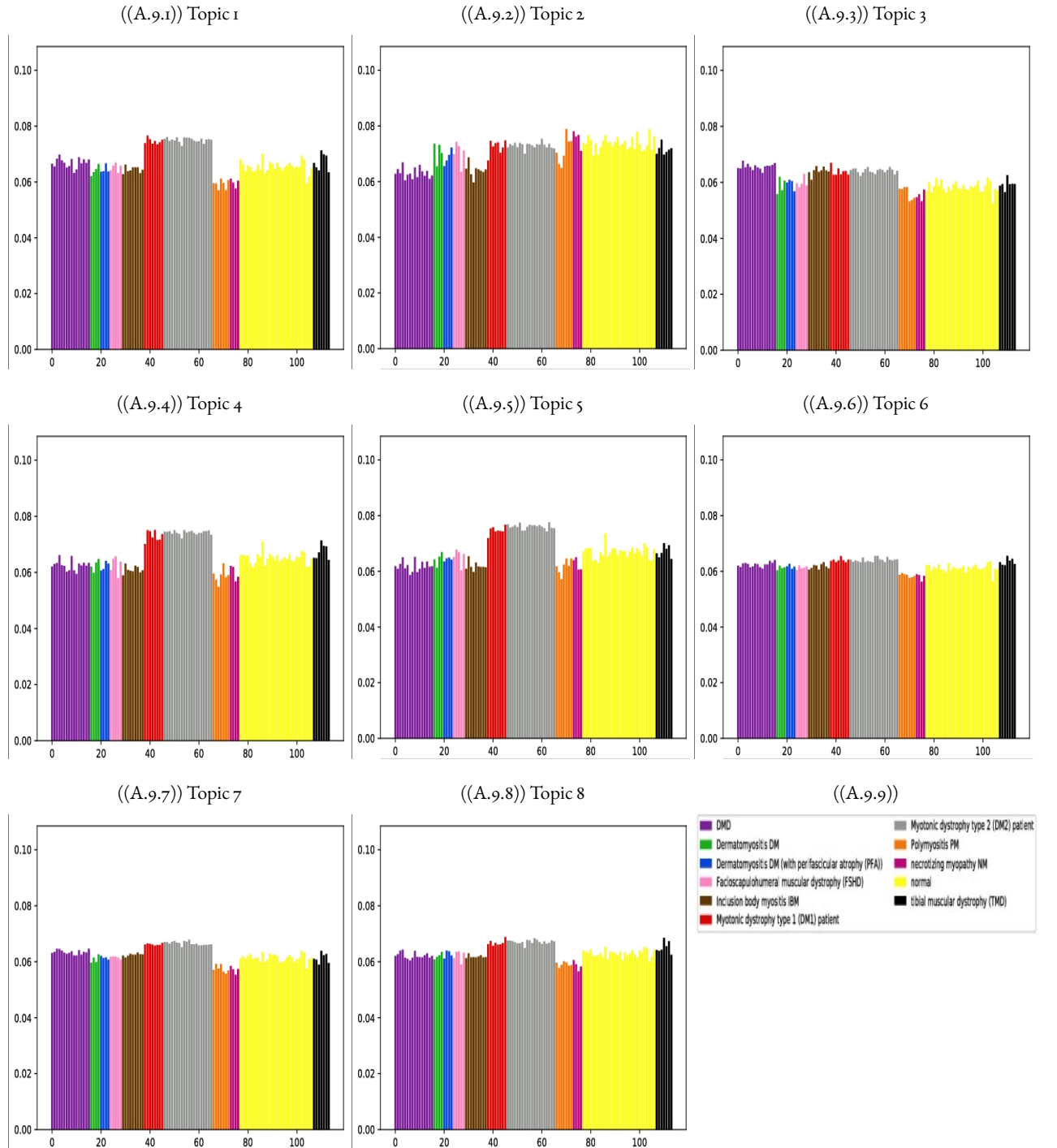


Figure A.9: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = \text{Remove Bin 1} = \sqrt{t_2} = \text{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

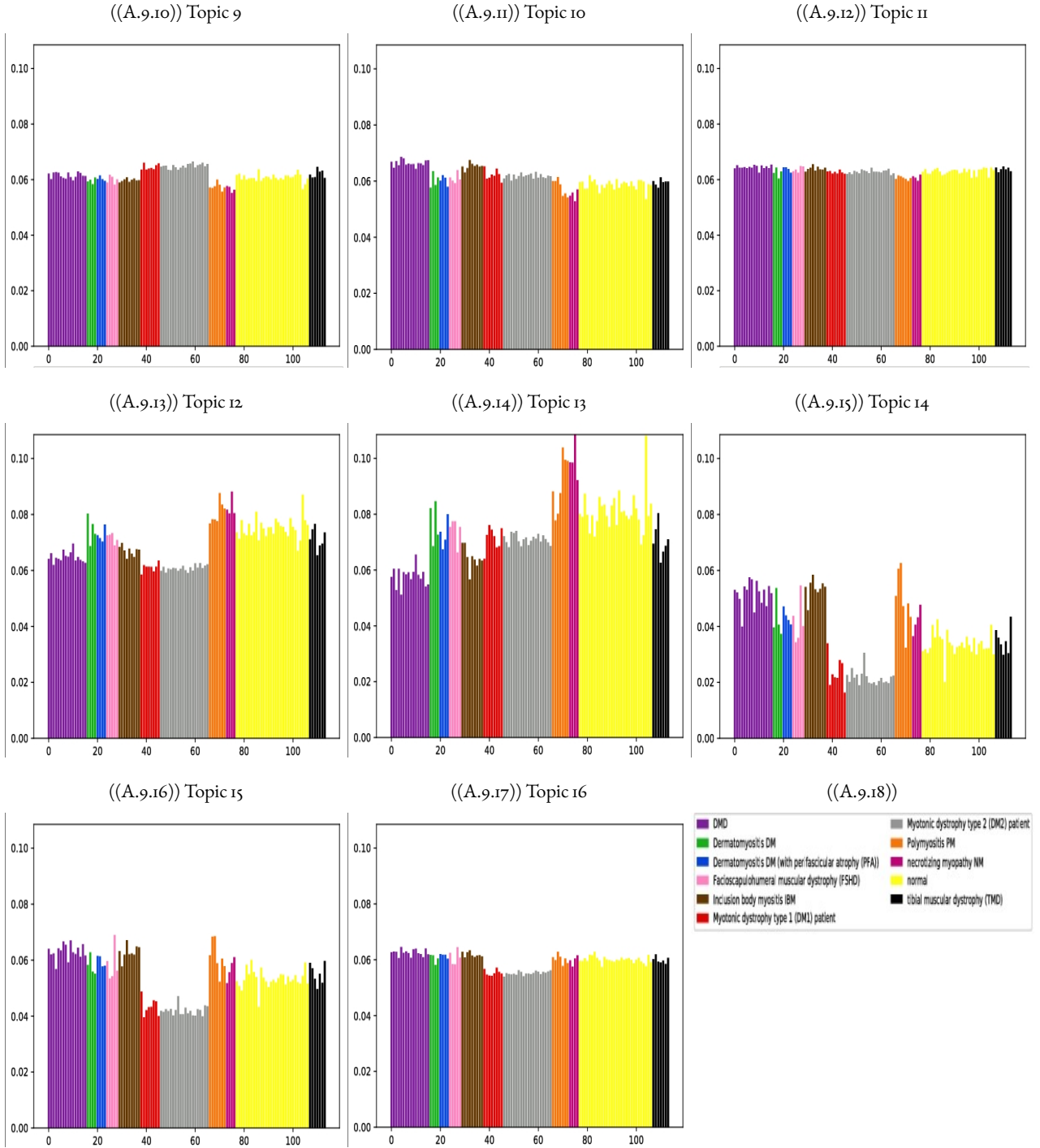


Figure A.9: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = \text{X}$ Remove Bin $t_1 = \sqrt{t_2} = \text{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

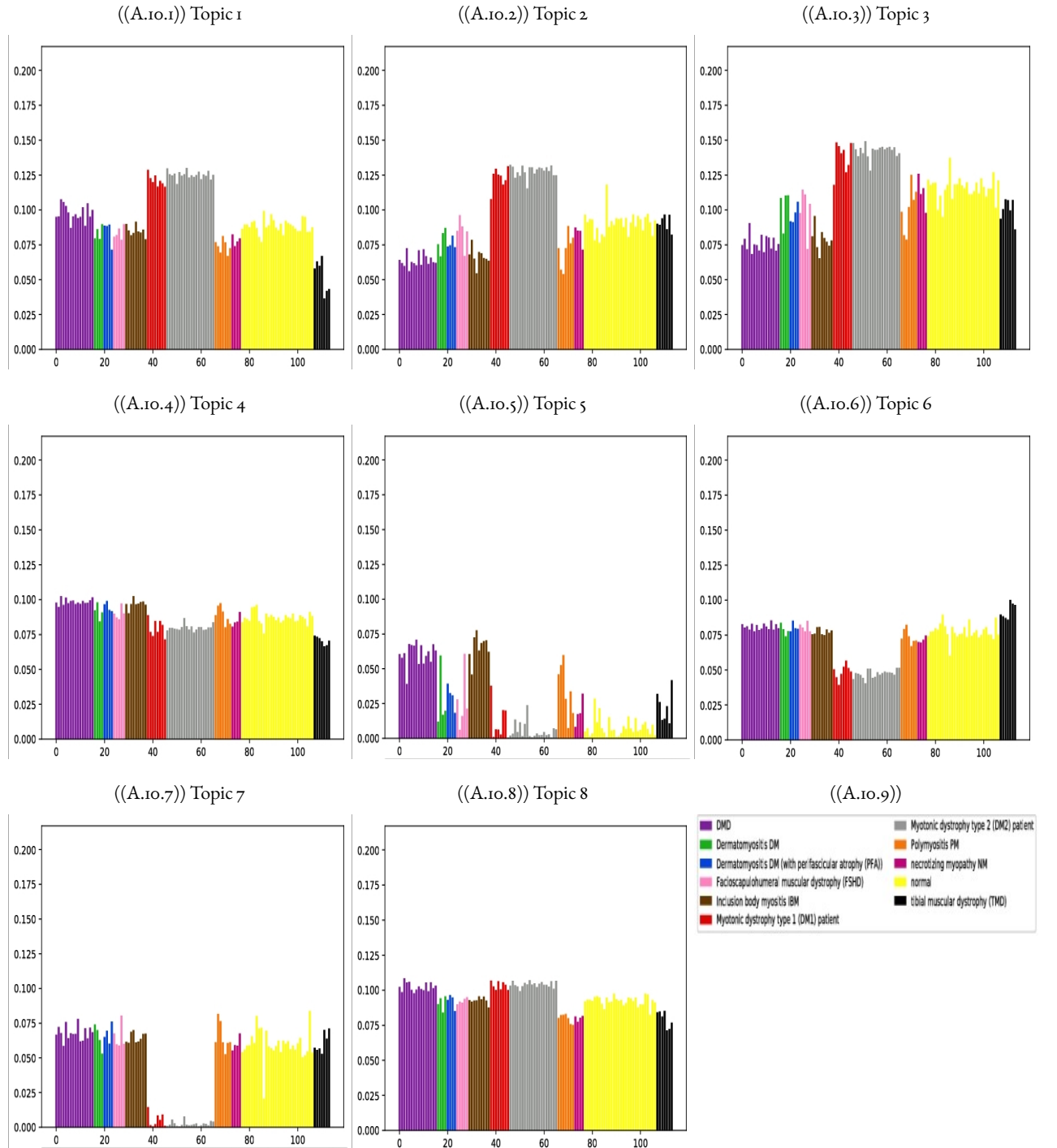


Figure A.10: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0$ Remove Bin $1 = \mathbf{x}_{t_2} = \mathbf{x}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

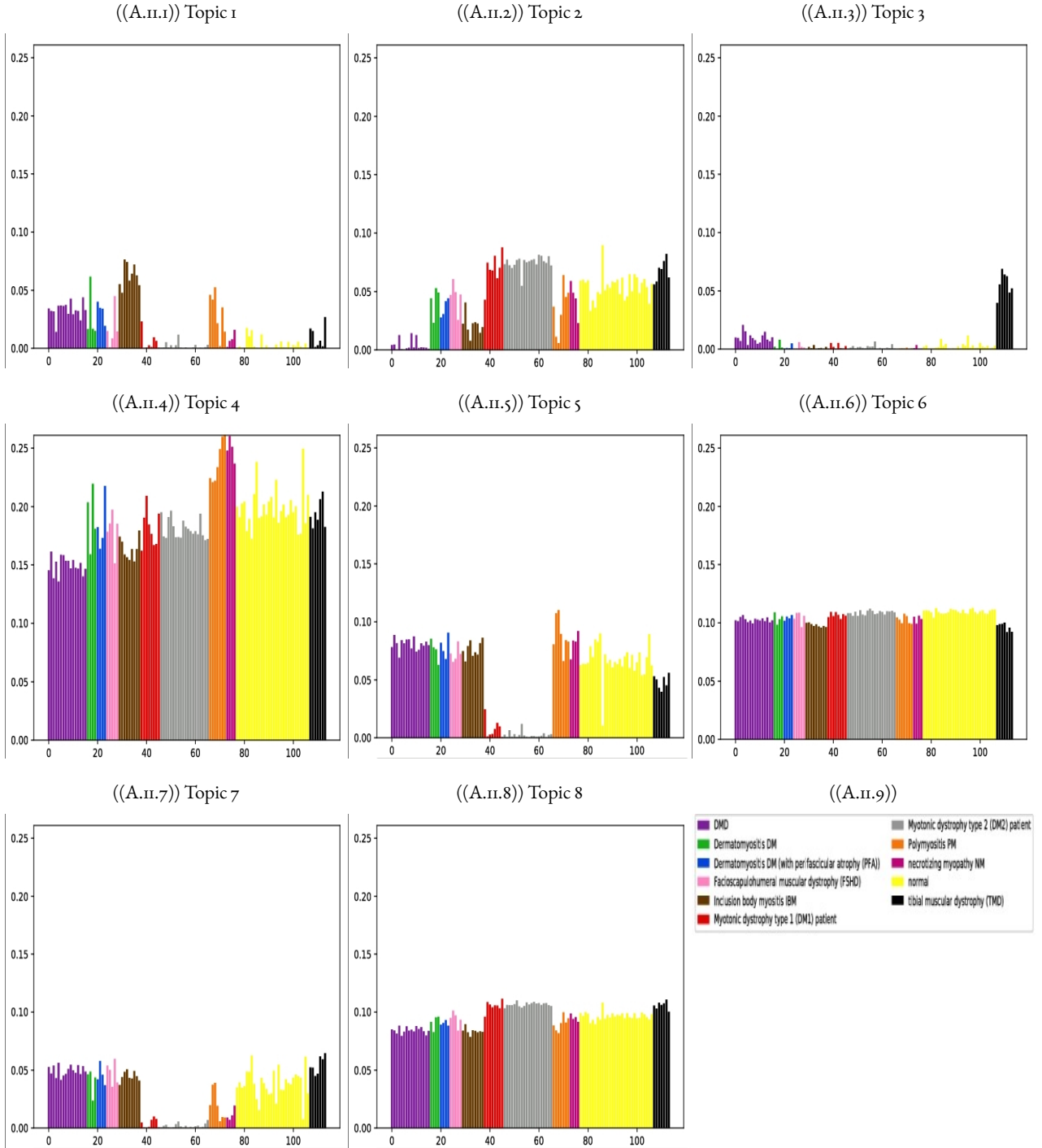


Figure A.11: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0$ Remove Bin $1 = \mathbf{x}_{t_2} = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

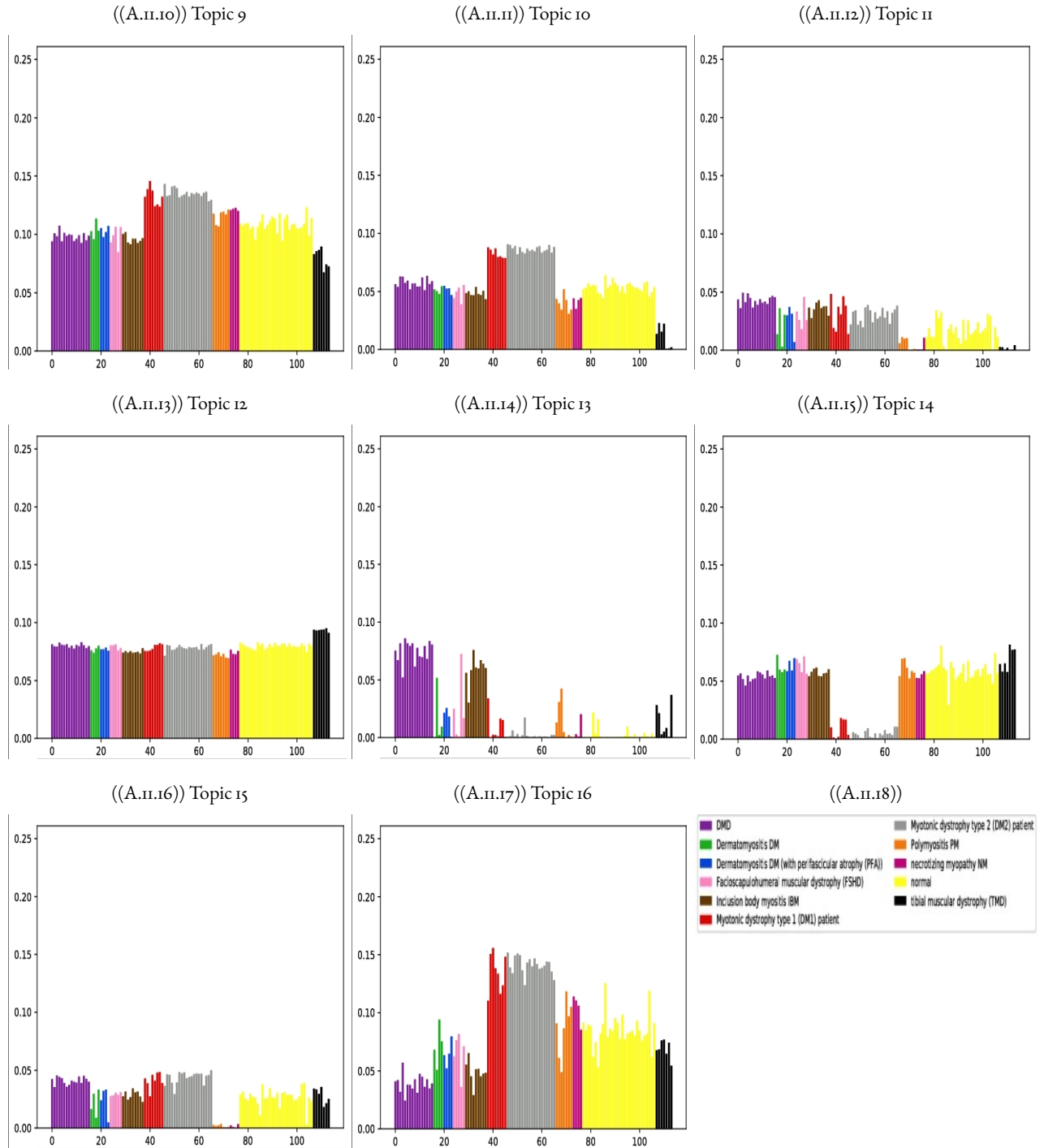


Figure A.11: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0$ Remove Bin $1 = \mathbf{X}_{t_2} = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

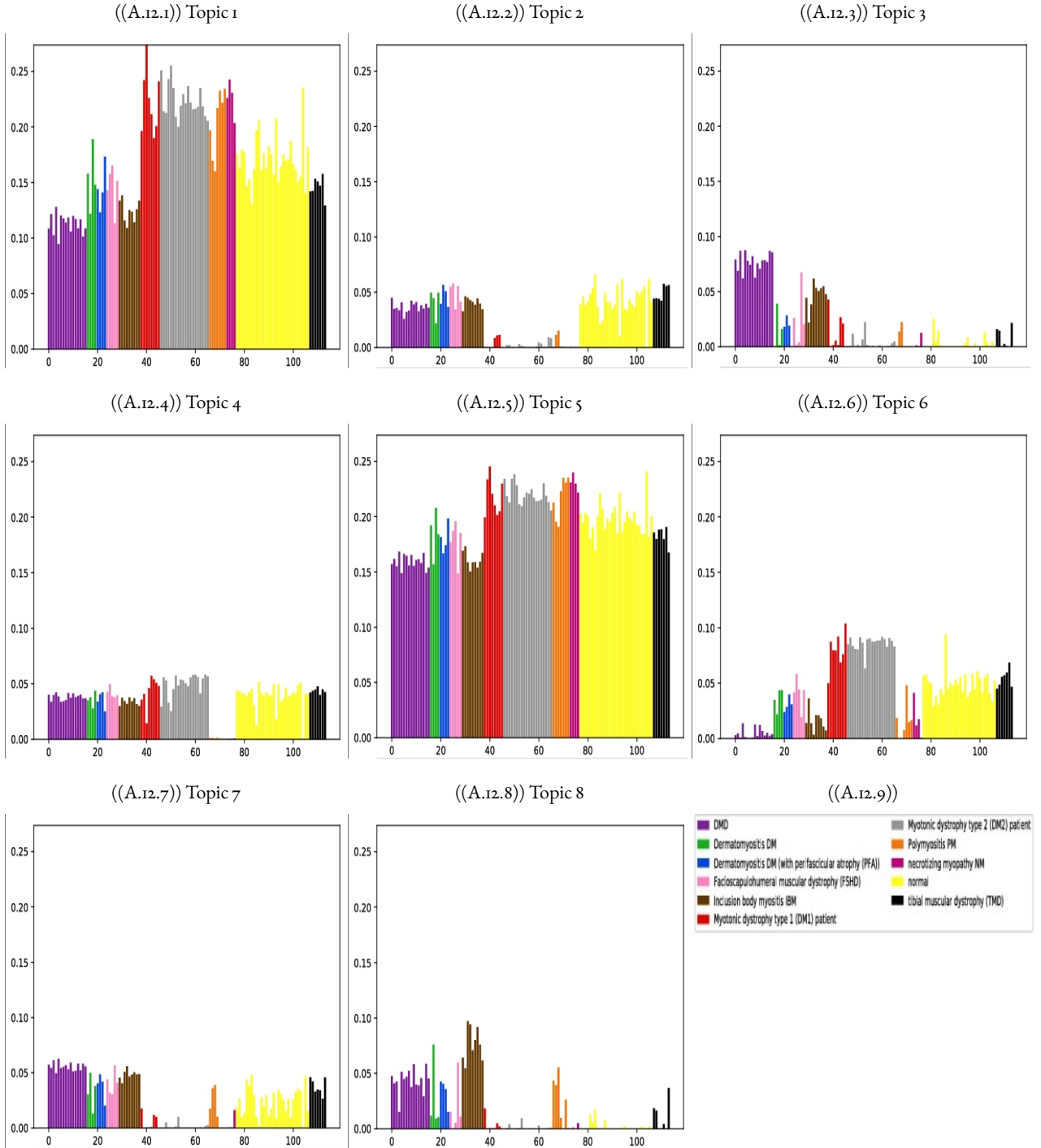


Figure A.12: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Repetition $t_1 = 0$ Remove Bin 1 = $\mathbf{x}_{t_2} = \mathbf{x}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

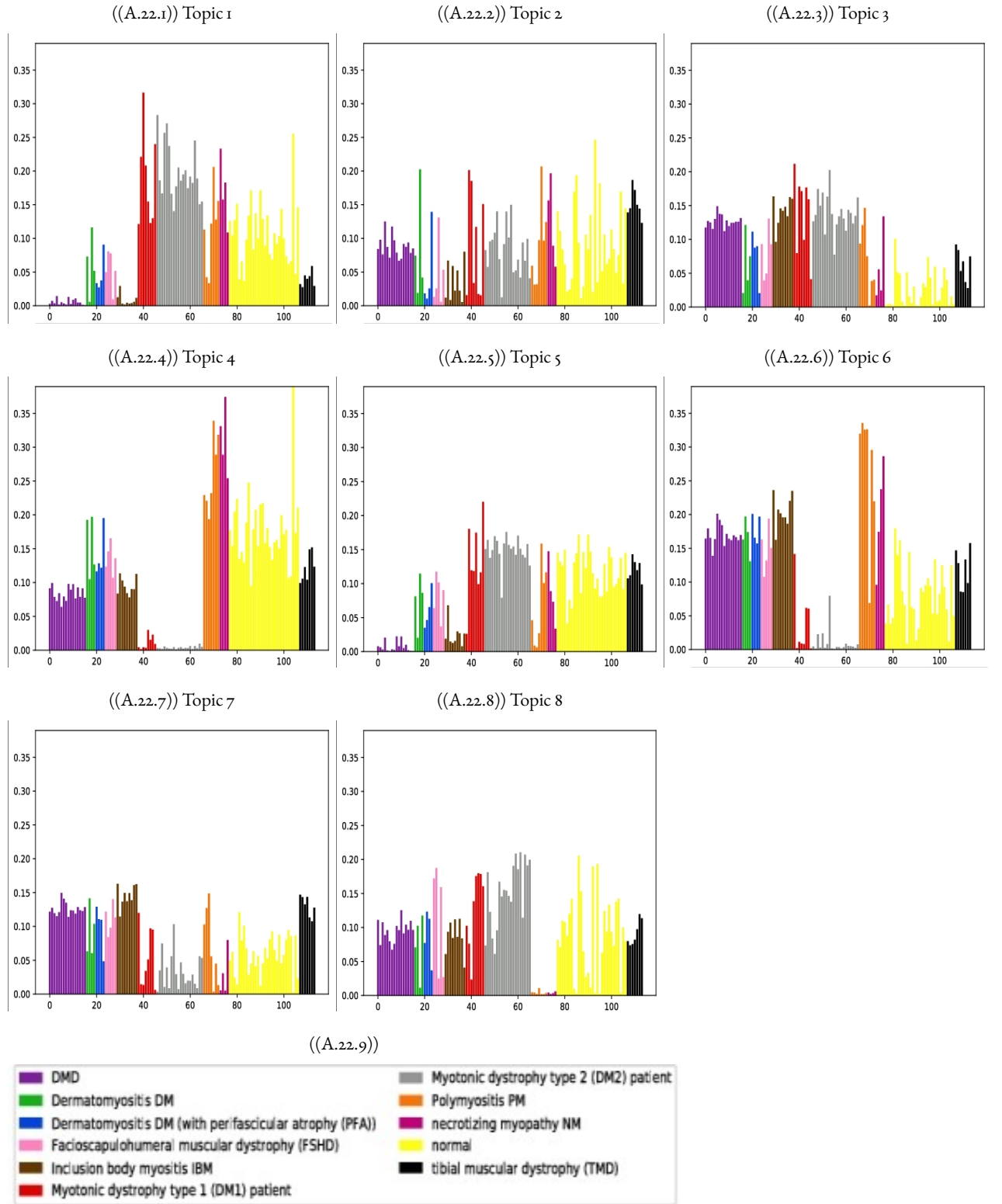


Figure A.22: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Median $t_M = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

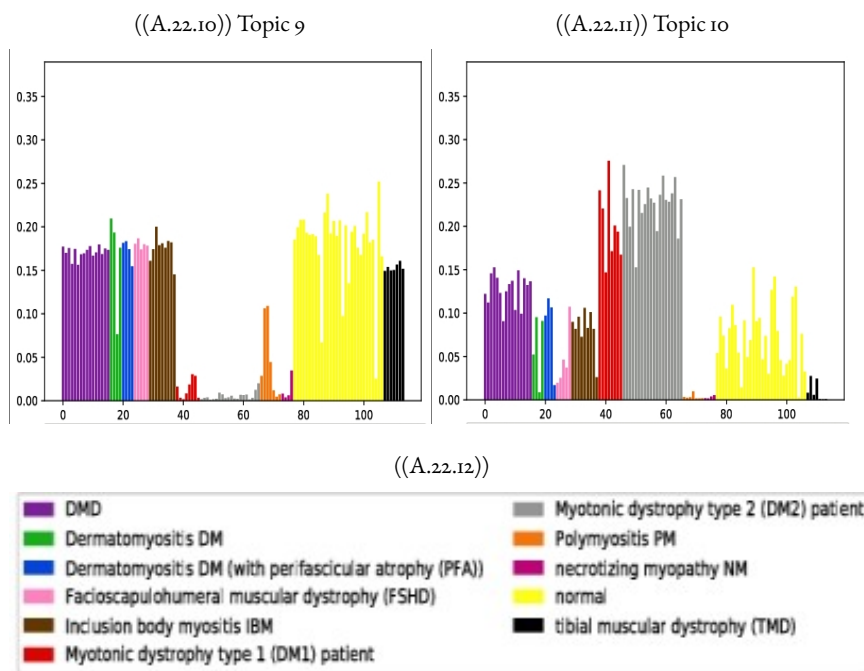


Figure A.22: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Median $t_M = \mathbf{x}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

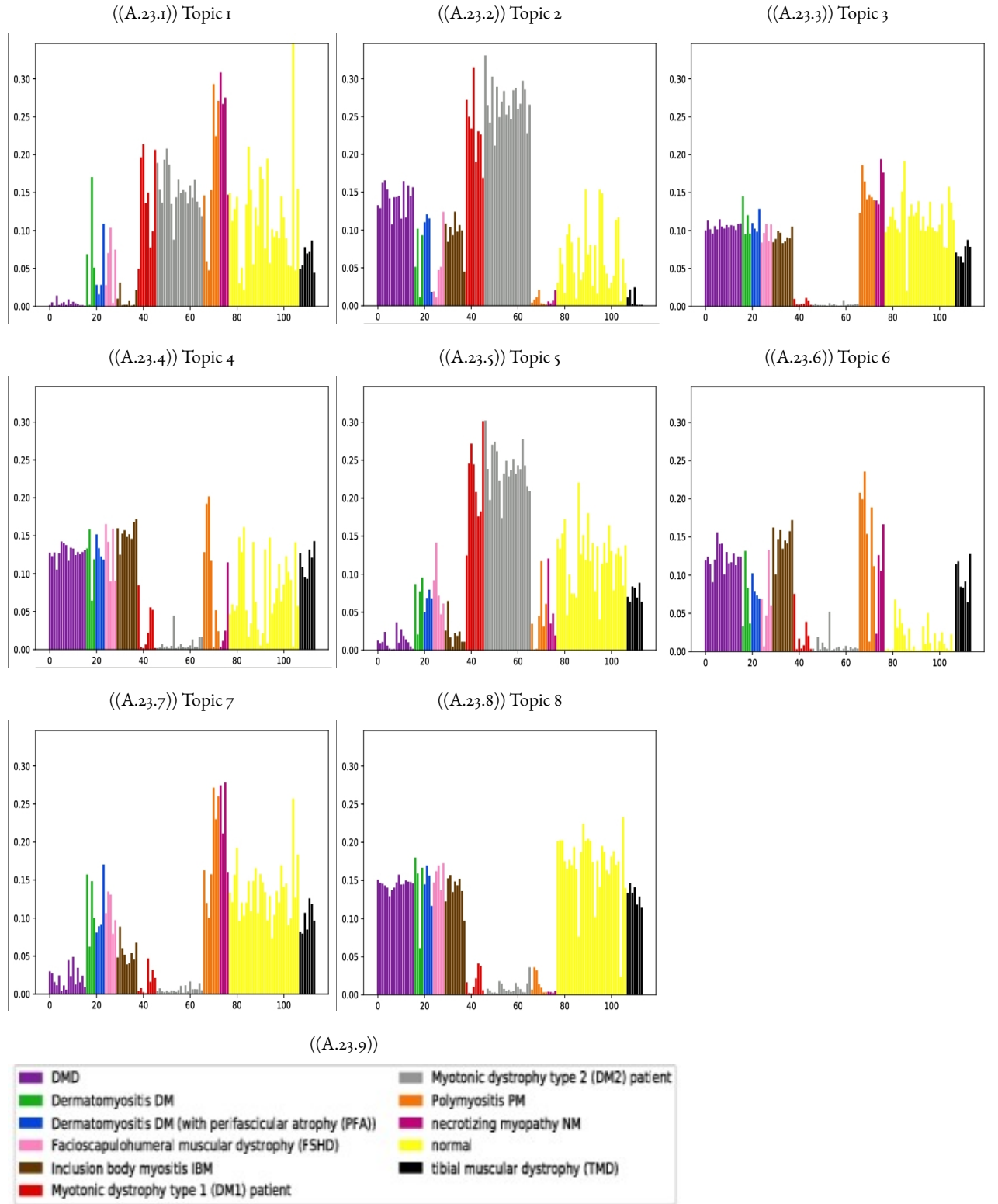


Figure A.23: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Median $t_M = 0$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

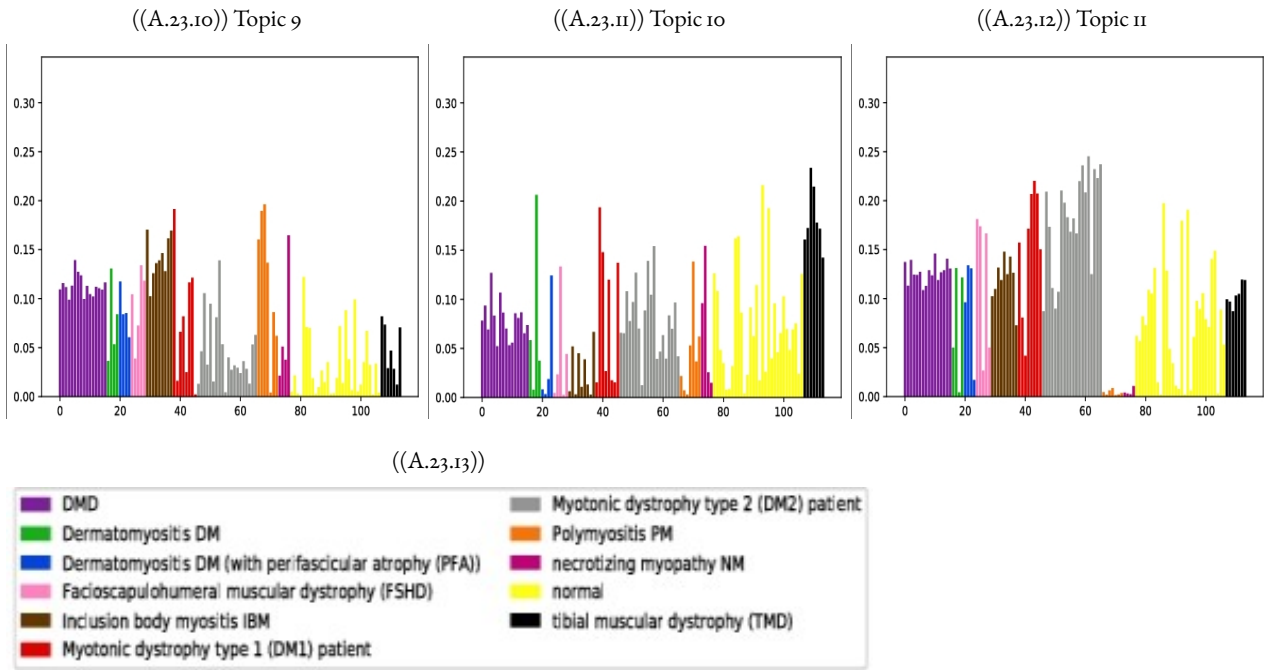


Figure A.23: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Median $t_M = 0$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

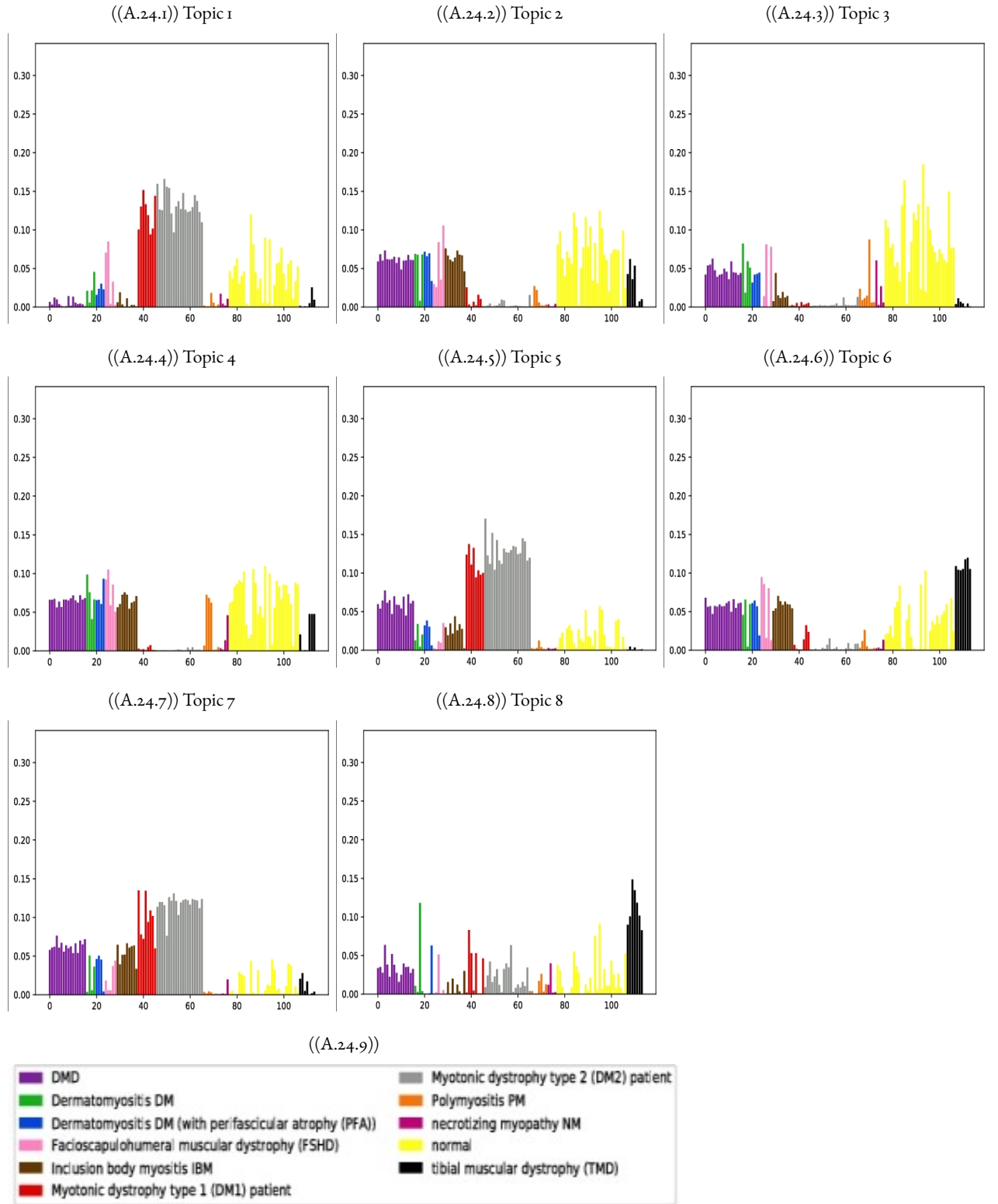


Figure A.24: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Median $t_M = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

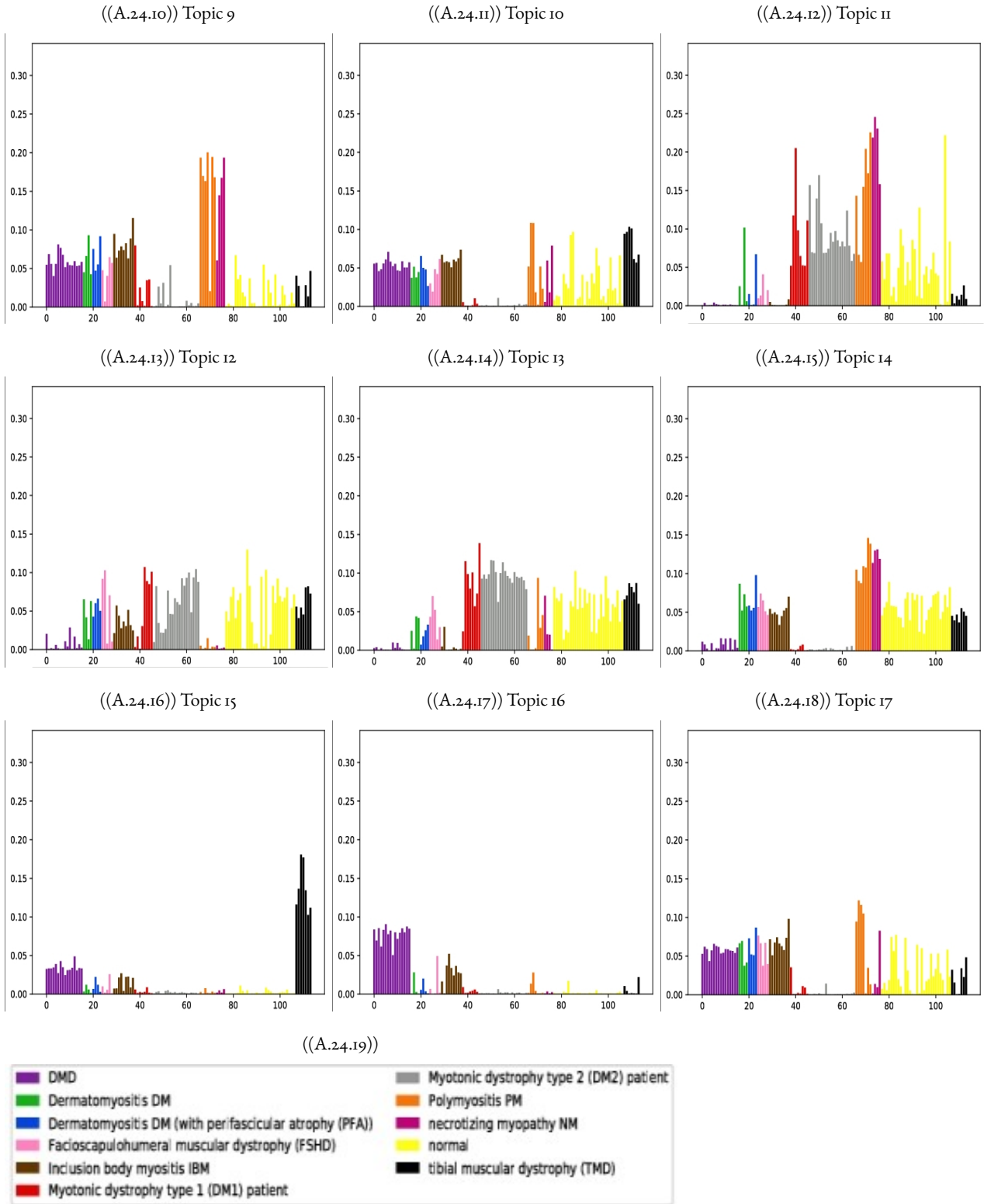


Figure A.24: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Median $t_M = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

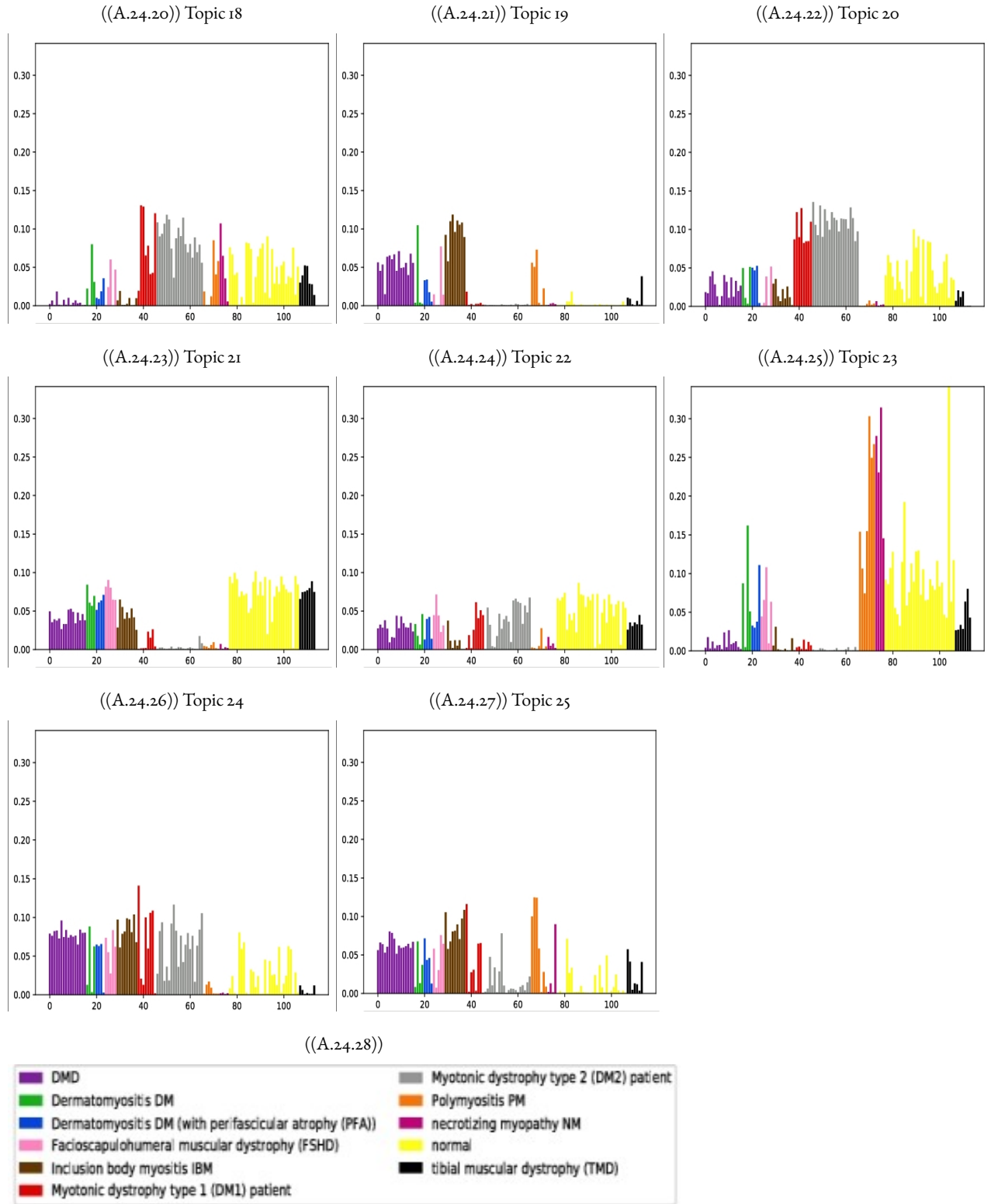


Figure A.24: The distribution over the samples for each topic on the Muscle Dataset obtained by LDA for Median $t_M = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

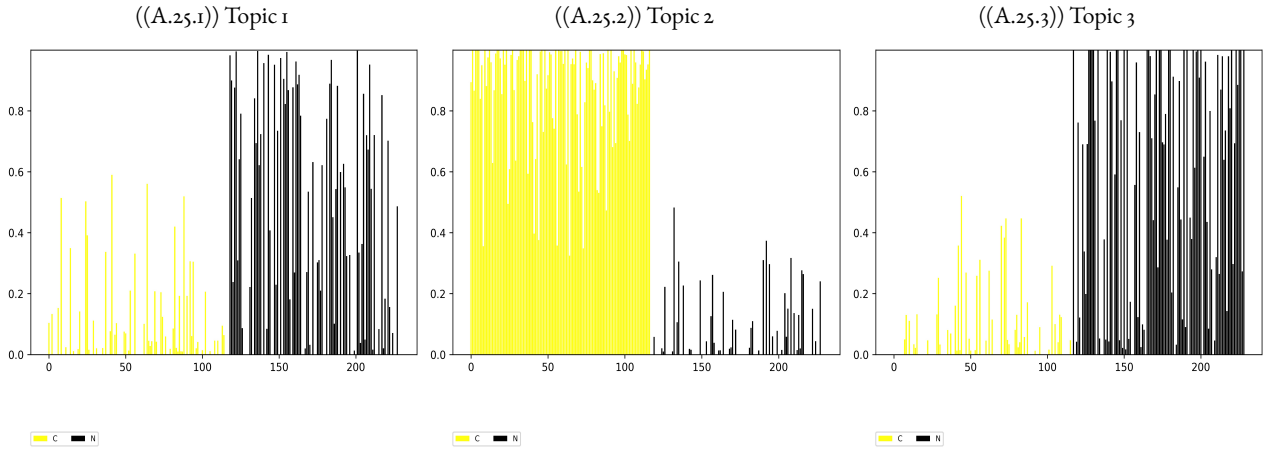


Figure A.25: The distribution over the samples for each topic on the TCGA Dataset obtained by LDA for Median with $t_M = 0.2$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

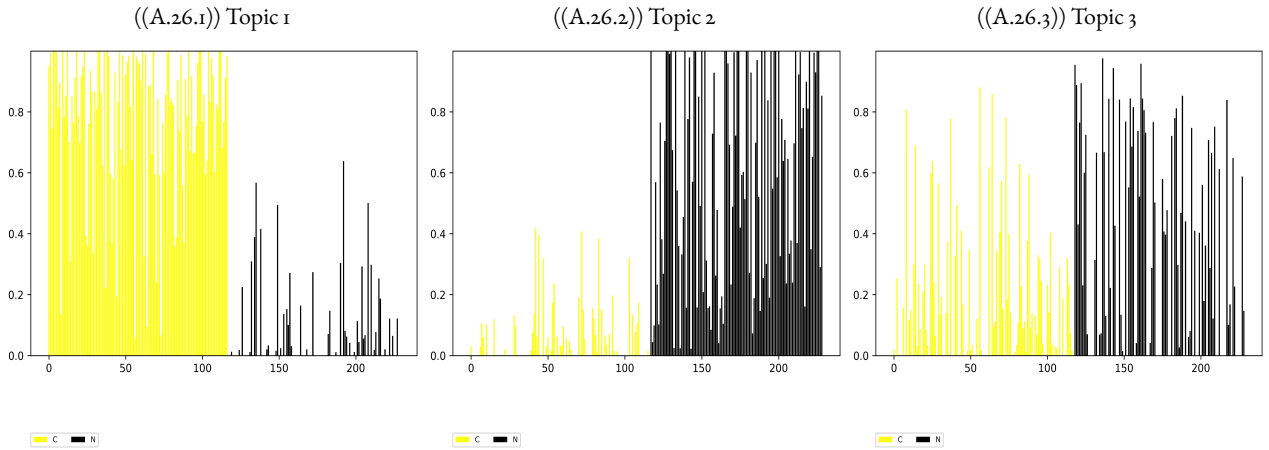


Figure A.26: The distribution over the samples for each topic on the TCGA Dataset obtained by LDA for Median with $t_M = \mathbf{X}$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

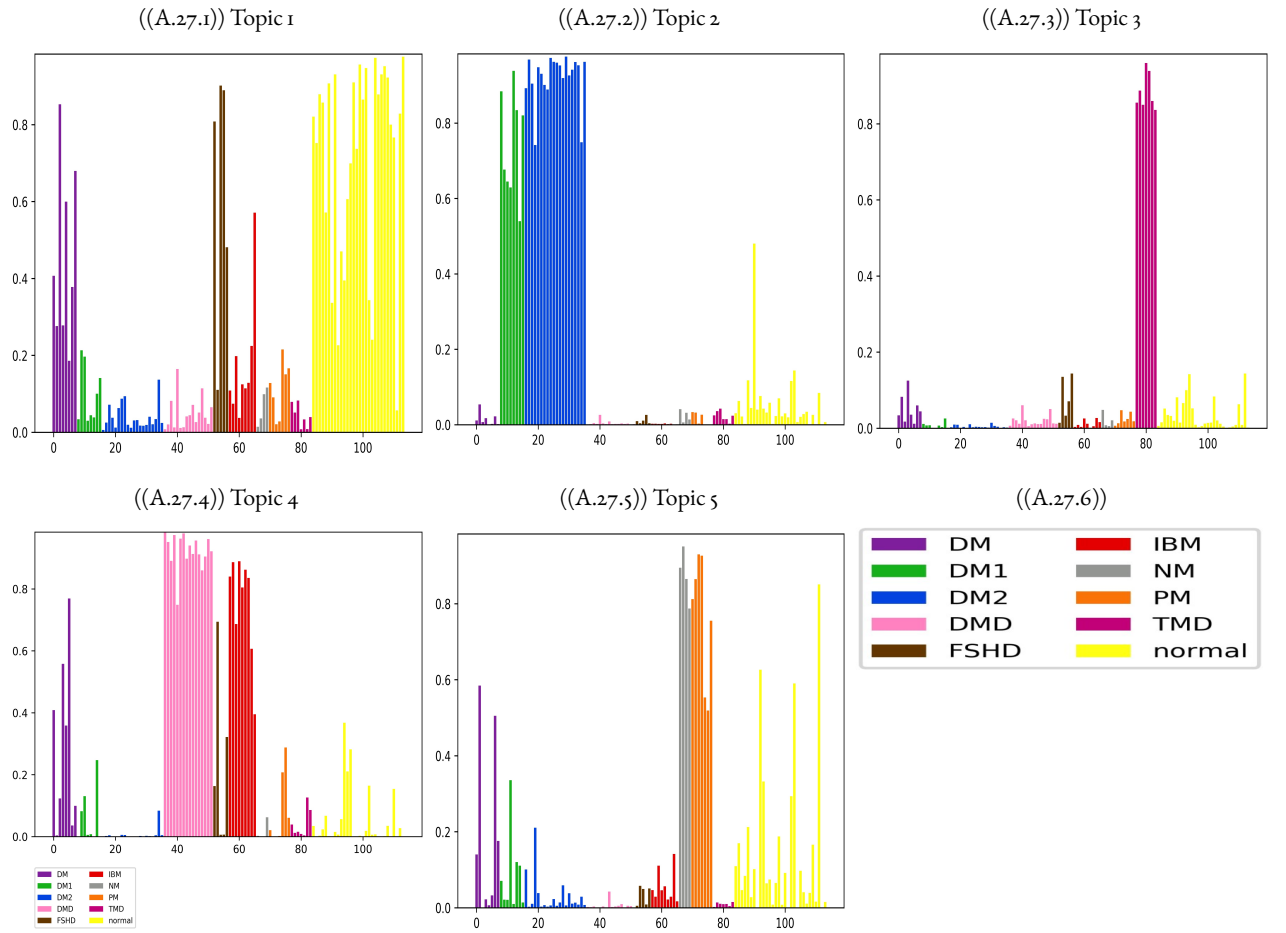


Figure A.27: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0$ and $K = 5$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

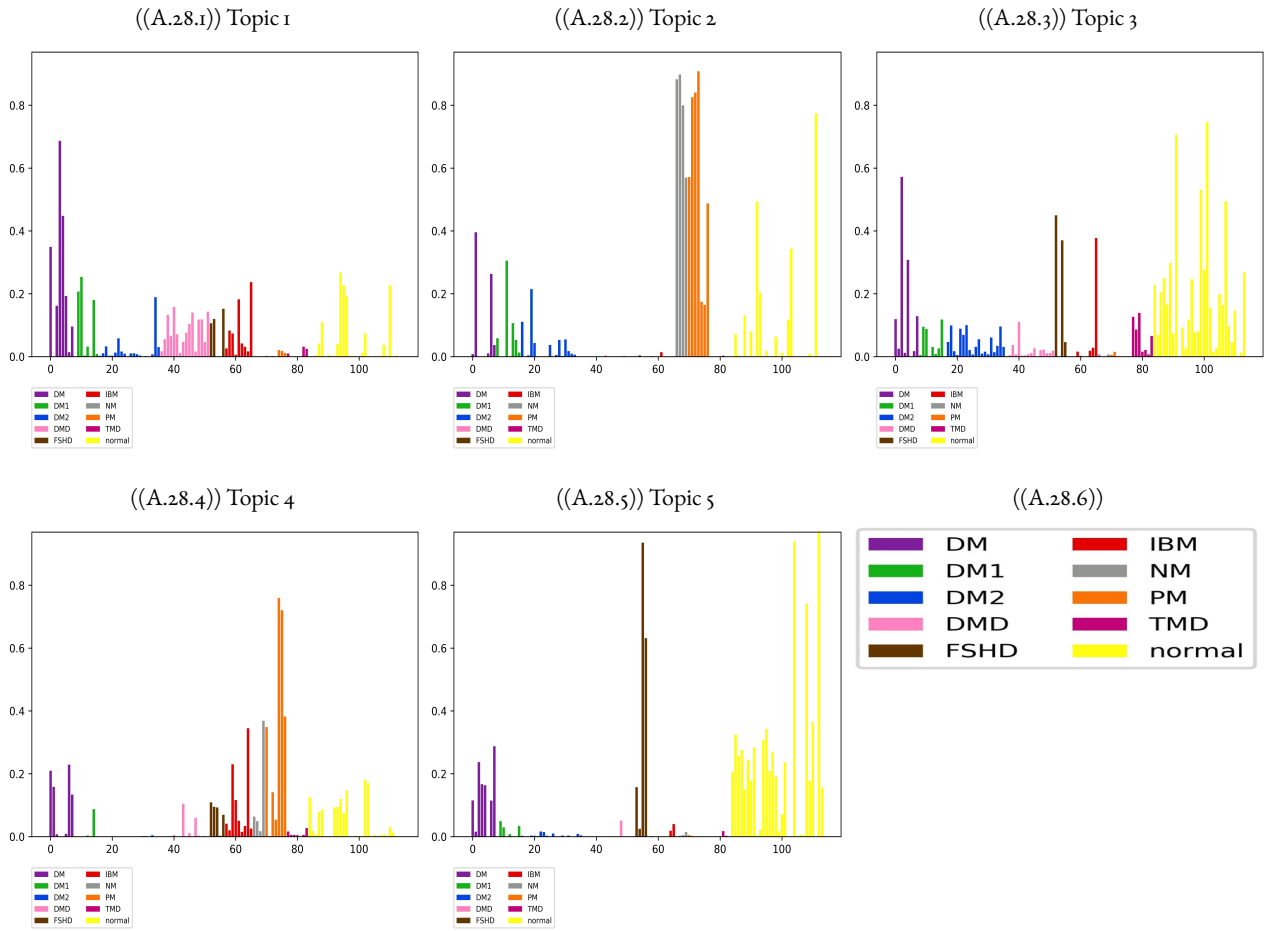


Figure A.28: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0$ and $K = 10$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

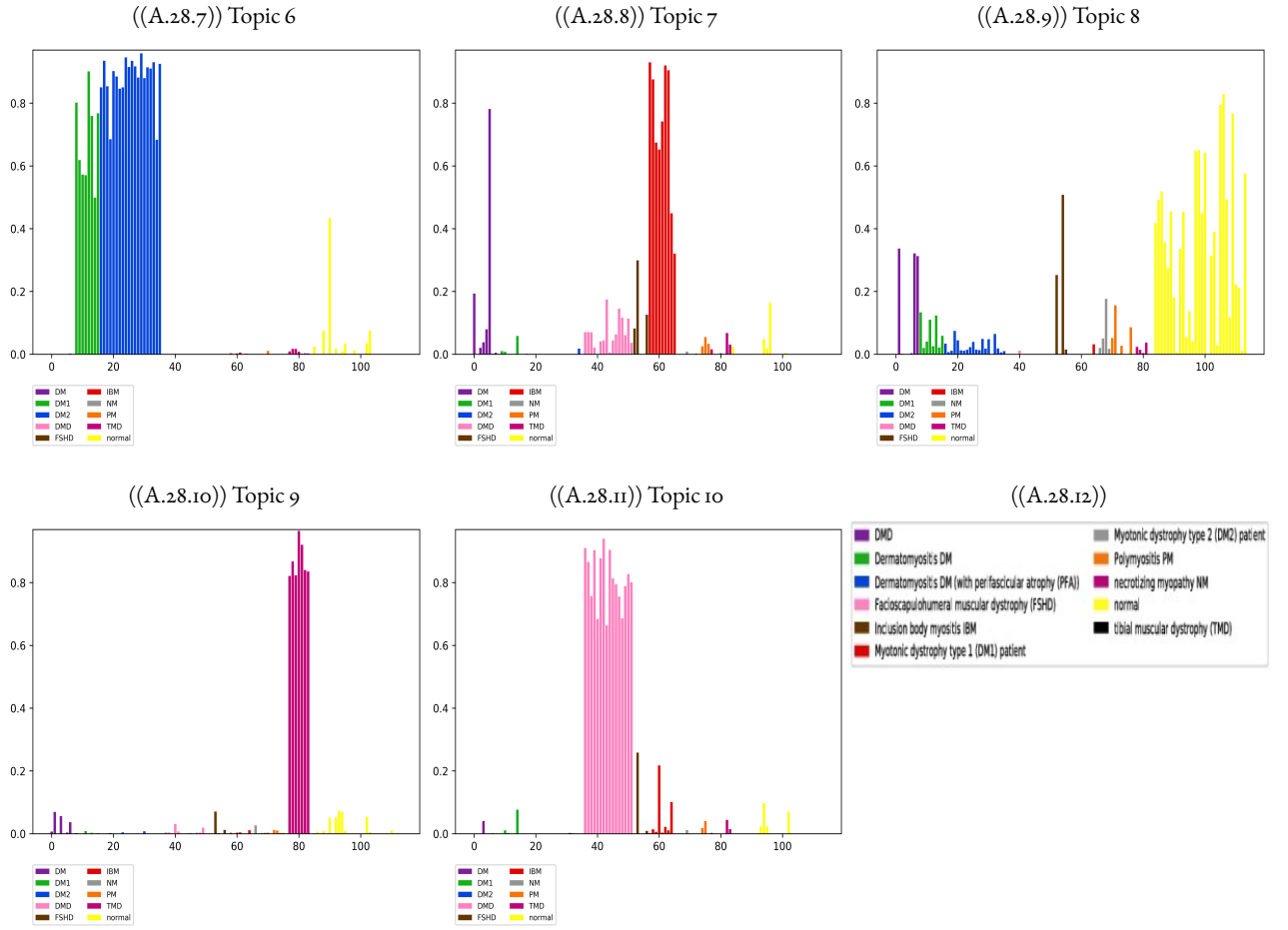


Figure A.28: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0$ and $K = 10$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

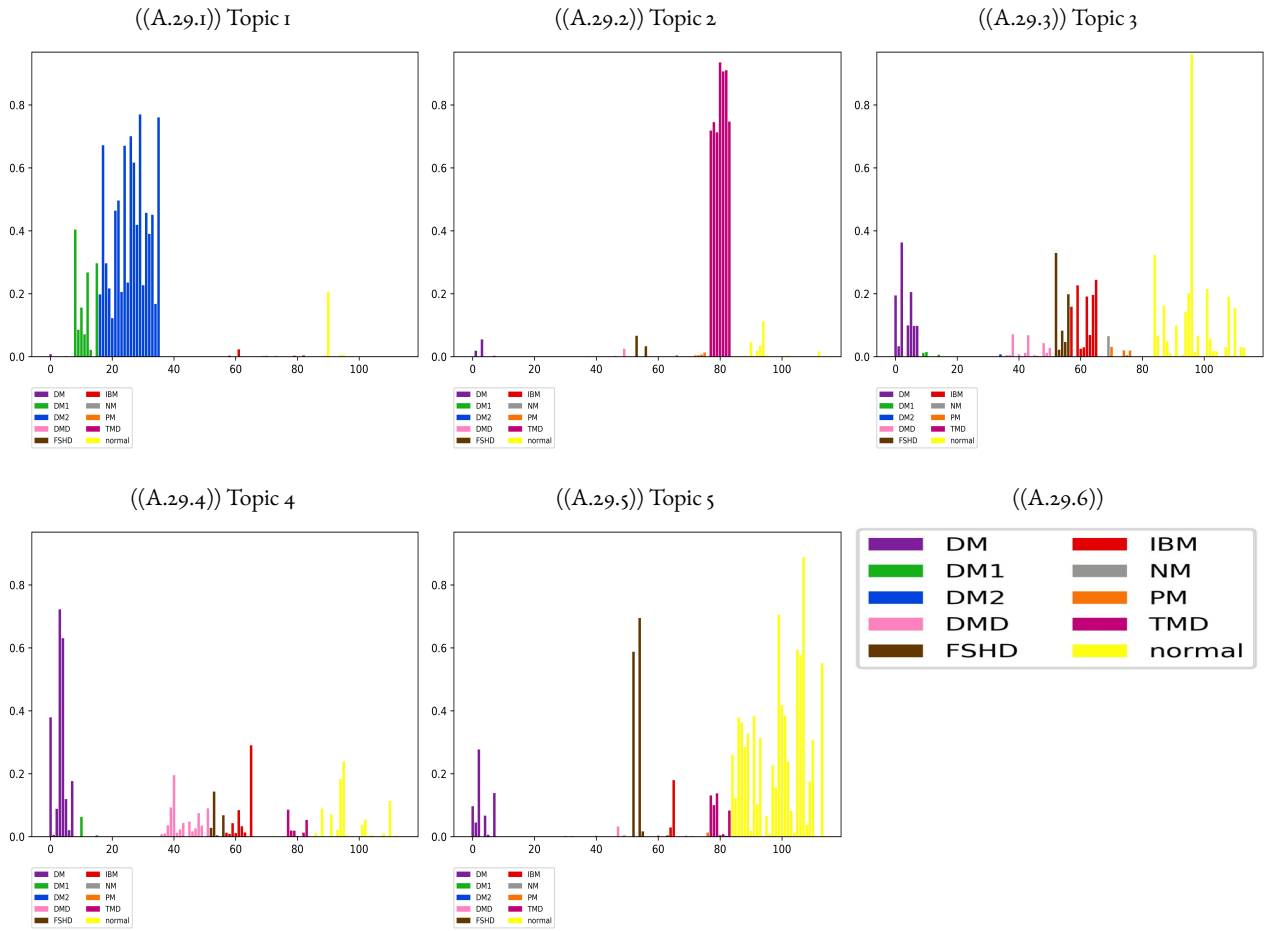


Figure A.29: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0$ and $K = 15$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

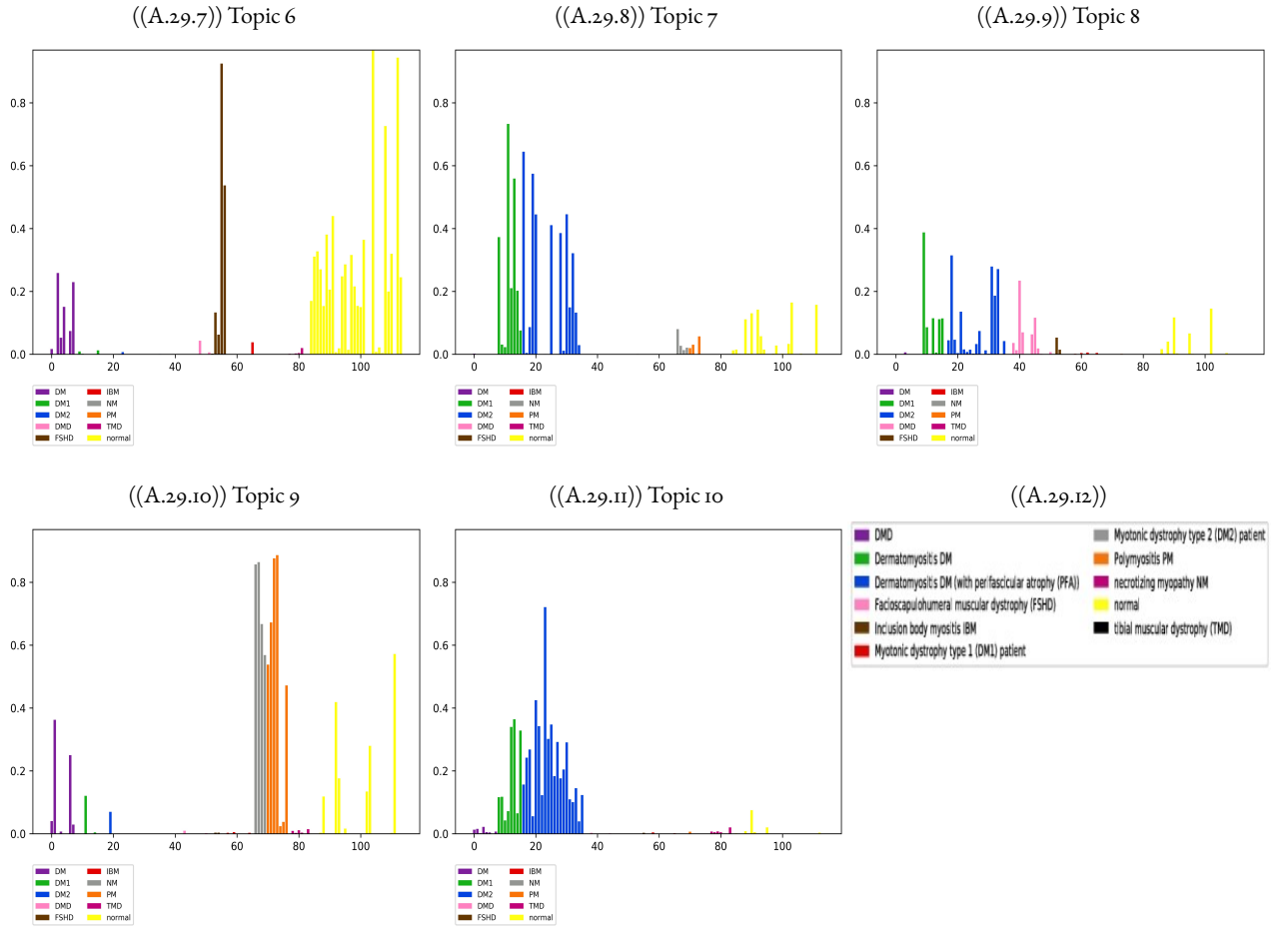


Figure A.29: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0$ and $K = 15$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

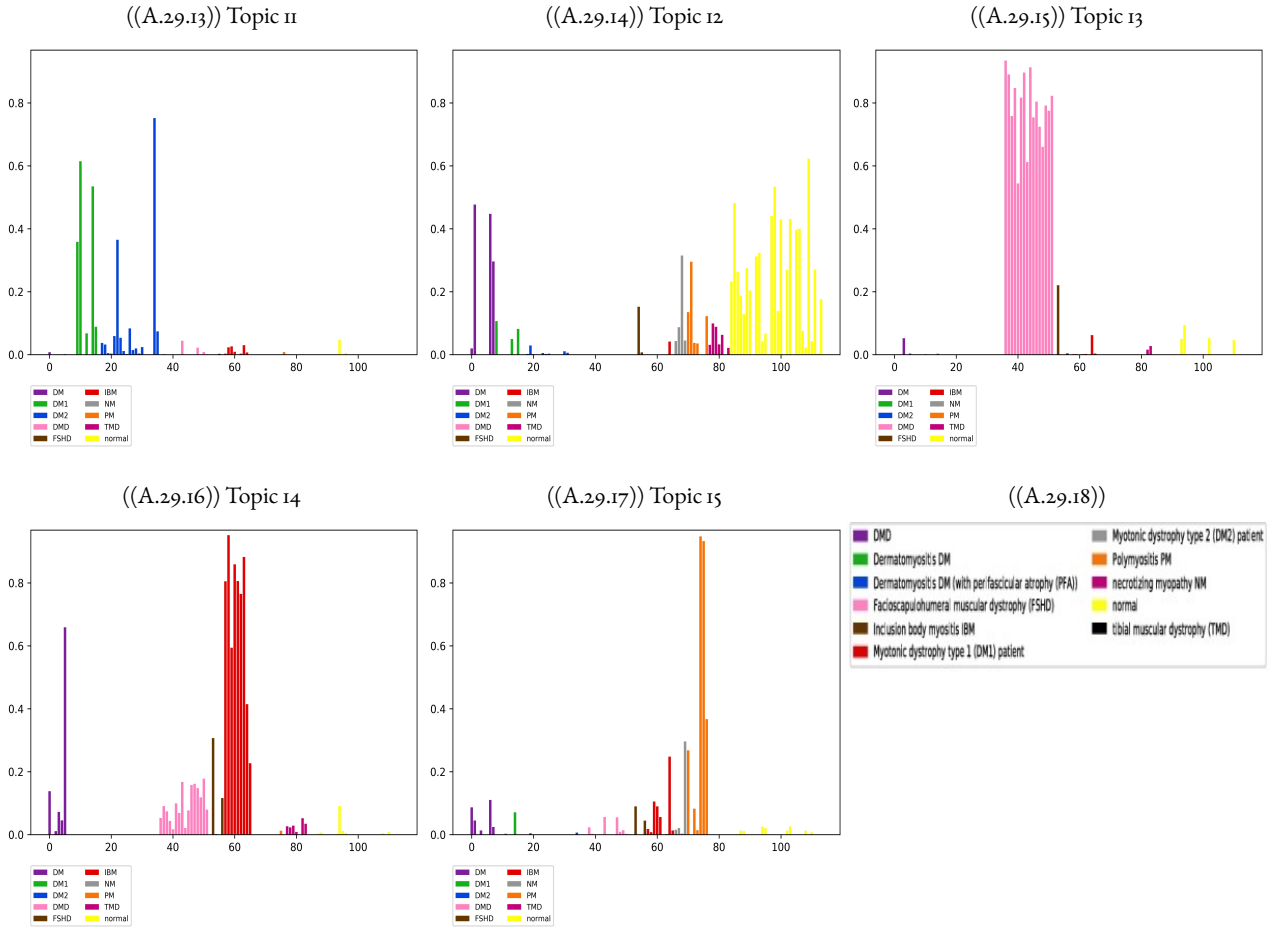


Figure A.29: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0$ and $K = 15$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

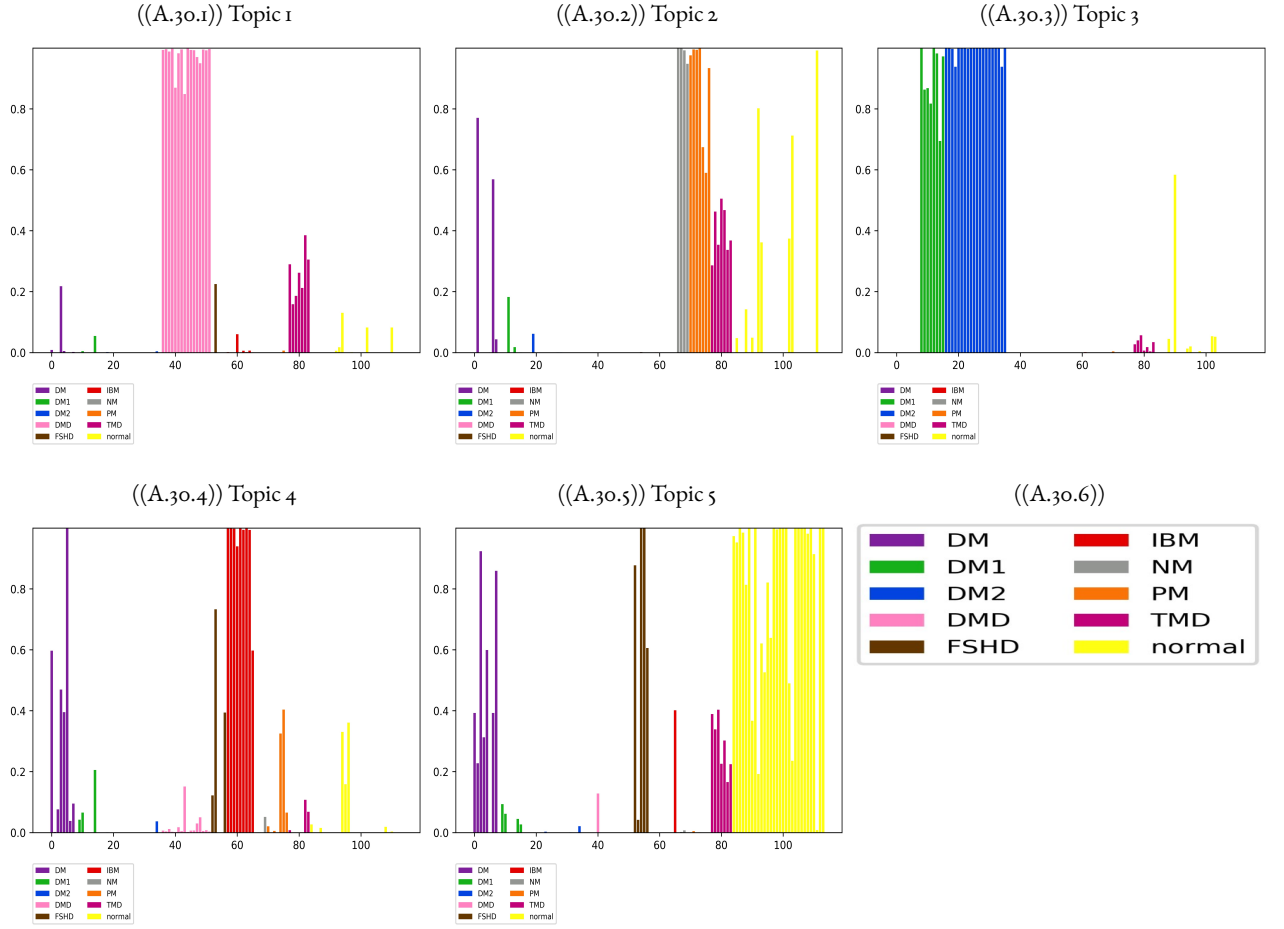


Figure A.30: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0.2$ and $K = 5$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

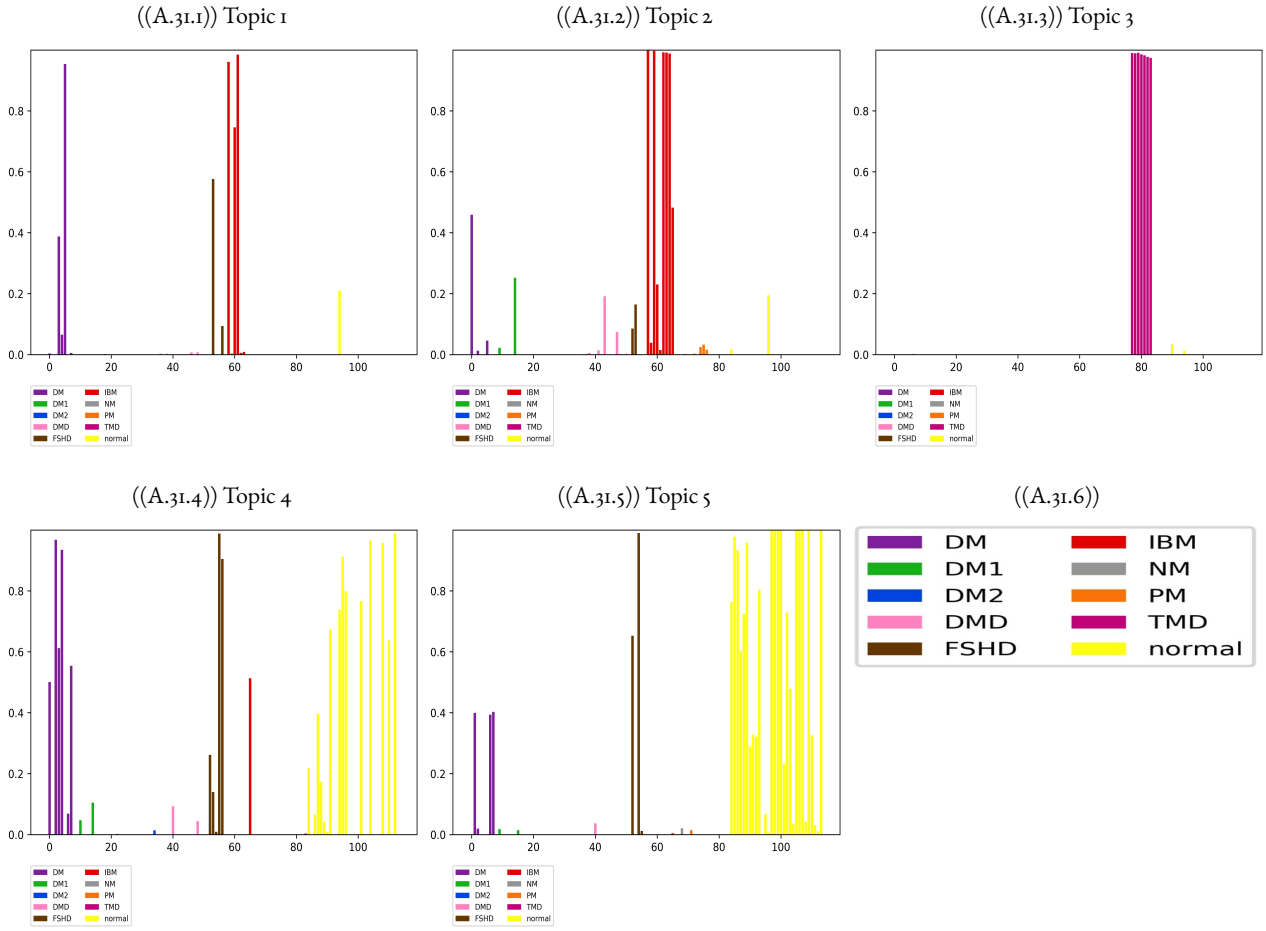


Figure A.31: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0.2$ and $K = 10$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

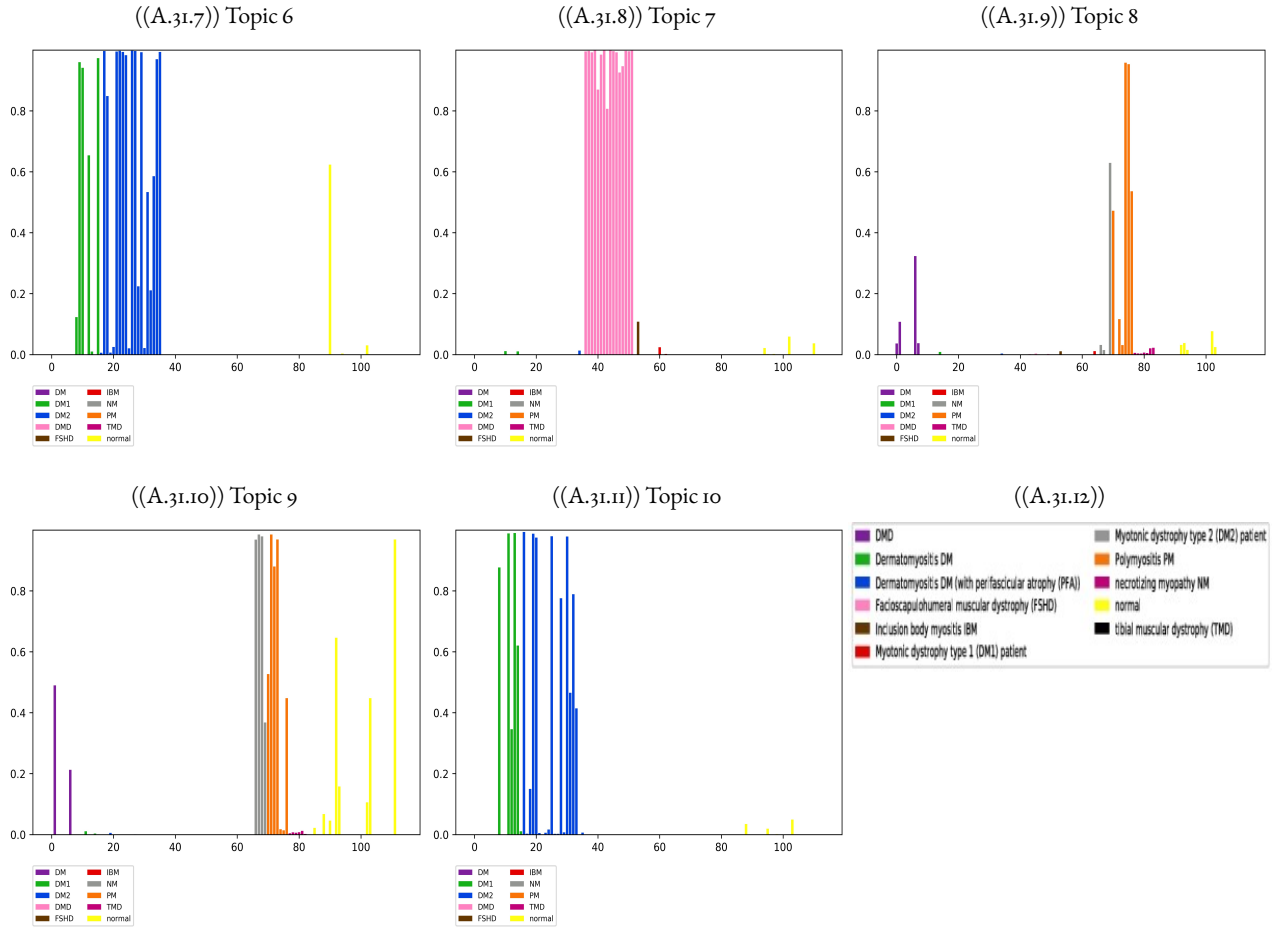


Figure A.31: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0.2$ and $K = 10$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

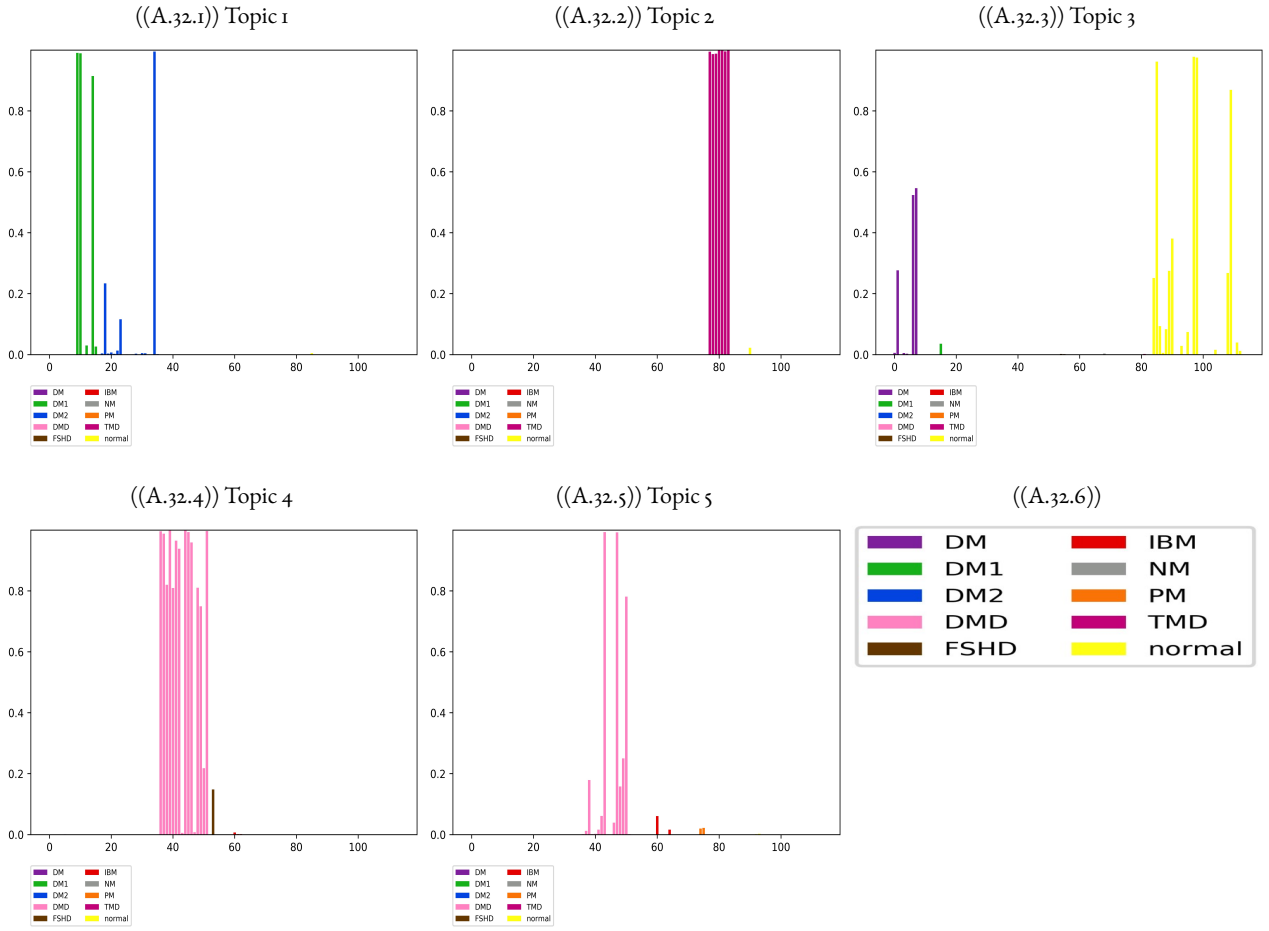


Figure A.32: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0.2$ and $K = 15$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

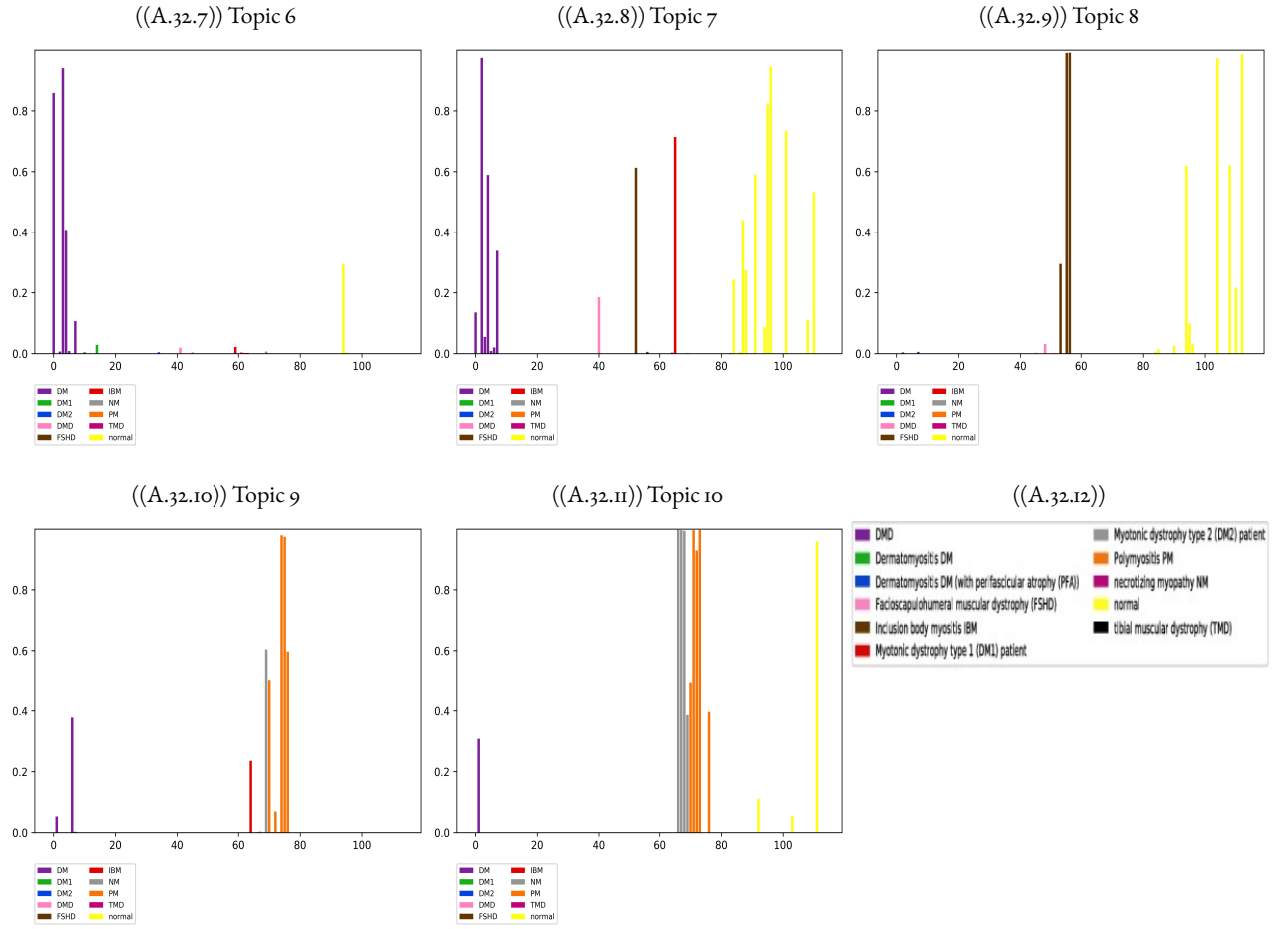


Figure A.32: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0.2$ and $K = 15$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

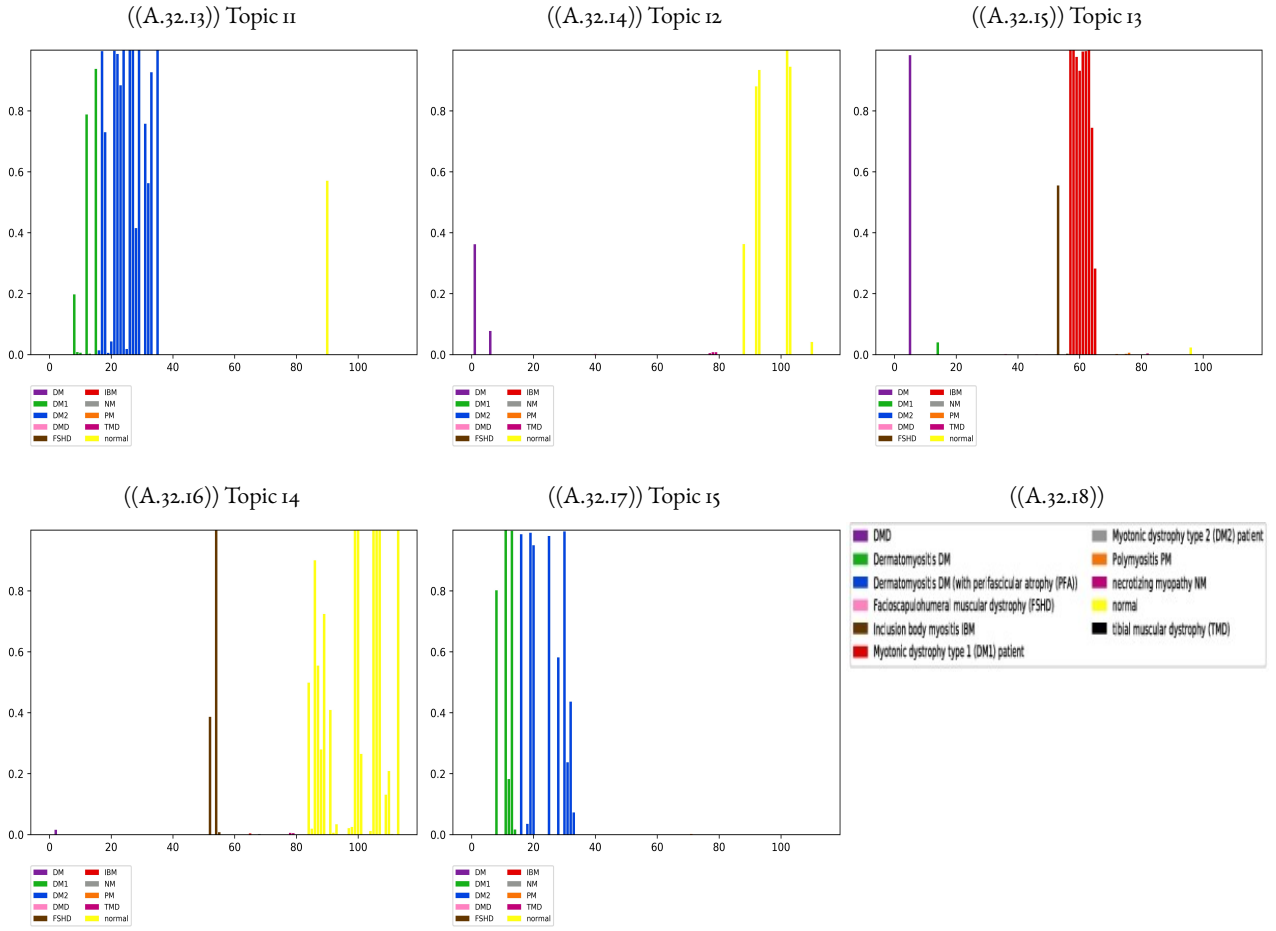


Figure A.32: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = 0.2$ and $K = 15$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

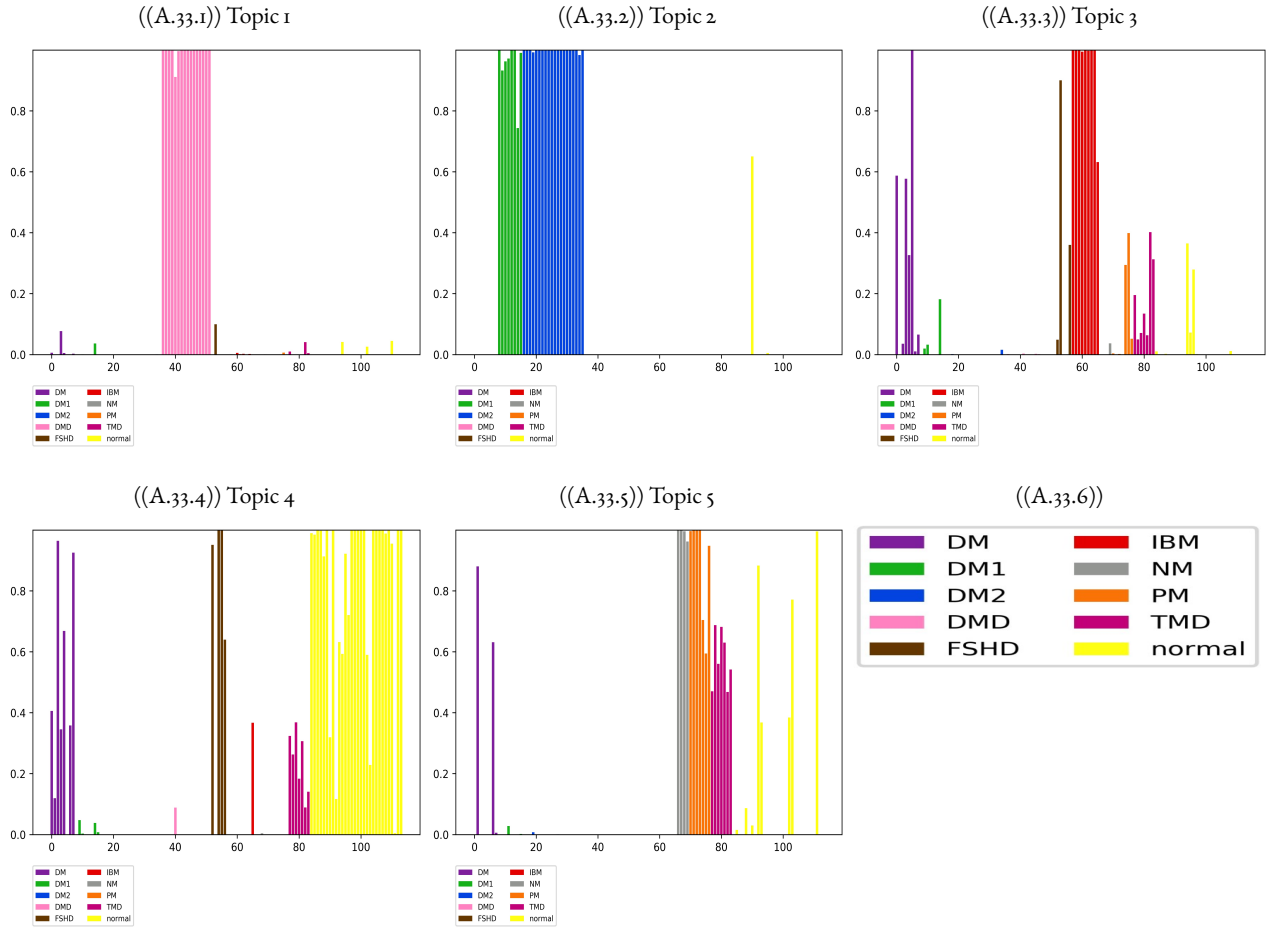


Figure A.33: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = \mathbf{X}$ and $K = 5$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

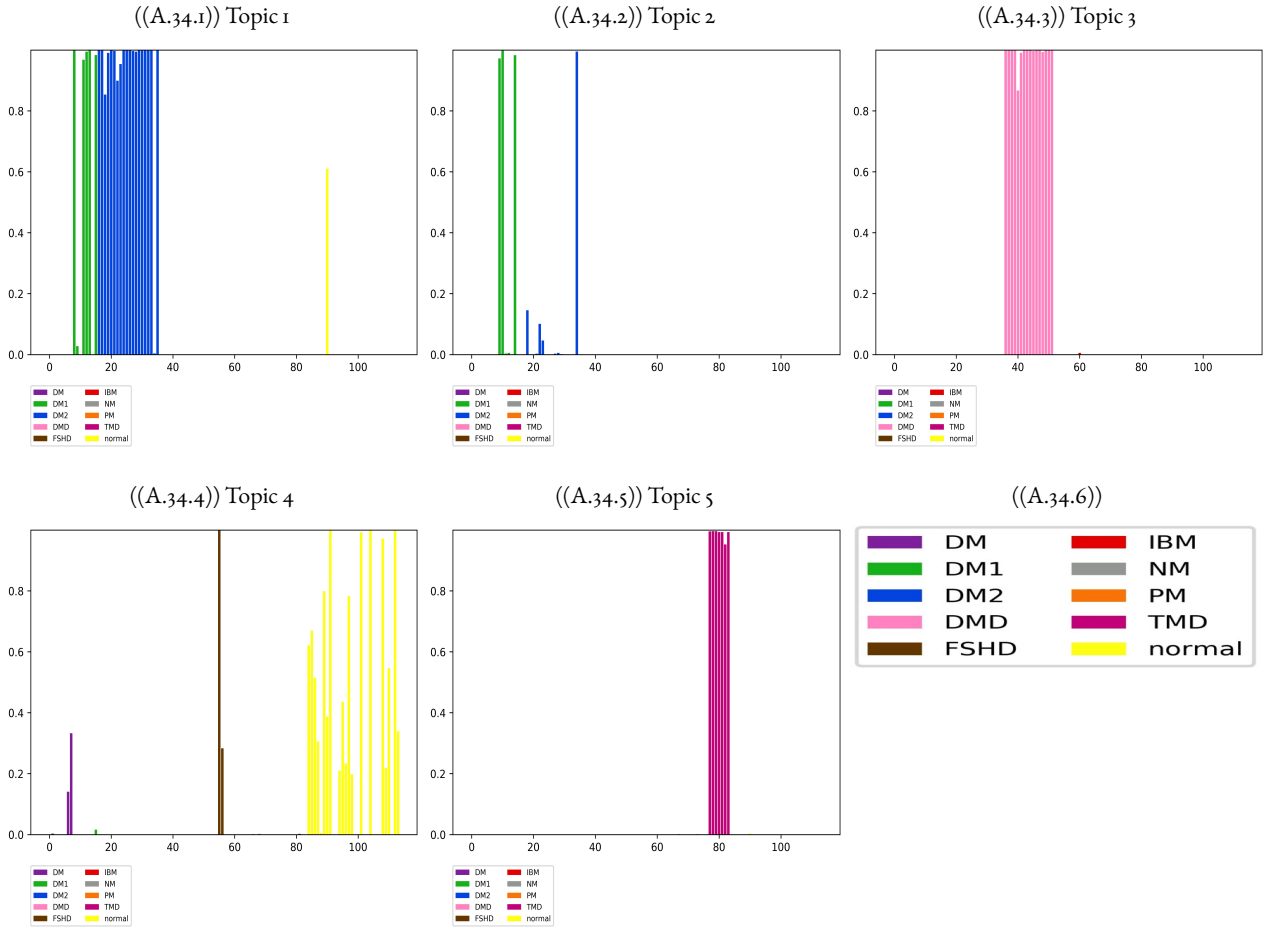


Figure A.34: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = \mathbf{X}$ and $K = 10$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

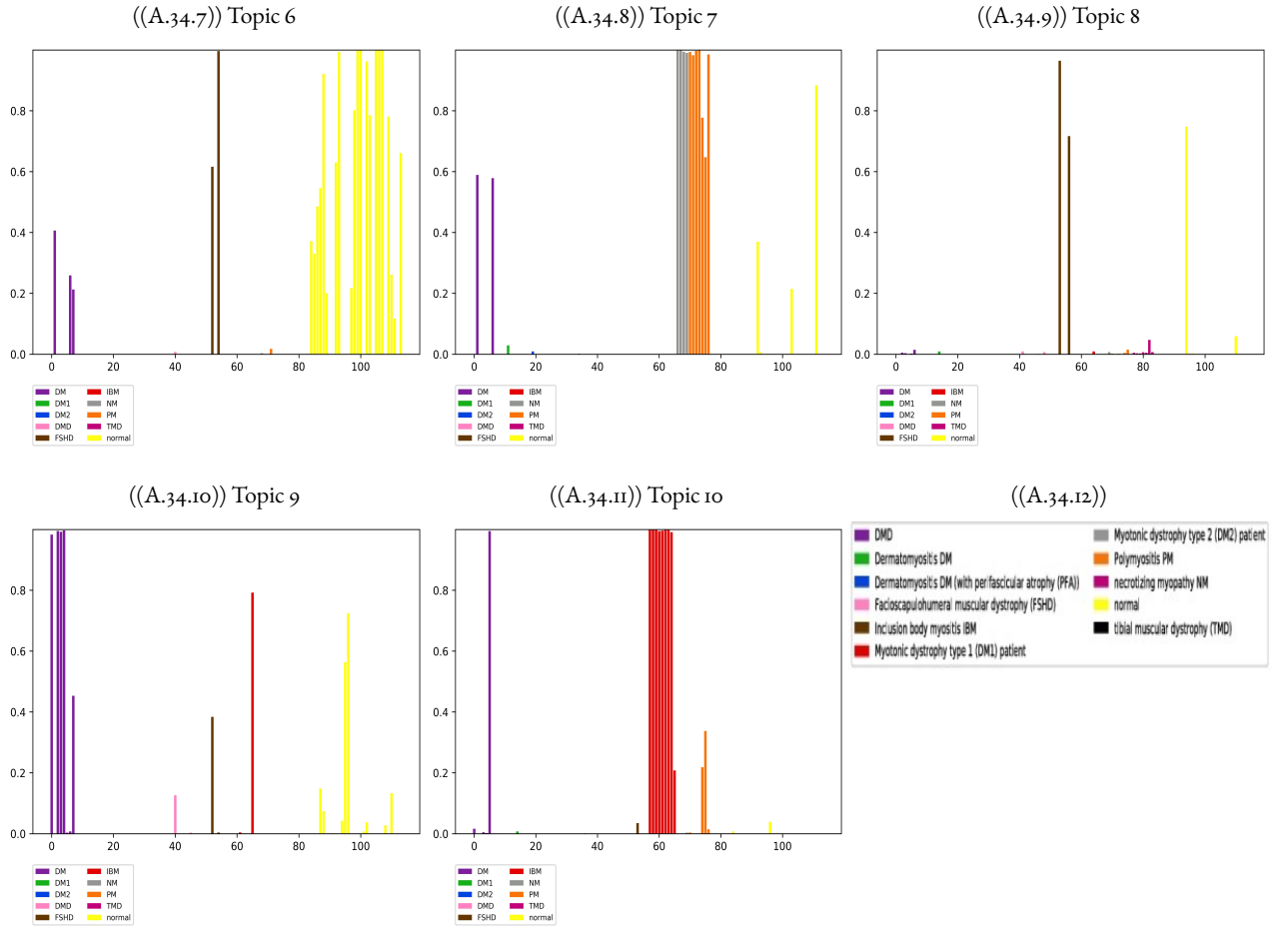


Figure A.34: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = \mathbf{X}$ and $K = 10$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

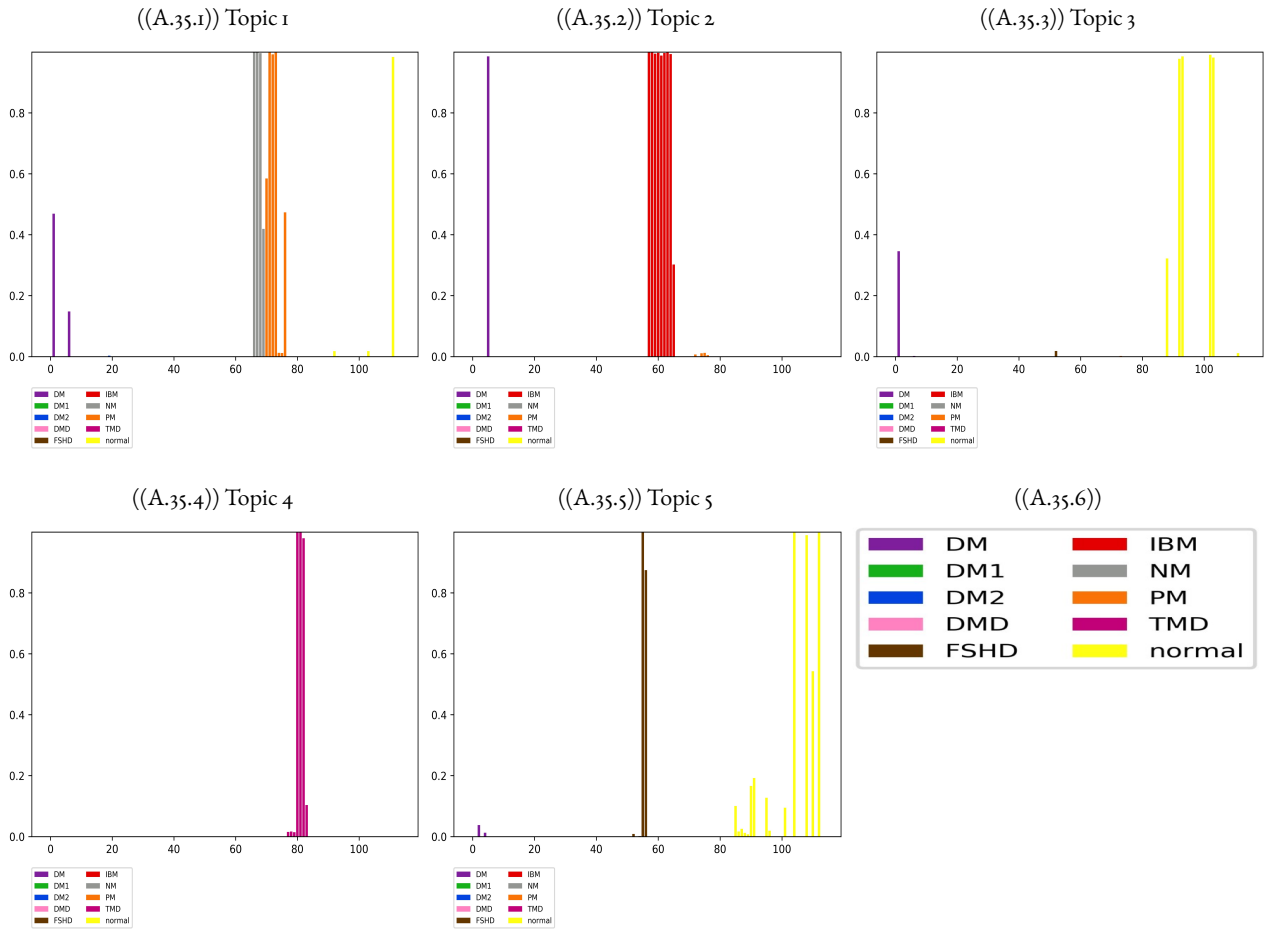


Figure A.35: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = \mathbf{X}$ and $K = 15$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

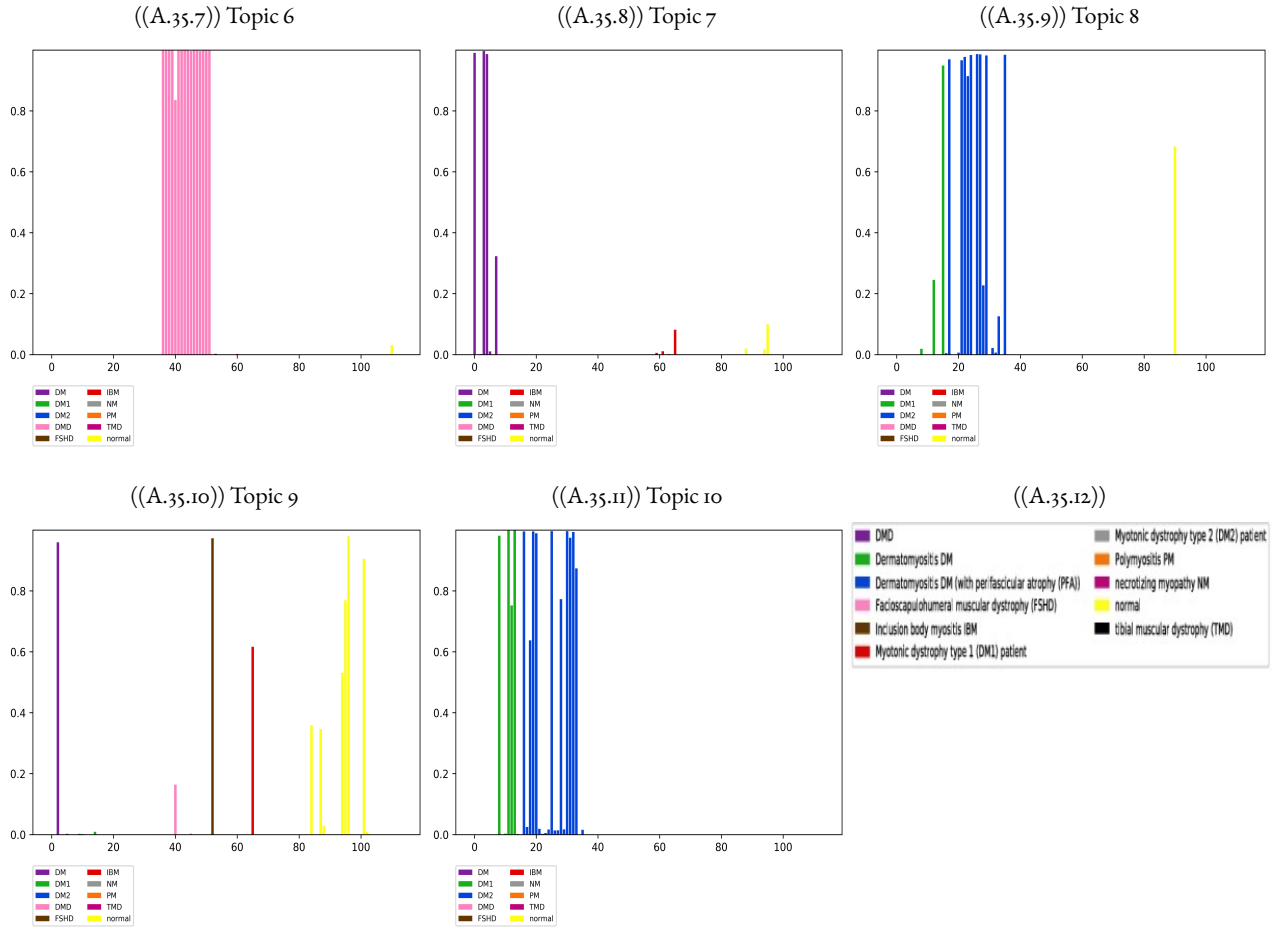


Figure A.35: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = \mathbf{x}$ and $K = 15$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

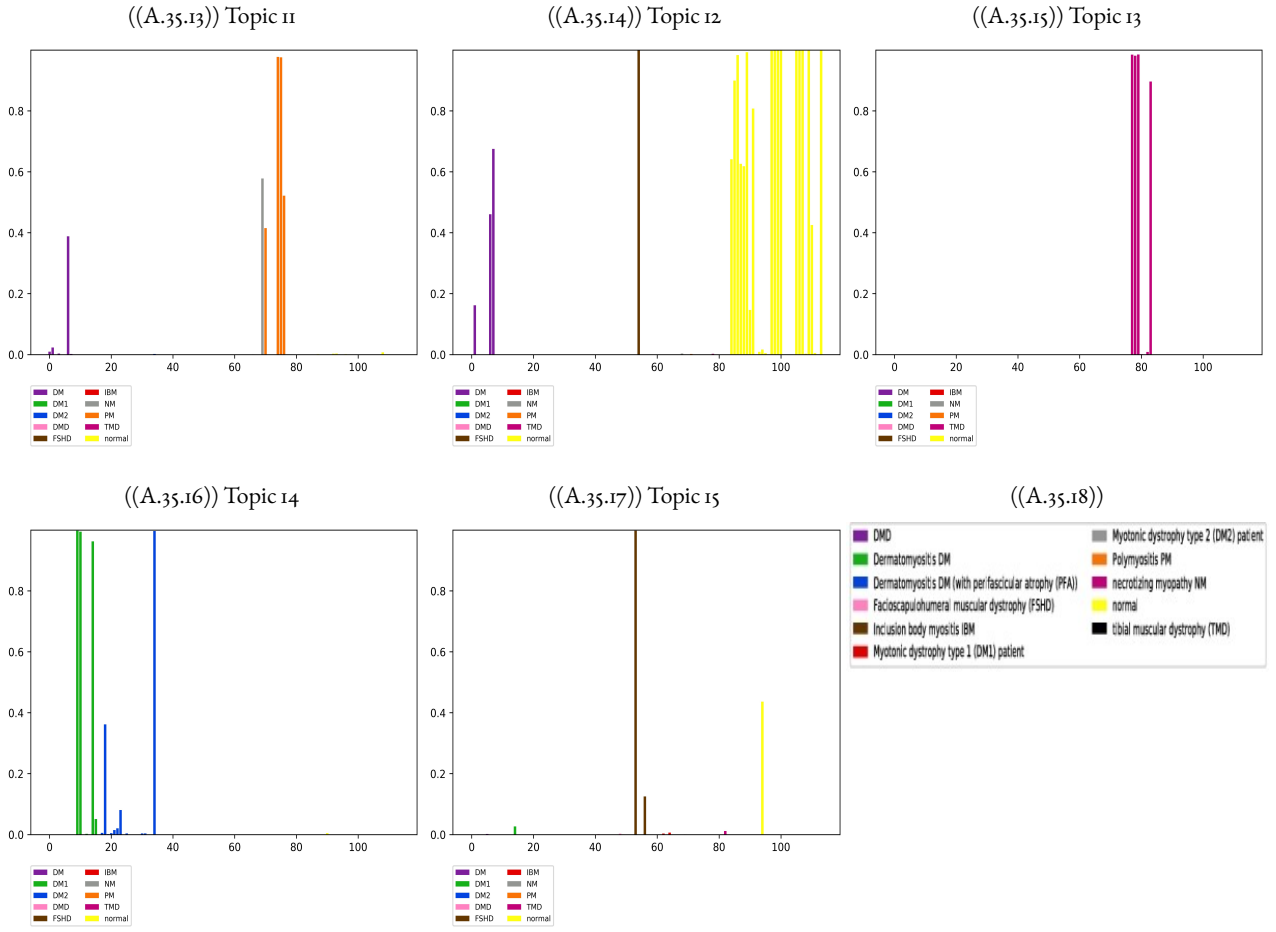


Figure A.35: The distribution over the samples for each topic on the Muscle Dataset obtained by LPD with $t_{LPD} = \mathbf{X}$ and $K = 15$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

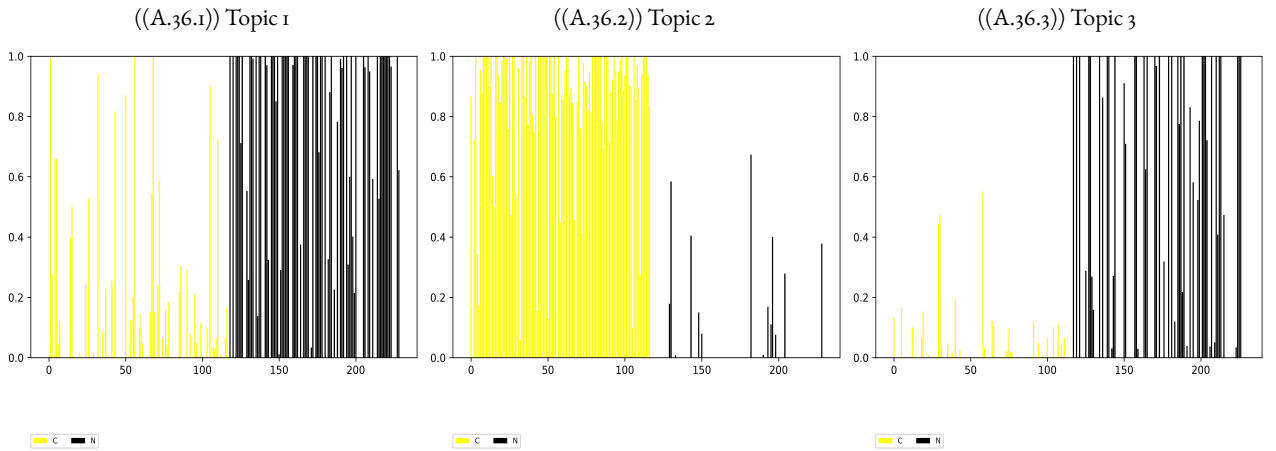


Figure A.36: The distribution over the samples for each topic on the TCGA Dataset obtained by LPD with $t_{LPD} = \mathbf{X}$ and $K = 3$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

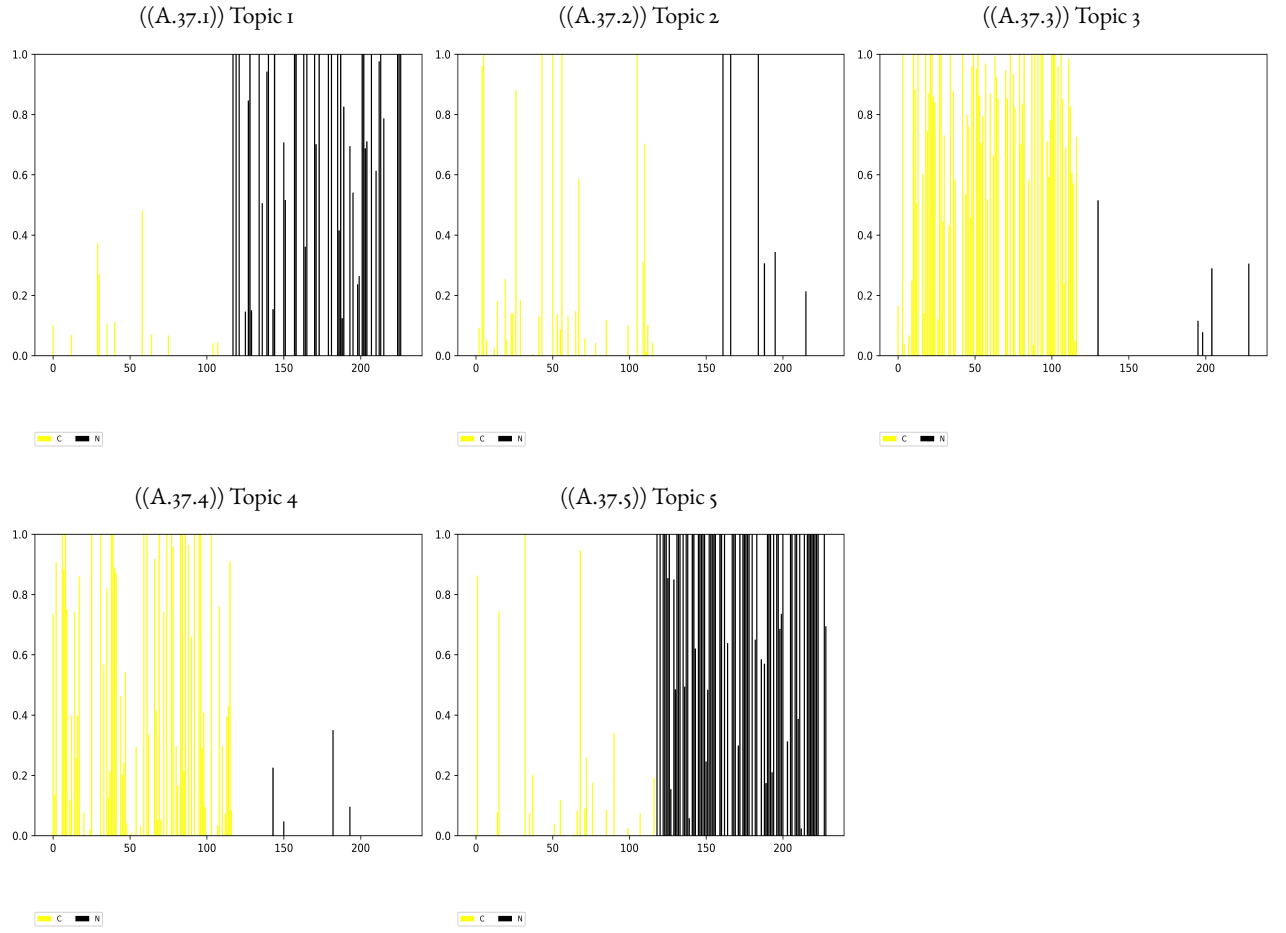


Figure A.37: The distribution over the samples for each topic on the TCGA Dataset obtained by LPD with $t_{LPD} = \lambda$ and $K = 5$. In each figure the bars are the samples and their corresponding probabilities for each topic. The bars are colored according to the disease of the sample.

A.3 Feature Selection using Topic Models Results

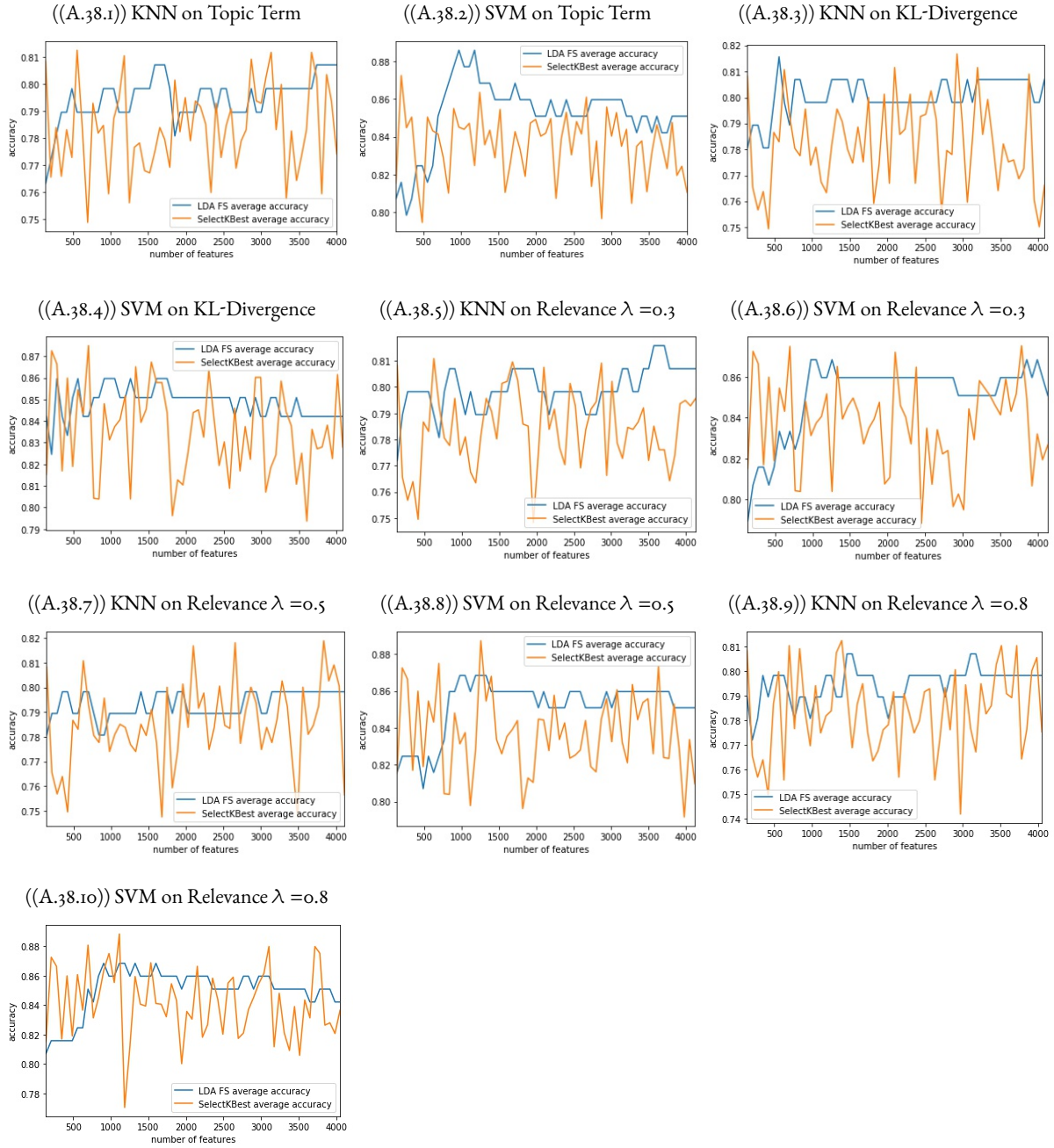


Figure A.38: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \mathbf{X} , $t_2 = \mathbf{X}$) on the GEM.

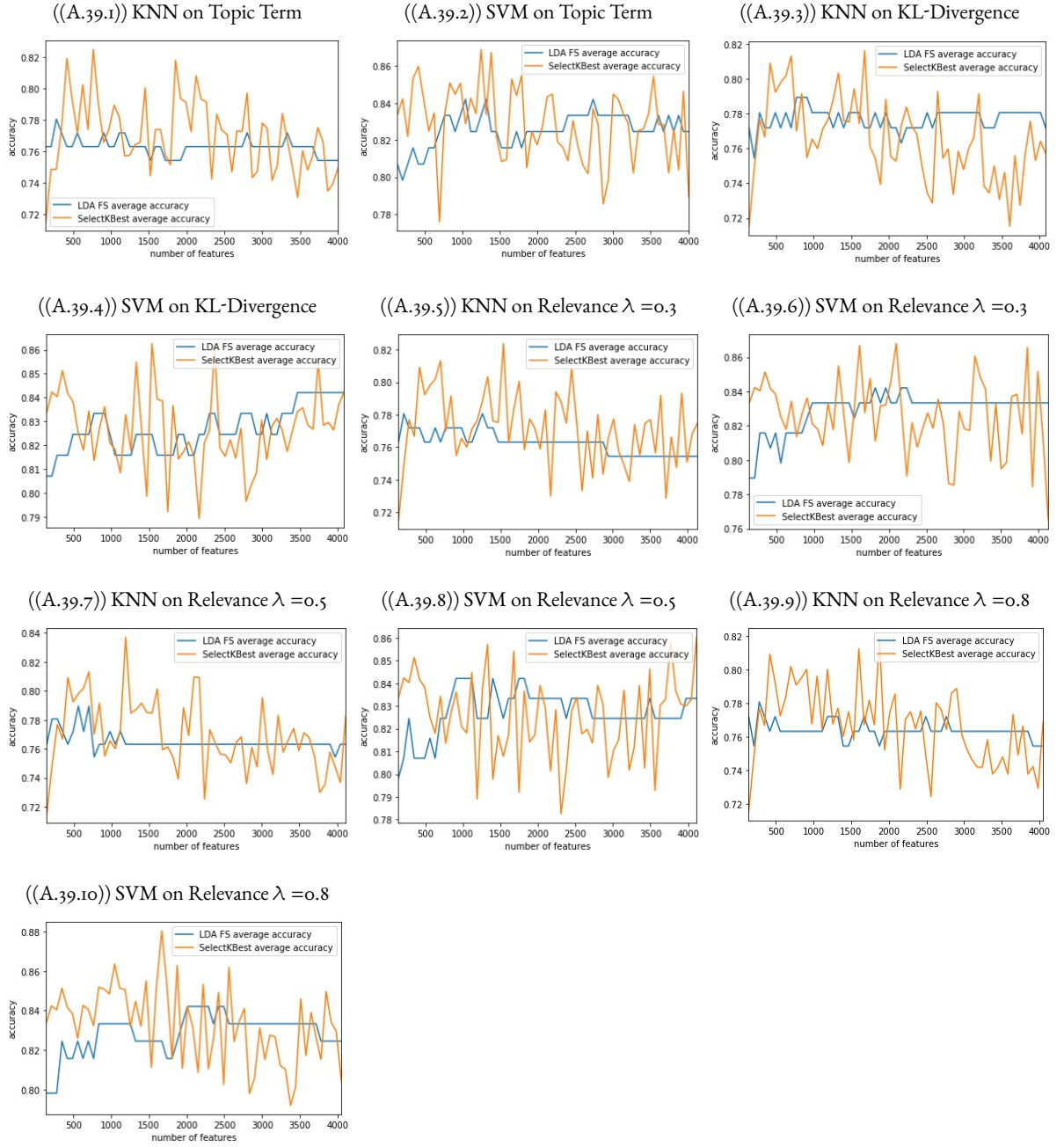


Figure A.39: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \mathbf{X} , $t_2 = \mathbf{X}$) on the count vector.

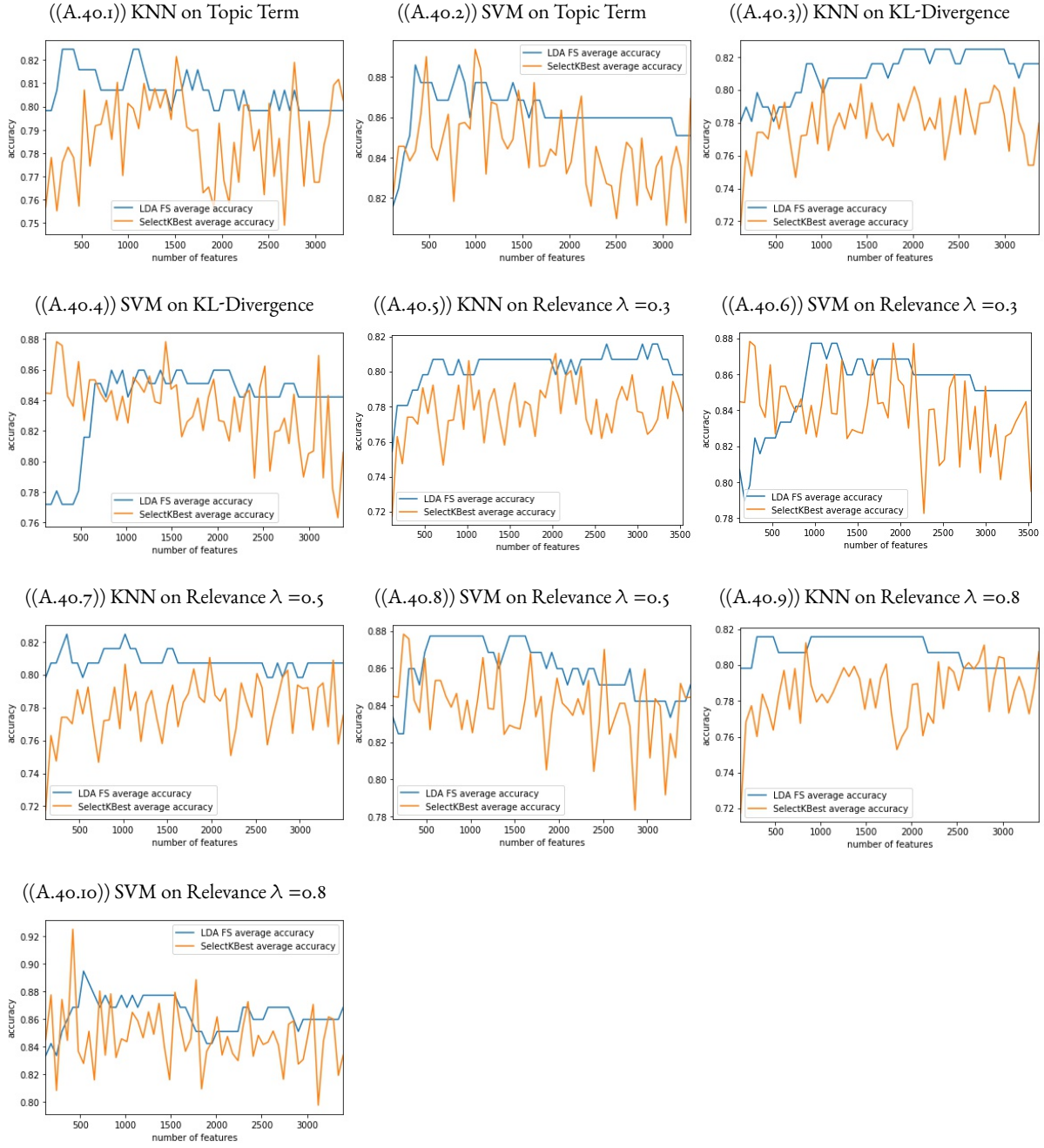


Figure A.40: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \mathbf{X} , $t_2 = 0.2$) on the GEM.

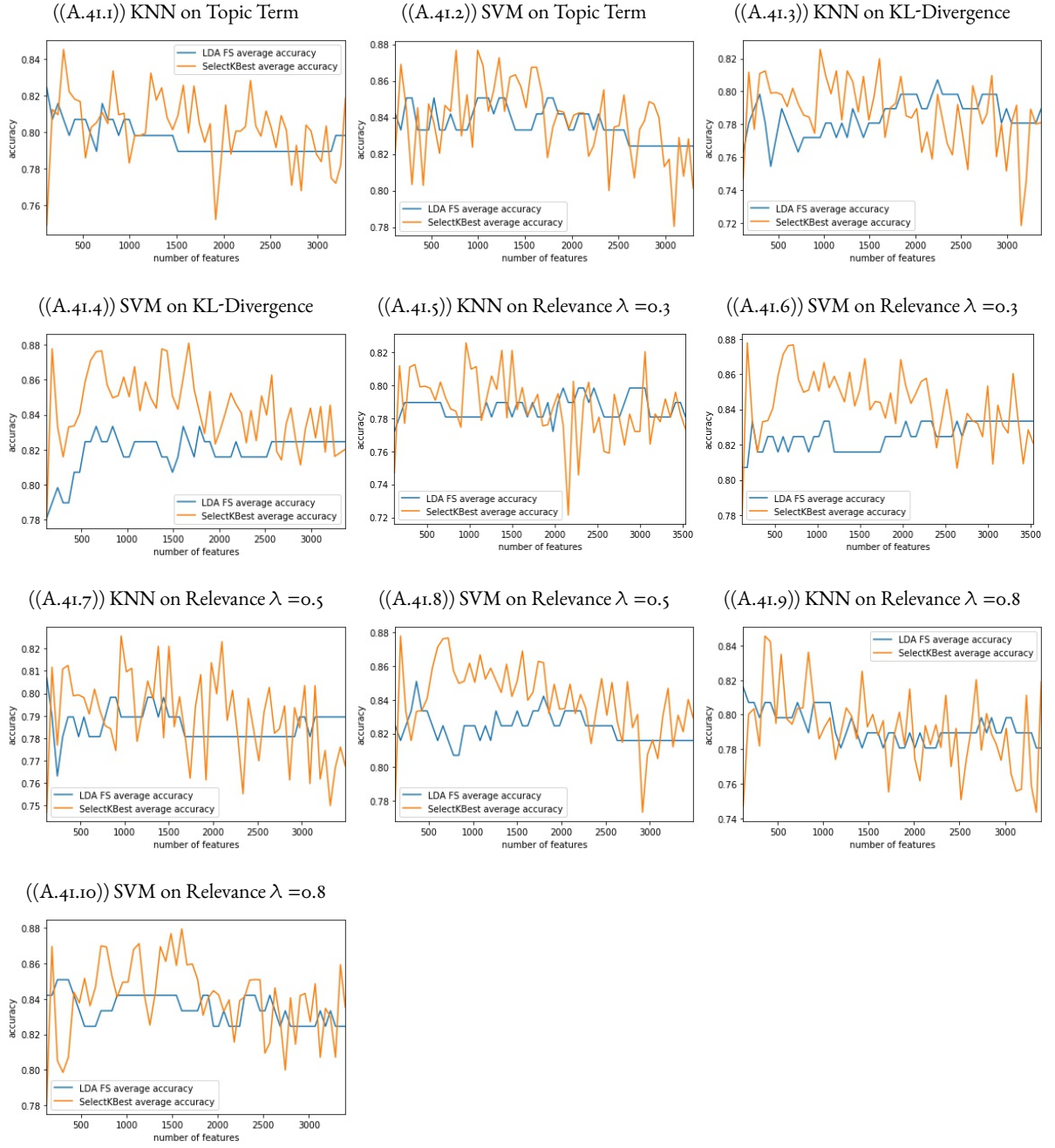


Figure A.41: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \mathbf{X} , $t_2 = 0.2$) on the count vector.



Figure A.42: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \checkmark , $t_2 = 0.0$) on the GEM.

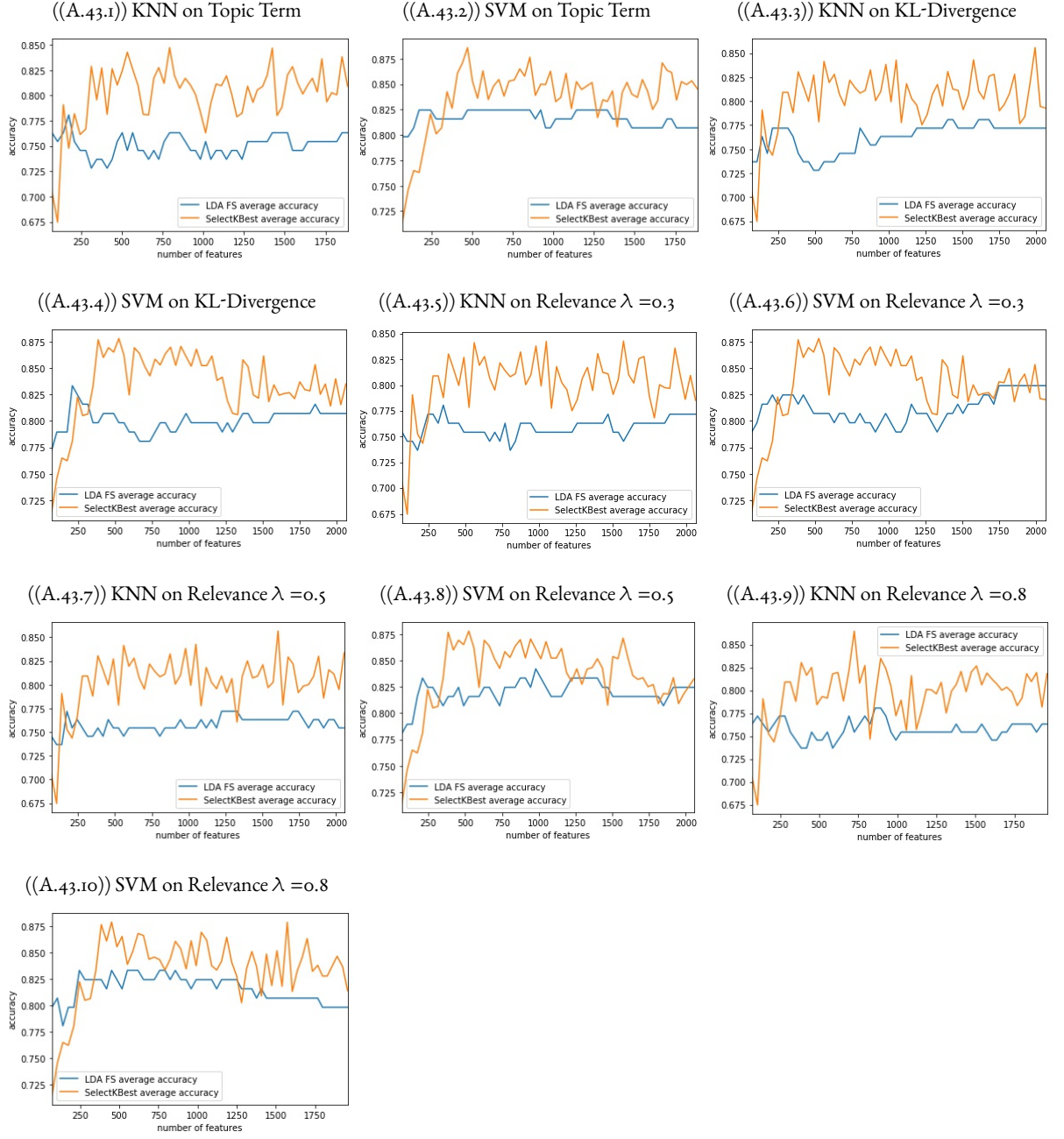


Figure A.43: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \checkmark , $t_2 = 0.0$) on the count vector.

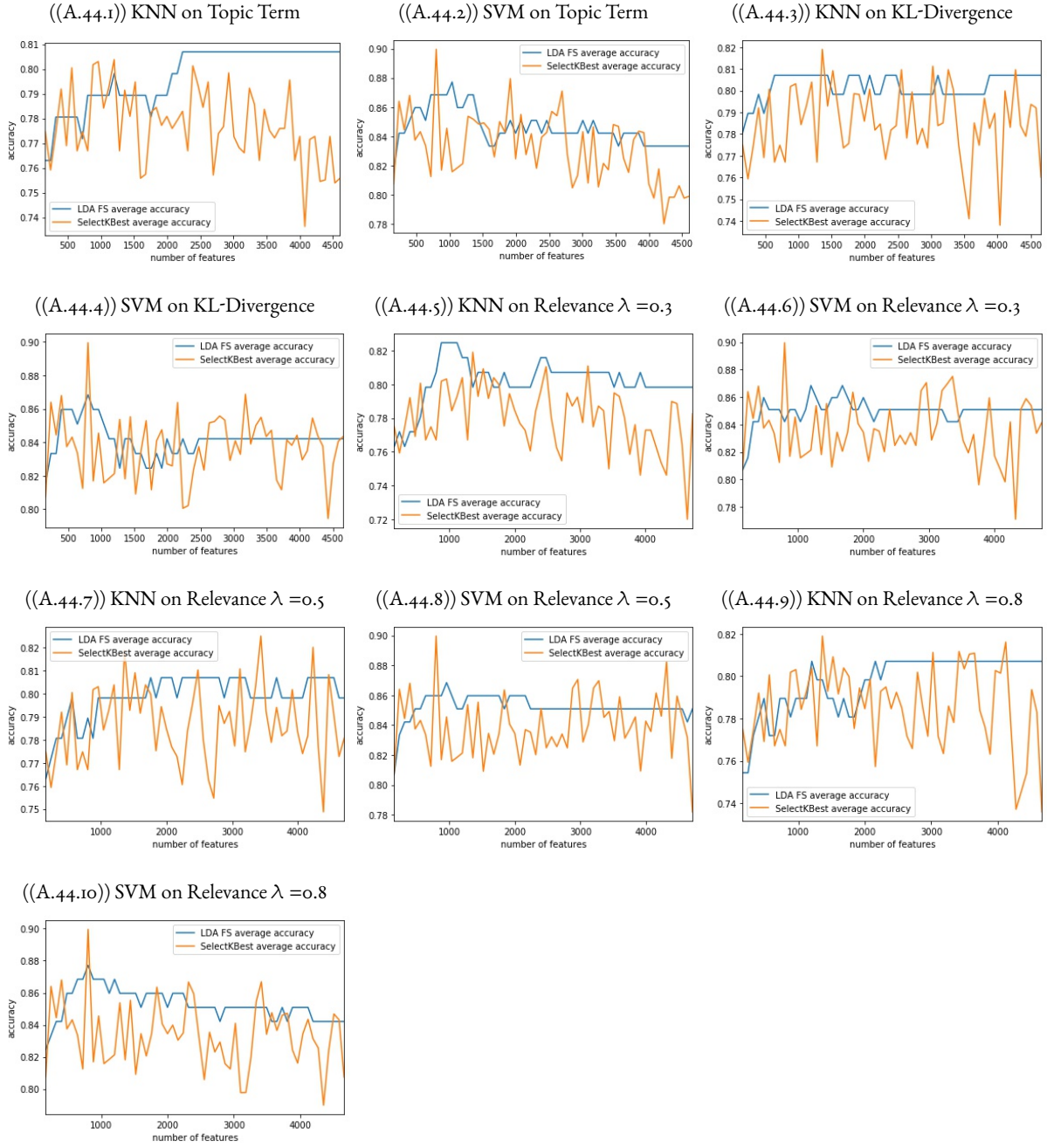


Figure A.44: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \checkmark , $t_2 = \mathbf{X}$) on the GEM.

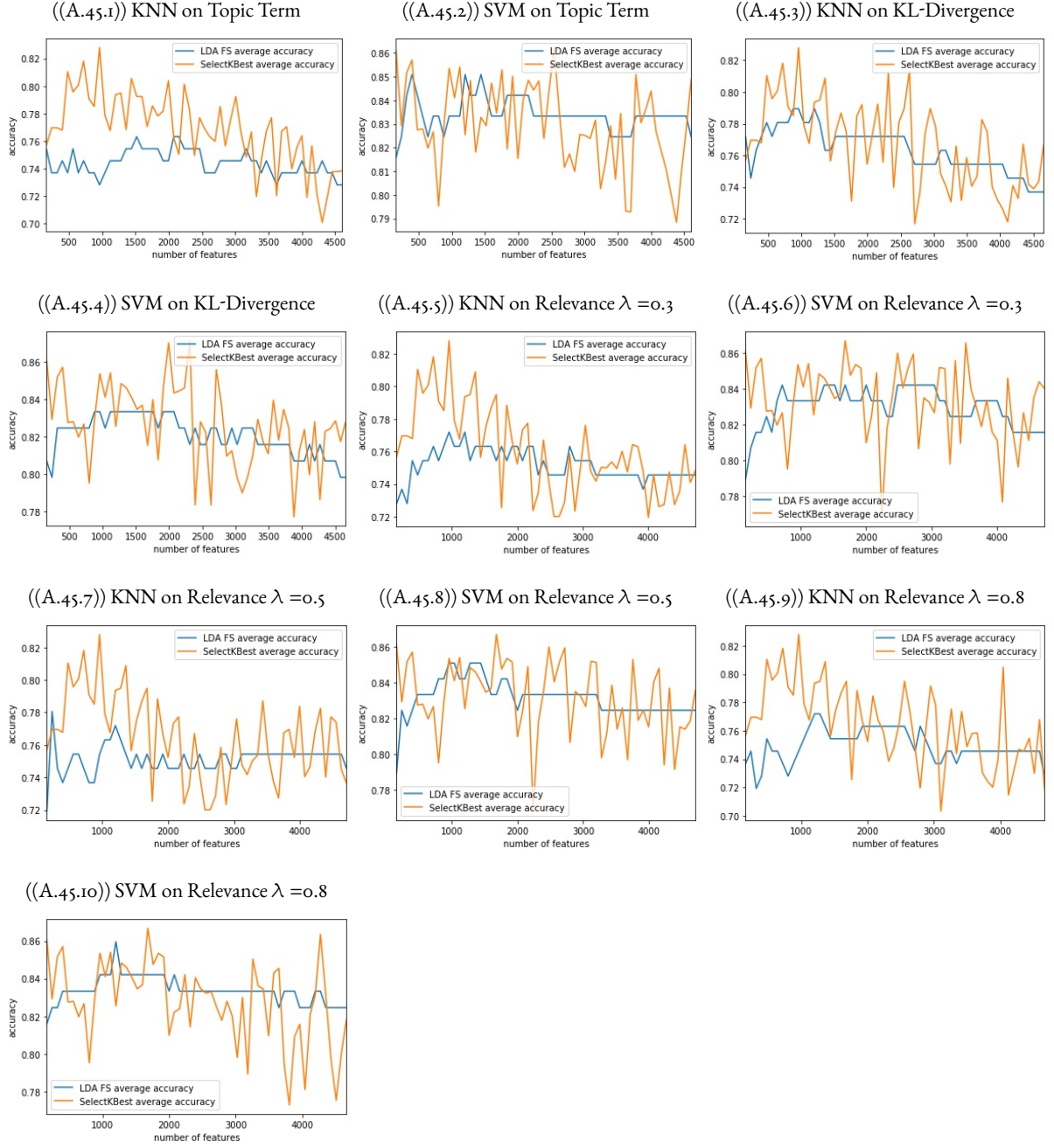


Figure A.45: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \checkmark , $t_2 = \mathbf{X}$) on the count vector.

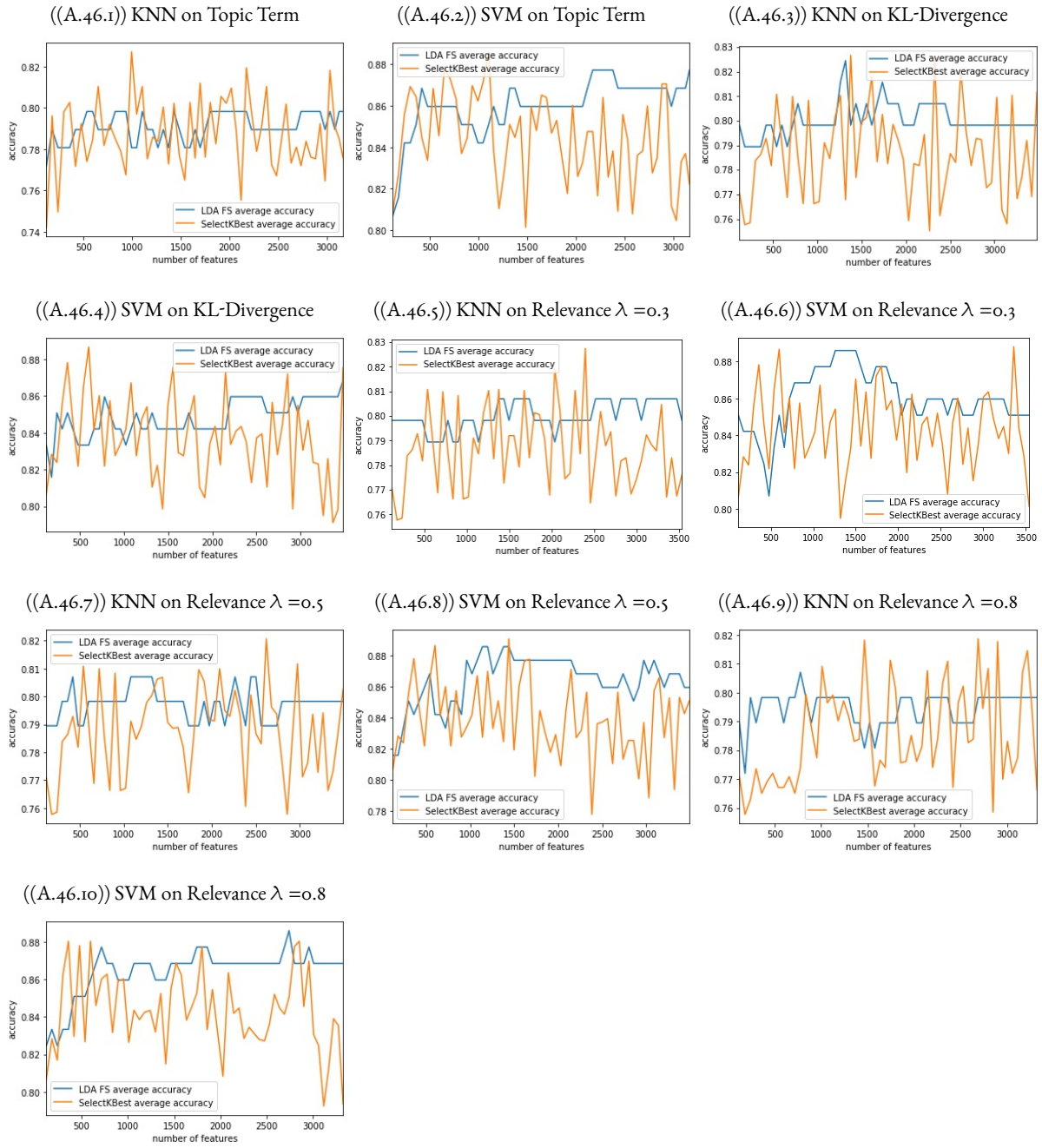


Figure A.46: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0, \text{RemoveBin}=\mathbf{X}, t_2 = \mathbf{X}$) on the GEM.

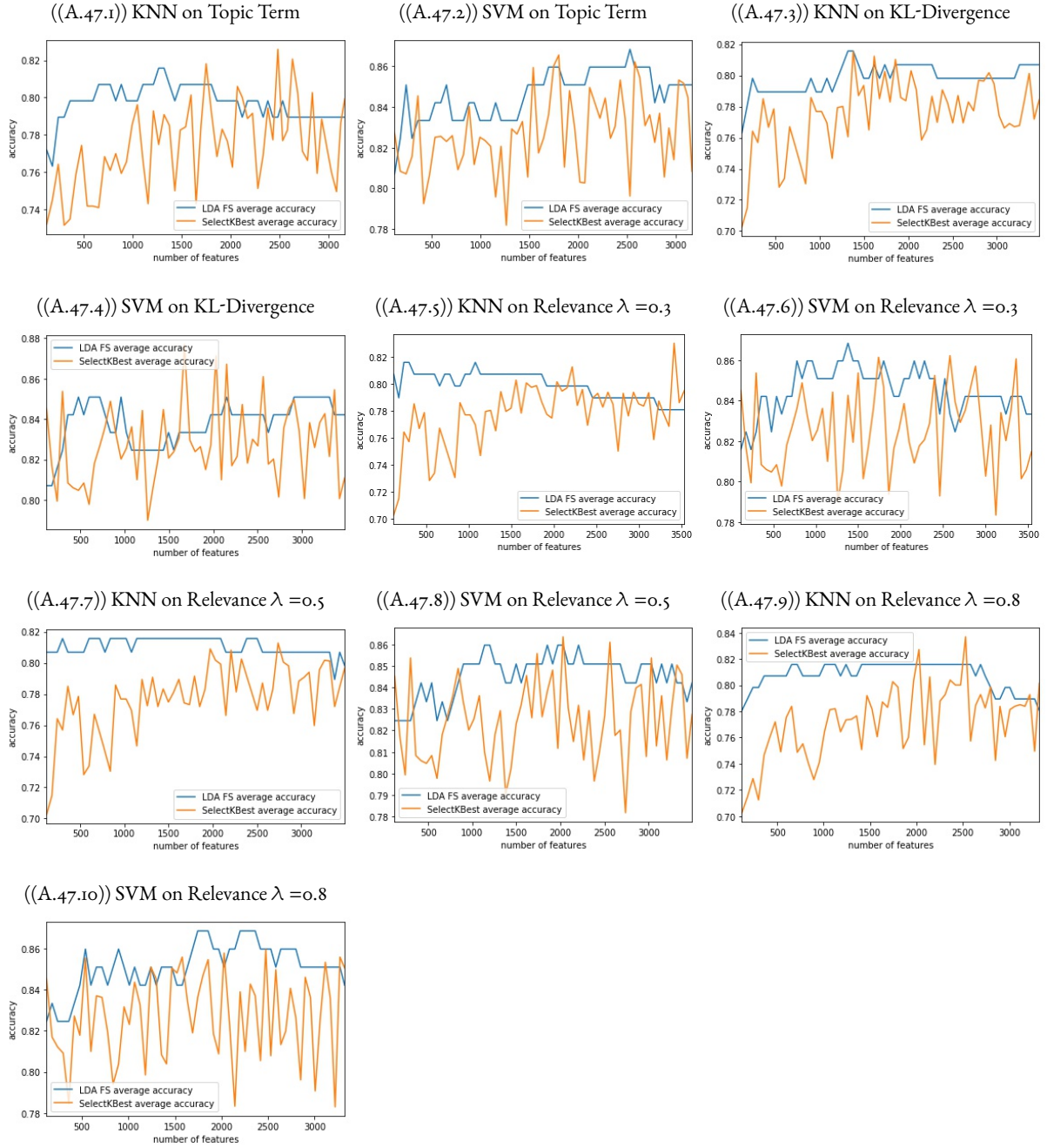


Figure A.47: Unsupervised Feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0$, RemoveBin= \mathbf{X} , $t_2 = \mathbf{X}$) on the count vector.

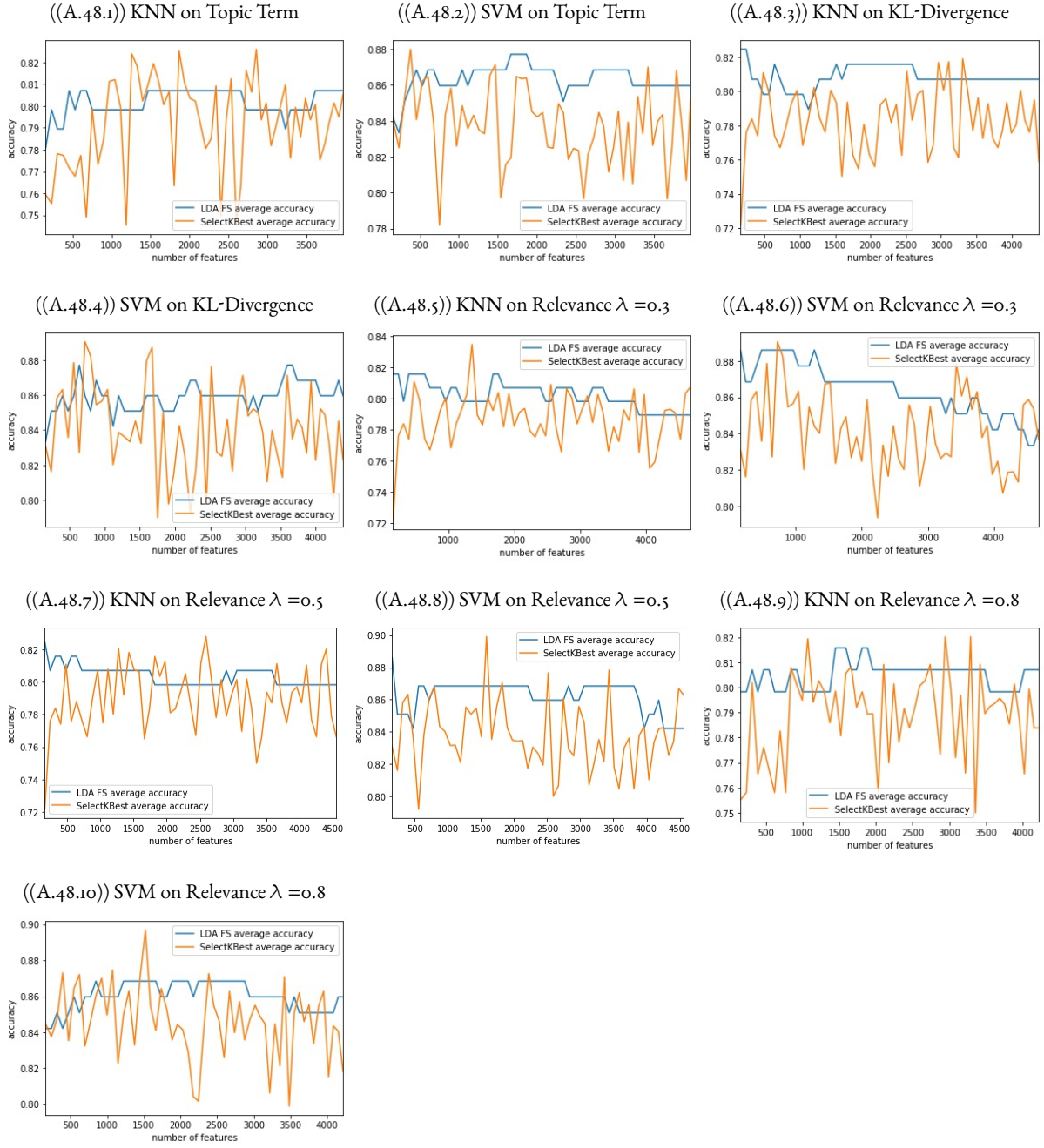


Figure A.48: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0$, RemoveBin= \mathbf{X} , $t_2 = 0.2$) on the GEM.

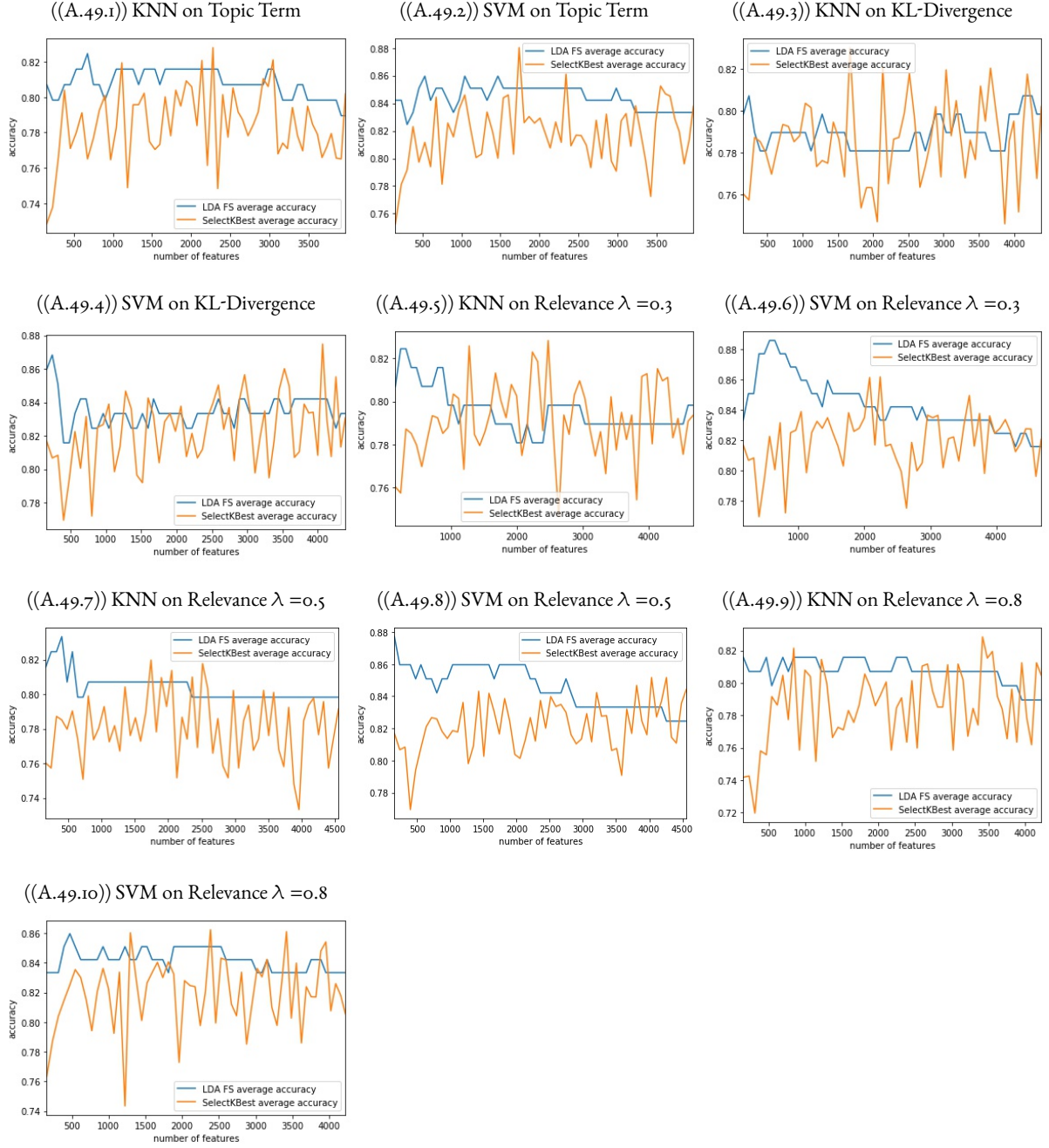


Figure A.49: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0, \text{RemoveBin}=\mathbf{X}, t_2 = 0.2$) on the count vector.

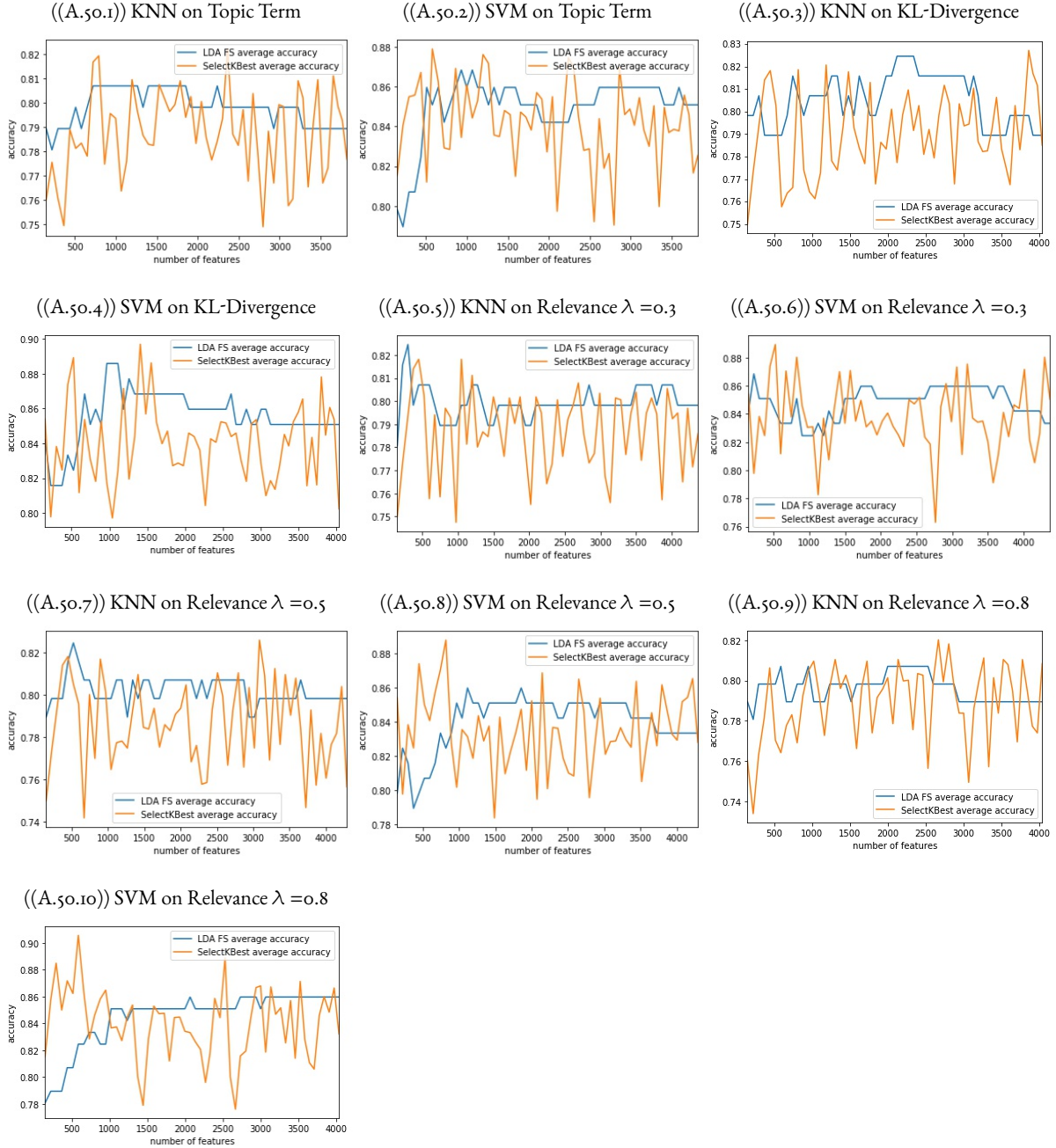


Figure A.50: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0, \text{RemoveBin}=\sqrt{\cdot}, t_2 = \times$) on the GEM.

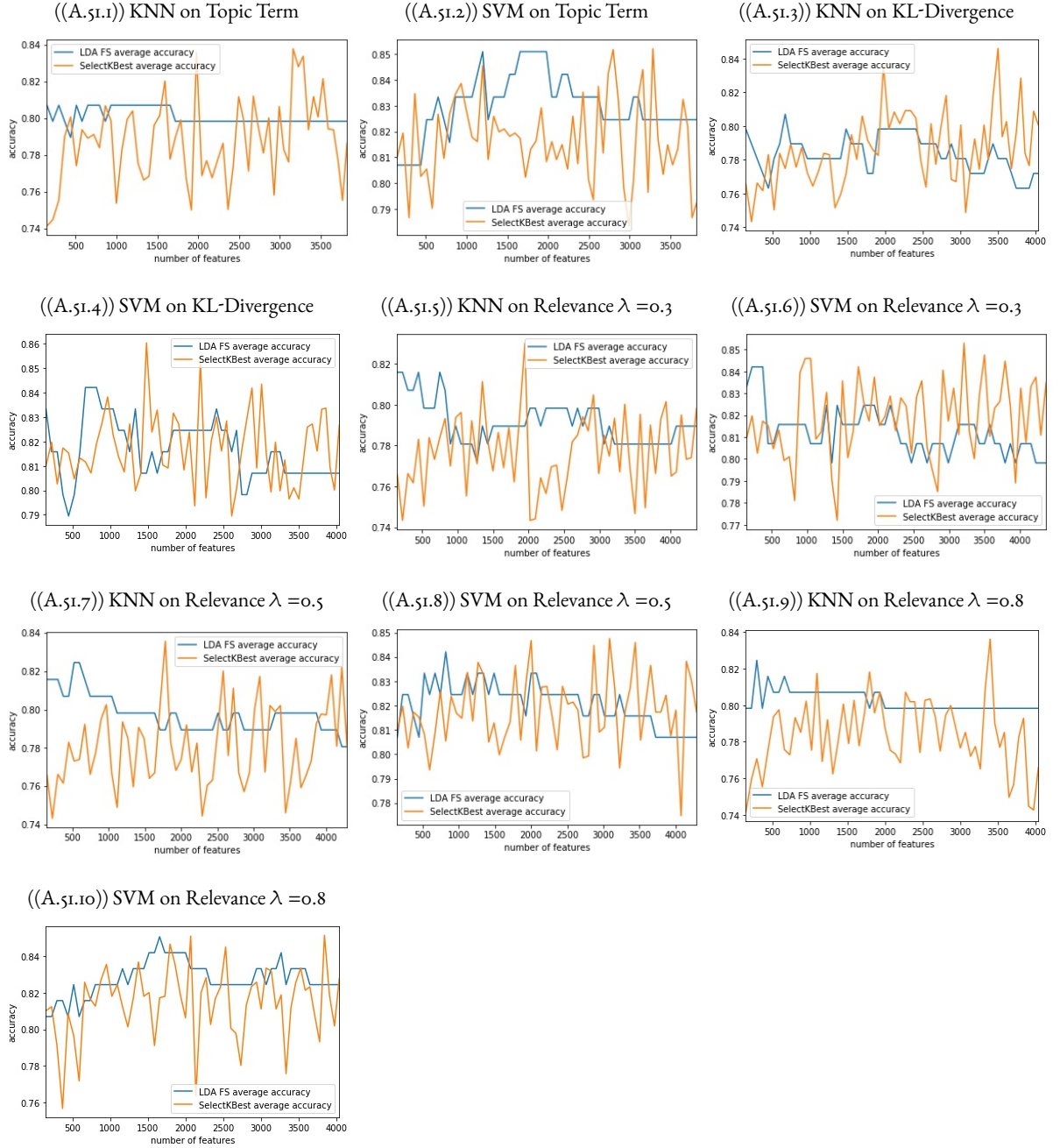


Figure A.51: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0, \text{RemoveBin}=\checkmark, t_2 = \times$) on the count vector.

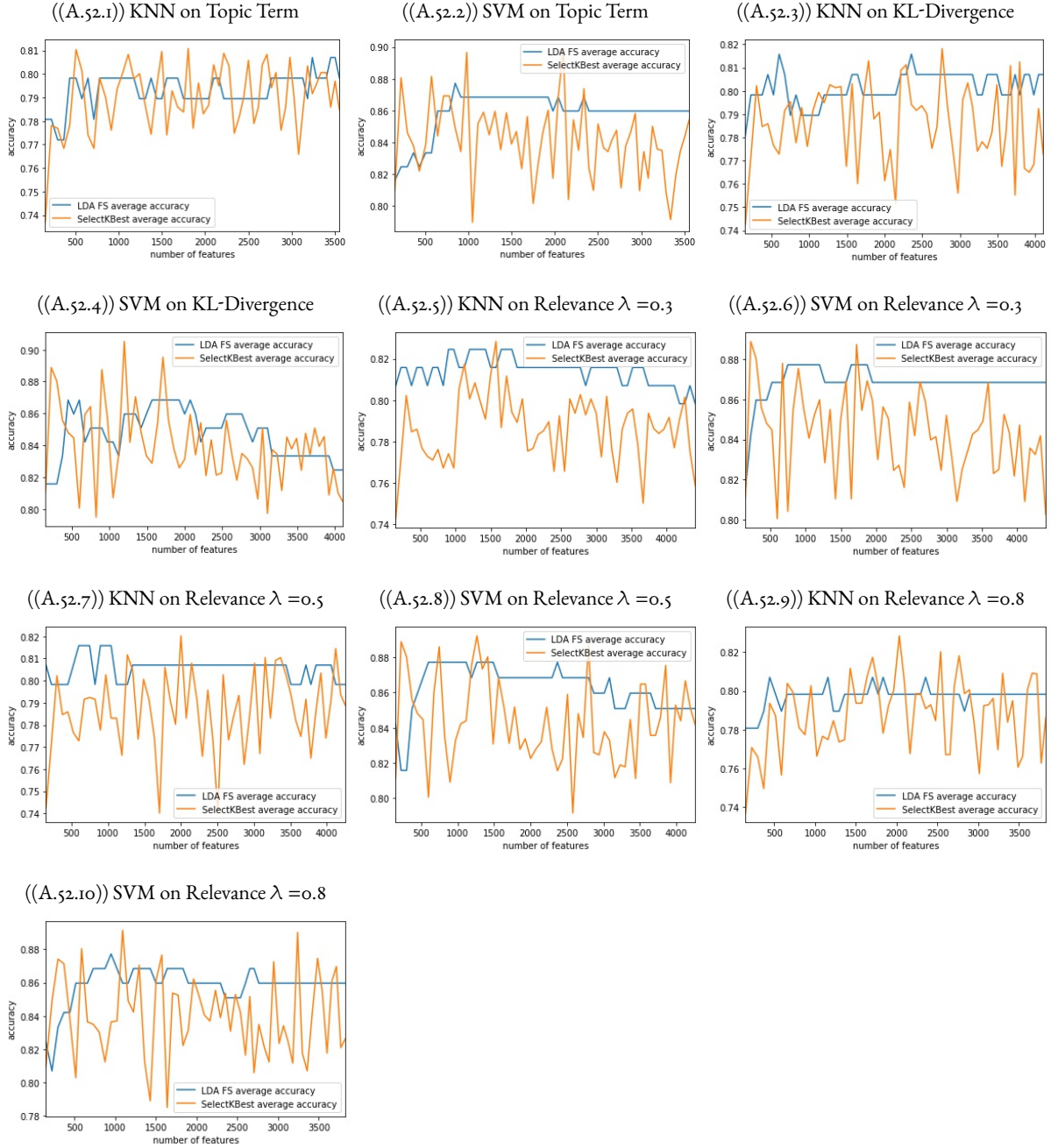


Figure A.52: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0.2$, RemoveBin= \mathbf{X} , $t_2 = \mathbf{X}$) on the GEM.

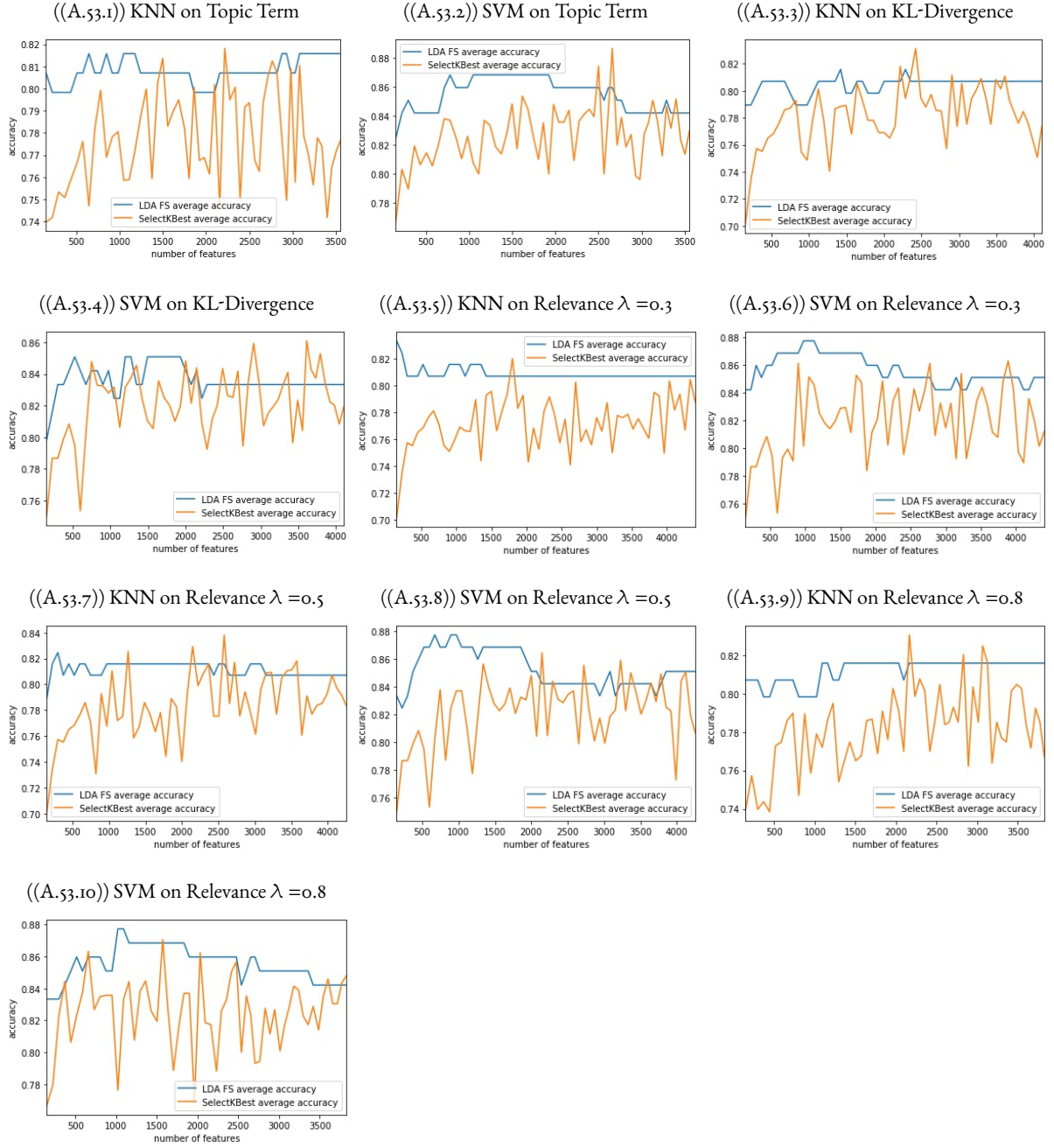


Figure A.53: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0.2, \text{RemoveBin}=\mathbf{X}, t_2 = \mathbf{X}$) on the count vector.

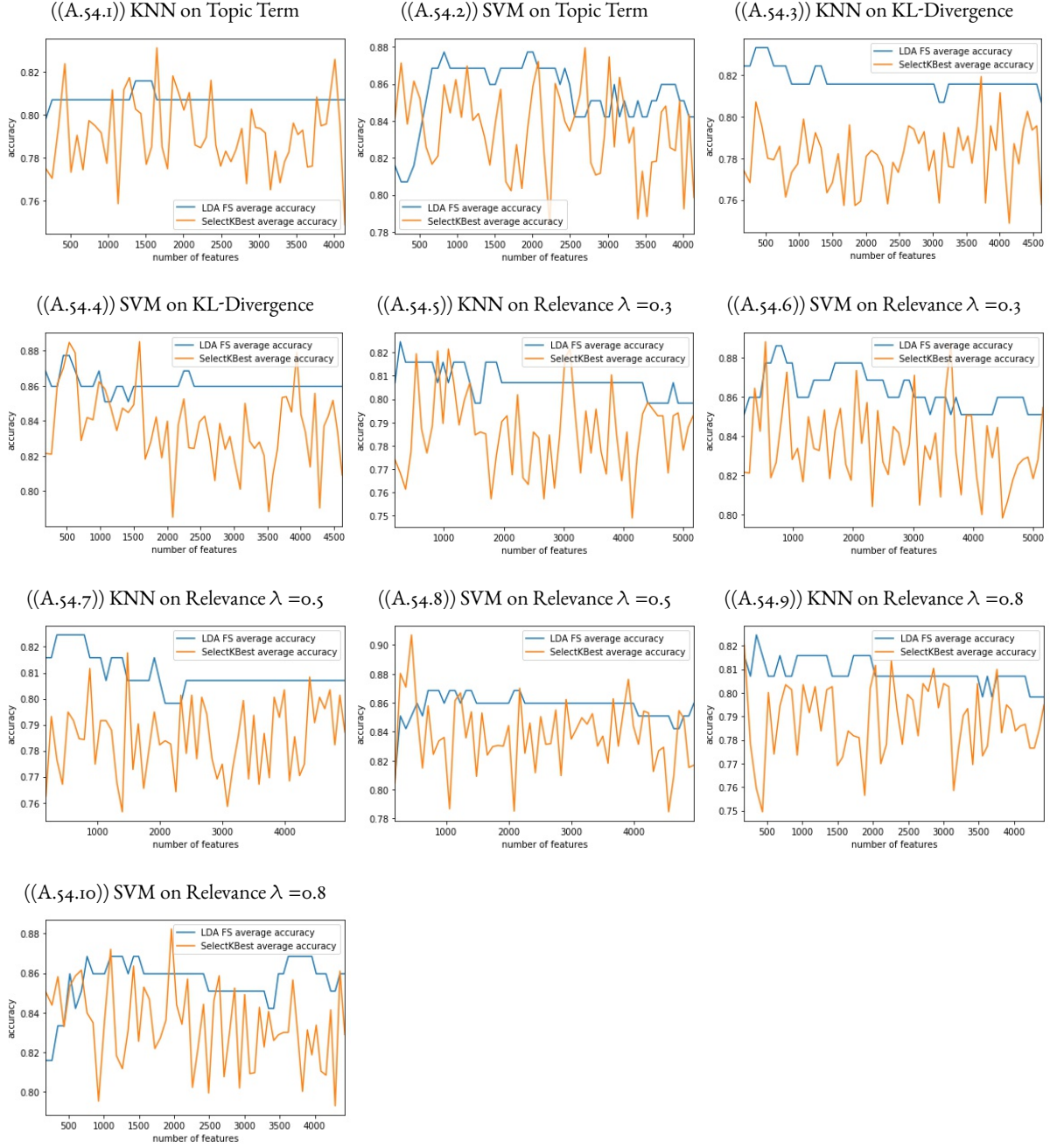


Figure A.54: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0.2$, RemoveBin= \mathbf{X} , $t_2 = 0.3$) on the GEM.

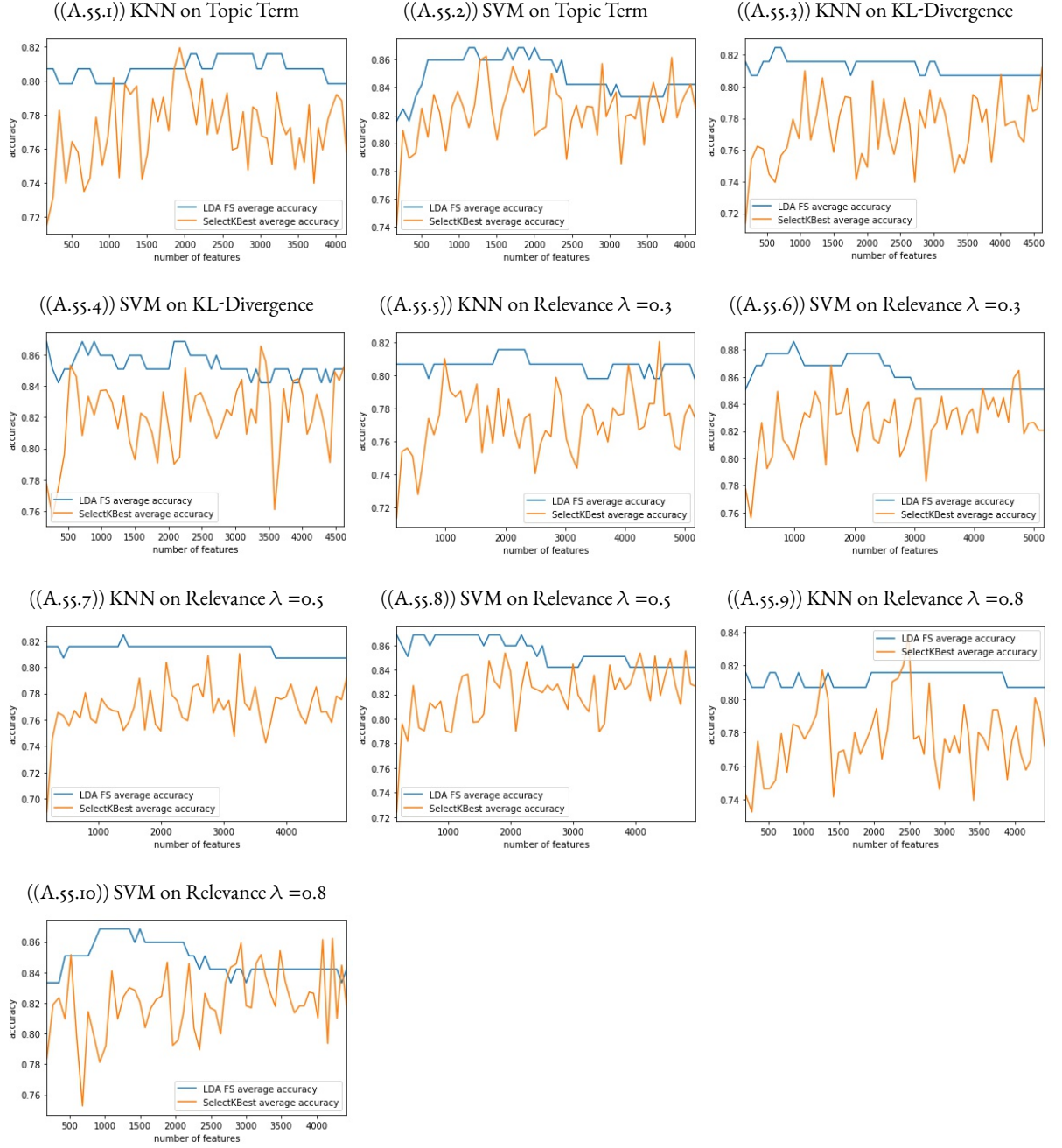


Figure A.55: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0.2$, RemoveBin= \mathbf{X} , $t_2 = 0.3$) on the count vector.

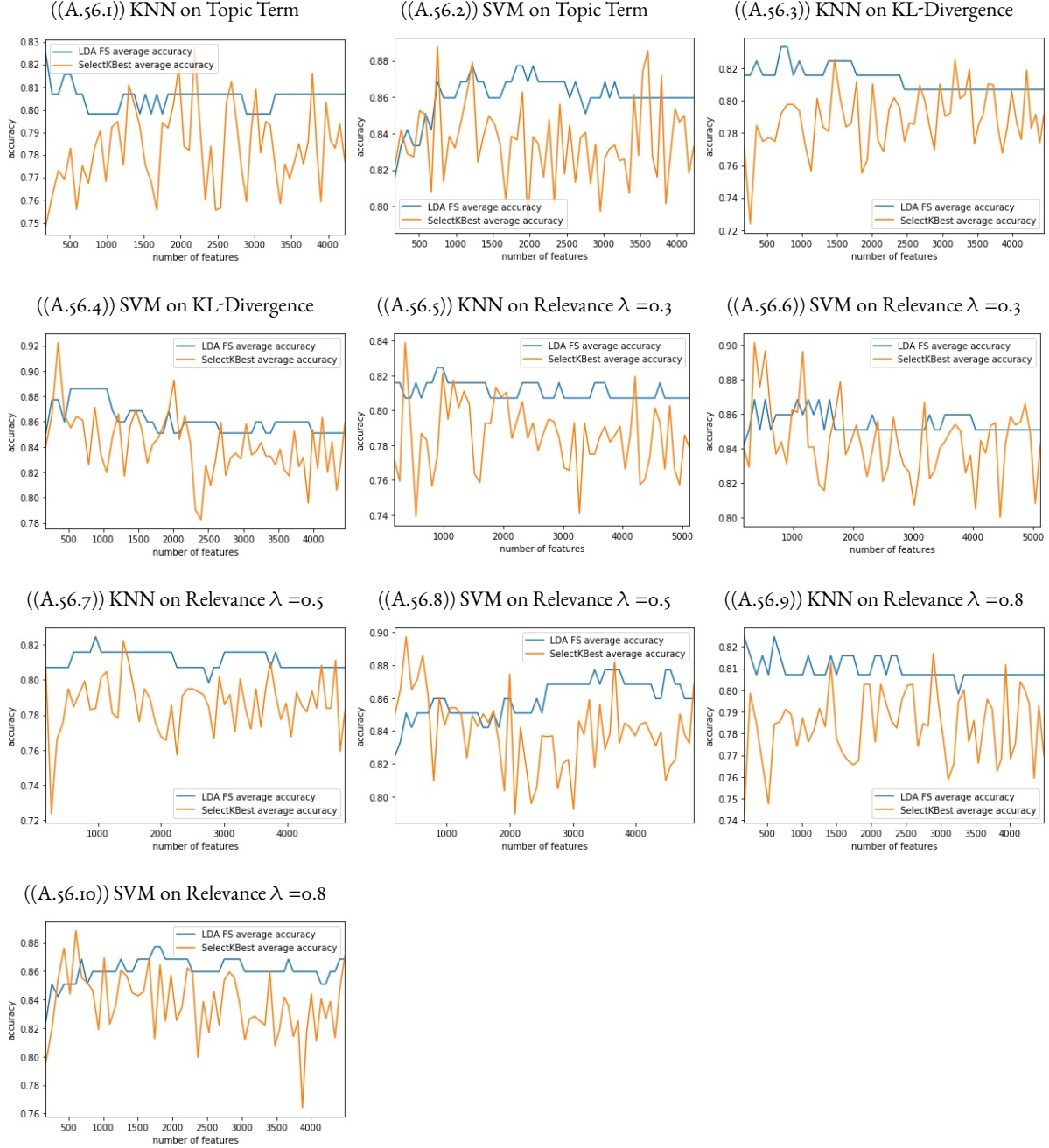


Figure A.56: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0.2, \text{RemoveBin}=\sqrt{\cdot}, t_2 = \mathbf{X}$) on the GEM.

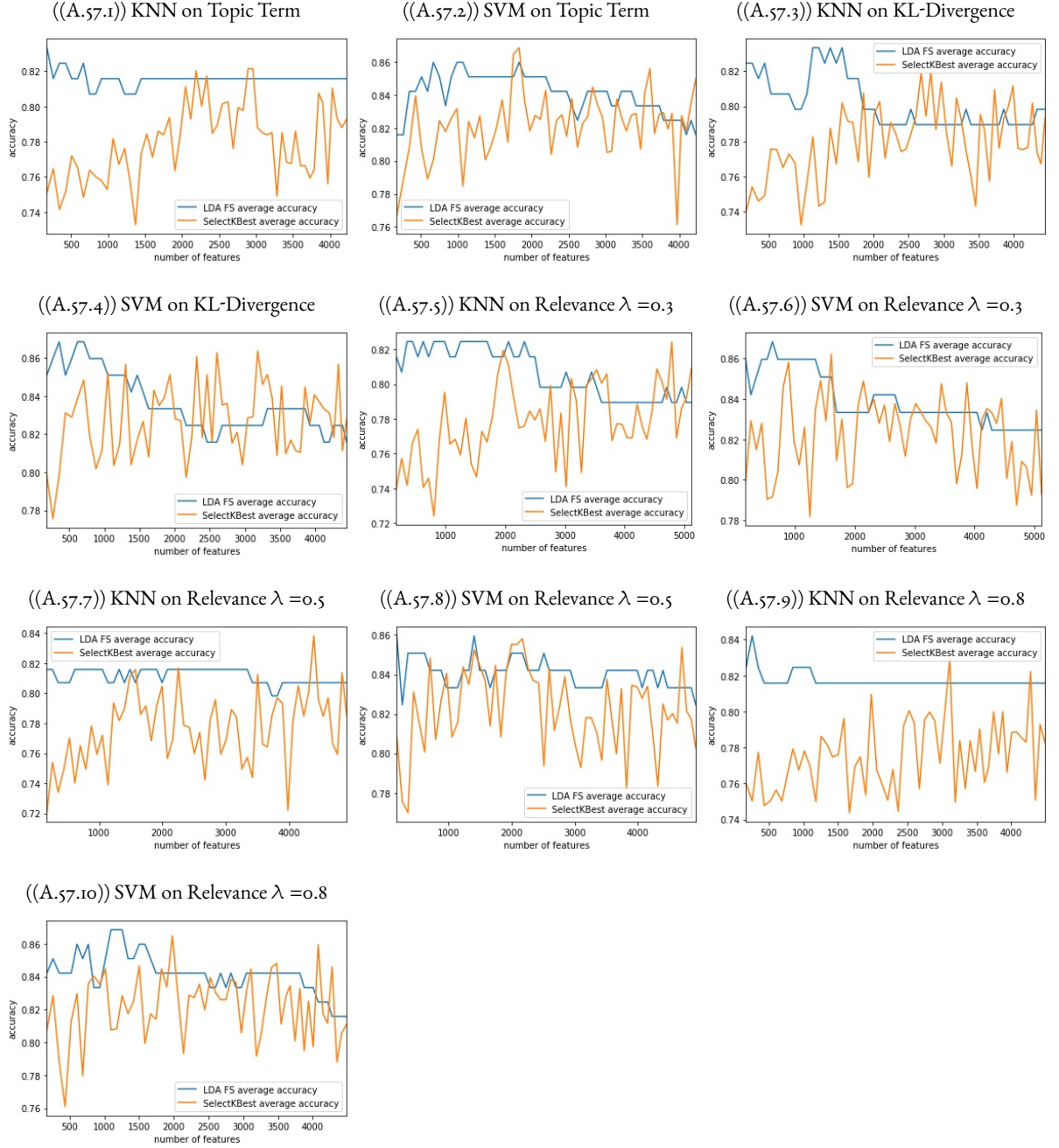


Figure A.57: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Repetition variant ($t_1 = 0.2$, RemoveBin= \checkmark , $t_2 = \times$) on the count vector.

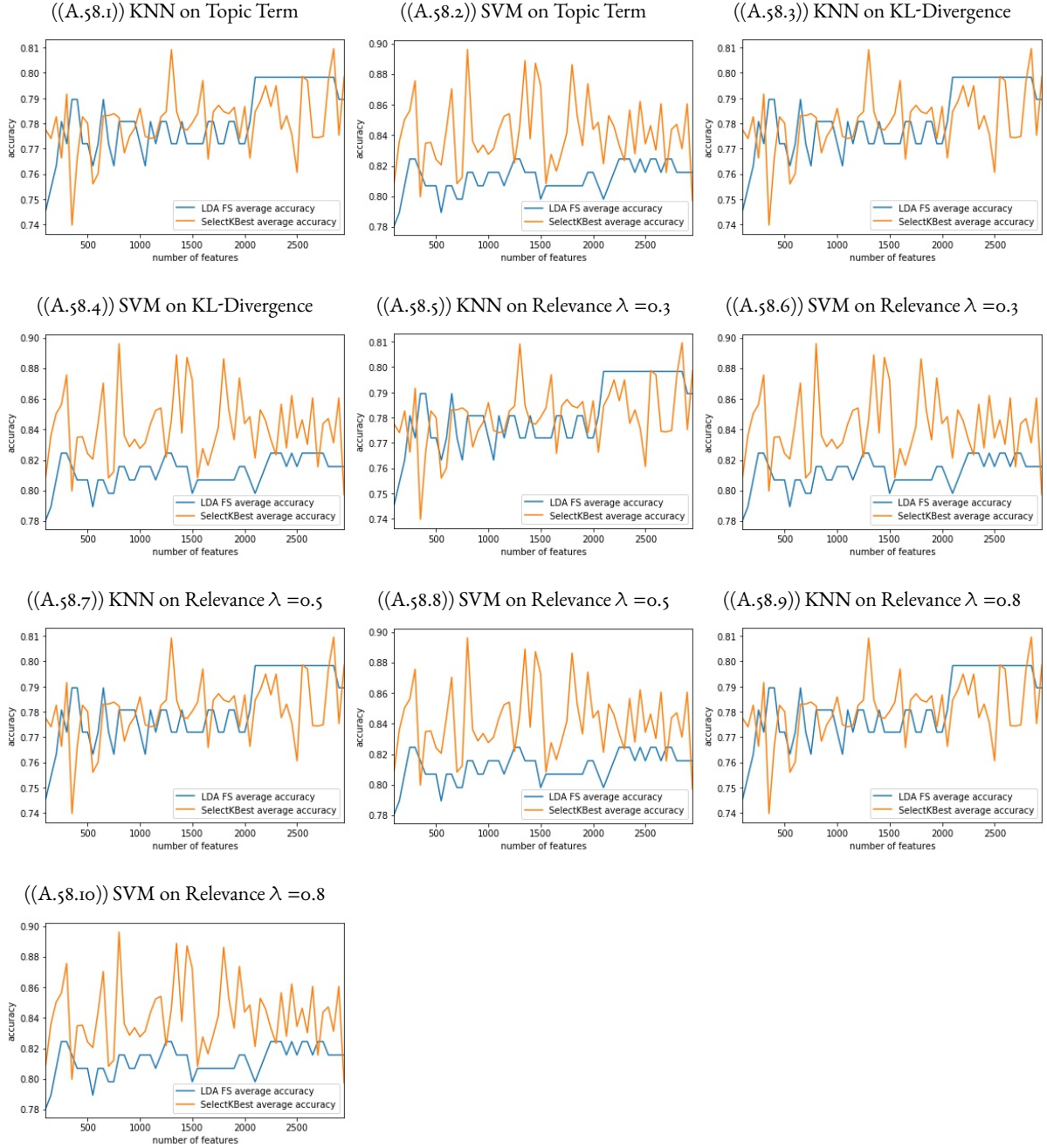


Figure A.58: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Median variant with $t_M = \mathbf{X}$ on the GEM.

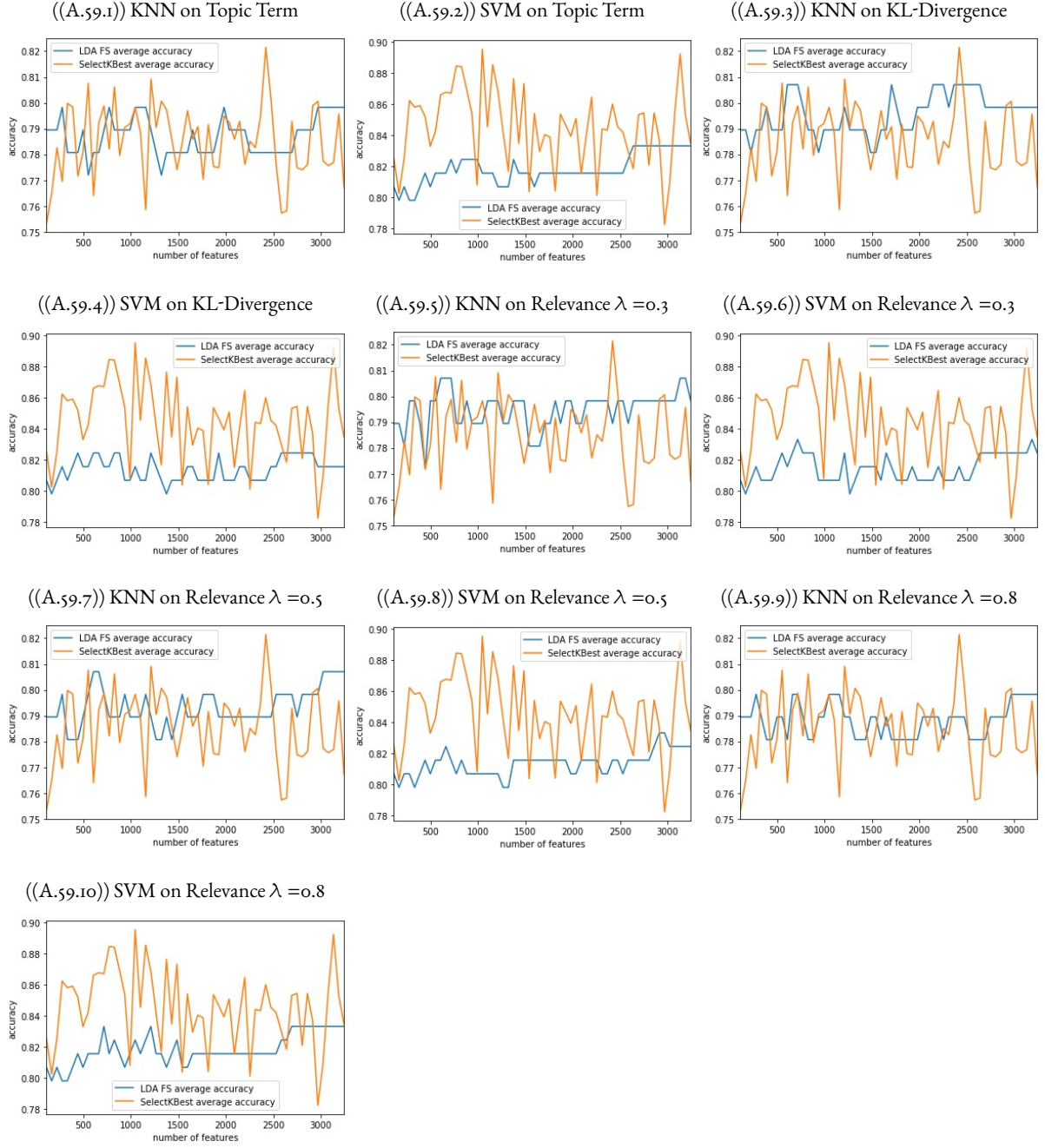


Figure A.59: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Median variant with $t_M = 0.0$ on the GEM.



Figure A.60: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Median variant with $t_M = 0.2$ on the GEM.



Figure A.61: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Bibin variant with $t_B = \mathbf{X}$ on the GEM.

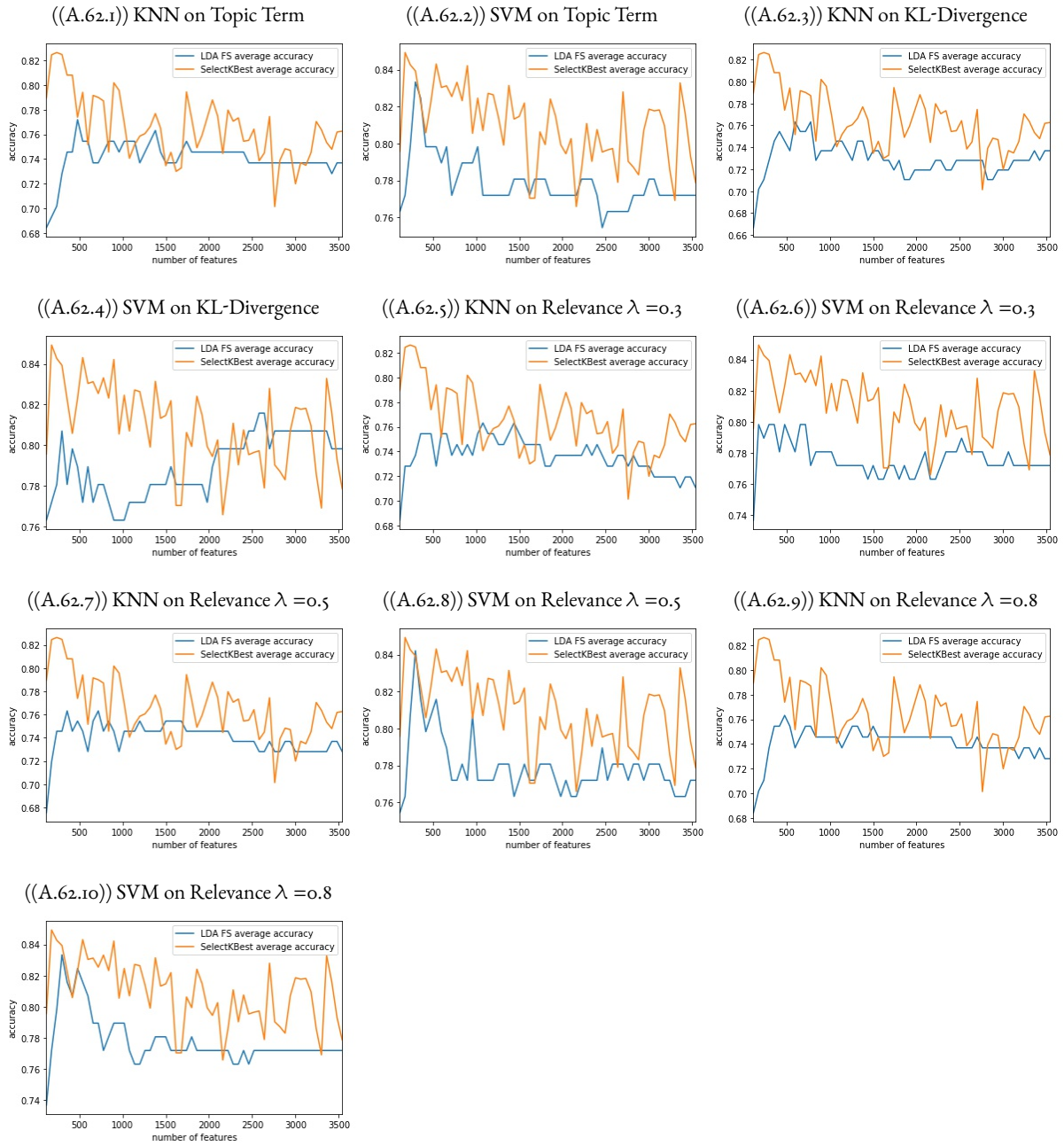


Figure A.62: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Bibin variant with $t_B = \mathbf{X}$ on the count vector.



Figure A.63: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Bibin variant with $t_B = 0$ on the GEM.



Figure A.64: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Bibin variant with $t_B = 0$ on the count vector.

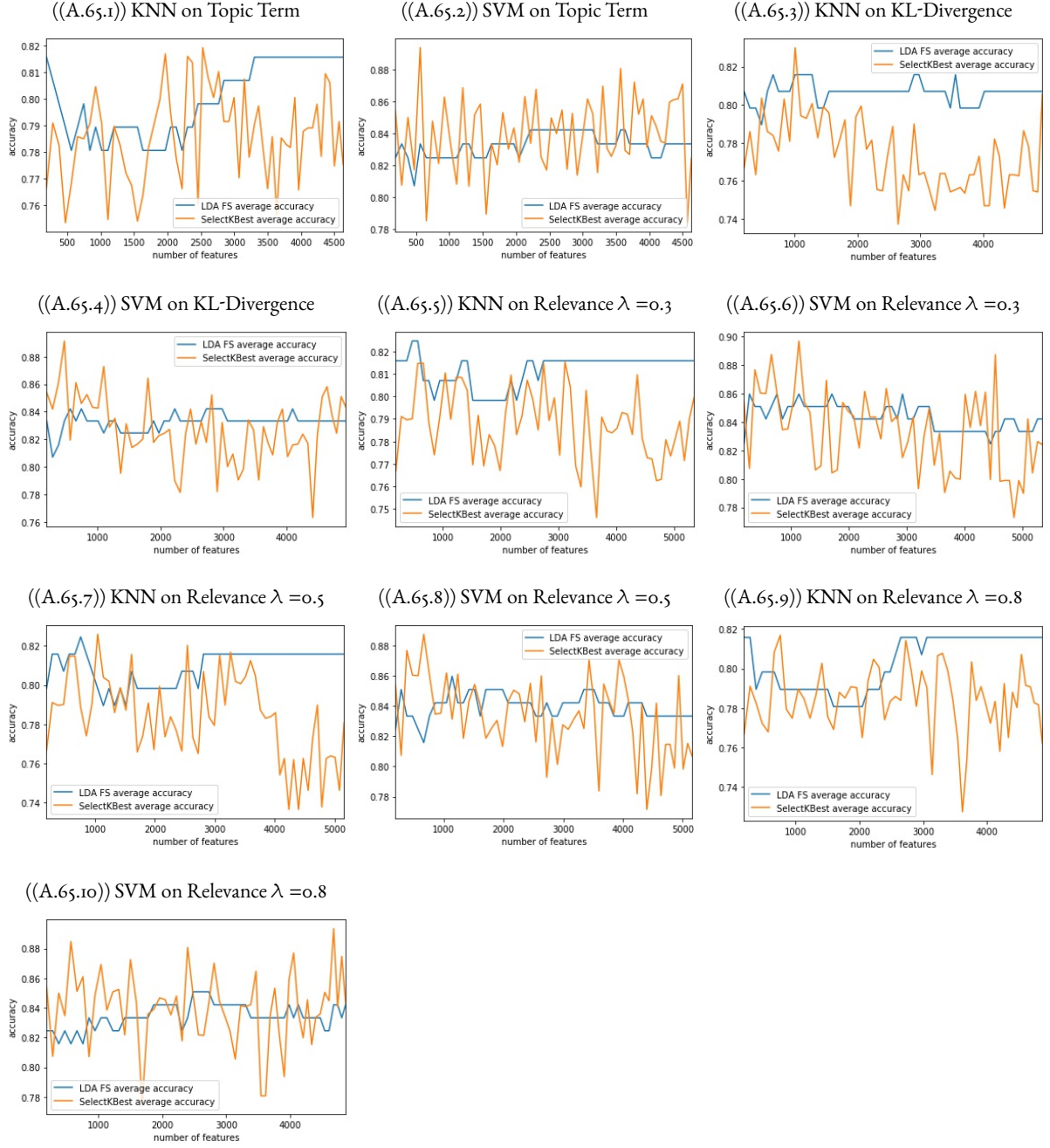


Figure A.65: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Bibin variant with $t_B = 0.2$ on the GEM.



Figure A.66: Unsupervised feature Selection using LDA and Univariate Feature Selection on MD Dataset using Bibin variant with $t_B = 0.2$ on the count vector.

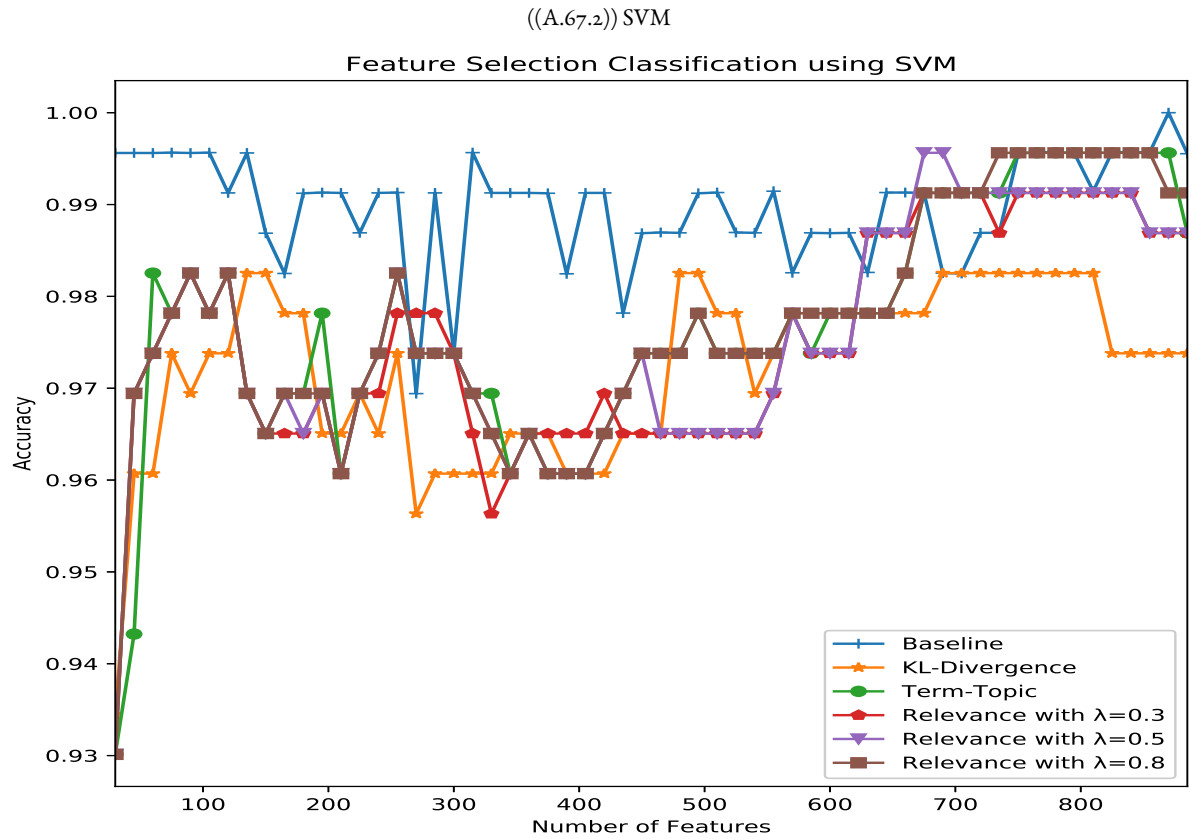
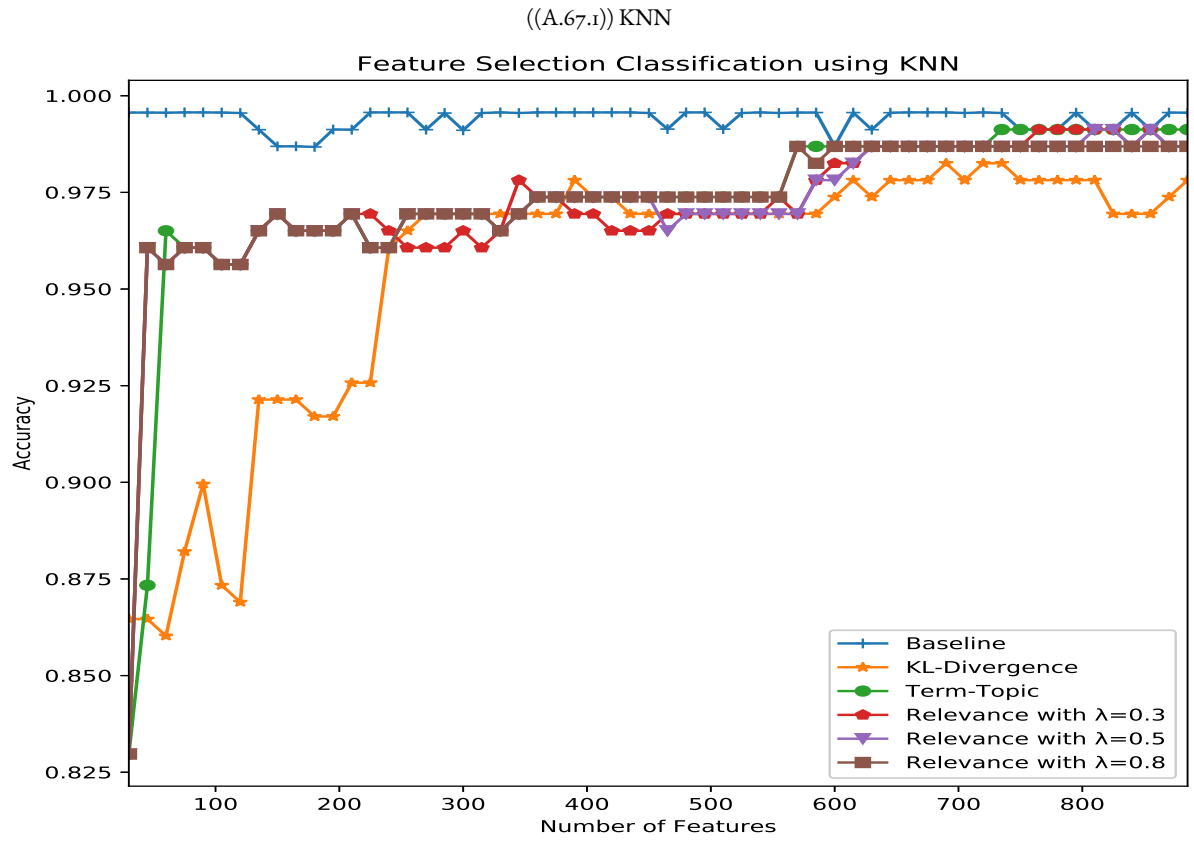


Figure A.67: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Bibin variant with $t_B = \mathbf{X}$ on the GEM.

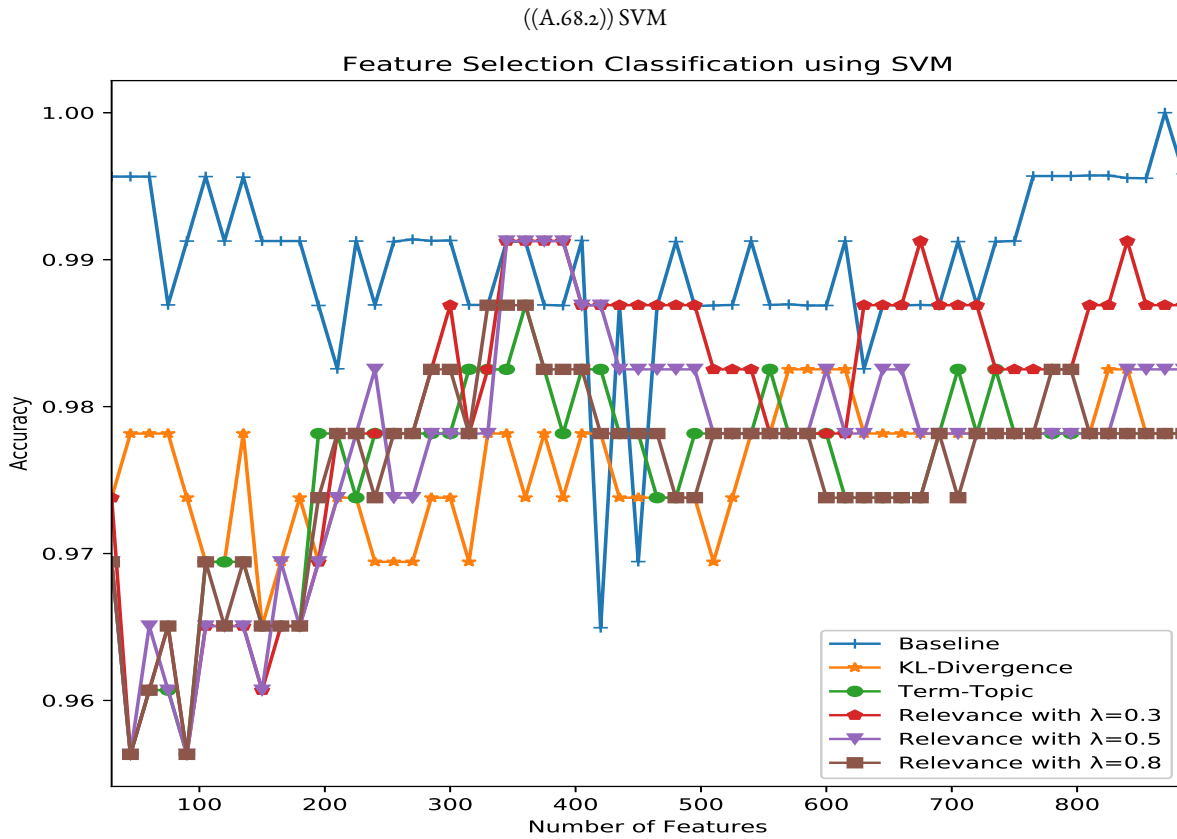
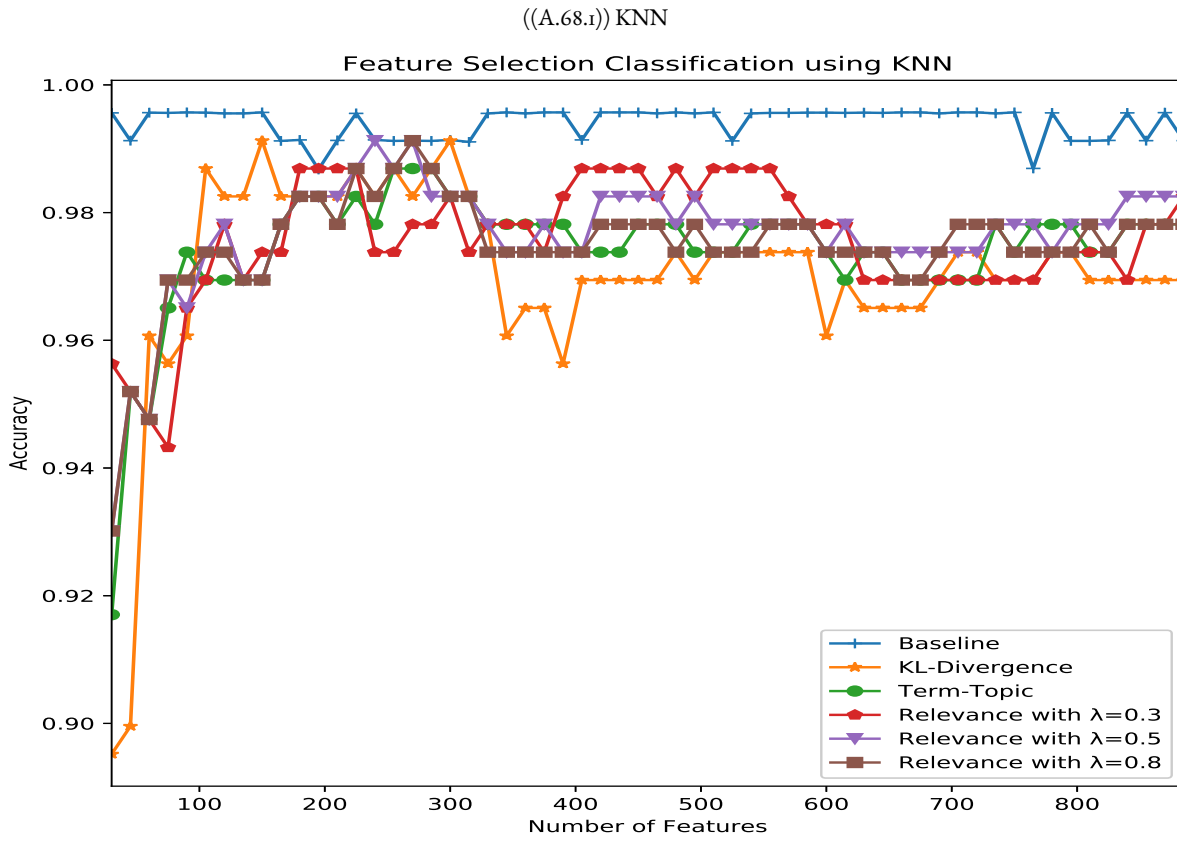


Figure A.68: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Bibin variant with $t_B = 0.2$ on the GEM.

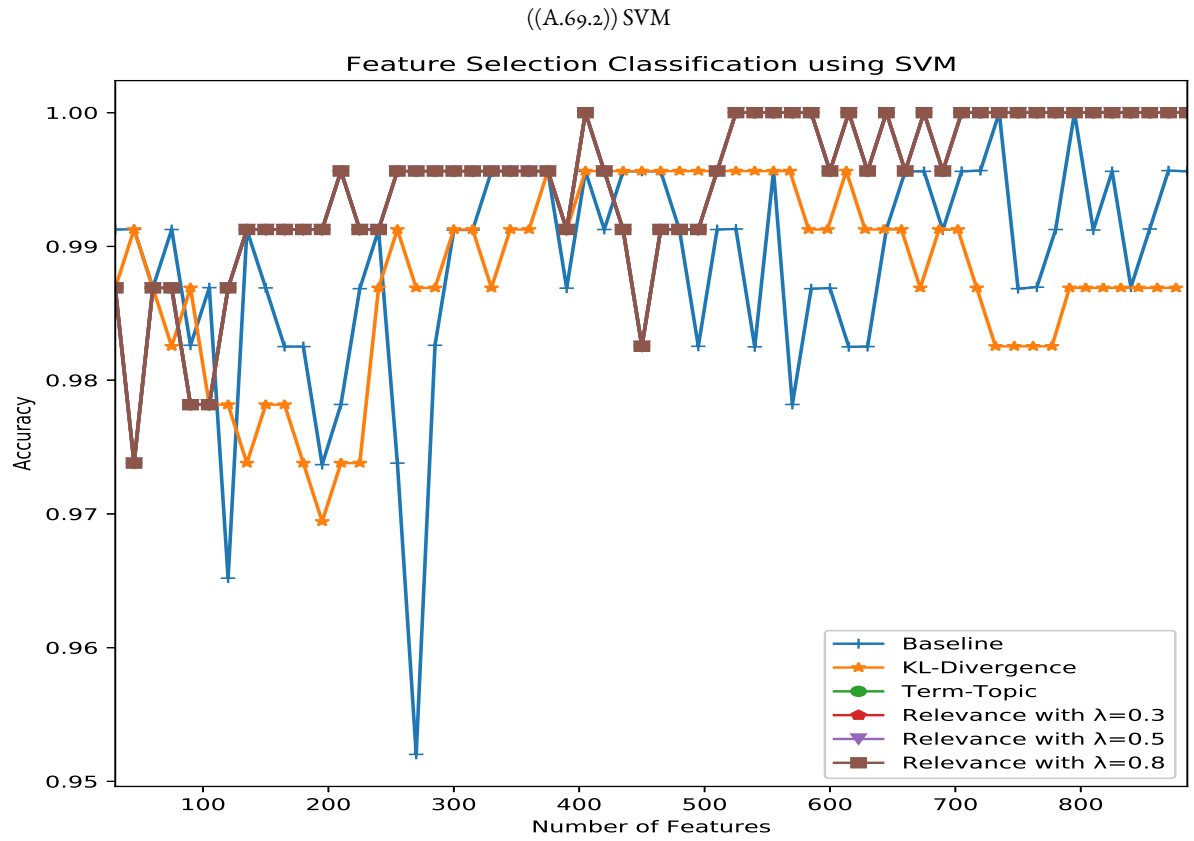
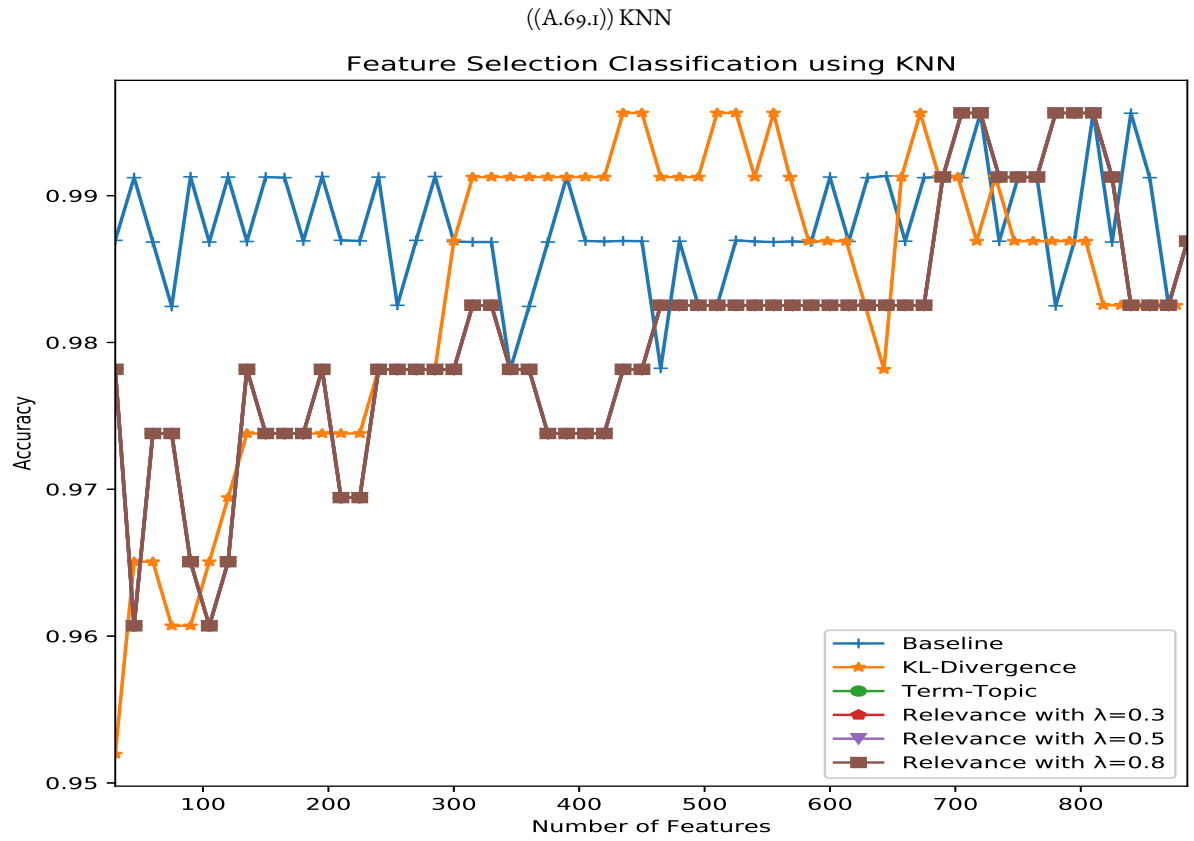


Figure A.69: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Median variant with $t_M = \mathbf{X}$ on the GEM.

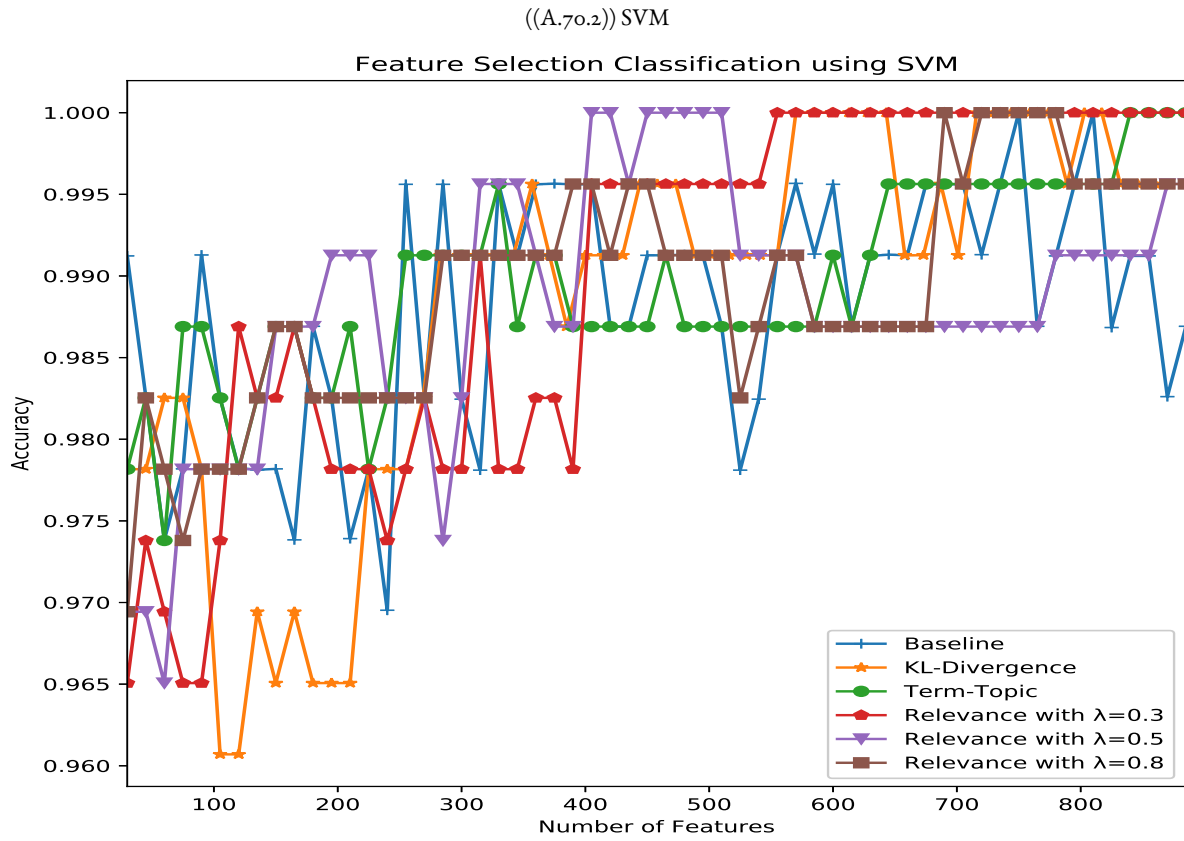
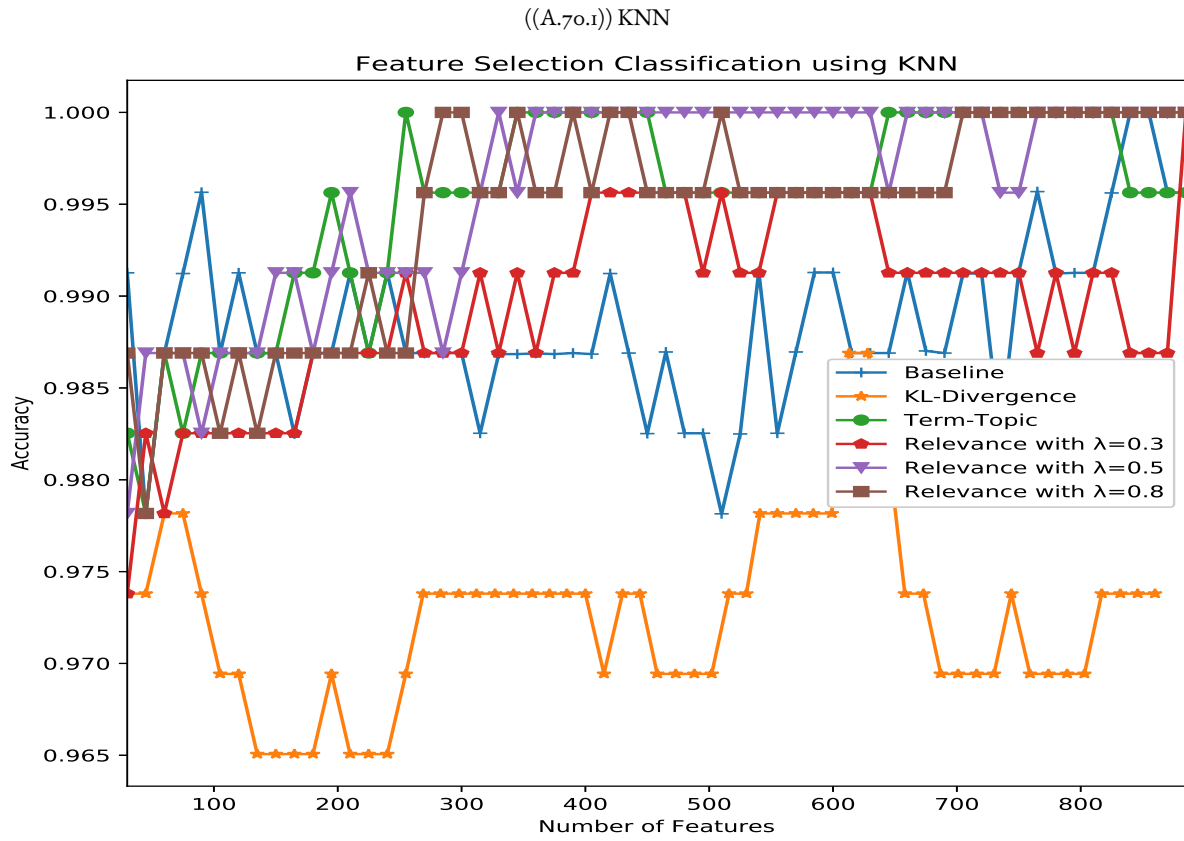


Figure A.70: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Median variant with $t_M = 0.2$ on the GEM.

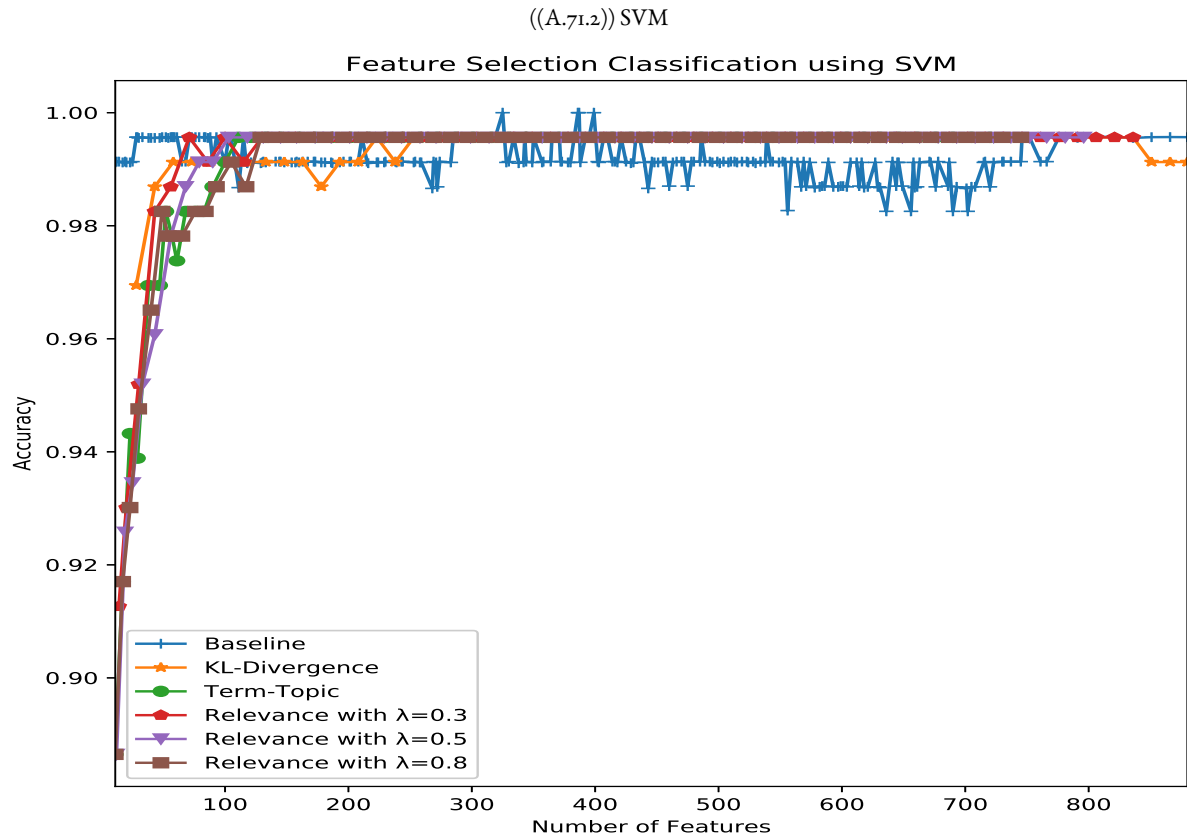
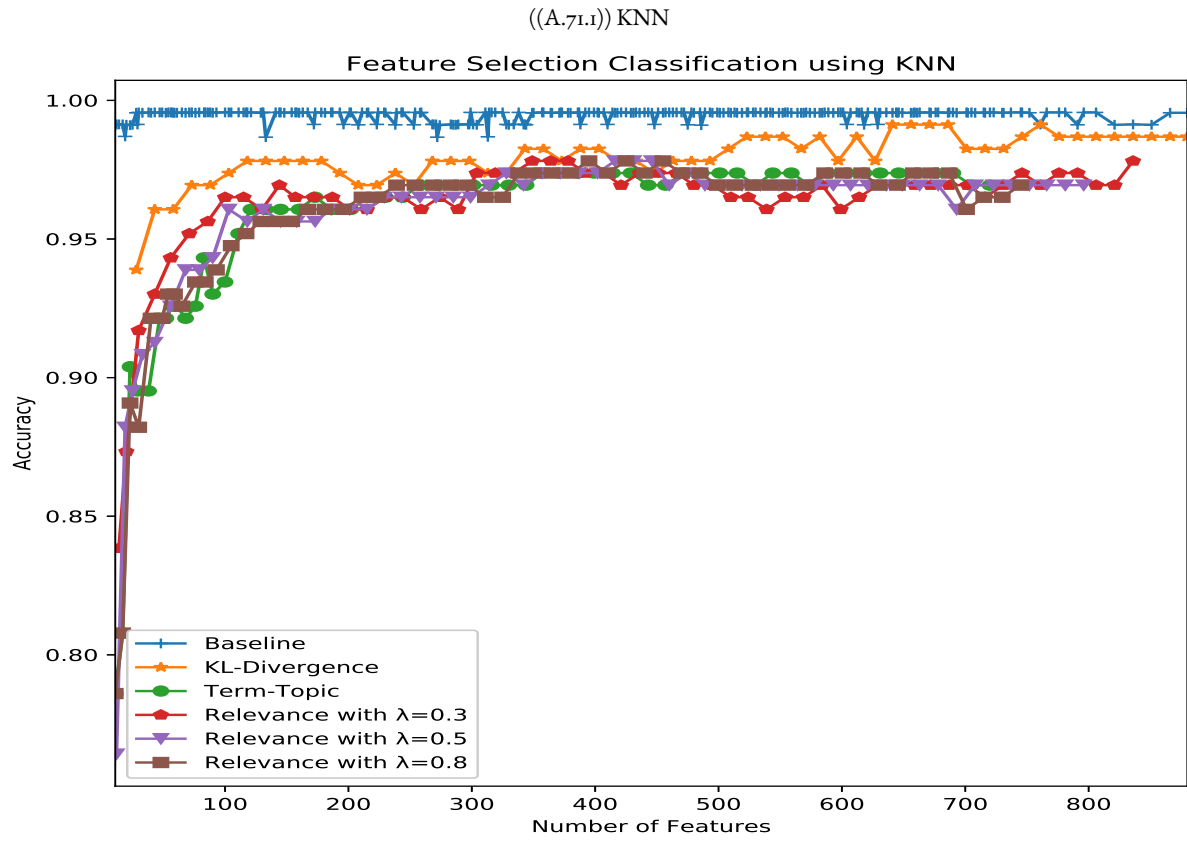


Figure A.71: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \mathbf{X} , $t_2 = \mathbf{X}$) on the GEM.

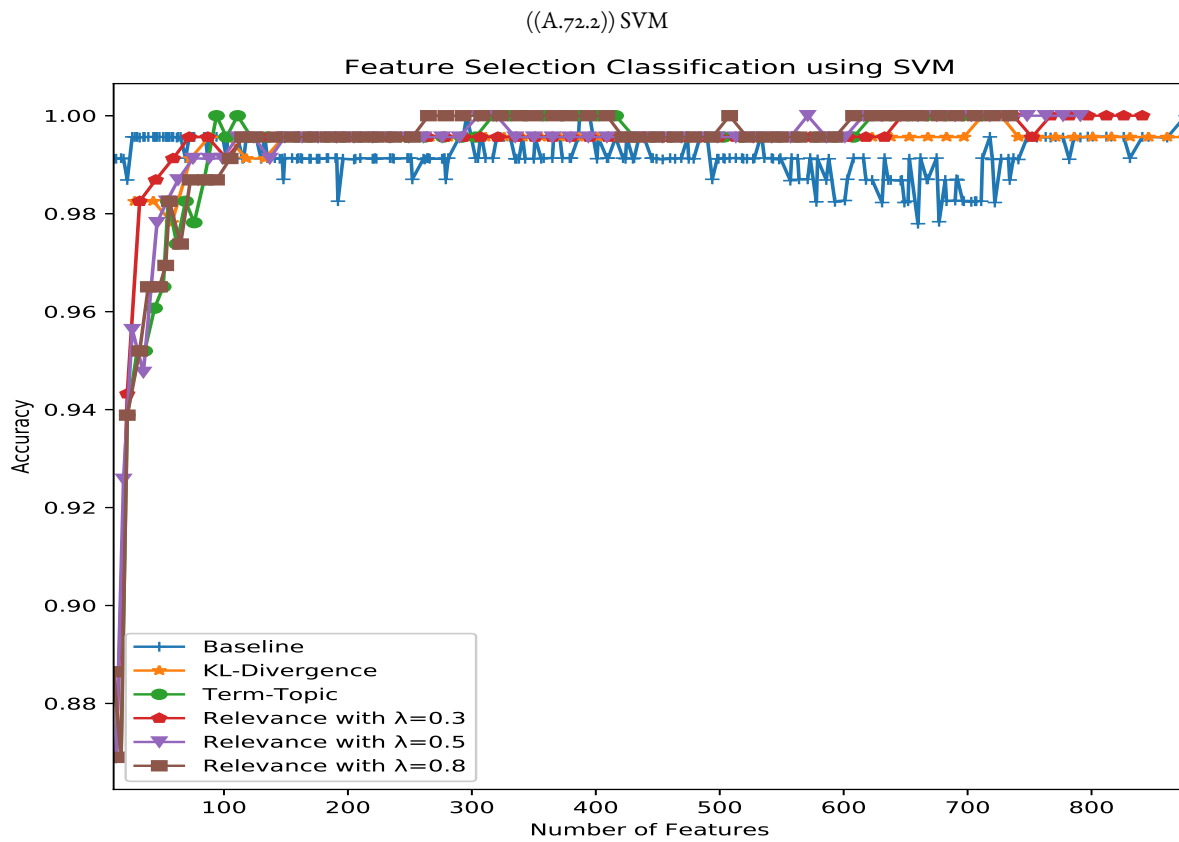
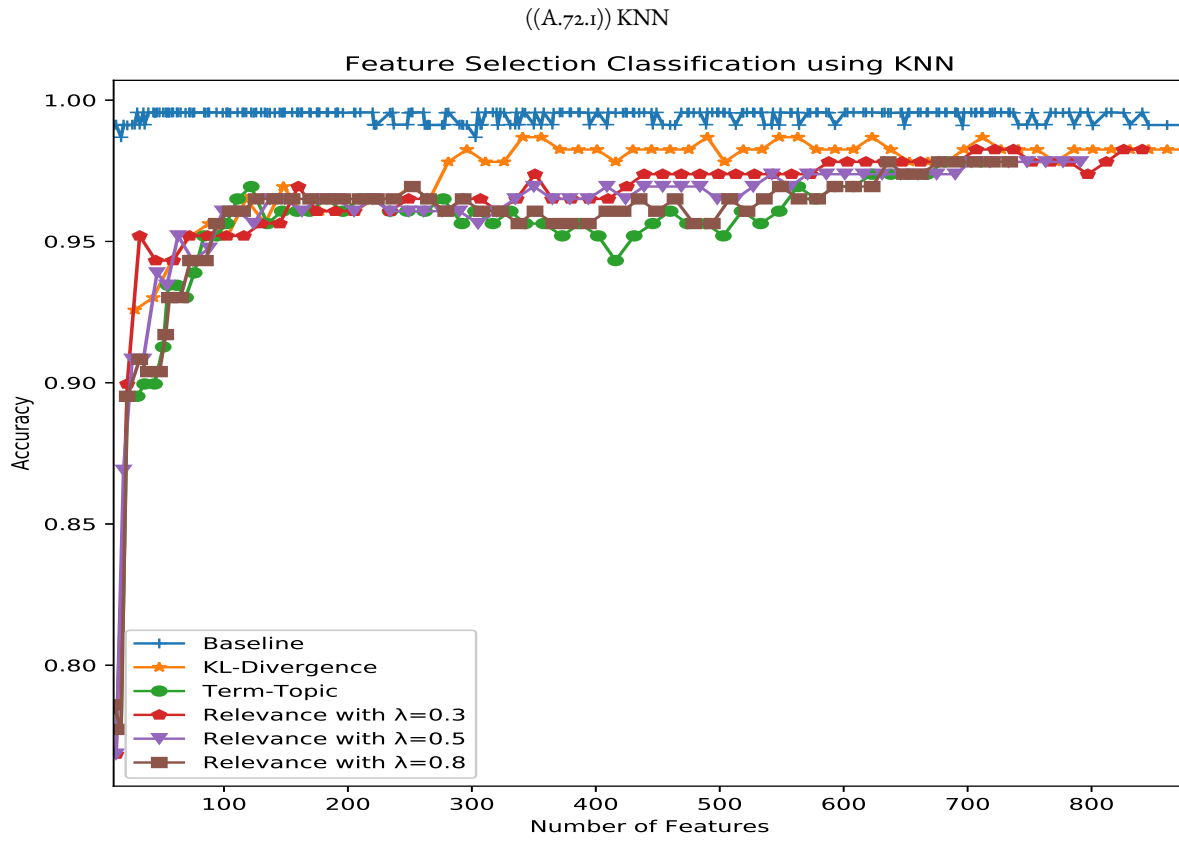


Figure A.72: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \mathbf{X} , $t_2 = 0.2$) on the GEM.

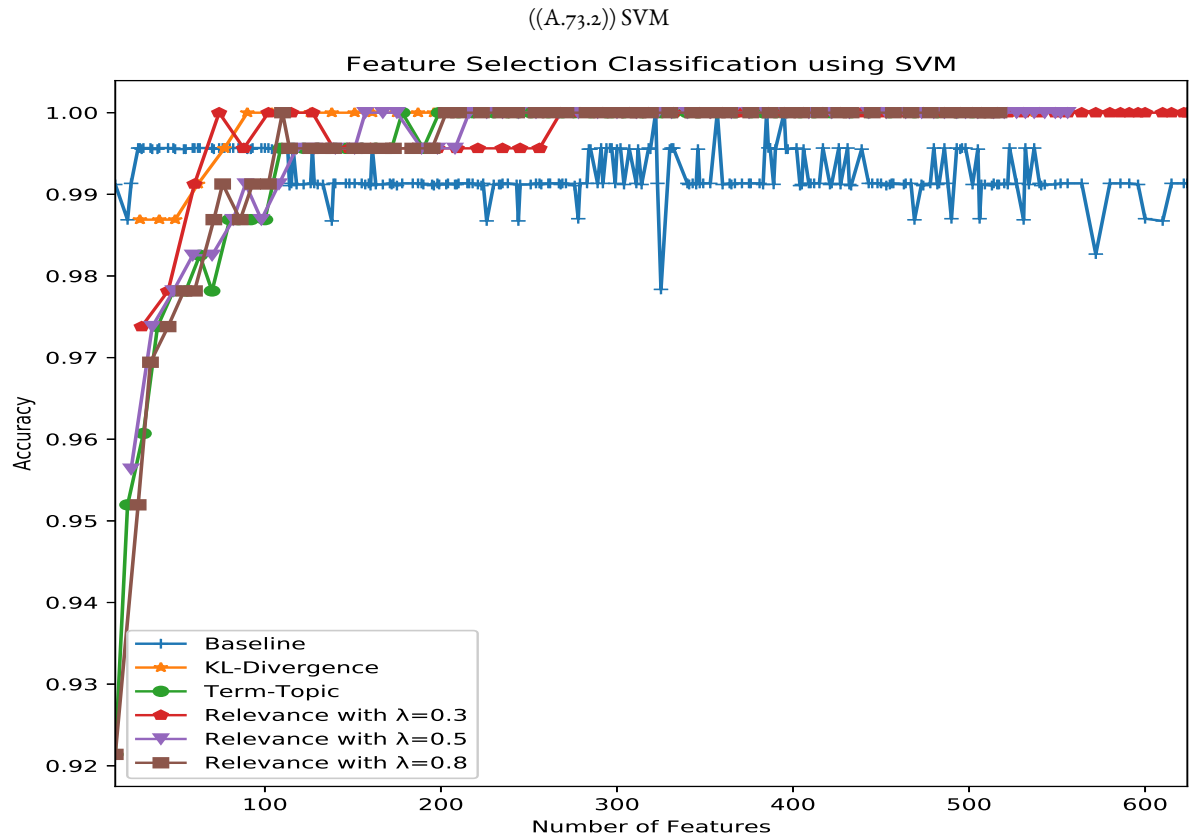
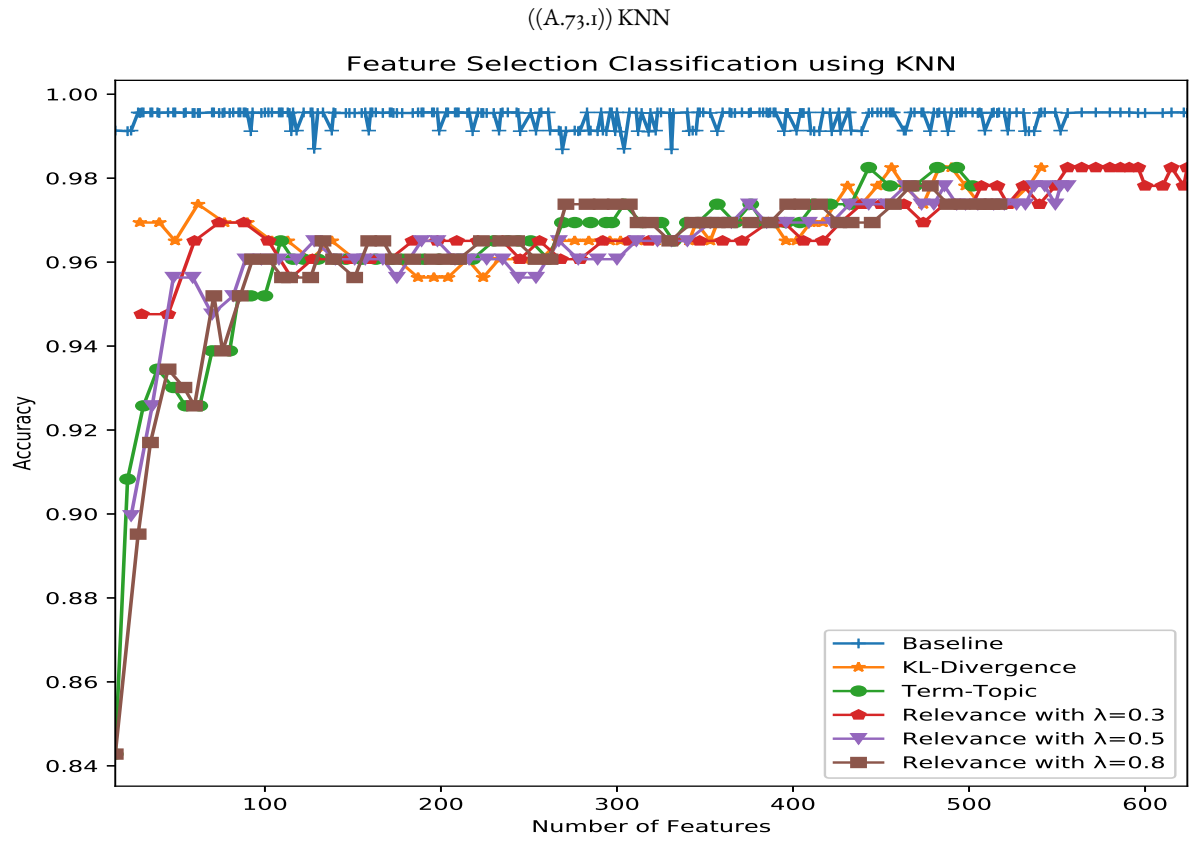


Figure A.73: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Repetition variant ($t_1 = \mathbf{X}$, RemoveBin= \checkmark , $t_2 = \mathbf{X}$) on the GEM.

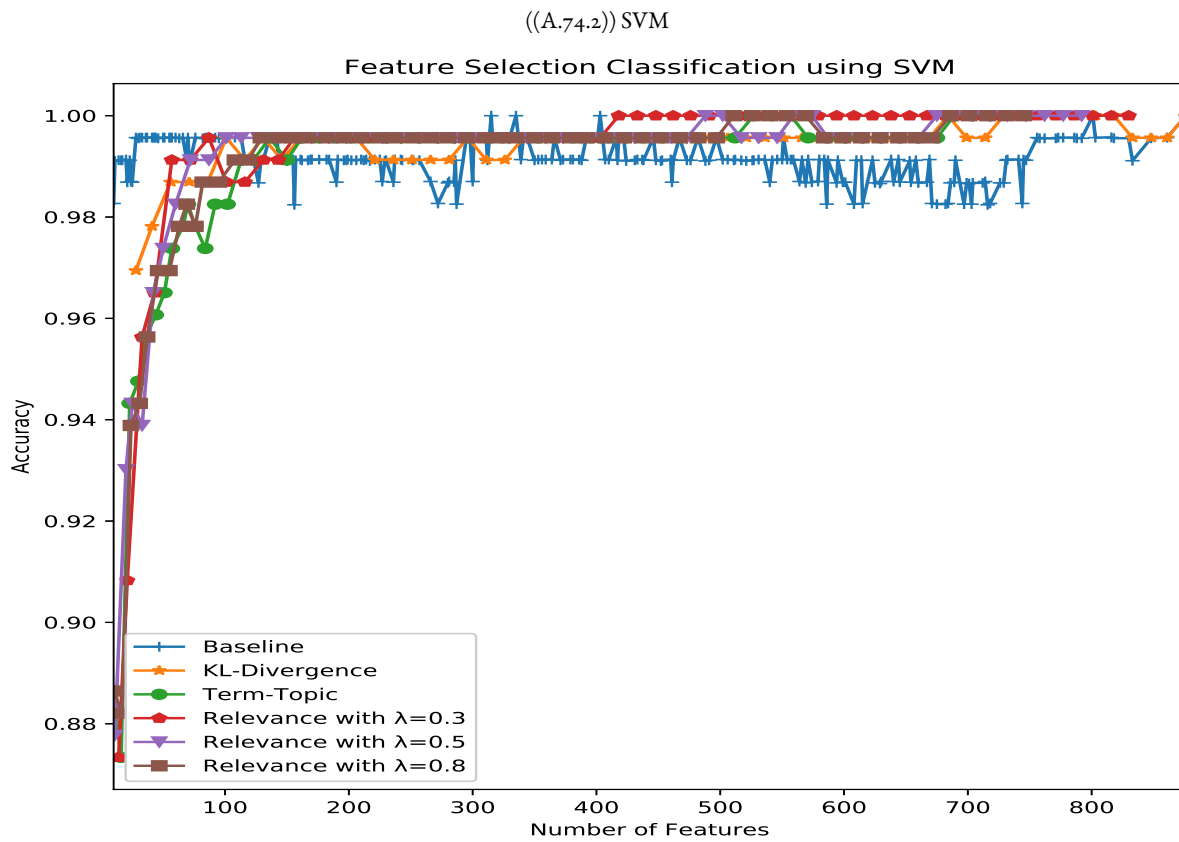
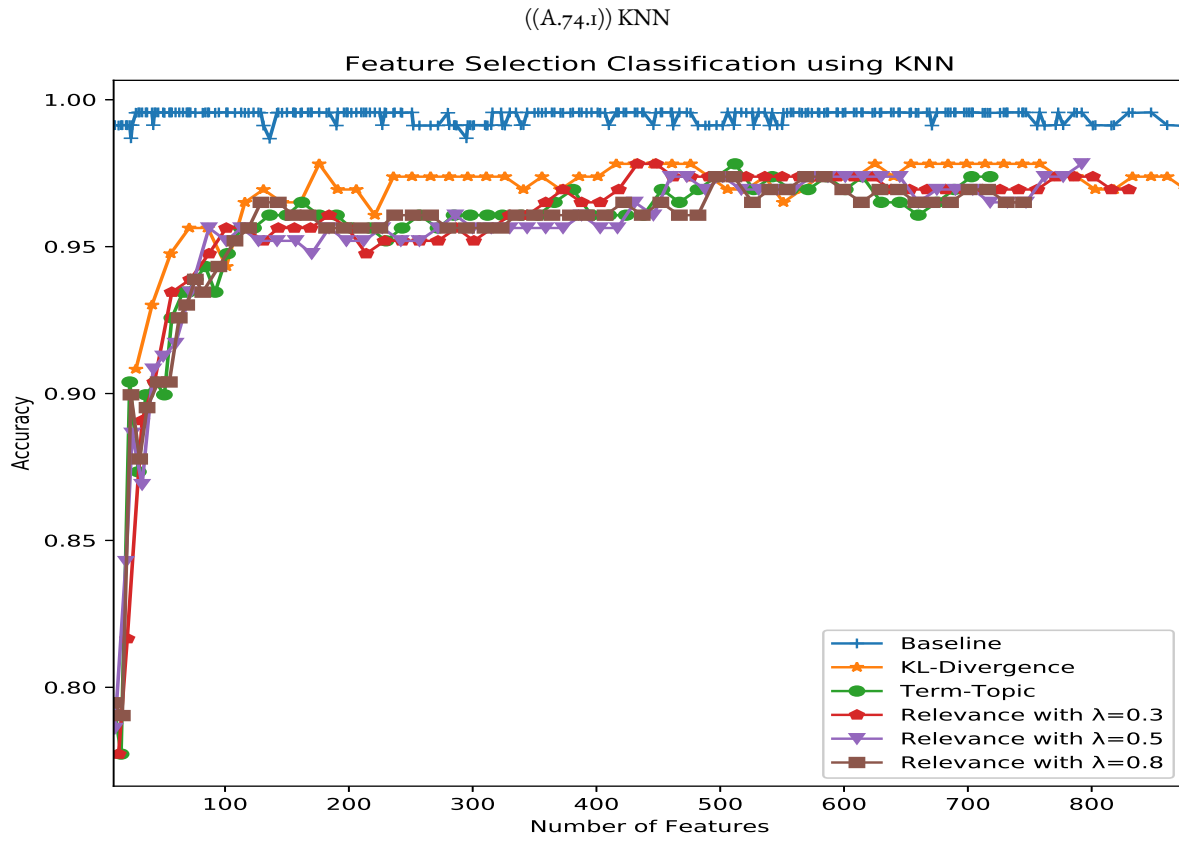


Figure A.74: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Repetition variant ($t_1 = 0.2, \text{RemoveBin}=\mathbf{X}, t_2 = \mathbf{X}$) on the GEM.

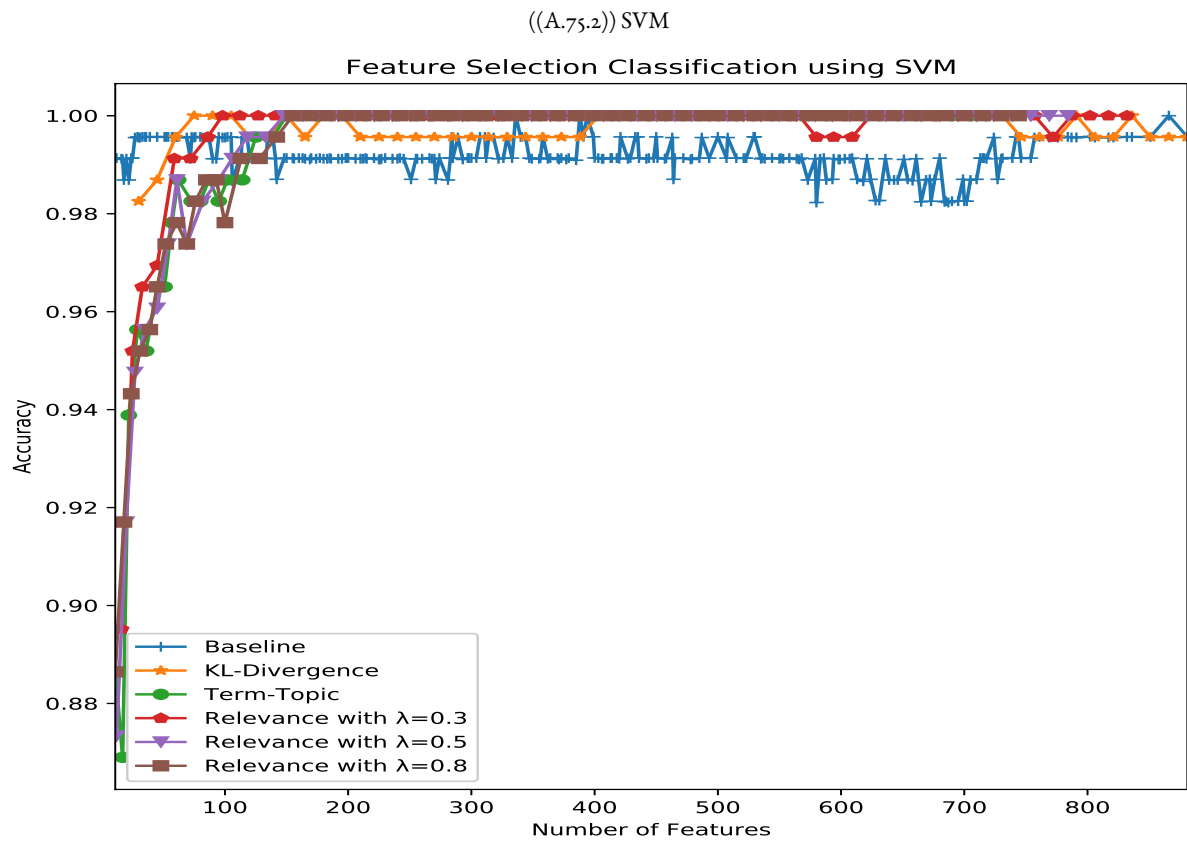
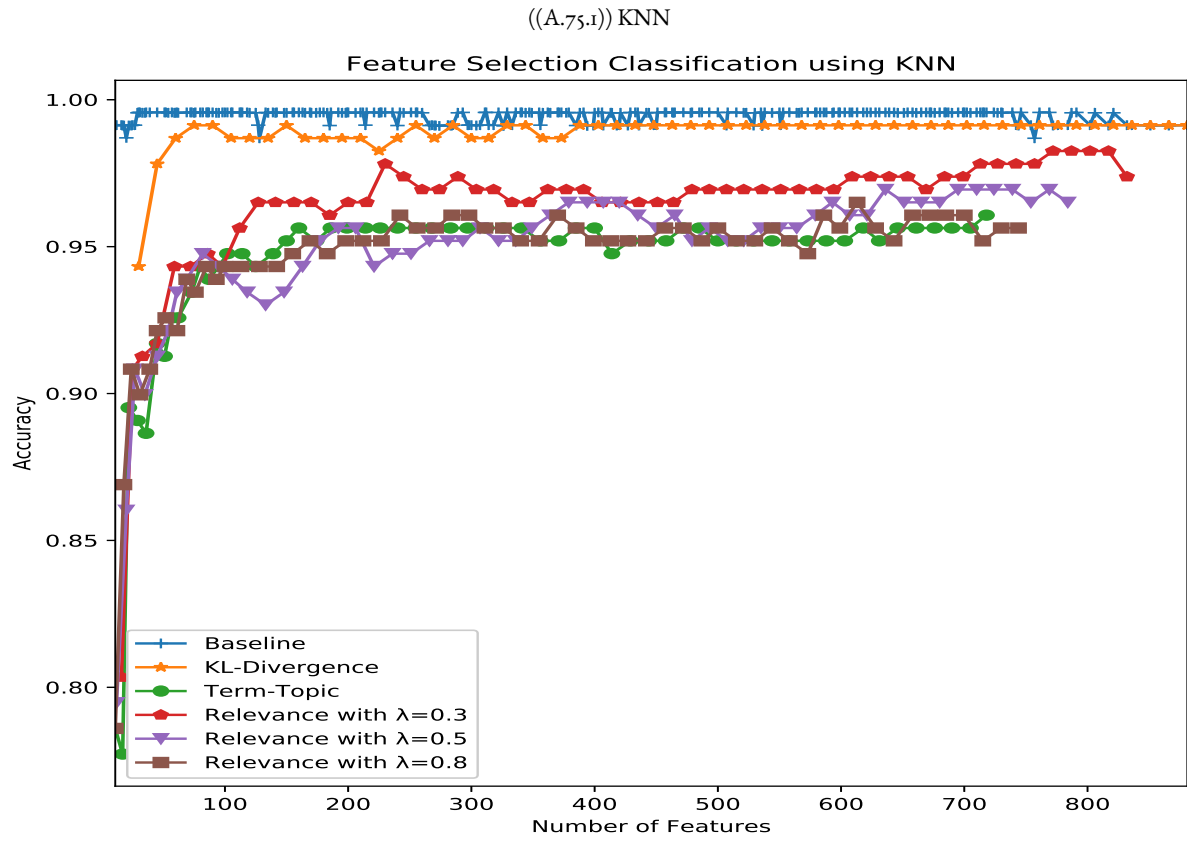


Figure A.75: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Repetition variant ($t_1 = 0.2$, RemoveBin= \mathbf{X} , $t_2 = 0.3$) on the GEM.

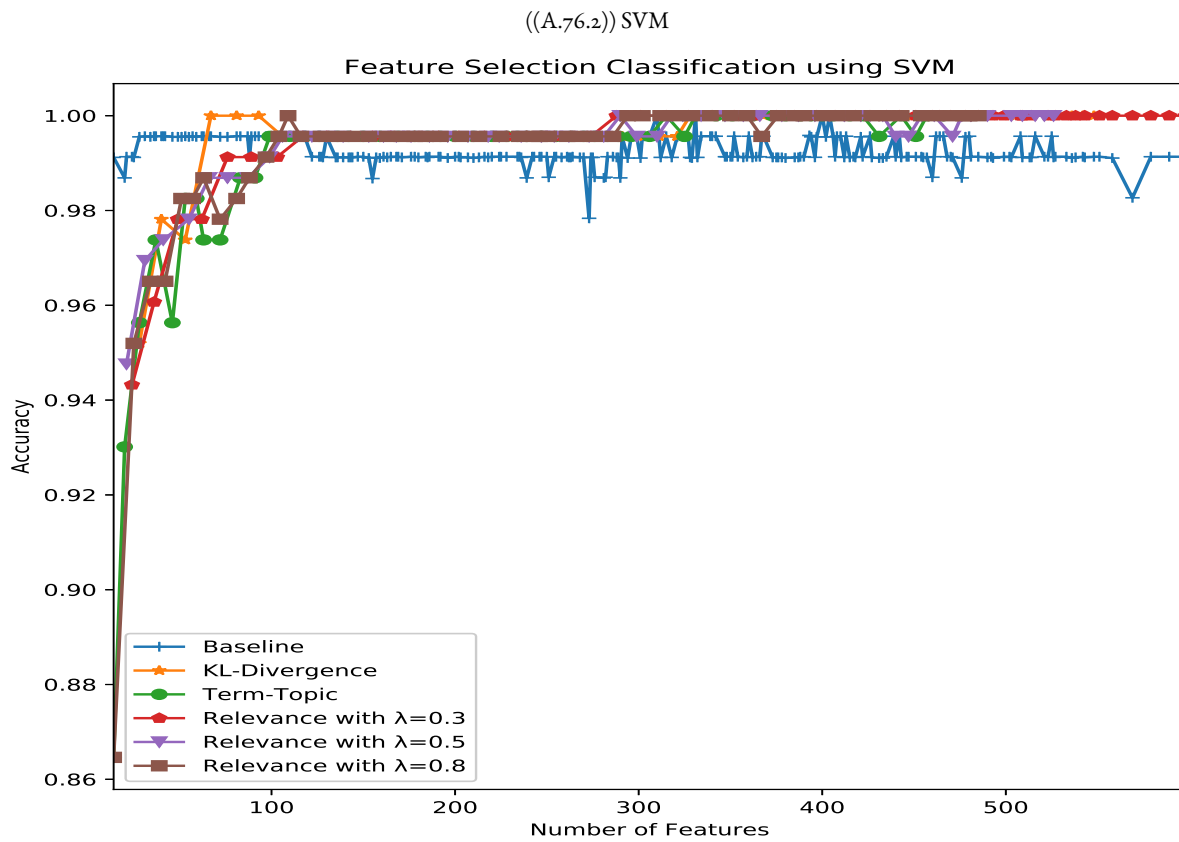
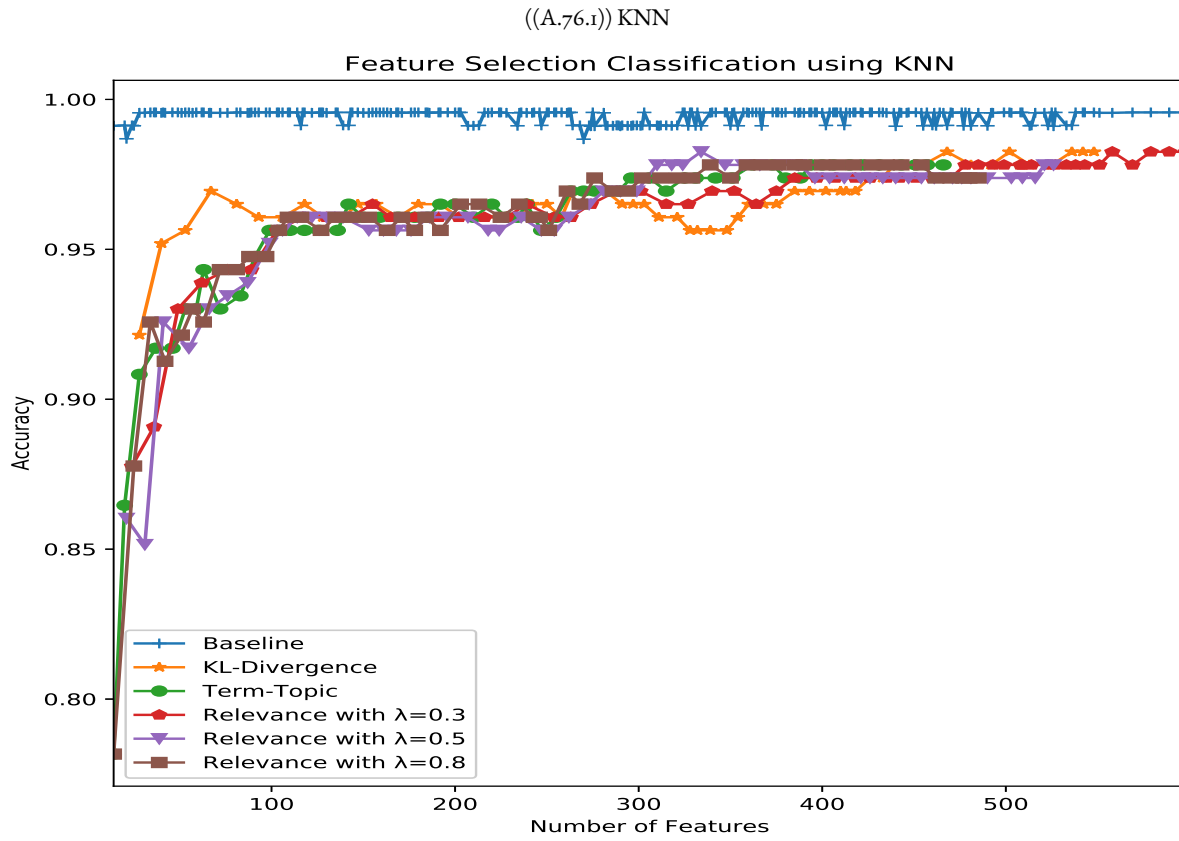


Figure A.76: Unsupervised Feature Selection using LDA and Univariate Feature Selection on TCGA Dataset Repetition variant ($t_1 = 0.2, \text{RemoveBin}=\checkmark, t_2 = \times$) on the GEM.

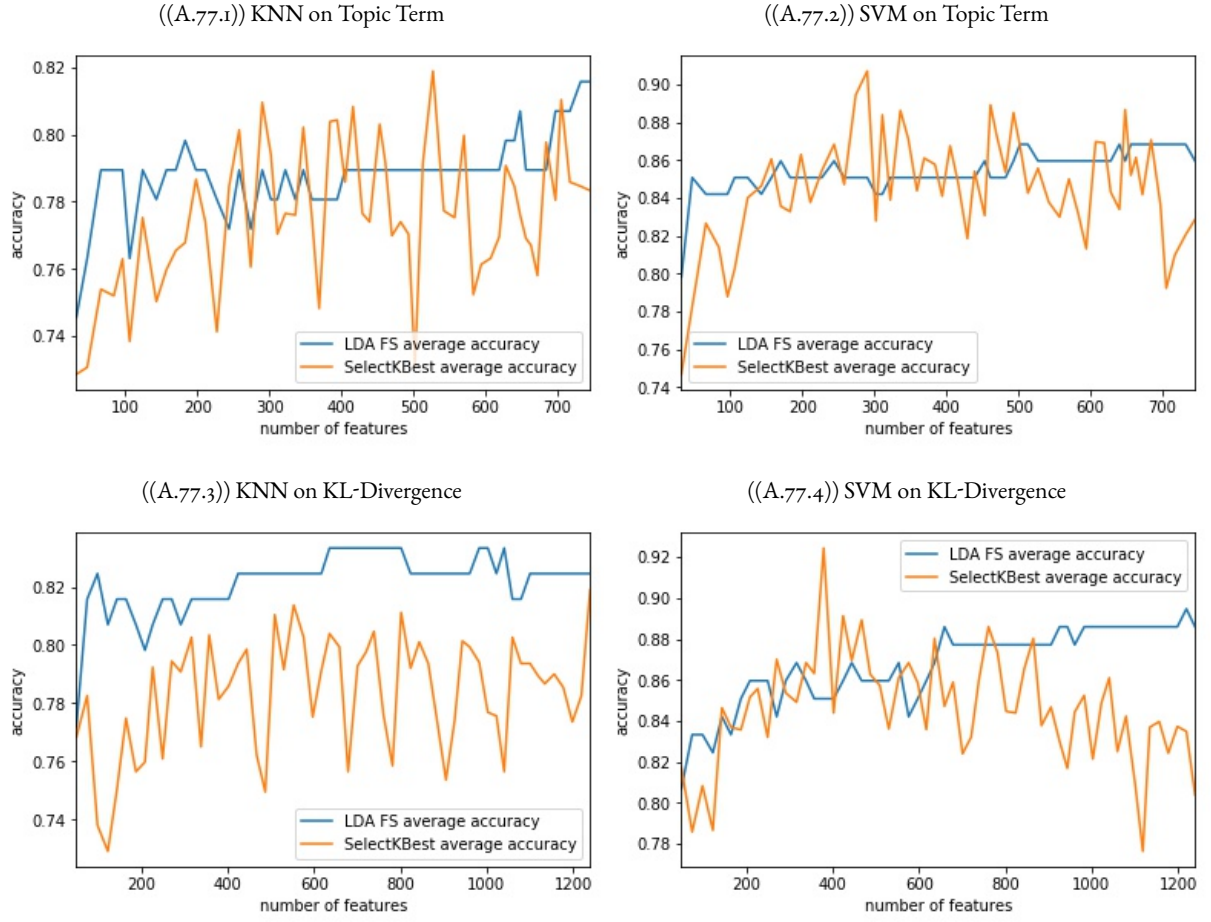


Figure A.77: Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset where $t_{LPD} = 0$ and the number of topics is $K = 5$ on the GEM.

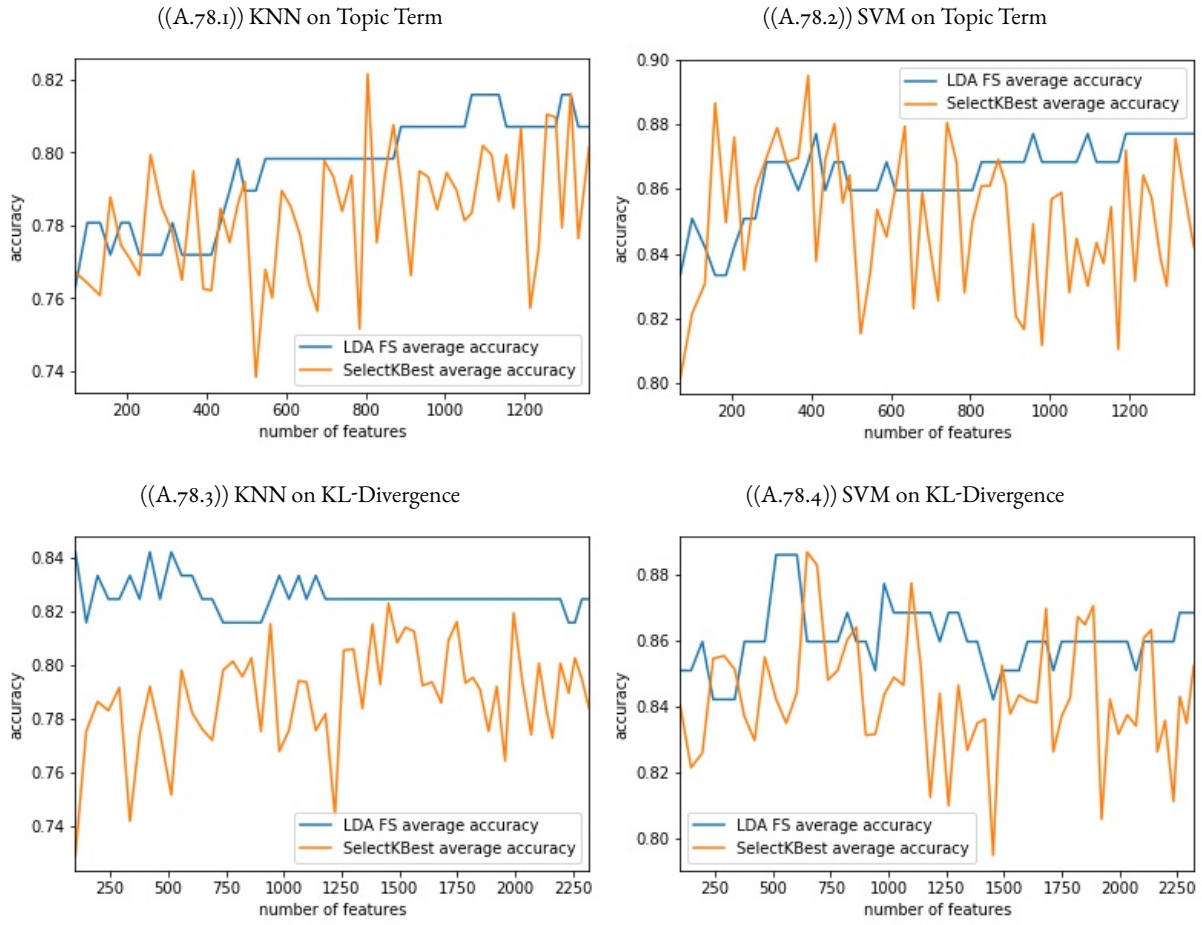


Figure A.78: Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset where $t_{LPD} = 0$ and the number of topics is $K = 10$ on the GEM.

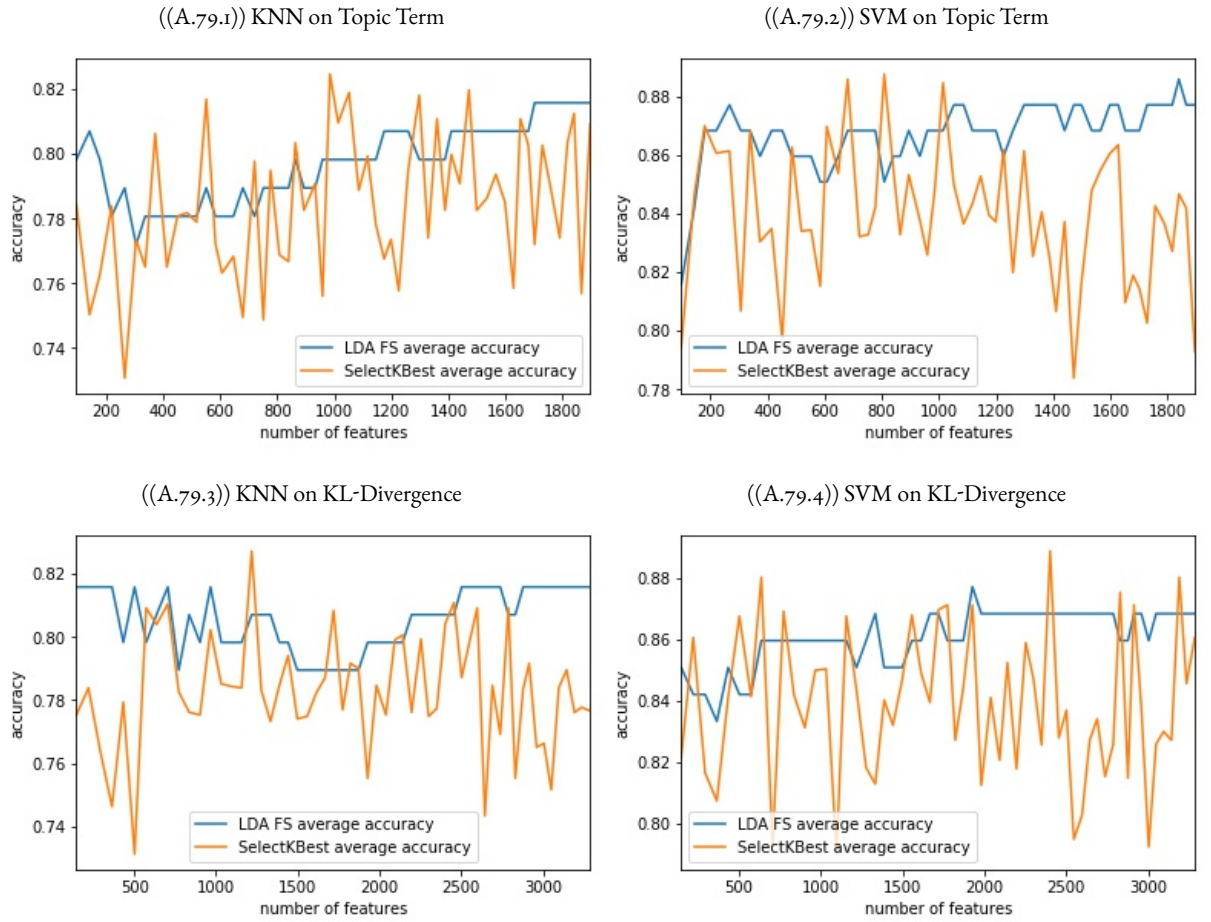


Figure A.79: Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset where $t_{LPD} = 0$ and the number of topics is $K = 15$ on the GEM.

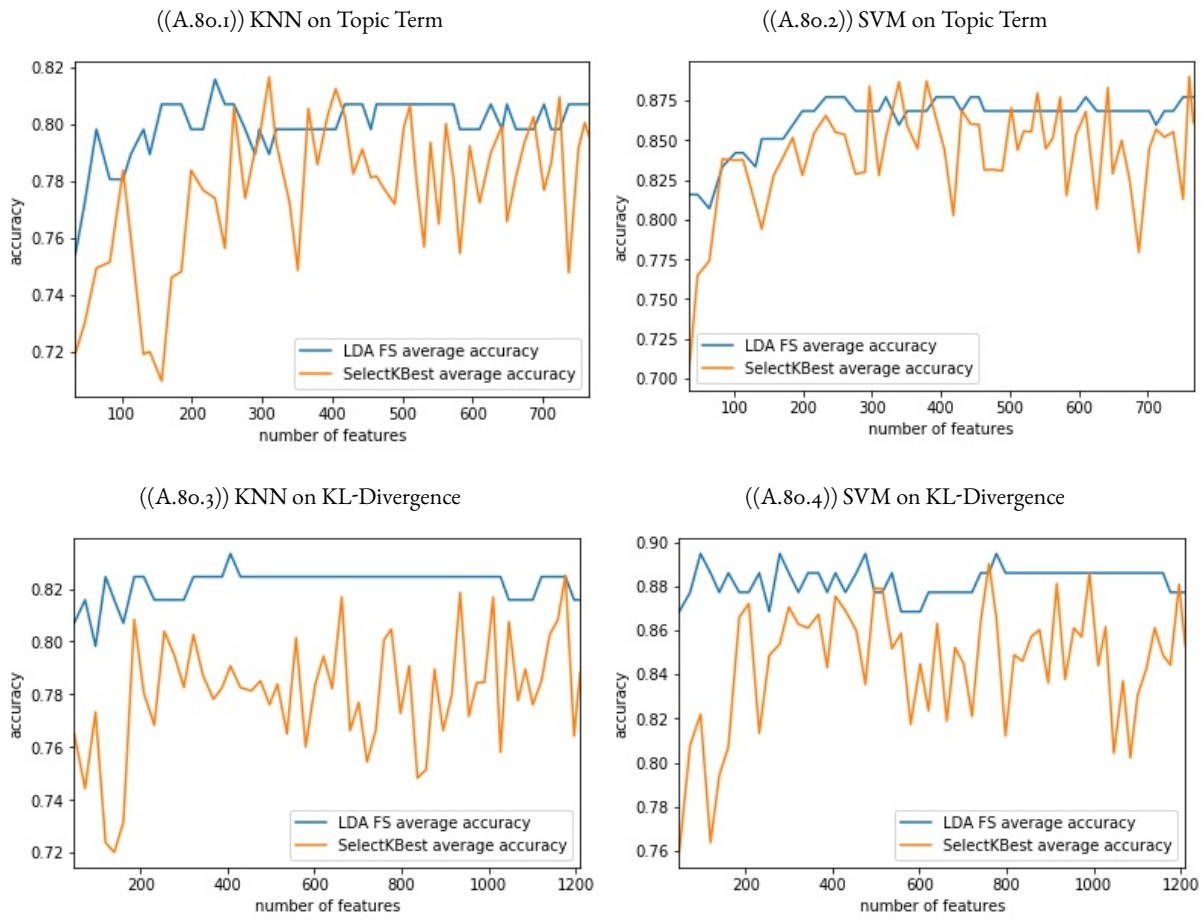


Figure A.8o: Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset where $t_{LPD} = 0.2$ and the number of topics is $K = 5$ on the GEM.

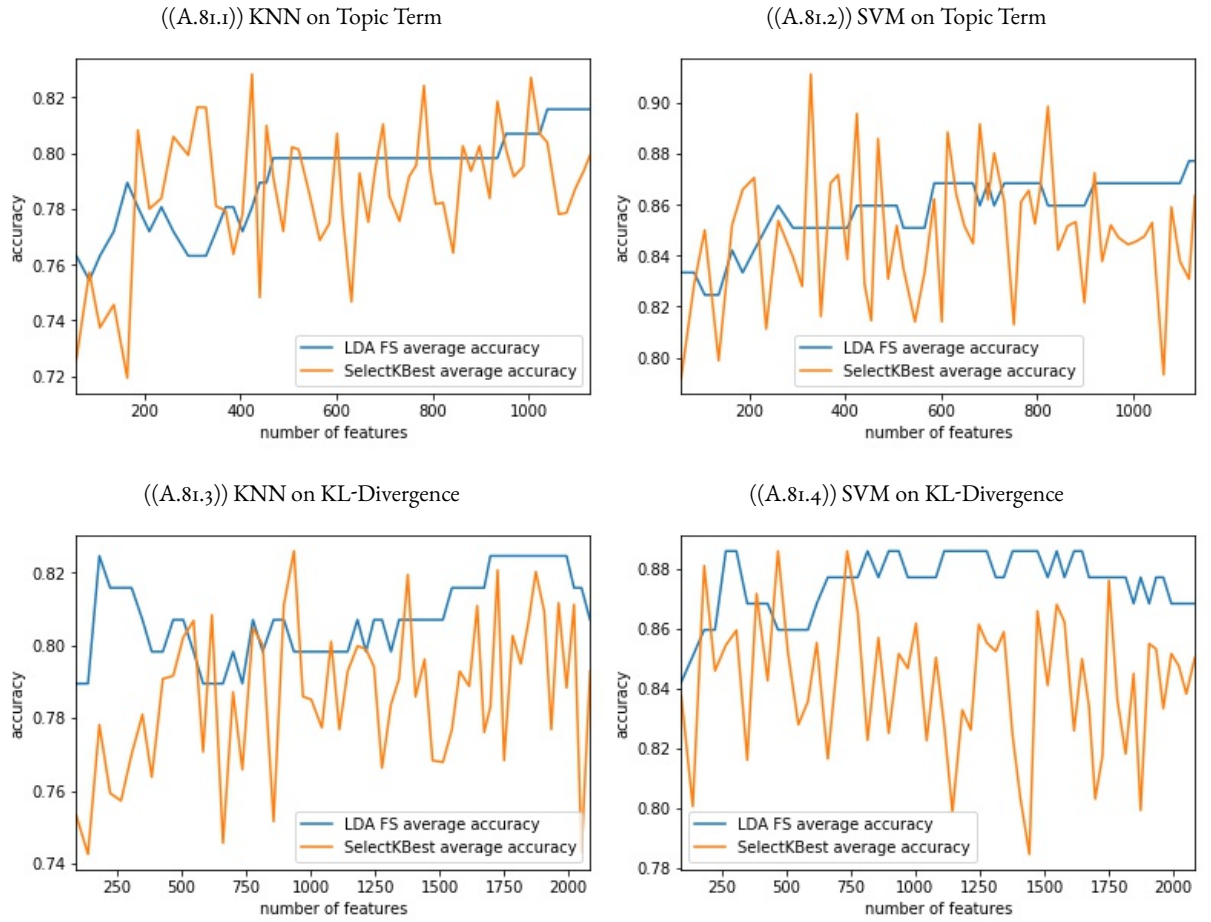


Figure A.81: Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset where $t_{LPD} = 0.2$ and the number of topics is $K = 10$ on the GEM.

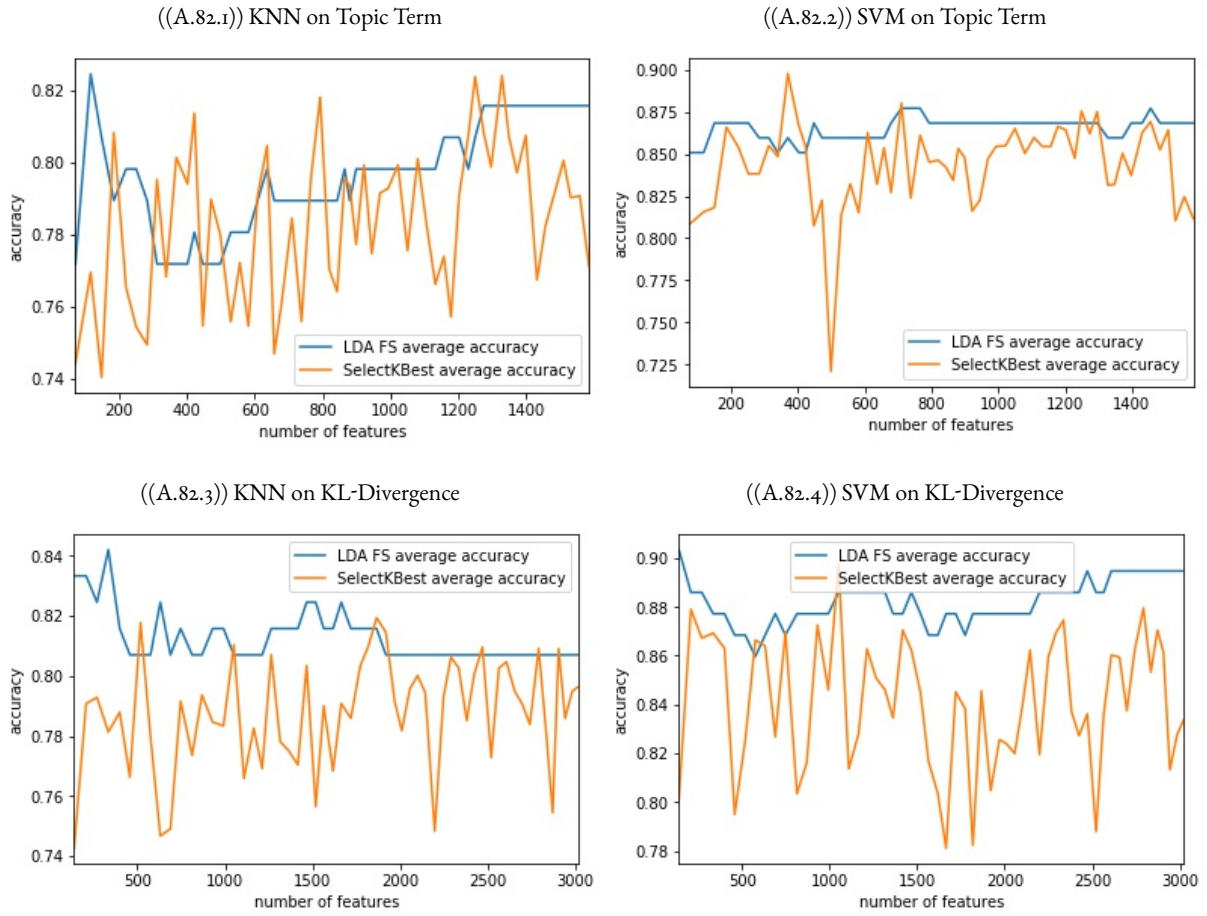


Figure A.82: Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset where $t_{LPD} = 0.2$ and the number of topics is $K = 15$ on the GEM.

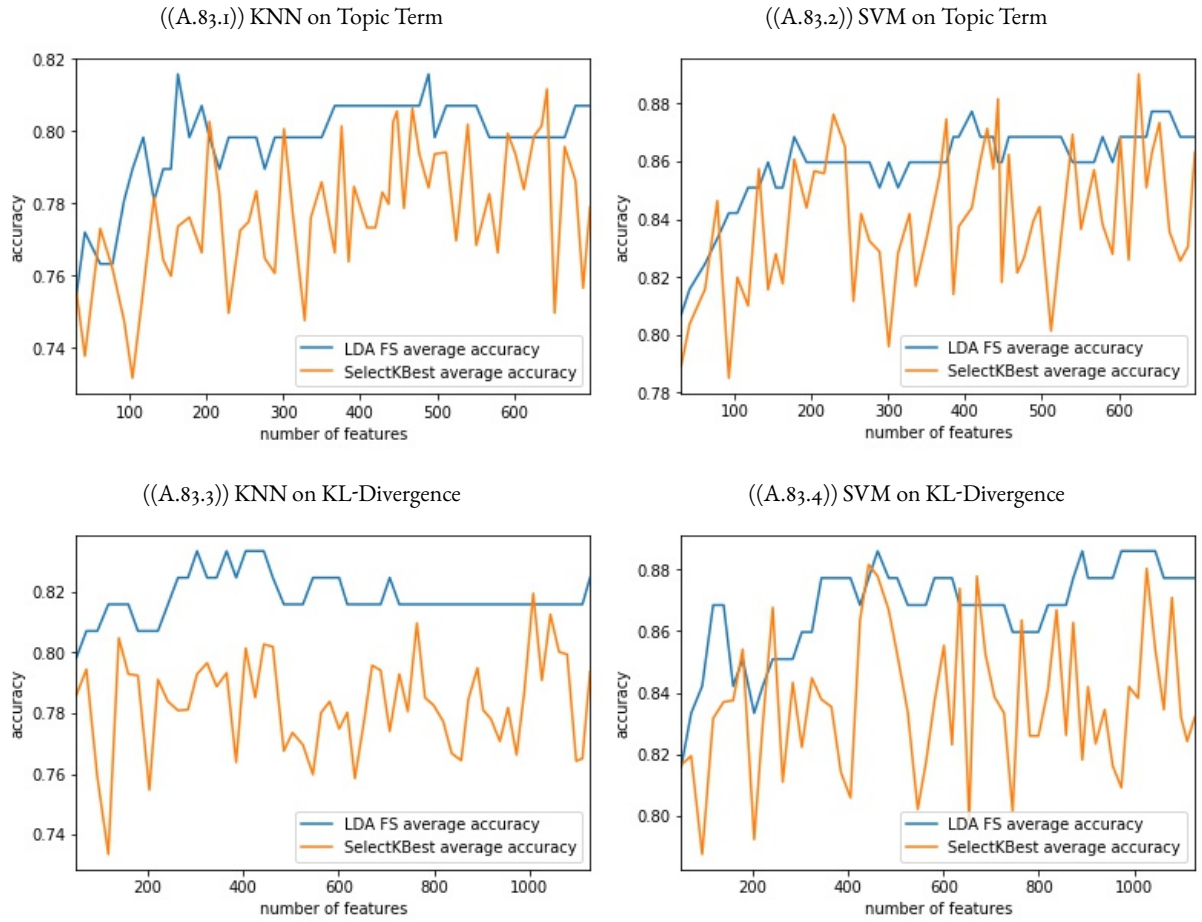


Figure A.83: Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset where $t_{LPD} = \chi$ and the number of topics is $K = 5$ on the GEM.

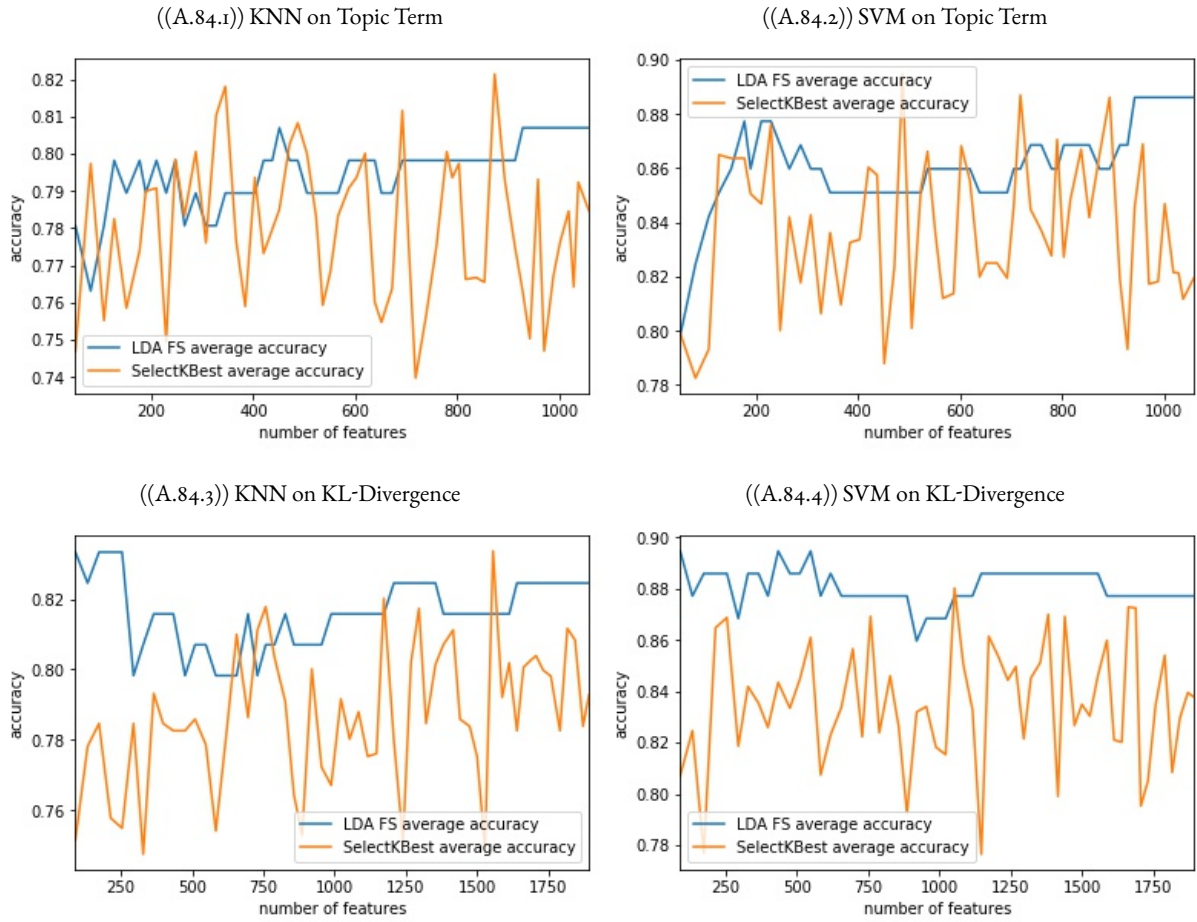


Figure A.84: Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset where $t_{LPD} = \mathbf{X}$ and the number of topics is $K = 10$ on the GEM.

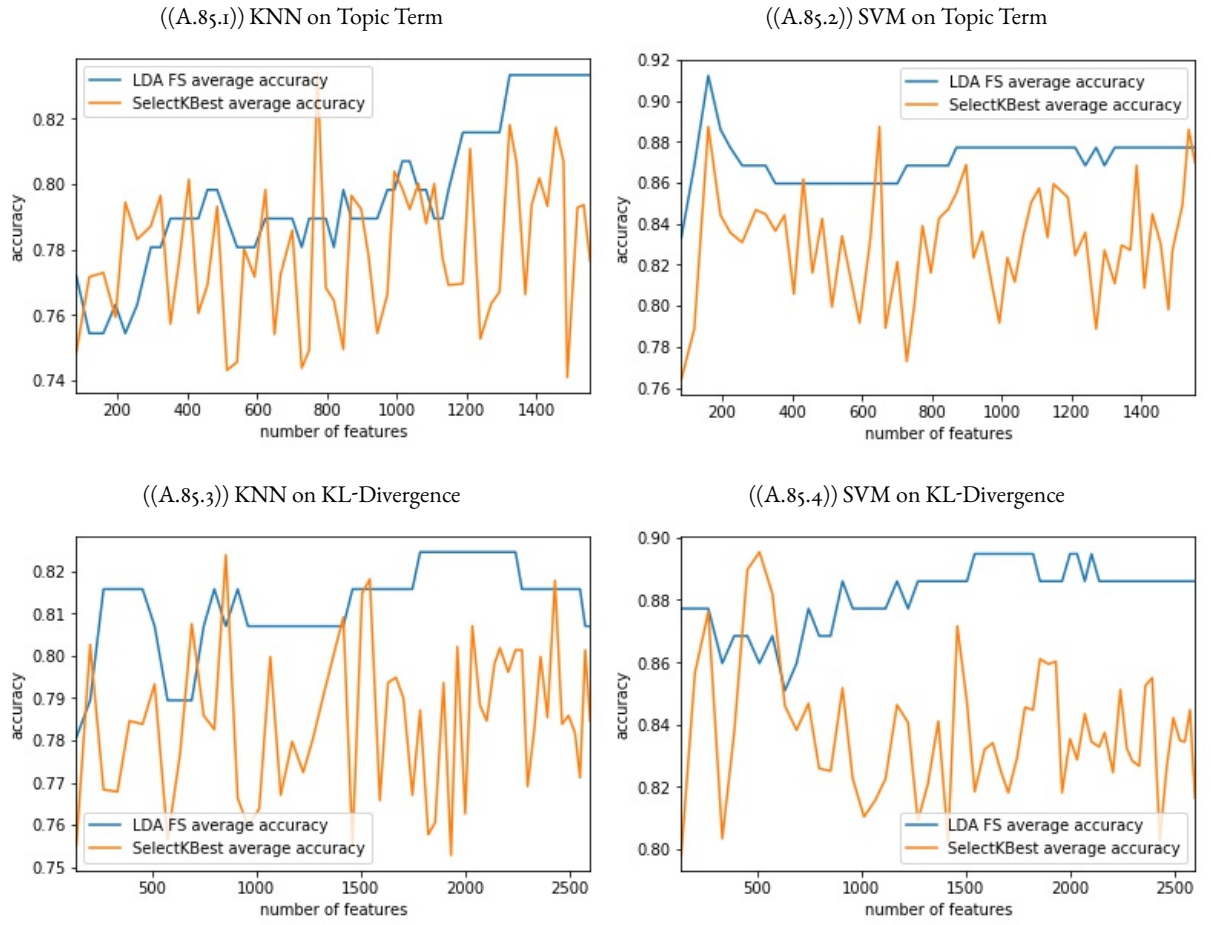


Figure A.85: Unsupervised feature Selection using LPD and Univariate Feature Selection on MD Dataset where $t_{LPD} = \mathbf{X}$ and the number of topics is $K = 15$ on the GEM.

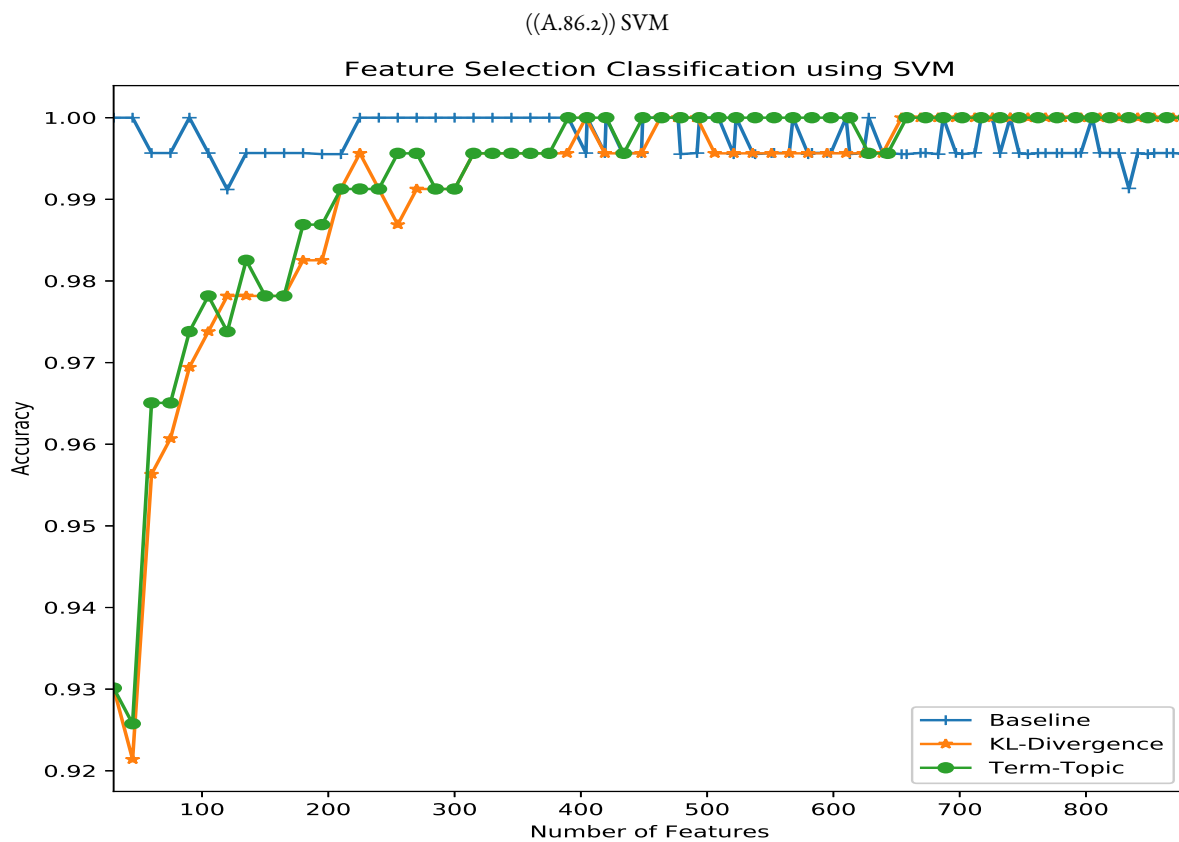
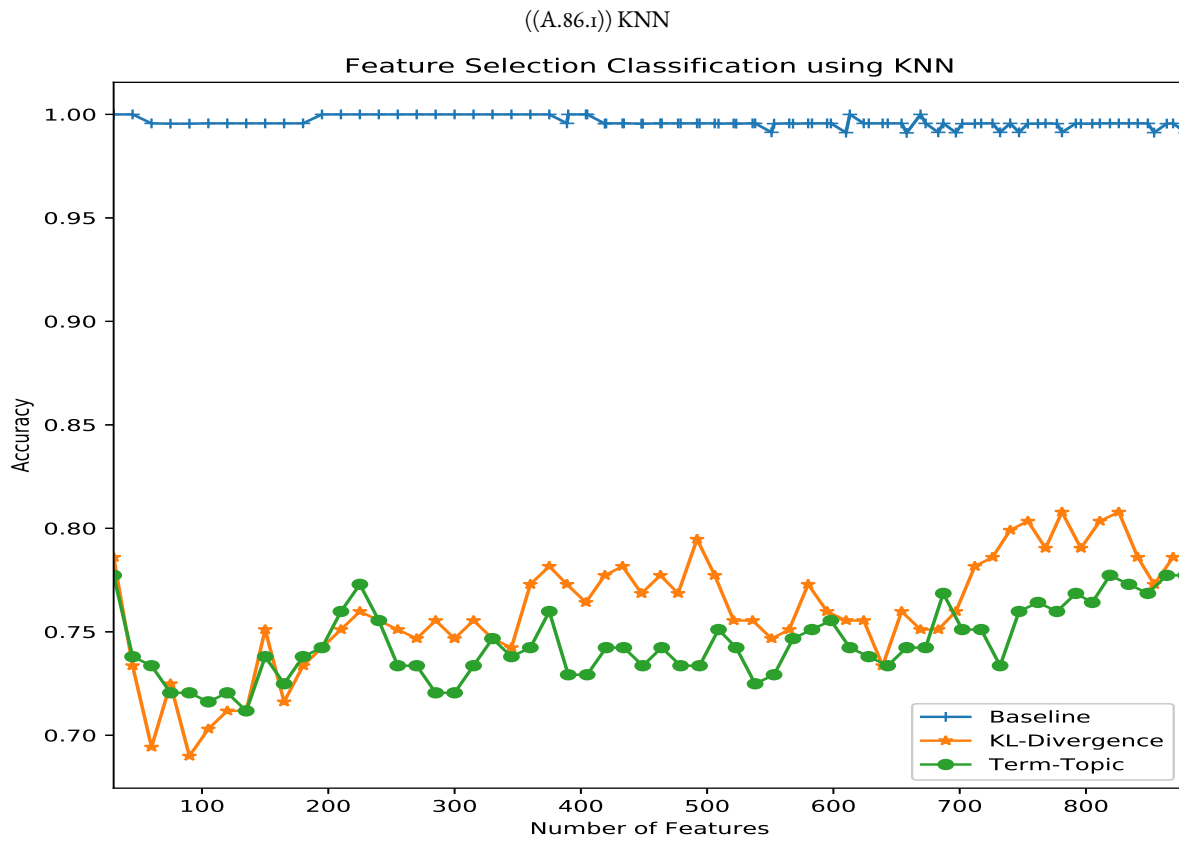


Figure A.86: Unsupervised Feature Selection using LPD and Univariate Feature Selection on TCGA Dataset where $K = 3$ on the GEM.

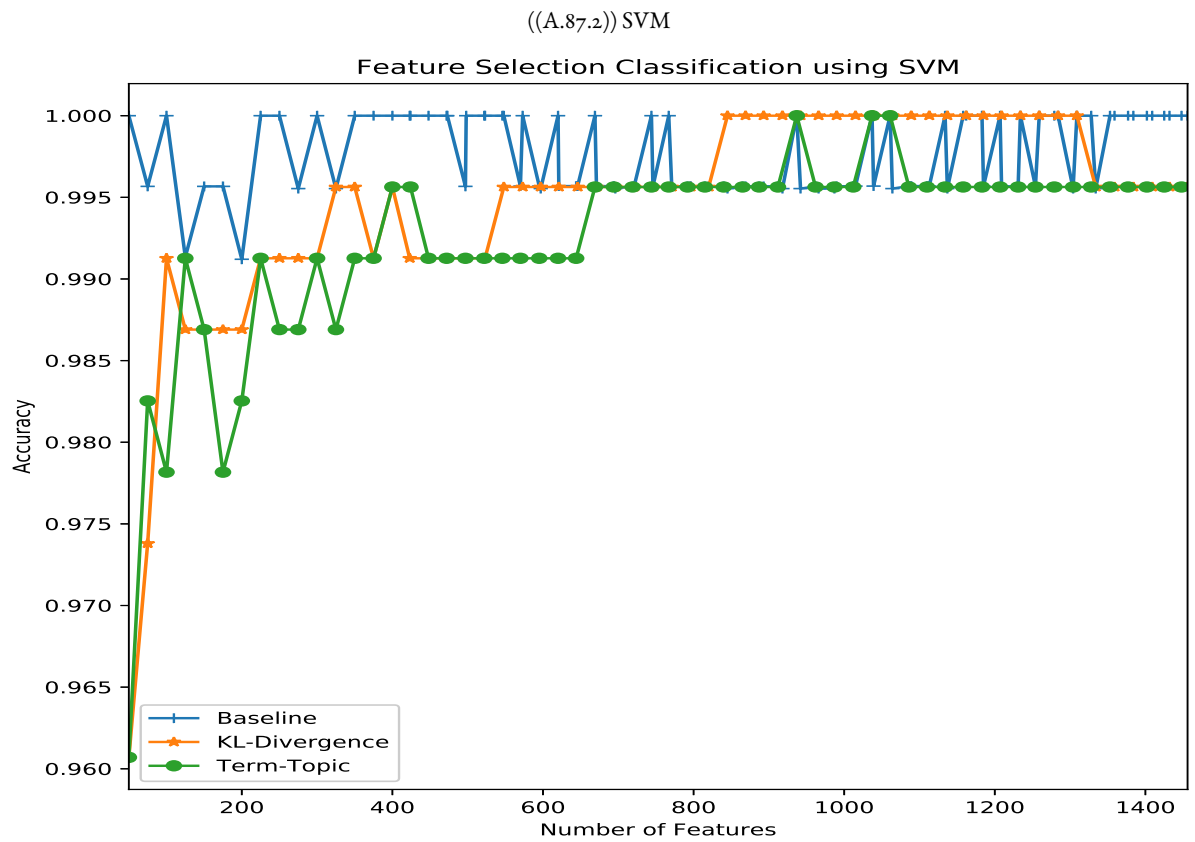
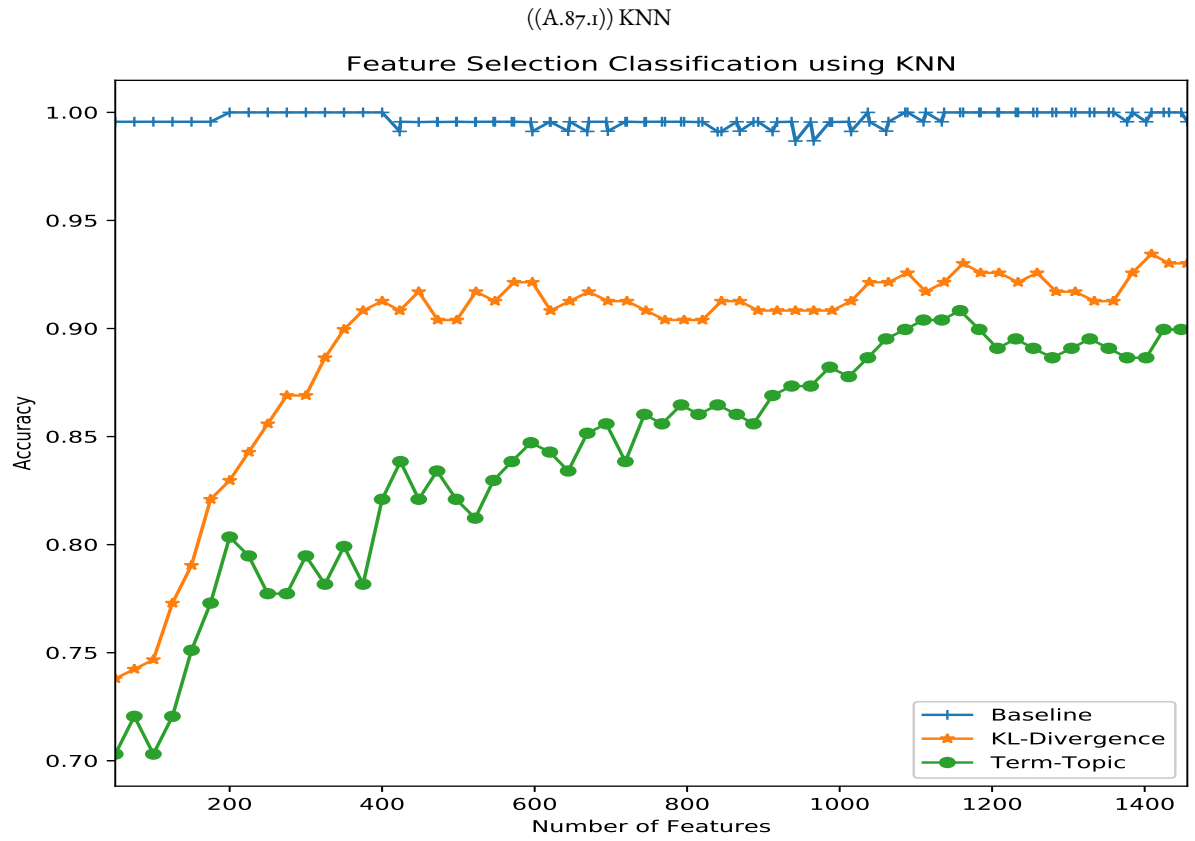


Figure A.87: Unsupervised Feature Selection using LPD and Univariate Feature Selection on TCGA Dataset where $K = 5$ on the GEM.

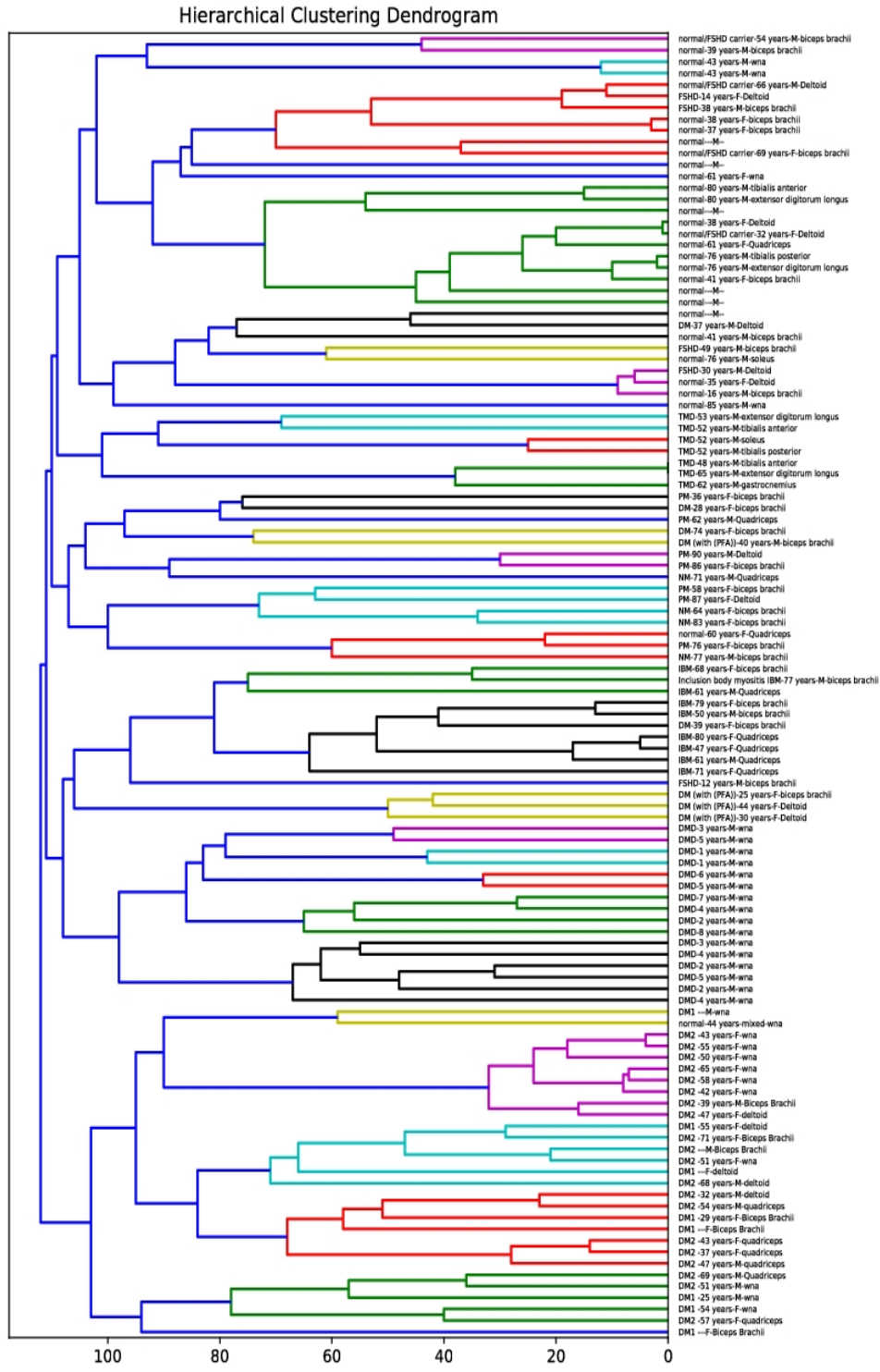


Figure A.88: Dendrogram of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Repetition Variant with ($t_1=\mathbf{X}$, Remove Bin 1 = $\mathbf{X}t_2=\mathbf{X}$, $h = 200$, $n_c = 8$) on the Relevance Matrix with $\lambda = 0.8$.

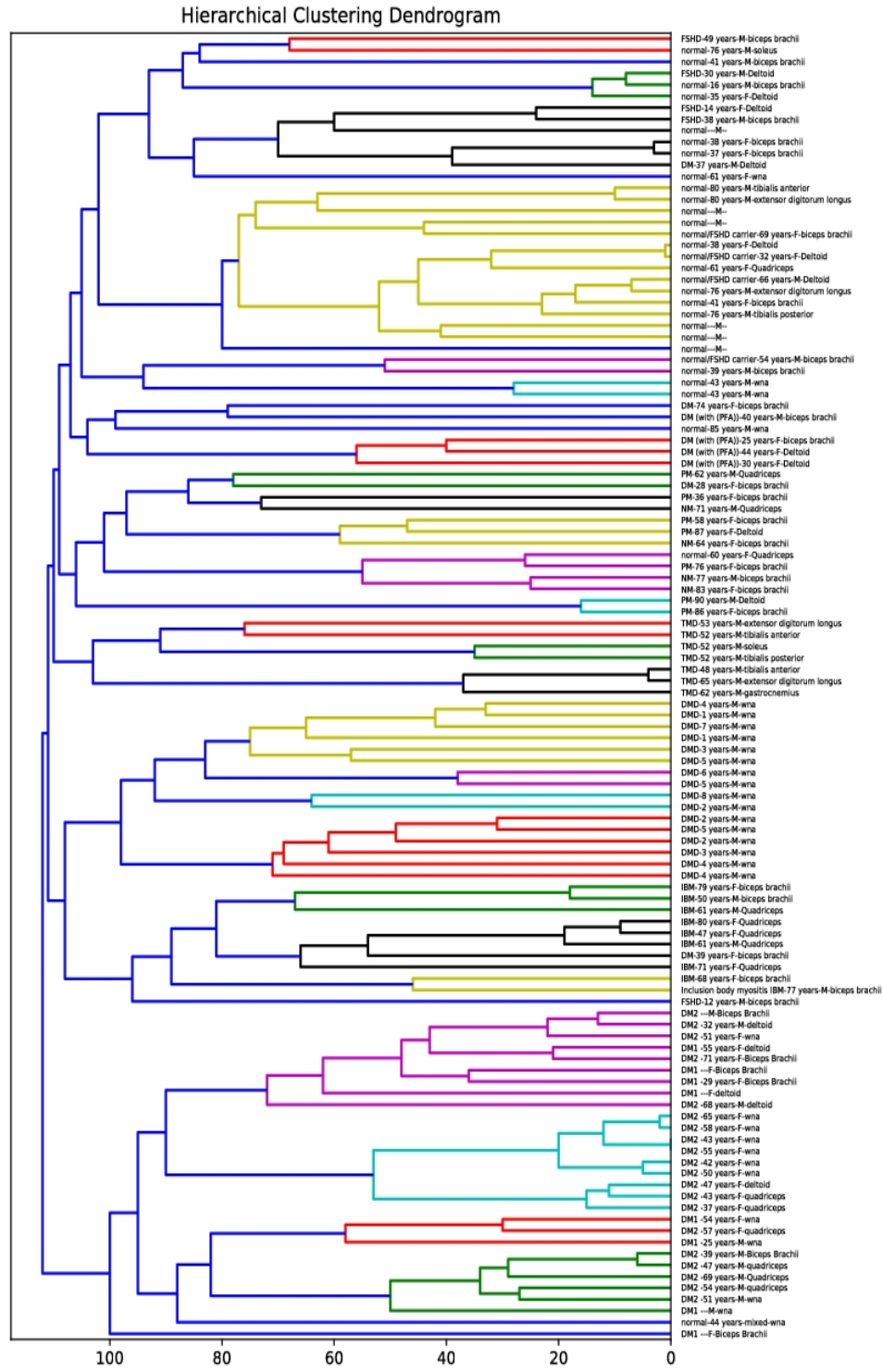


Figure A.89: Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Repetition Variant with ($t_1=\mathbf{X}$, Remove Bin 1 = $\mathbf{X}t_2 = 0.2$, $b = 100$, $n_c = 7$) on the Relevance Matrix with $\lambda = 0.3$.

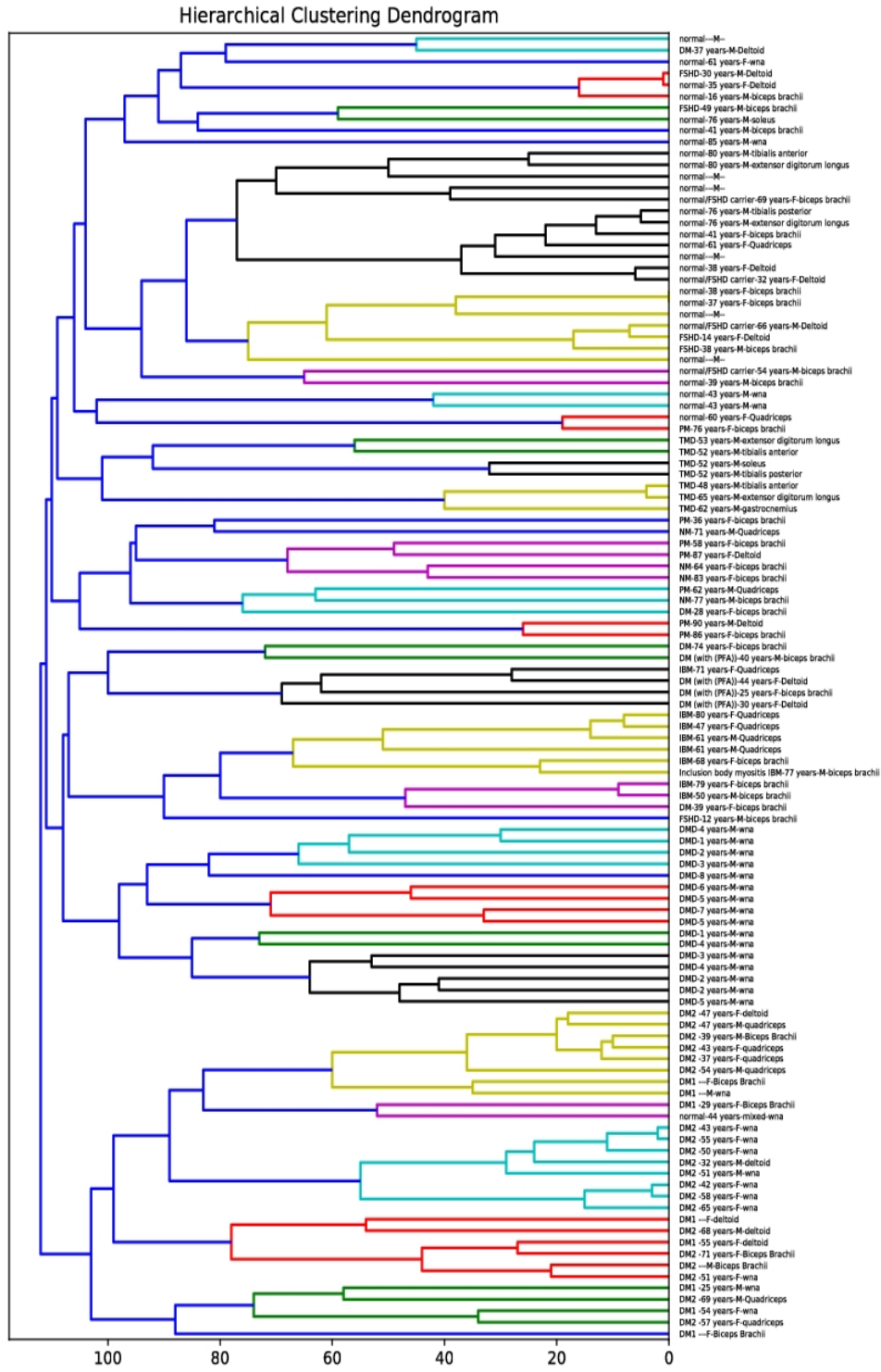


Figure A.90: Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Repetition Variant with ($t_1=\mathbf{X}$, Remove Bin 1 = $\sqrt{t_2} = 0.2$, $h = 50$, $n_c = 7$) on the Relevance Matrix with $\lambda = 0.5$.

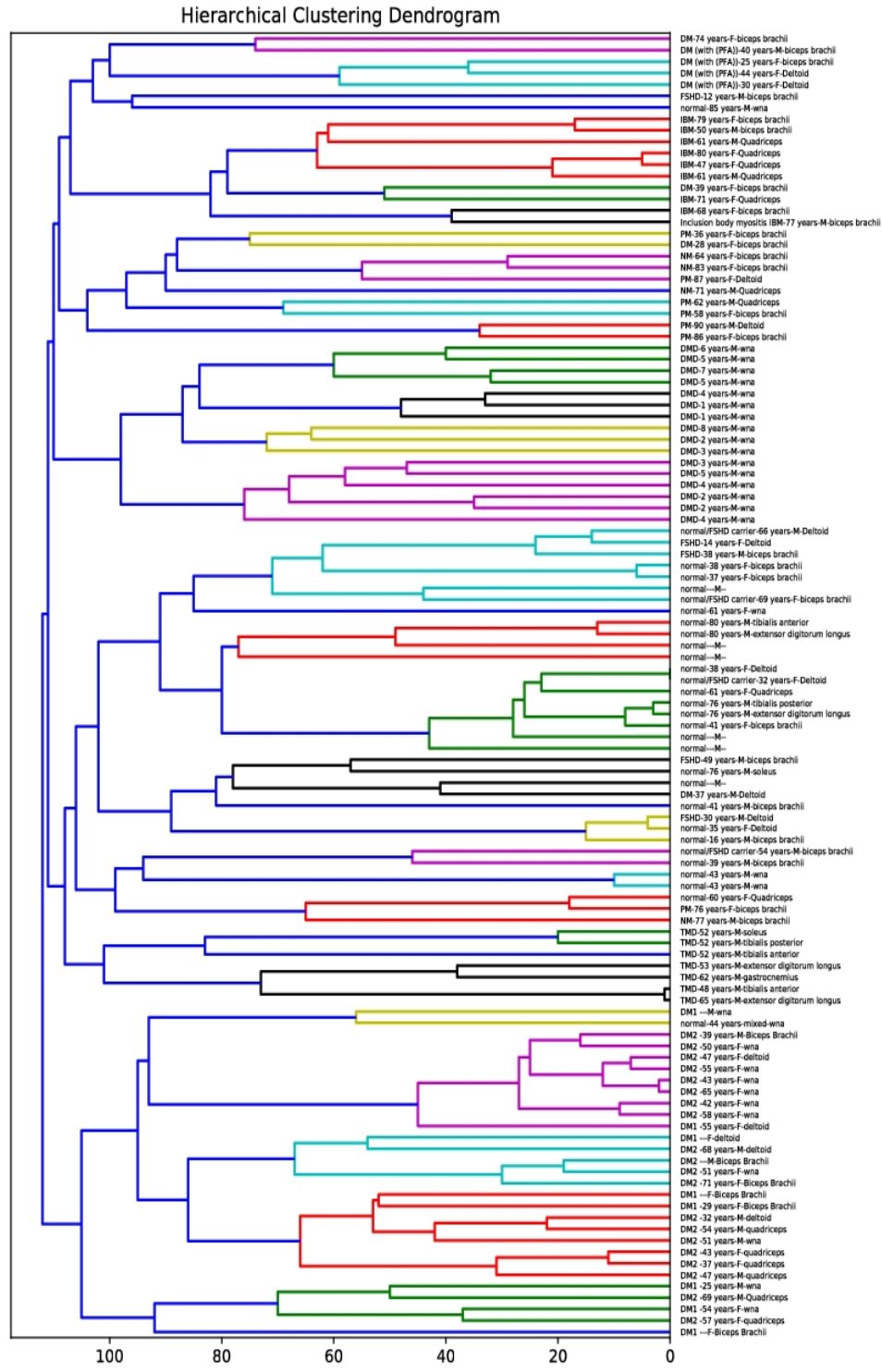


Figure A.91: Dendrogram of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Repetition Variant with $(t_1=\mathbf{X}, \text{Remove Bin } 1 = \sqrt{t_2}\mathbf{X}, b = 100, n_c = 7)$ on the KL-Divergence Matrix.

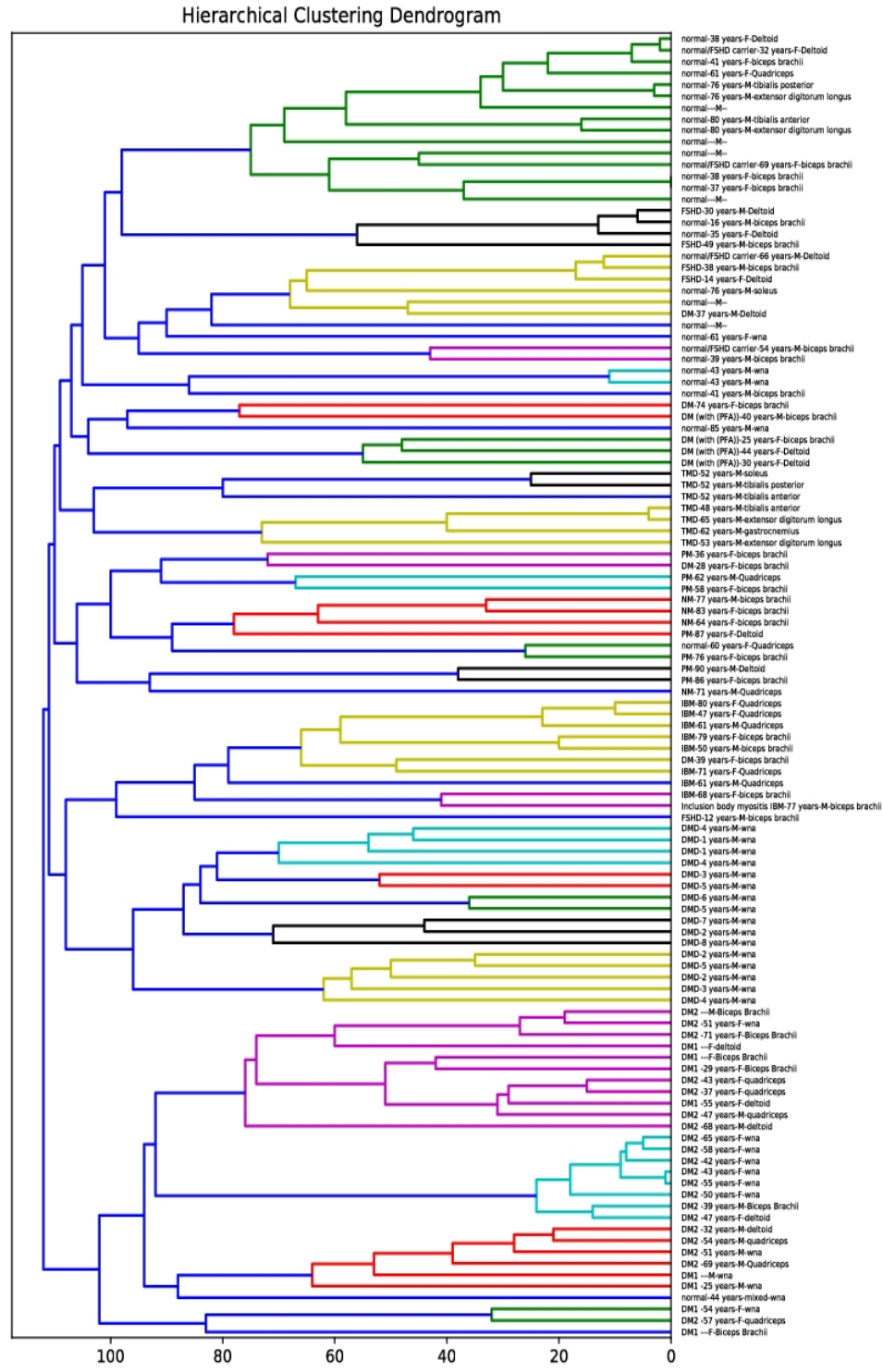


Figure A.92: Dendrogram of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Repetition Variant with ($t_1 = 0.0$, Remove Bin $I = \mathbf{X}t_2 = \mathbf{X}$, $h = 100$, $n_c = 8$) on the Relevance Matrix with $\lambda = 0.3$.



Figure A.93: Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Repetition Variant with ($t_1 = 0$, Remove Bin 1 = \mathbf{X} , $t_2 = 0.2$, $h = 150$, $n_c = 7$) on the KL-Divergence Matrix.

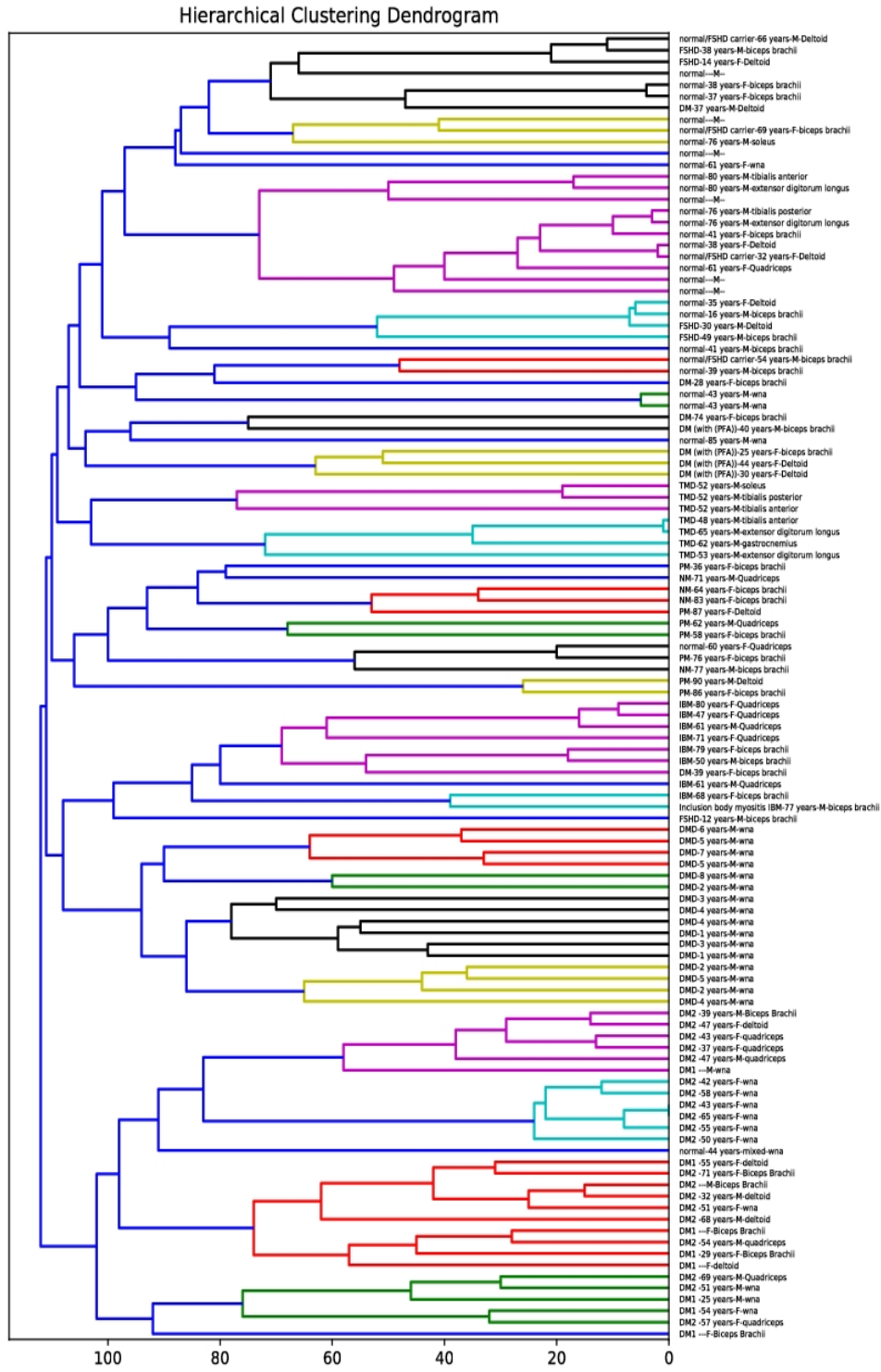


Figure A.94: Dendrogram of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Repetition Variant with ($t_1 = 0$, Remove Bin $1 = \sqrt{t_2} = \mathbf{X}$, $b = 150$, $n_c = 7$) on the KL-Divergence Matrix.

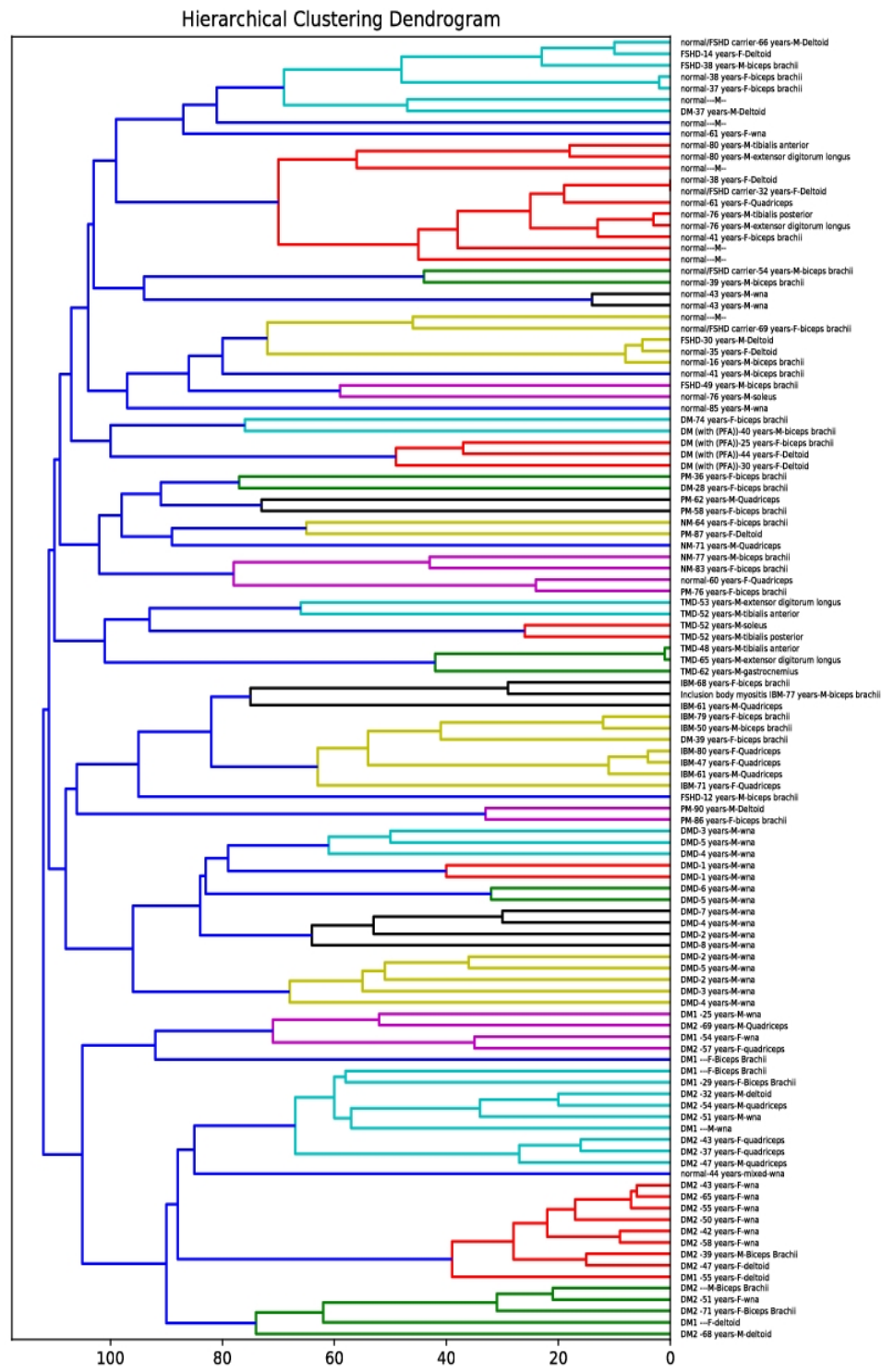


Figure A.95: Dendrogram of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Repetition Variant with ($t_1 = 0.2$, Remove Bin 1 = $\mathbf{X}_{t_2} = \mathbf{X}$, $h = 150$, $n_c = 8$) on the Topic-Term Matrix.

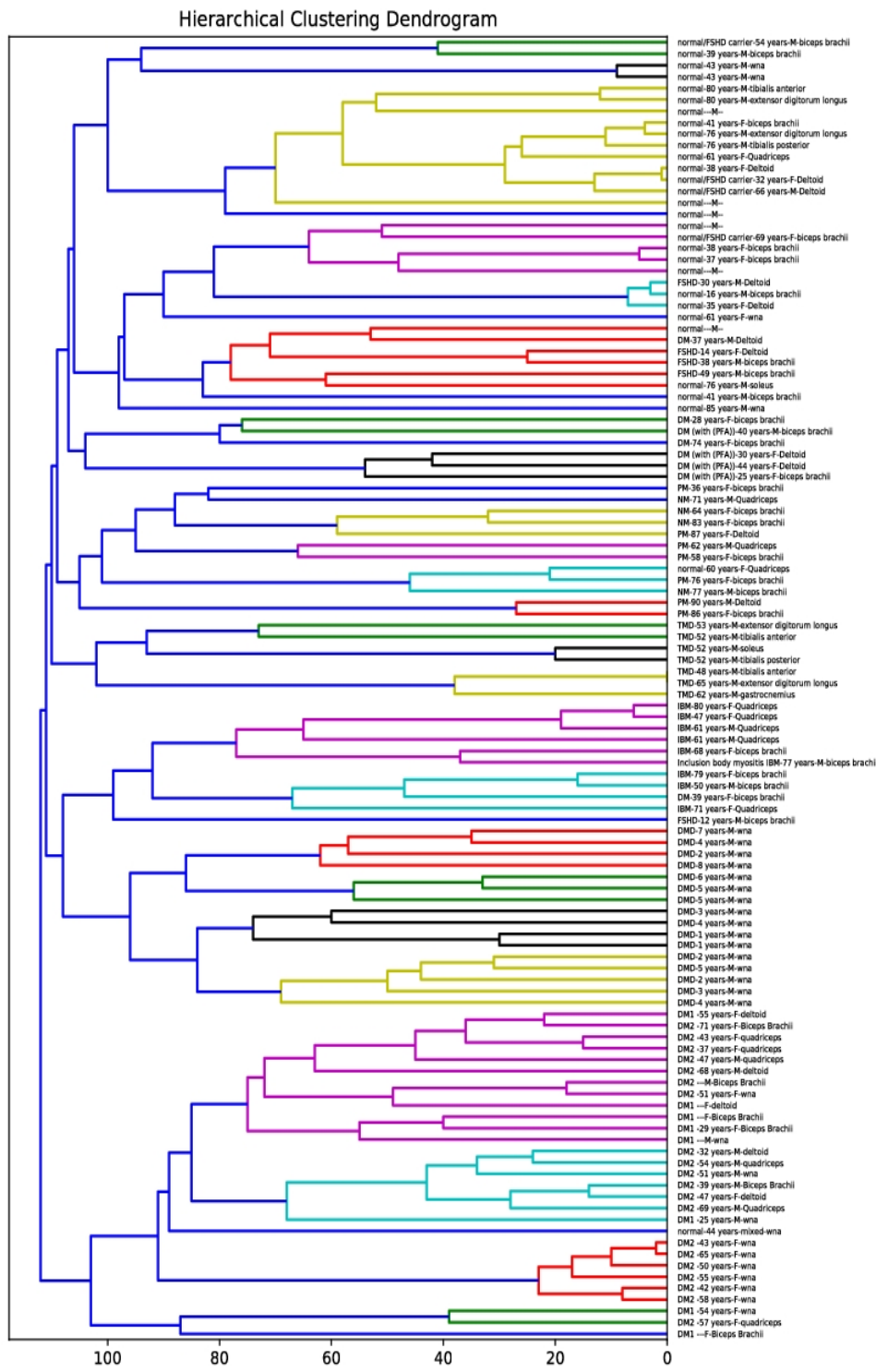


Figure A.96: Dendrogram of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Repetition Variant with ($t_1 = 0.2$, Remove Bin 1 = \mathbf{X} , $t_2 = 0.3$, $h = 100$, $n_c = 7$) on the Relevance Matrix with $\lambda = 0.3$.

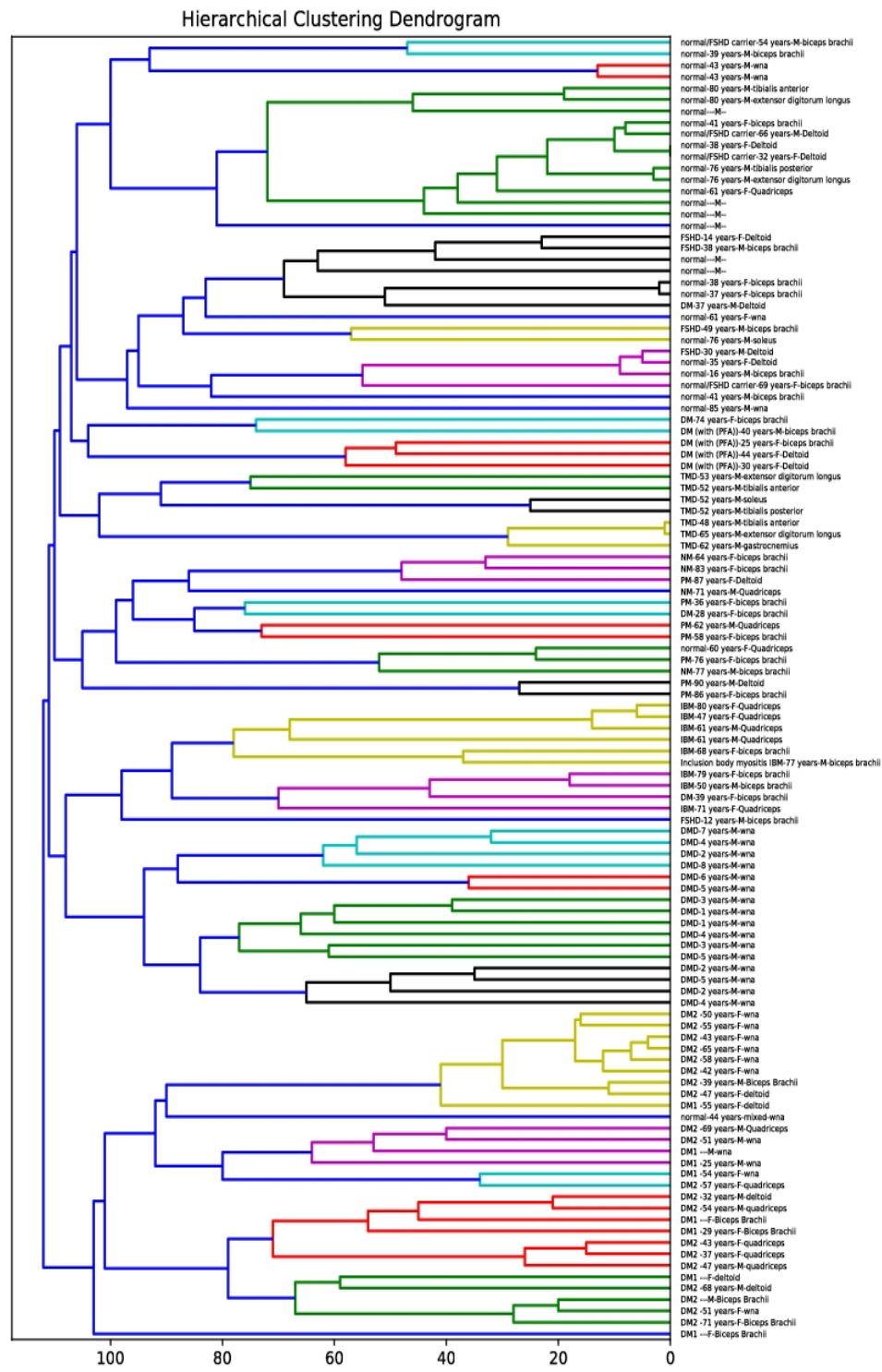


Figure A.97: Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Repetition Variant with ($t_1 = 0.2$, Remove Bin $1 = \sqrt{t_2} = \mathbf{X}$, $b = 200$, $n_c = 11$) on the KL-Divergence Matrix.

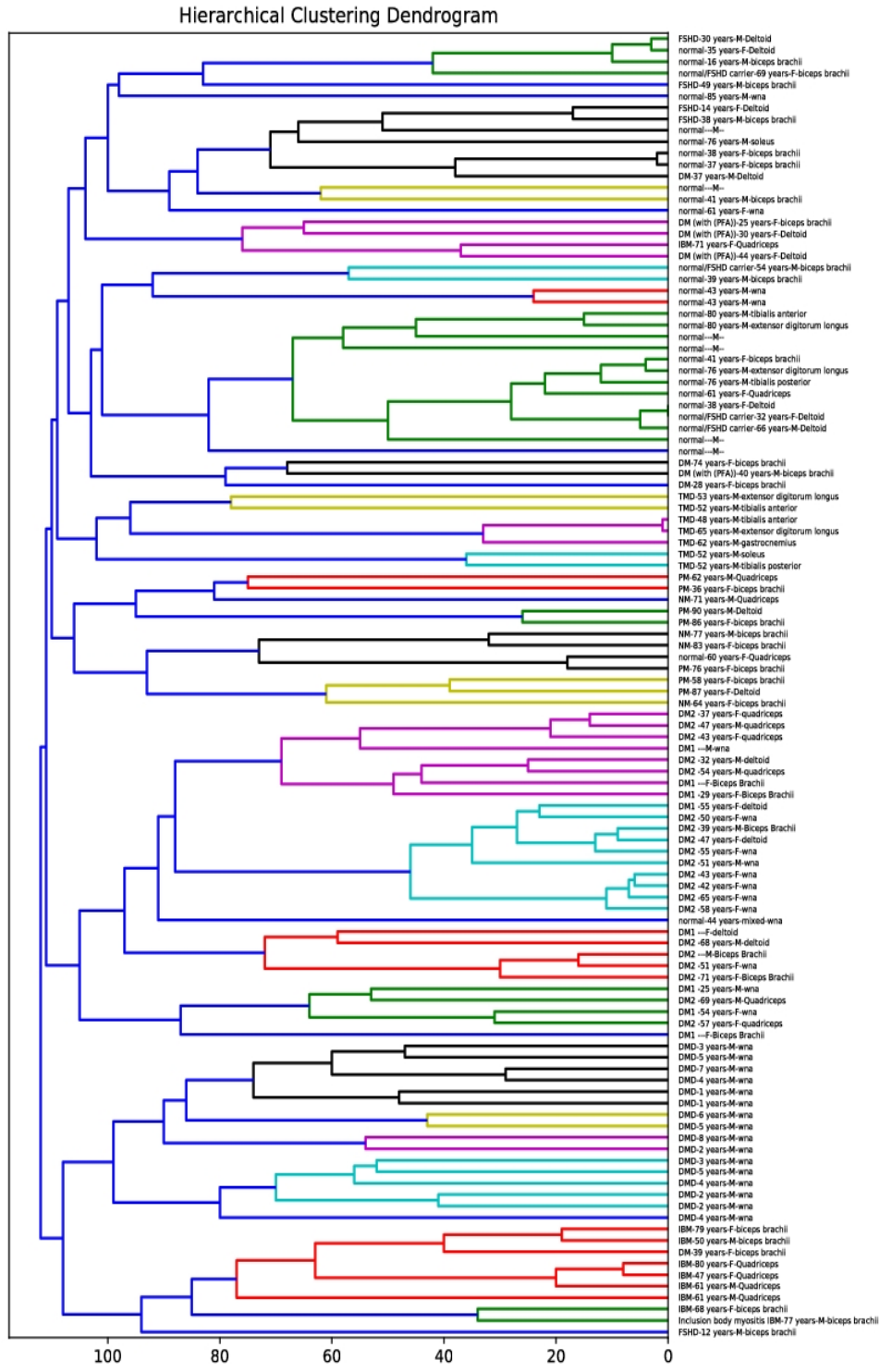


Figure A.98: Dendrogram of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Median Variant with $t_M = \mathbf{X}$, $h = 150$, $n_c = 11$ on the KL-Divergence Matrix.

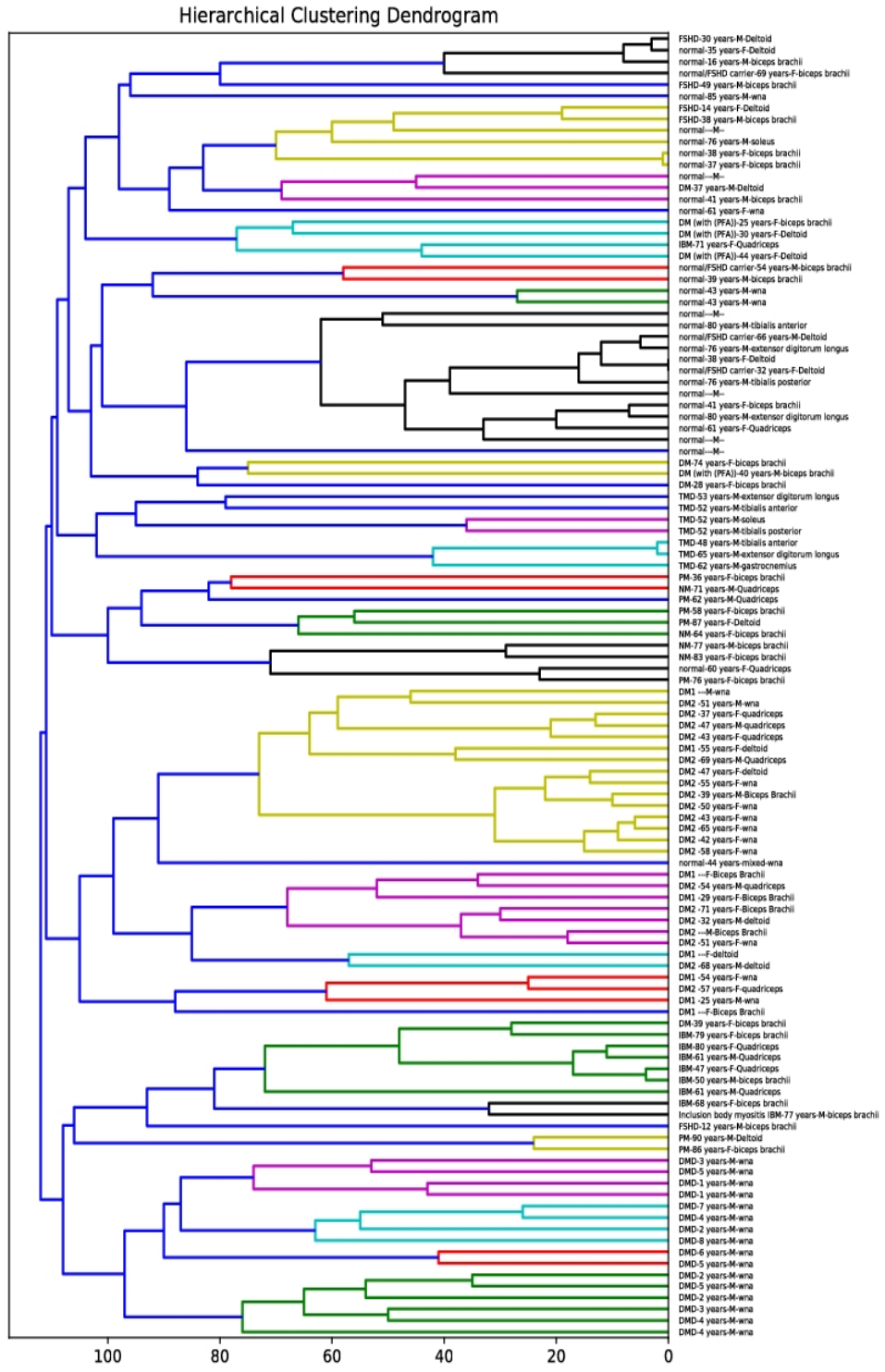


Figure A.99: Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Median Variant with $t_M = 0.2$, $h = 100$, $n_c = 11$ on the Topic-Term Matrix.

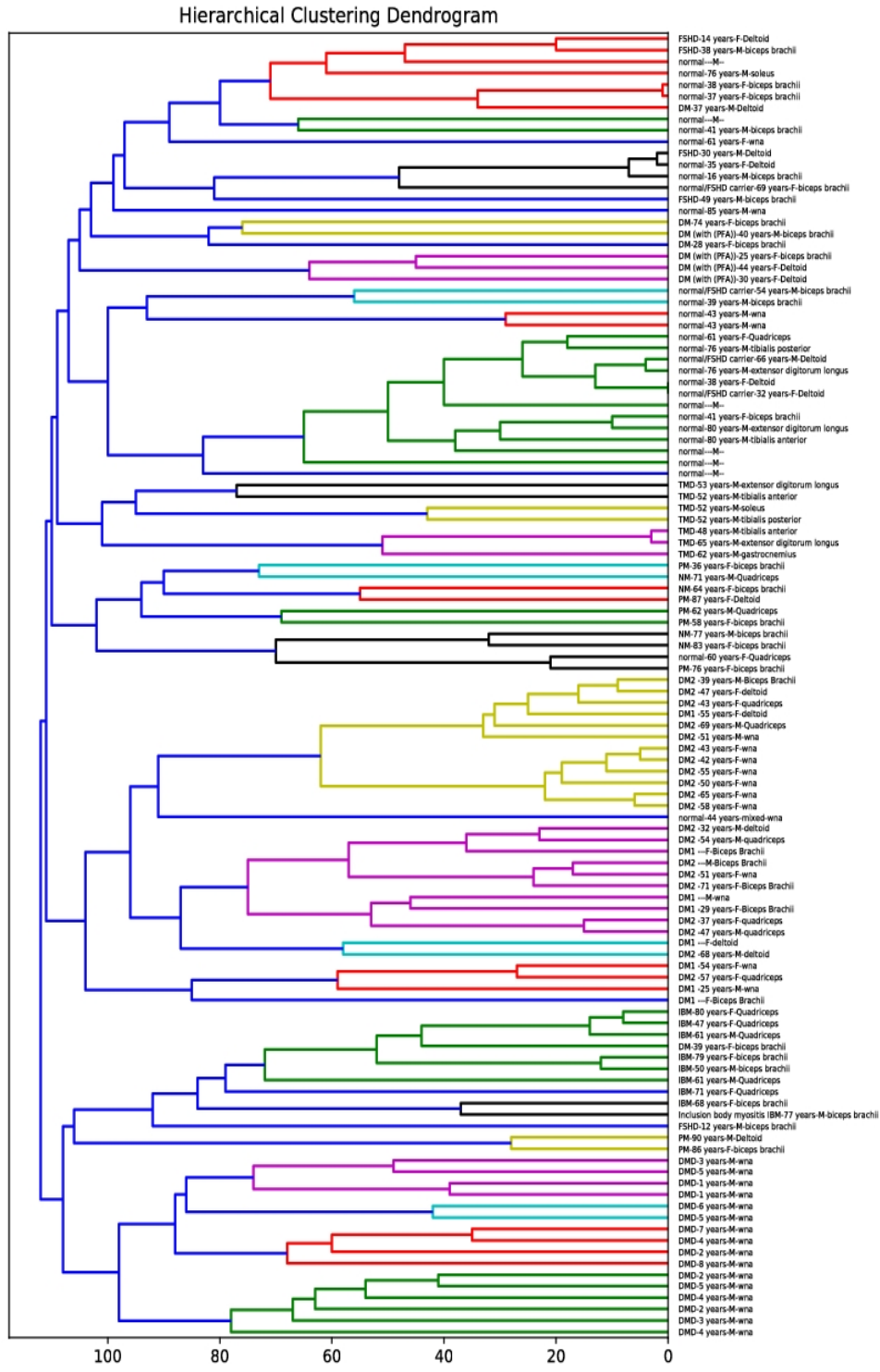


Figure A.100: Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Median Variant with $t_M = 0.0$, $b = 200$, $n_c = 11$ on the Relevance Matrix with $\lambda = 0.3$.



Figure A.10r: Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Bibin Variant with $t_B = \mathbf{X}$, $h = 50$, $n_c = 7$ on the Topic-Term Matrix.

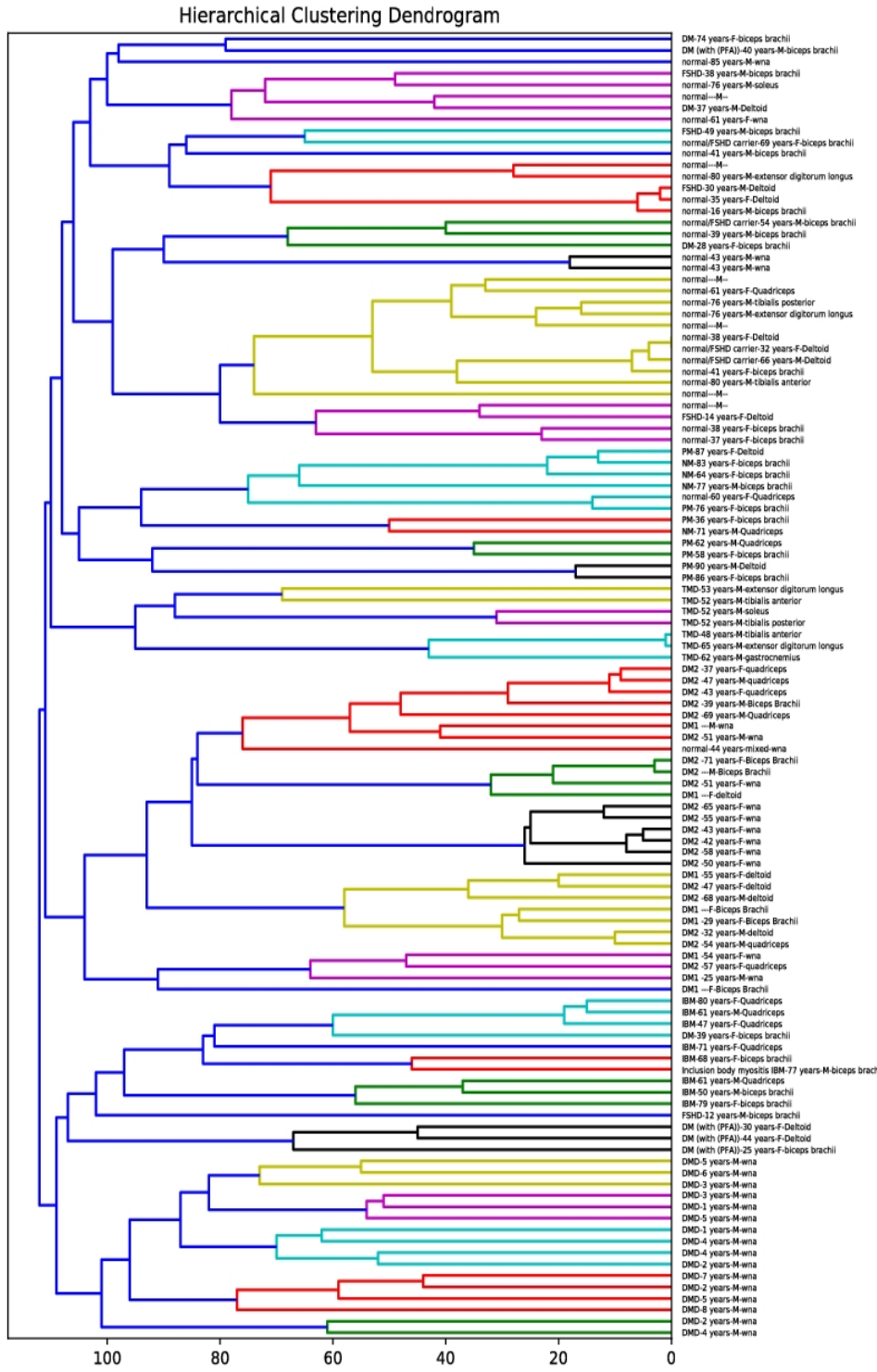


Figure A.102: Dendrogramm of HAC using Unsupervised Feature Selection with LDA on the Muscle Dataset with Bibin Variant with $t_B = 0.2$, $h = 50$, $n_c = 7$ on the Relevance Matrix with $\lambda = 0.3$.

Bibliography

- [Altman, 1992] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.
- [Arthur and Vassilvitskii, 2007] Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.
- [Aziz et al., 2017] Aziz, R., Verma, C., and Srivastava, N. (2017). Dimension reduction methods for microarray data: a review. *AIMS Bioengineering*, 4(2):179–197.
- [Barrett et al., 2011] Barrett, T., Troup, D., Wilhite, S., Ledoux, P., Evangelista, C., Kim, I., Tomashevsky, M., Marshall, K., Phillippy, K., Sherman, P., Muerter, R., Holko, M., Ayanbule, O., Yefanov, A., and Soboleva, A. (2011). Ncbi geo: Archive for functional genomics data sets - update. *Nucleic acids research*, 39:D1005–10.
- [Bellman, 2015] Bellman, R. E. (2015). *Adaptive control processes: a guided tour*, volume 2045. Princeton university press.
- [Berkhin, 2006] Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer.
- [Berrar et al., 2009] Berrar, D. P., Dubitzky, W., and Granzow, M. (2009). *A Practical Approach to Microarray Data Analysis*. Springer Publishing Company, Incorporated, 1st edition.
- [Biba et al., 2007] Biba, M., Esposito, F., Ferilli, S., Di Mauro, N., and Basile, T. M. A. (2007). Unsupervised discretization using kernel density estimation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI’07*, pages 696–701, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Bicego et al., 2010] Bicego, M., Lovato, P., Ferrarini, A., and Delledonne, M. (2010). Biclustering of expression microarray data with topic models. In *2010 20th International Conference on Pattern Recognition*, pages 2728–2731.
- [Bicego et al., 2010] Bicego, M., Lovato, P., Oliboni, B., and Perina, A. (2010). Expression microarray classification using topic models. In *Proceedings of the 2010 ACM Symposium on Applied Computing, SAC ’10*, pages 1516–1520, New York, NY, USA. ACM.

- [Bicego et al., 2012] Bicego, M., Lovato, P., Perina, A., Fasoli, M., Delledonne, M., Pezzotti, M., Polverari, A., and Murino, V. (2012). Investigating topic models’ capabilities in expression microarray data classification. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 9:1831–1836.
- [Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- [Blei, 2012a] Blei, D. M. (2012a). Probabilistic topic models. *Commun. ACM*, 55(4):77–84.
- [Blei, 2012b] Blei, D. M. (2012b). Probabilistic topic models. *Commun. ACM*, 55(4):77–84.
- [Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.
- [Braschi et al., 2018] Braschi, B., Denny, P., Gray, K., Jones, T., Seal, R., Tweedie, S., Yates, B., and Bruford, E. (2018). Genenames. org: the hgnc and vgnc resources in 2019. *Nucleic acids research*, 47(D1):D786–D792.
- [Brazma et al., 2001] Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C., et al. (2001). Minimum information about a microarray experiment (miame)—toward standards for microarray data. *Nature genetics*, 29(4):365.
- [Chen et al., 2012] Chen, X., He, T., Hu, X., Zhou, Y., An, Y., and Wu, X. (2012). Estimating functional groups in human gut microbiome with probabilistic topic models. *IEEE transactions on nanobioscience*, 11(3):203–215.
- [Chen et al., 2011] Chen, X., Hu, X., Lim, T. Y., Shen, X., Park, E., and Rosen, G. L. (2011). Exploiting the functional and taxonomic structure of genomic data by probabilistic topic modeling. *IEEE/ACM transactions on computational biology and bioinformatics*, 9(4):980–991.
- [Coelho et al., 2010] Coelho, L. P., Peng, T., and Murphy, R. F. (2010). Quantifying the distribution of probes between subcellular locations using unsupervised pattern unmixing. *Bioinformatics*, 26(12):i7–i12.
- [Crick, 1958] Crick, F. H. (1958). On protein synthesis. In *Symp Soc Exp Biol*, volume 12, page 8.
- [Dai et al., 2005] Dai, M., Wang, P., Boyd, A. D., Kostov, G., Athey, B., Jones, E. G., Bunney, W. E., Myers, R. M., Speed, T. P., Akil, H., et al. (2005). Evolving gene/transcript definitions significantly alter the interpretation of genechip data. *Nucleic acids research*, 33(20):e175–e175.
- [Deerwester et al., 1990] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS*, 41:391–407.
- [Dey et al., 2017] Dey, K., Joyce Hsiao, C., and Stephens, M. (2017). Visualizing the structure of rna-seq expression data using grade of membership models. *PLOS Genetics*, 13:e1006599.
- [Dogma,] Dogma, C. Central Dogma of molecular biology. Accessed: 2019-10-15.
- [Georgara et al., 2018] Georgara, A., Ntiniakou, T., and Chalkiadakis, G. (2018). Learning hedonic games via probabilistic topic modeling. In *EUMAS*.
- [Griffiths and Steyvers, 2004] Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

- [Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.
- [Haykin et al., 2009] Haykin, S. S. et al. (2009). *Neural networks and learning machines/Simon Haykin*. New York: Prentice Hall.
- [Hira and Gillies, 2015] Hira, Z. M. and Gillies, D. F. (2015). A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics*, 2015.
- [Hofmann, 1999] Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI’99, pages 289–296, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Hospedales et al., 2012] Hospedales, T., Gong, S., and Xiang, T. (2012). Video behaviour mining using a dynamic topic model. *International journal of computer vision*, 98(3):303–323.
- [Johnson et al., 2007] Johnson, W. E., Li, C., and Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127.
- [Kho et al., 2017] Kho, S. J., Yalamanchili, H. B., Raymer, M. L., and Sheth, A. P. (2017). A novel approach for classifying gene expression data using topic modeling. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, ACM-BCB ’17, pages 388–393, New York, NY, USA. ACM.
- [Kinsella et al., 2011] Kinsella, R. J., Kähäri, A., Haider, S., Zamora, J., Proctor, G., Spudich, G., Almeida-King, J., Staines, D., Derwent, P., Kerhornou, A., et al. (2011). Ensembl biomarts: a hub for data retrieval across taxonomic space. *Database*, 2011.
- [Kolesnikov et al., 2014] Kolesnikov, N., Hastings, E., Keays, M., Melnichuk, O., Tang, A., Williams, E., Dylag, M., Kurbatova, N., Brandizi, M., Burdett, T., Megy, K., Pilicheva, E., Rustici, G., Tikhonov, A., Parkinson, H., Petryszak, R., Sarkans, U., and Brazma, A. (2014). Arrayexpress update—simplifying data submissions. *Nucleic acids research*, 43.
- [Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86.
- [Lazar et al., 2012] Lazar, C., Taminiau, J., Menganck, S., Steenhoff, D., Coletta, A., Molter, C., de Schaetzen, V., Duque, R., Bersini, H., and Nowe, A. (2012). A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(4):1106–1119.
- [Leek et al., 2012] Leek, J. T., Johnson, W. E., Parker, H. S., Jaffe, A. E., and Storey, J. D. (2012). The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics*, 28(6):882–883.
- [Liu et al., 2016] Liu, L., Tang, L., Dong, W., Yao, S., and Zhou, W. (2016). An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus*, 5(1):1608.

- [Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.
- [Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.
- [Malatras et al., 2019] Malatras, A., Duguez, S., and Duddy, W. (2019). Muscle gene sets: A versatile methodological aid to functional genomics in the neuromuscular field. *Skeletal Muscle*, 9.
- [Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- [McCallum, 2002] McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- [Mitchell et al., 1997] Mitchell, T. M. et al. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877.
- [NCI, 2019] NCI (2019). The cancer genome atlas program.
- [Ng and Jordan, 2002] Ng, A. Y. and Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848.
- [Parmigiani et al., 2003] Parmigiani, G., Garrett, E. S., Irizarry, R. A., and Zeger, S. L. (2003). The analysis of gene expression data: an overview of methods and software. In *The analysis of gene expression data*, pages 1–45. Springer.
- [Pearson, 2006] Pearson, H. (2006). What is a gene? *Nature*, 441(7092):398–401.
- [Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- [Perina et al., 2010] Perina, A., Lovato, P., Murino, V., and Bicego, M. (2010). Biologically-aware latent dirichlet allocation (balda) for the classification of expression microarray. In *IAPR International Conference on Pattern Recognition in Bioinformatics*, pages 230–241. Springer.
- [Piccolo et al., 2012] Piccolo, S. R., Sun, Y., Campbell, J. D., Lenburg, M. E., Bild, A. H., and Johnson, W. E. (2012). A single-sample microarray normalization method to facilitate personalized-medicine workflows. *Genomics*, 100(6):337–344.
- [Piccolo et al., 2013] Piccolo, S. R., Withers, M. R., Francis, O. E., Bild, A. H., and Johnson, W. E. (2013). Multiplatform single-sample estimates of transcriptional activation. *Proceedings of the National Academy of Sciences*, 110(44):17778–17783.
- [Pitman, 1995] Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability theory and related fields*, 102(2):145–158.

- [Řehůřek and Sojka, 2010] Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. <http://is.muni.cz/publication/884893/en>.
- [Rogers et al., 2005] Rogers, S., Girolami, M., Campbell, C., and Breitling, R. (2005). The latent process decomposition of cDNA microarray data sets. *IEEE/ACM transactions on computational biology and bioinformatics*, 2(2):143–156.
- [Rokach and Maimon, 2005] Rokach, L. and Maimon, O. (2005). *Clustering Methods*, pages 321–352. Springer US, Boston, MA.
- [Sievert and Shirley, 2014] Sievert, C. and Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA. Association for Computational Linguistics.
- [Tarca et al., 2006] Tarca, A. L., Romero, R., and Draghici, S. (2006). Analysis of microarray experiments of gene expression profiling. *American journal of obstetrics and gynecology*, 195(2):373–388.
- [Teh et al., 2005] Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2005). Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392.
- [Verleysen and François, 2005] Verleysen, M. and François, D. (2005). The curse of dimensionality in data mining and time series prediction. In *International Work-Conference on Artificial Neural Networks*, pages 758–770. Springer.
- [Wang, 2010] Wang, C. (2010). C++ implementation of hierarchical dirichlet process for topic modeling. <https://github.com/blei-lab/hdp>.
- [Yalamanchili et al., 2017] Yalamanchili, H. B., Kho, S. J., and Raymer, M. L. (2017). Latent dirichlet allocation for classification using gene expression data. In *2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 39–44. IEEE.
- [Zhao et al., 2014] Zhao, W., Zou, W., and J Chen, J. (2014). Topic modeling for cluster analysis of large biological and medical datasets. *BMC bioinformatics*, 15 Suppl 11:S11.