Machine Learning for Genomic Data

Undergraduate Thesis

Author: Akankshita Dash
Supervisor: Assoc. Prof. Vincent Tan
Co-Supervisor: Dr. Huili Guo

A thesis submitted in partial fulfilment of the requirement for the degree of Bachelor of Science with Honours in Applied Mathematics

Department of Mathematics
Faculty of Science
National University of Singapore

8th March, 2019

Acknowledgements

First and foremost, I would like to express my utmost gratitude to my supervisors; **Assoc. Professor Vincent Tan**, for introducing me to machine learning in MA4270, and **Dr. Huili Guo** for breaking down complex bioinformatics concepts into simple ones for me to understand. I am extremely grateful for their guidance during the entire period of my honours project. Throughout the year, they have been understanding of my strengths and weaknesses and tolerant of my mistakes, guiding me along when I hit roadblocks, all the while suggesting new ideas and patiently helping me clarify my doubts. I have learnt so much from both of them, both about academia and life, and without their constant guidance, this thesis would not have been possible. I couldn't have asked for better mentors. Thank you.

I would also like to thank the people in Dr. Huili Guo's lab, especially Indrik and Shawn, for being so helpful when I was struggling with certain concepts. I also express my appreciation to Zhao Yue for being a helpful companion on this journey, and wish him well for his FYP.

To my friends - Spatika, Srishti, Megha, Bushra and Kai Ting - thank you for your support and encouragement. You have listened to my never-ending rants, put up with my idiosyncrasies, seen me at my lowest points and yet still stuck by me.

Last but not the least, I would like to thank my family. Despite living so far, you have always managed to be there for me. I'm so grateful for your support and encouragement throughout my undergraduate years. To my sister and brother-in-law; despite your busy lives and the personal tragedy you faced last year, you continued lending a listening ear to my insignificant complaints. To my Mom and Dad - you have always been my pillars of support.You have never pressured me and always encouraged me to do my best. From sending me home-cooked meals, visiting me in Singapore when I couldn't make it home, and offering me life advice when I was at crossroads, you have helped me up when I felt like I was on the edge of failure. Thank you so much.

Akankshita Dash

Abstract

This report explores the application of machine learning techniques on short time-series gene expression data. Although standard machine learning algorithms work well on longer time-series', they often fail to find meaningful insights from fewer timepoints.

In this report, we explore model-based clustering techniques. We combine popular unsupervised learning techniques like K-Means, Gaussian Mixture Models, Bayesian Networks, Hidden Markov Models with the well-known Expectation Maximization algorithm. K-Means and Gaussian Mixture Models are fairly standard, while Hidden Markov Model and Bayesian Networks clustering are more novel ideas that suit time-series gene expression data.

Contents

1 Introduction **2**
1.1 Machine Learning 2
1.2 Genomics 3
1.2.1 Key Definitions and Terms 3
1.2.2 Central Dogma 4
1.2.3 Virus-Host Interaction 5
1.2.4 Hand, Foot, and Mouth Disease 5
1.2.4.1 GTPases 6
1.2.5 Objective 7
1.3 Data Generation through Genome Sequencing Techniques 7
1.3.1 Methodology 7

2 Literature Review **9**
2.1 Clustering Time Series Gene Expression Values 9
2.2 Algorithms and Models 10
2.2.1 Expectation Maximisation (EM) Algorithm 10
2.2.1.2 Limitations 11
2.2.2 K-Means 11
2.2.2.1 Limitations 12
2.2.3 Gaussian Mixture Models 12
2.2.3.1 Covariance Matrix 14
2.2.3.2 Limitations 14
2.2.4 Hidden Markov Models 15
2.2.4.1 HMM-based Clustering 17
2.2.4.2 Limitations 18
2.2.5 Bayesian Networks 18
2.2.5.1 Limitations 19
2.2.6 Hierarchical Clustering 19
2.3 Evaluation Metrics for Model Selection 21
2.3.1 Information Criteria 22
2.3.1.1 BIC and AIC 22
2.3.2 Elbow Method 22
2.3.3 Misclassification Error (ME) Distance 23
2.3.4 Gene Ontology 24

3 Exploratory Analysis 26
3.1 Data Description 26
3.1.1 Data Preprocessing 26
3.2 Initial Analysis 27

4 Results 29
4.1 Feature Engineering 29
4.2 Model-Based Clustering Results 31
4.2.1 K-Means 31
4.2.2 Gaussian Mixture Models 32
4.2.3 HMM-based Clustering 34
4.2.4 Bayesian Networks 35
4.3 Performance Evaluation 36
4.4 Biological Results 37

5 Conclusion 39
5.1 Summary 39
5.1.1 Biological Experiments 39

6 Future Work 40
6.1 Algorithms 40
6.1.1 Tuning 40
6.1.2 Different Algorithms 40

A Appendix 41
A.1 K-Means 41
A.1.1 BIC Plots 41
A.2 Gaussian Mixture Models 41
A.2.1 BIC Plots 41
A.2.2 Trend Plots 42
A.3 Bayesian Networks 42
A.3.1 Heatmaps for Complete Datasets 42
A.4 HMM-based Clustering 43
A.4.1 BIC Plots 43
A.4.2 Trend plots 43

1 Introduction

Machine learning is a field of artificial intelligence that uses statistical techniques to give a computer the ability to learn from data, without explicitly programming it to do so.

Genomics is a branch of molecular biology concerned with the structure, function, evolution, and mapping of an organism's complete set of DNA.

In this project, we apply machine learning techniques to *viral* genomic data recovered from human cells, and attempt to derive useful insights about the **EV-A71 virus**, also known as hand-foot-and-mouth disease (HFMD).

1.1 Machine Learning

With the large volume of data available nowadays, it is impossible to apply statistical techniques and derive useful insights without making use of a computer.

Considered one of the biggest innovations since the microchip, machine learning enables humans to build models and capture patterns within data. It allows one to define simple rules to improve the performance of these models. The applications of machine learning are manifold - from spam filters in e-mail to online banking, it is omnipresent. Machine learning's presence is growing, and its use cases continue increasing.

A machine learning model can be defined by its training data and the set of hypotheses functions.

Machine learning is broadly divided into the following 3 categories:-

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

In **supervised learning**, the training data are labelled, i.e. each data point is attached with its true value, or *ground truth.* Typical problems include *classification* (binary/multi-class) and *regression.* E.g. If given a training dataset consisting of pictures of cats and dogs, our model accurately predicts whether a new image is that of a cat or a dog. This is a *classification* problem. Another simple example is of a dataset with people's heights and weights. Given a new person's height, we should be able to accurately predict weight. This is a *regression* problem.

In **unsupervised learning**, unlike supervised learning, the training data are unlabelled. The number of labels may or may not be specified beforehand. Unsupervised learning is commonly used as an initial exploratory technique to draw useful patterns from the dataset. E.g. Given many different books, classifying them into different categories, or *clusters*, can help us identify the genres they belong to.

Reinforcement learning is different from both supervised and unsupervised learning. It focuses on maximising reward potential from the data by balancing the trade-off between exploration and exploitation. E.g. In recommendation engines, the items showed to a particular user may ordered based on how much they liked previous products or how *different* they are from previous products. This is done to maximise the number of clicks, i.e. the *reward.*

Since the data that we have in this project is unlabelled, our focus is on *unsupervised learning.*

1.2 Genomics

Genomes contain the complete set of genes present in an organism, and genomics describe how this genetic information is stored and interpreted in the cell.

1.2.1 Key Definitions and Terms

- **Nucleobases**: Components of nucleotides, that are the building blocks of DNA - Adenine (A), Thymine (T), Cytosine (C), Guanine (G). Additionally, Uracil (U) is used instead of Thymine (T) for RNA.

- **Codons**: A sequence of three DNA or RNA nucleotides that corresponds with a specific amino acid or stop signal during protein synthesis. For eukaryotes, there are a total of 64 such codons, which map to 20 amino acids (and stop signals) through the *genetic code.*

- **Amino acid**: Organic molecules that make up the building block of proteins. There are 20 canonical amino acids.

- **DNA**: Deoxyribonucleic acid, a double helix molecule, that contains the genetic material for all organisms on Earth (including viruses).

- **RNA**: Ribonucleic acid, a single helix, polymeric molecule that is essential in various biological roles in coding, decoding, regulation and expression of genes. Biologically active RNAs, including transport, ribosomal and small nuclear RNA (tRNA, rRNA, snRNAs) fold into unique structures guided by complementary pairing between nucleotide bases. Cellular organisms use messenger RNA (mRNA) nucleotides to direct synthesis of specific proteins.

- **Protein**: Sequences of amino acids that carry out the majority of cellular functions such as motility, DNA regulation, and replication.
- **Transcription**: The process in which a DNA sequence of a gene is rewritten, or *transcribed*, to make an RNA molecule. (In eukaryotes, the RNA polymerase makes a strand of mRNA from DNA).
- **Translation**: The sequence of the mRNA is decoded, or *translated*, to specify an amino acid sequence (i.e. protein).
- **Gene**: A region of DNA (deoxyribonucleic acid) coding either for the messenger RNA encoding the amino acid sequence in a polypeptide chain or for a functional RNA molecule.
- **Chromosome**: Cellular structures that contain genes. A chromosome comprises of a single DNA molecule that may be either circular or linear.
- **Virus**: A small infectious agent that replicates only inside the living cells of other organisms.
- **Gene Expression**: The generation of a functional gene product from the information encoded by a gene, through the processes of transcription and translation. Gene products are often proteins. [1]

1.2.2 Central Dogma

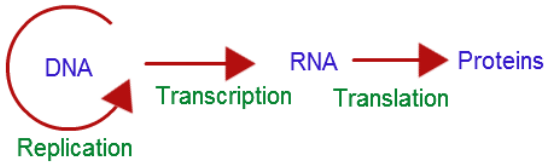


Figure 1: Central dogma of Molecular Biology

The central dogma of biology states that: DNA is *transcribed* by RNA polymerases into mRNA (messenger RNA), which is read by ribosomes to generate protein through *translation*. [9]

---

[1] Non-protein coding genes can encode functional RNA, including ribosomal RNA (rRNA), transfer RNA (tRNA) or small nuclear RNA (snRNA)

In this project, our focus is on translation. More formally, we will be studying the translational interaction between the EV-A71 virus and its host cell, by observing gene expression values at different timepoints.

1.2.3 Virus-Host Interaction

When a host gets infected with a virus, the virus hijacks the host cell's translation machinery, which affects protein production.

In the context of EV-A71, the viral replication process begins when a virus infects its host by attaching to the host cell and penetrating the cell wall or membrane. The viral genome then hijacks the host cell's translation mechanism, forcing it to replicate the viral genome by producing viral proteins to make new capsids (the protein shell of a virus that protects it). The viral particles are then assembled into virions. These virions burst out of the host cell during a process called lysis, killing the host cell. Some viruses take a portion of the host's membrane during the lysis process to form an envelope around the capsid. [1]

After infection with any virus, overall translation of all genes decreases.

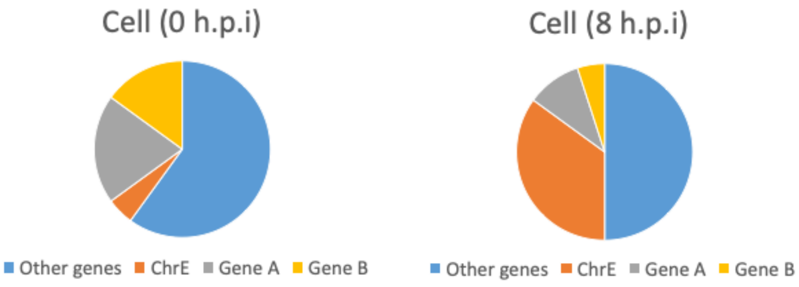


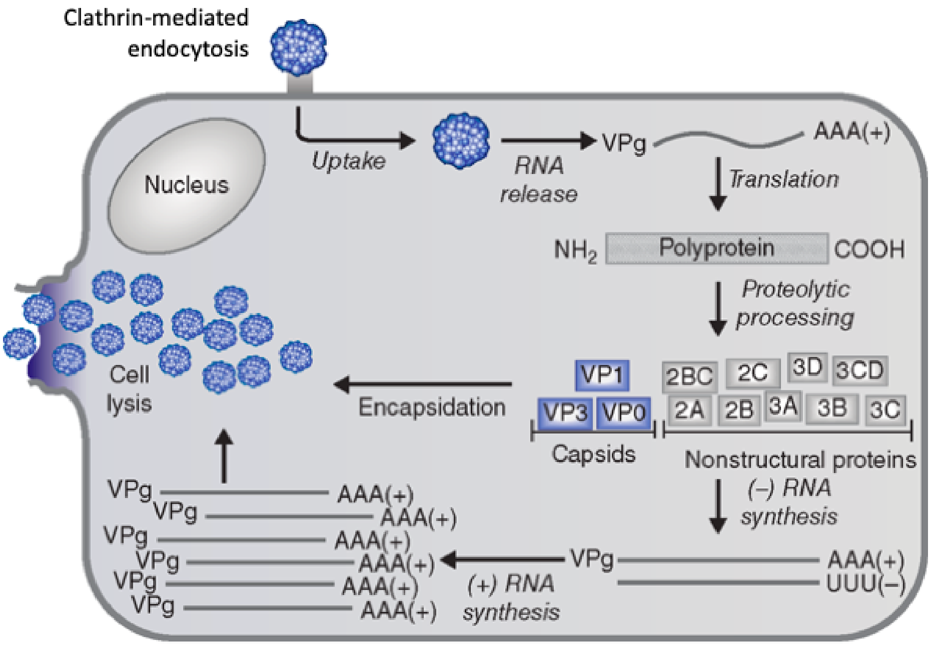
Figure 2: EV-A71 (chrE) infection over time*

*where Chromosome E (chrE) specifies the viral genes

In Figure 2, Following viral replication, the new viruses may go on to infect new hosts. Many viruses such as influenza, chicken pox, AIDS, the common cold, and rabies cause diseases in humans.

1.2.4 Hand, Foot, and Mouth Disease

Hand, foot, and mouth disease (HFMD) is a common infection caused by a group of viruses. The one we will be examining in this report is the enterovirus A71, or the EV-A71 virus. HFMD mostly affects small children, occasionally causing symptoms to develop in adults. It typically begins with a fever, which is followed by rashes and bumps on other parts of the body. Currently, there is no treatment that specifically targets the disease, nor is there a vaccine that is approved for use in Singapore [3].


Clathrin-mediated endocytosis
Uptake
RNA release
VPg
AAA(+)
Translation
Nucleus
NH$_2$
Polyprotein
COOH
Proteolytic processing
VP1
VP3
VP0
2BC
2C
3D
3CD
2A
2B
3A
3B
3C
Capsids
Nonstructural proteins
Cell lysis
Encapsidation
(–) RNA synthesis
VPg
AAA(+)
UUU(–)
(+) RNA synthesis
VPg
AAA(+)

Source: Brooks GF, Carroll KC, Butel JS, Morse SA, Mietzner TA: *Jawetz, Melnick, & Adelberg's Medical Microbiology*, 26th Edition: www.accessmedicine.com

Figure 3: EV-A71 life cycle

1.2.4.1 GTPases

Viruses depend on host cells for propagation and dissemination. Host cell mechanisms that allow viral entry, facilitate viral replication, and enable viral egress, are targeted for exploitation by viral pathogens [2].

Guanosine-5'-triphosphate (GTP) is a purine nucleoside triphosphate. It is one of the building blocks needed for the synthesis of RNA during the transcription process, and is involved in energy transfer within the cell. GTPases are a class of proteins that bind and hydrolyze GTP. GTPases include Rab, Ras, Rac, Ran, Rho, Arf, and Sar proteins.

Our focus is on the proteins Rab and Rho. Emerging evidence demonstrates that viruses have evolved numerous strategies to modulate Rab proteins' functions [2].

In polioviruses, the 2C protein possesses weak GTPase activity. The 2C protein in EV-A71 viruses plays an important role in viral replication [27]. Our hypothesis is that a certain member protein of GTPase is important for establishing the infection in EV-A71 viruses, and could be linked to Rho GTPases. We focus on the translation level changes in attempt to find information supporting our conjecture.

1.2.5 Objective

When a virus infects a cell, most of the processes in the cell get shut down and some proteins do not get generated. There are groups of:-

1. Genes that escape this shut down by the virus
2. Genes that aid in the shut down
3. Genes that are not influenced by the virus

[1] is the most important, as these are the ones that the virus needs for its own production.

Thus, our objective is to identify these groups from the set of genes that we have available. We do this by using unsupervised learning techniques on gene expression data to generate clusters.

1.3 Data Generation through Genome Sequencing Techniques

RNA sequencing refers to techniques used to determine the sequence of RNA molecules. It includes high-throughput shotgun sequencing of cDNA molecules obtained by reverse transcription from RNA, and next-generation sequencing technologies to sequence the RNA molecules within a biological sample in an effort to determine the primary sequence and relative abundance of each RNA molecule.

For this project, we primarily use gene expression data that has been generated using the following techniques:-

- **RNASeq**: RNASeq uses next-generation sequencing (NGS) to reveal the presence and total quantity of RNA in a biological sample at a given moment.
- **Ribosome Profiling (RPF)**: RPF uses specialised messenger RNA (mRNA) sequencing to determine which mRNAs are being actively translated. Unlike RNASeq, which sequences all of the mRNA of a given sequence present in a sample, RPF only targets mRNA sequences that are being actively translated.

1.3.1 Methodology

We posit that host factors regulated in viral translation may be important for establishing viral infection. The infection and analysis was carried out as follows -

1. Infect cells at various timepoints
2. Collect lysates post infection (cold synced) at various timepoints - 0 hours post infection (h.p.i), 2 h.p.i, 4 h.p.i, 6 h.p.i, 8 h.p.i
3. Perform clustering analysis to identify relevant genes
4. Validate changes in genes
5. Test for functions of genes in viral infection

2 Literature Review

2.1 Clustering Time Series Gene Expression Values

Clustering techniques are essential in the data mining process, as they reveal natural structures and identify interesting patterns in the underlying data.

For gene expression data observed at different time-points, clustering can help identify biologically relevant groups of genes and samples.

Through the principle of guilt-by-association [23], if we find a known cluster that contains some uncharacteristic genes, we can hypothesise that these genes are also significant.

As mentioned in [11], there is no one-size-fits-all solution to clustering, or even a consensus of what an ideal clustering should be like. Clusters may have arbitrary shapes and sizes, and each clustering criterion imposes a certain structure on the data. If the data happen to conform to the requirements of a particular criterion, the true clusters are recovered, however, very rarely is this the case. Each algorithm imposes its own set of biases on the clusters it determines. Most (reasonable) clustering algorithms would yield similar results on synthetically constructed datasets; however, in practice, they can give widely differing results on real-world,noisy, gene expression data.

For temporal gene expression data, the most commonly used clustering algorithms are [12, 16, 24] are

1. Model Based Clustering
    - K-Means
    - Gaussian Mixture Models
    - Hidden Markov Models
    - Bayesian Networks
2. Hierarchical Clustering
3. Self Organising Maps

For all our model based clustering techniques, we determine partitions by using the expectation-maximisation (EM) algorithm (described in 2.2.1) for maximum likelihood estimates of model parameters .

In this report, we will use hierarchical clustering for exploratory analysis, and apply all the model based clustering techniques to the different datasets we are provided with.

2.2 Algorithms and Models

2.2.1 Expectation Maximisation (EM) Algorithm

The expectation maximisation (EM) algorithm is a general technique for finding maximum likelihood solutions for probabilistic models having latent variables [5]. EM is used for all the model based techniques covered in this report.

2.2.1.1 Procedure

Consider a probabilistic model in which we collectively denote all the observed variables by $\mathbf{X}$ and the hidden variables by $\mathbf{Z}$. The join distribution is governed by a set of parameters, denoted θ. We try to maximise the likelihood function, that is given by

$$p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta) \tag{1}$$

If $\mathbf{Z}$ is continuous, we replace the summation with integration as appropriate. The maximum likelihood estimator is

$$\theta = \arg\min_{\theta} f(\theta)$$

Direct optimisation of $p(\mathbf{X}|\theta)$ is difficult due to the presence of latent variables, so we attempt to optimise the expected value of the log-likelihood of the data given our model. We are now ready to define the general EM algorithm in algorithm 1.

Algorithm 1 Expectation Maximisation

Initialisation: For $l = 0$, make an initial estimate of the parameters θ.
E Step (l^{th} iteration): Evaluate p($\mathbf{Z}|\mathbf{X}$,θ), i.e find the conditional probability distribution of the latent variables given our current model $\theta^{(l)}$ and the data $\mathbf{X}$.
M Step (l^{th} iteration): Update θ as

$$\theta^{(l+1)} = \arg\min_{\theta} u(\theta, \theta^{(l)} \tag{2}$$

where

$$u(\theta, \theta^{(l)}) = -\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{(l)}) \log p(\mathbf{X}, \mathbf{Z}|\theta) + \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{(l)}) \log p(\mathbf{Z}|\mathbf{X}, \theta^{(l)}) \tag{3}$$

End: Terminate when one of the following criteria is met

1. Iterate till l reaches the maximum number of iterations specified
2. Till the parameter estimates stop changing
3. Till the likelihood function stop changing

2.2.1.2 Limitations

The EM Algorithm typically converges to the local optima. Thus, it needs to be run multiple times in order to find the best fitting model and its parameters.

2.2.2 K-Means

K-Means is an algorithm to identify groups, or clusters, of data points in a multidimensional space [5].

We group data points based on their feature similarities according to a given distance metric (Euclidean Distance in our case), by minimising a given cost function.

Given a set of data points $D = \{x_t\}_{t=1}^{n}$ and fixed number of clusters $K \leq n$, the algorithm works as follows [26]-

Algorithm 2 K-Means Clustering

```
**Step 0, Initialisation:** For $j = 1, .., K$, initialise centers ${\mu_j}^{(1)}$ randomly.
**Step $l \in \mathbb{N}$, Assignment (E) Step:** Assign each point $x_t$ to its closest mean
    $${j_t}^{(l)} = \arg\min_j ||x_t - \mu_j||^2$$
**Step $l \in \mathbb{N}$, Update (M) Step:** Recompute $\mu_j$'s as means of the assigned points:
    $$\mu_j^{(l+1)} = \frac{1}{|\{t : j_t^{(l)} = j\}|} \sum_{t : j_t^{(l)} = j} x_t$$
**End:** Terminate when cluster assignments $j_t$ do not change.
```

For a given partition $D = \cup_{j=1}^{K} D_j$, where

$$D_j := \{x \in D : j = \arg\min_j ||x - \mu'_j||^2\}$$

this algorithm minimises the overall squared error:

$$J(D_1, \ldots, D_n, \mu_1, \ldots, \mu_n) = \sum_{j=1}^{K} \sum_{x \in D_j} ||x - \mu_j||^2 \tag{4}$$

where $D_1, D_2, \ldots, D_n$ are the clusters and $\mu_1, \mu_2, \ldots, \mu_n$ are cluster means. The two steps of updating the clusterings and re-estimating the centers correspond to the E step and M step of the EM Algorithm respectively.

Equation (2.2.2) is also known as the cost function.

The main problem of k-means is its dependency on the initially chosen centers. If the centers are chosen arbitrarily, then the algorithm converges very slowly or results in poor clustering. However, this can be solved by using the **K-Means++** [4] algorithm, which selects centers that are far apart from each other in a probabilistic manner.

Through K-Means++, we choose the initial K centers as follows [26]-

1. Take a random data point μ_1 chosen uniformly from dataset D.
2. Calculate new center μ_i, by choosing $x \in D$ with probability $\frac{D(x)^2}{\sum_{x' \in D} D(x')^2}$, (where $D(x)$ denotes the shortest distance from the data point x to its closest, already chosen center)
3. Repeat the previous step till we have K centers.

2.2.2.1 Limitations

Due to its ease of implementation and termination in a finite number of steps, K-Means is one of the most popular (and usually one of the first) algorithms applied for unsupervised learning problems.

However, due to the nature of the distance function chosen (Euclidean), K-means doesn't capture temporal dependencies in the data. Thus, it is not entirely suitable for our time series dataset. Moreover, k-means does hard clustering, i.e. a point can belong to only one cluster, while in real life data, a point can belong to more than one cluster.

2.2.3 Gaussian Mixture Models

Gaussian mixture models (GMMs) are mixtures, or linear combinations, of models with Gaussian distributions. They are widely used in data mining, statistical analysis and pattern recognition problems [5].

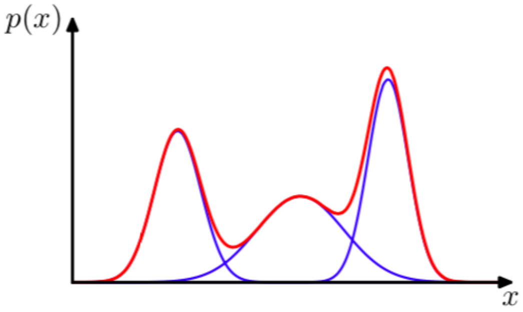


Figure 4: Mixture of 3 Gaussians in one dimension

A Gaussian mixture model is represented by

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) \tag{5}$$

where the equation above represents a mixture of K Gaussians, $\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$ is a *component* of the mixture model (each having its own mean μ_k and covariance Σ_k, and π_k are the component weights, such that $\sum_k \pi_k = 1$.

To use GMMs for clustering, we assume that our data are generated i.i.d from a mixture of K Gaussians, and we apply EM to estimate the K means, covariances, and weights, and thus determine the clusterings. We achieve this by introducing a latent variable z in (5). This latent variable has a 1-of-K representation - thus $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$, and find $p(\mathbf{z})$ and $p(\mathbf{x}|\mathbf{z})$, to get

$$p(\mathbf{x}) = \sum_{z} p(\mathbf{z}) p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) \tag{6}$$

where $p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$ and $p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k} \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)^{z_k}$.

Another quantity that is important is the conditional probability of $\mathbf{z}$ given $\mathbf{x}$, also known as the *responsibility* that component k takes for producing the observation $\mathbf{x}$, denoted by

$$\gamma(z_k) = p(z_k = 1|x) = \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)} \tag{7}$$

These formulas containing latent variable z help us formulate EM for GMM, given in 3.

The clusterings are indicated by the latent variable $\mathbf{z}$.

Algorithm 3 EM for Gaussian Mixture Models

```
Init: Initialise the means and covariances $\mu_k$ and $\Sigma_k$ and mixing co-efficients $\pi_k$, and evaluate the initial value of the log-likelihood.
Step l, E step: Evaluate the responsibilities from (7) using the current parameter values.
Step l, M Step: Re-estimate the parameters using the current responsibilities.
    $\mu_k^{(l+1)} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{n,k}) \mathbf{x}_n$
    $\Sigma_k^{(l+1)} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{n,k}) (\mathbf{x}_n - \mu_k^{(l+1)})(\mathbf{x}_n - \mu_k^{(l+1)})^T$
    $\pi_k^{(l+1)} = \frac{N_k}{N}$
where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$
Evaluate the log-likelihood
    $\log p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \log\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\}$
and terminate if one of the following criteria are met-
    1. Paramater estimates have converged (stopped changing).
    2. Log likelihood has converged.
    3. Maximum umber of iterations $l$ have been reached.
otherwise return to Step 2.
```

2.2.3.1 Covariance Matrix

Having different settings of the covariance matrix for each component help in restricting the number of parameters that need to be estimated (thus optimising the runtime of the algorithm). However, some of the settings (diagonal, tied and spherical) restrict the contour of the clusters obtained, as seen in figure 5. In the diagonal setting, the elliptical clusters obtained are aligned with the co-ordinate axes, while the spherical clusters are circular, while all clusters have the same shape in the tied setting. Assuming we have k components, here are 4 different covariance settings, summarised in the table 1.

2.2.3.2 Limitations

As discussed for previous models, GMM converges to a local optima, so we need to run the model multiple times to determine the best fit. Moreover, GMM takes

Covariance	Description	Parameters
Full	Each component has its own general covariance matrix	$\frac{KD(D+1)}{2}$
Diagonal	Each component has a diagonal covariance matrix	KD
Spherical	Each component has a single variance	K
Tied	All components share the same general covariance matrix	$\frac{D(D+1)}{2}$

Table 1: Covariance Settings for GMM

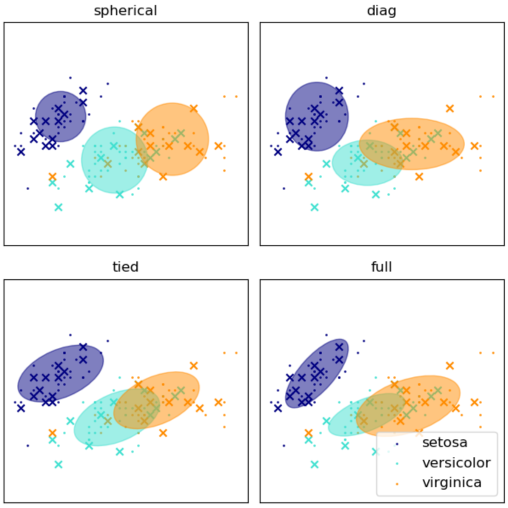


Figure 5: Different covariance settings for the 2D Iris dataset

a long time to run - however, we can indicate different parameter settings for the covariance matrix, based on what we know (or assume) about the data. This helps keeps the number of parameters that we are estimating under control and reduces the computation time.

2.2.4 Hidden Markov Models

In a first-order Markov chain describing a sequence of possible events, the probability of each event depends only on the state attained in the previous event.

A hidden Markov model can be viewed as a Markov chain of latent variables, with each observation conditioned on the state of the corresponding latent variable. The states of the Markov chain are not observed directly, i.e. the Markov chain

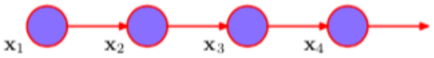


Figure 6: First-order Markov Chain

itself remains hidden [5].

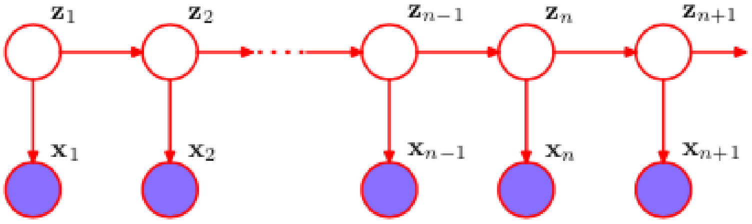


Figure 7: Hidden Markov Model

A single time slice of HMM corresponds to a mixture distribution, with component densities given by $p(x|z)$. Therefore HMMs can also be interpreted as an extension of a mixture model in which the choice of mixture component for each observation is not selected independently but depends on the choice of component for the previous observation [5].

HMMs are widely used in speech recognition, natural language modelling, on-line handwriting recognition and for the analysis of biological sequences such as proteins and DNA [5]. Thus, we can use a Hidden Markov Model (HMM) to model temporal dependencies in the data.

An HMM contains 3 essential components - observed sequence of data x_i, hidden states z_i, transition probabilities $p(z_i|z_{i-1})$ and emission probabilities $p(x_i|z_i)$.

The hidden states in an HMM are analogous to latent variables in GMMs, responsible for generating the given observations. The probability distribution of a hidden state z_n depends upon the state of the previous hidden state z_{n-1}. This is encoded by a transition probability matrix. For an HMM with M hidden states, the transition probability matrix has $M(M-1)$ independent parameters.

The emission probabilities are conditional distributions of the observed variables are $p(x_n|z_n, \phi)$, where ϕ is a set of parameters governing the distribution. The ϕ may be given by Gaussians if the elements of x are continuous variables, or by conditional probability tables if x is discrete [5, 13].

There are 3 basic problems tackled by hidden Markov models [26, 13] -

1. Given an observation sequence $\mathbf{x}$ and a model λ, compute the probability $p(x|\lambda)$. This is achieved using the Forward-Backward algorithm.

2. Given an observation sequence $\mathbf{x}$, compute the most likely observed state sequence. This is done using the Viterbi algorithm.

3. Adjust the model parameters of λ to maximise $p(x|\lambda)$ using the Baum-Welch algorithm.

2.2.4.1 HMM-based Clustering

We can apply the EM algorithm defined in Algorithm 1 to do clustering using HMMs.

The EM procedure for HMM based clustering is fairly complex, as we need to use a sub-EM procedure (Baum-Welch algorithm) in the M step, K times.

Starting from an initial collection of HMMs, EM iteratively finds cluster assignments that maximise the joint likelihood of the clustering.

Initialisation: Given sequences O^i, where $i \in [1{,}n]$, we randomly generate K HMMs $\lambda_1, \lambda_2, ..., \lambda_k$ and a partitioning $C = C_1, C_2, ...C_k$, where the assignment to clusters is done to maximise the following objective function:

$$f(C) = \prod_{k=1}^{K} \prod_{i \in C_k} L(O^i|\lambda_k) \tag{8}$$

where $L(O^i|\lambda_k)$ defines the likelihood function, i.e. the probability density for generating sequence O^i by model λ_k.

Assuming that we have a collection of K HMMs, our algorithm is as follows:

1. **Iteration** ($t \in \{1{,}2{,}...\}$):
   (a) **E Step:** Generate a new partitioning of the sequences by assigning each sequence O^i to the model k for which the likelihood $L(O^i|\lambda_k)$ is maximal.
   (b) **M Step:** Re-estimate parameters using the Baum-Welch algorithm [13]: ${\lambda_1}^t, {\lambda_2}^t...{\lambda_k}^t$ using their start parameters from the previous iteration and the sequences assigned to each model.

2. **Stop**, if one of the following takes place -
   (a) If the improvement of the objective function is below a given threshold
   (b) There are no more re-assignments
   (c) The number of iterations is reached

2.2.4.2 Limitations

The biggest limitation with HMM-based clustering is convergence to a local optima, so the algorithm needs to be run multiple times with different random initialisations to find the best fit. However, like GMM training, HMM based clustering takes significant computation time - especially due to the Baum-Welch EM procedure in the M step of the clustering.

2.2.5 Bayesian Networks

Bayesian networks are probabilistic graphical models that represent a set of variables and their conditional dependencies via directed acyclic graphs.

In [26], we studied the Chow-Liu algorithm to learn tree-structured graphical models. In [21], the authors combined Chow-Liu Trees and Expectation Maximisation to derive a model-based clustering method, called the CL Multinet Classifier.

If there are K classes, then K CL trees are learned. Each tree distribution Pr_k approximates the joint probability distribution of the attributes, given a specific class. The root attribute has no parents, while the other attributes can have at most ONE parent, i.e. if the attributes are expressed as r.vs $\{X_1, X_2, ..., X_n\}$, then $\Pi_{X_i} = \{X_{j \neq i}\}$ has only one attribute, and $\Pi_{X_{root}} = \{\emptyset\}$.

An inherent advantage of applying Bayesian networks case is that we can infer the structure behind each cluster too. For each cluster, we can see which timepoints affect the other.

Using Bayes Theorem, the posterior probability of a Chow Liu Tree can be expressed as

$$Pr(C = k|X_1, X_2, ..., X_n) = \frac{Pr(C = k)Pr(X_1, X_2, ..., X_n|C = k)}{\sum_{k'} Pr(C = k')Pr(X_1, X_2, ..., X_n|C = k')} \tag{9}$$

We observe that the denominator is constant with respect to the class, so we express it as $\frac{1}{\beta}$.

Then, the previous formula can be re-written as

$$Pr(C = k|X_1, X_2, ..., X_n) = \beta * Pr(C = k) \prod_{i=1}^{n} Pr_k(X_i|\Pi_{X_i}) \tag{10}$$

The edge weights of a Chow Liu tree are encoded by the *mutual information*, which is derived from the empirical probability distributions observed from the data.

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} Pr(x, y) \log \frac{Pr(x, y)}{Pr(X)Pr(y)} \tag{11}$$

A maximum weighted spanning tree (either Prim's/Kruskal's) algorithm is then used to find the Chow-Liu tree for each k^{th} class.

Unsupervised training of the CL Trees (henceforth referred to as CL Multinets) is carried out assuming that the data have been generated from a mixture of K Bayesian networks. This mixture can be described as

$$Pr(X) = \sum_{k=1}^{K} \alpha_k f_k(X) \tag{12}$$

where

$$\sum_{k=1}^{K} \alpha_k = 1, \quad \alpha_k \geq 0 \tag{13}$$

The α_k are the mixing coefficients and the f_k are the joint probability distributions of the CL Multinets. Given N observations, K clusters (CL Multinets), and initial partitions P^0=$\{\mathrm{P}^0_0$,P^0_1,...,$\mathrm{P}^0_K\}$, the EM procedure for the m^{th} iteration is detailed in algorithm 4 -

2.2.5.1 Limitations

Although Bayesian Networks are directed acyclic graphs and thus use fewer parameters than HMMs, unlike HMMs, we need to discretise the data in order to use the empirical frequencies to estimate joint probability distributions. Discretisation also results in loss of information, as the resultant dataset is dependent upon the number of bins chosen. Moreover, just like HMM-based clustering, there exist only general information theoretic criteria to determine the number of clusters K, which is not always be applicable [21].

2.2.6 Hierarchical Clustering

All the previous models that we covered can be classified under *partitional clustering*. The other category, hierarchical clustering, seeks to build a hierarchy of clusters using a given distance metric for datum and a similarity criteria for the clusters obtained. However, unlike partitional clustering, there exist no definitive statistical techniques determine the optimal number of clusters - instead, we need to examine the resulting hierarchical structure (dendogram) and visually determine the optimal number of clusters [18].

There are two approaches to hierarchical clustering -

- Agglomerative: Each datum initially represents a cluster of its own. Clusters are then successively merged until the desired cluster structure is obtained.

Algorithm 4 Expectation Maximisation for Bayesian Networks

E-Step: For $r = 1, 2, ..., N$ and $k = 1, 2, ..., K$ compute the posterior probabilities for x^r belonging to P_k.

$$t_k^m(x') = \frac{\alpha_k^m \prod_{i=1}^n Pr_k^m(x_i^r|\Pi_{x_i^r})}{\sum_{k'=1}^K \alpha_{k'}^m \prod_{i=1}^n Pr_{k'}^m(x_i^r|\Pi_{x_i^r})} \tag{14}$$

C-Step: Update the partition P^m=$\{\mathrm{P}_0^m$,P_1^m,...,$\mathrm{P}_K^m\}$ by assigning each x^r to the cluster that provides the max posterior probability.

M-Step: For $k = 1, ..., K$, maximise the Classification Maximum Likelihood (CML) criteria and re-estimate the parameters θ^m using the new partitions P_k^m.

$$CML = \sum_{k=1}^{K} \sum_{x^r \in P_k} \log \prod_{i=1}^{n} Pr_k(x_k^r|\Pi_{x_i^r}) + \sum_{k=1}^{K} n_k \log \alpha_k \tag{15}$$

where n_k are the number of samples belonging to the k^{th} cluster, and α_k are the cluster weights.

The parameters for the second term are re-estimated (maximised) using the following formula:

$$\alpha_k^{m+1} = \frac{n_k}{N}, \quad k = 1, ..., K \tag{16}$$

The first term (after some manipulation) contains the mutual information computed by the observed data (empirical distributions). Using the maximum weighted spanning tree algorithm, we maximise (15), resulting in a new tree distribution.

Terminate: If any of the following criteria are reached:

1. Maximum number of iterations reached.
2. CML converges.
3. Parameters converge.

- Divisive: All objects initially belong to one cluster. This main cluster is then divided into sub-clusters, which are successively sub-divided into their own sub-clusters, and this process continues until the desired cluster structure is obtained.

We set the distance metric (for within cluster distances) as Euclidean. The linkage criterion in hierarchical clustering specifies the difference between any two sets (clusters) obtained. It determines which clusters to merge by calculating the differences between two clusters as a function of the pairwise distances between observations. The different types of linkages are-

- Single: the distance between two clusters is equal to the shortest distance from any member of one cluster to any member of the other cluster.

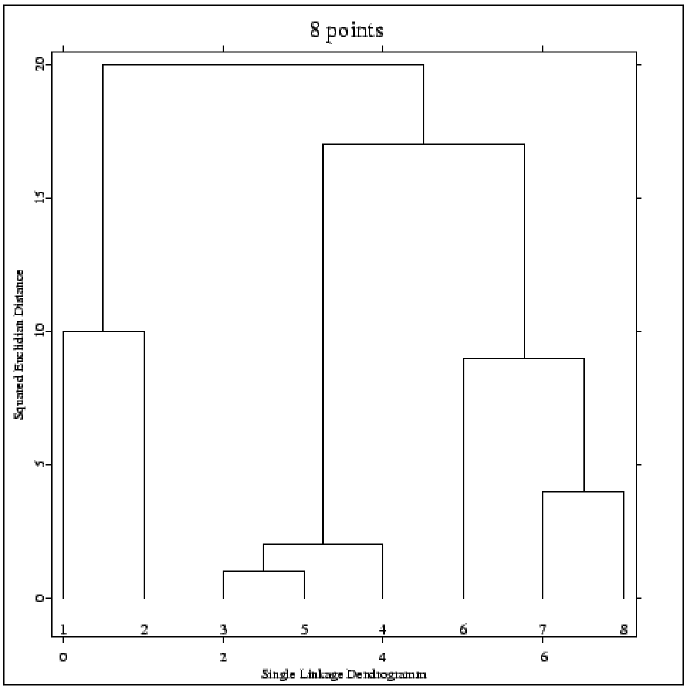


Figure 8: Hierarchical Clustering Dendogram for Single-Linkage

- Average: the distance between two clusters is equal to the average distance from any member of one cluster to any member of the other cluster.

- Complete: the distance between two clusters is equal to the longest distance from any member of one cluster to any member of the other cluster.

- Ward: the distance is defined as the error function of the unified cluster minus the error functions of the individual clusters, where the error function is the average distance of each data point in a cluster to the cluster centroid.

For this report, we use agglomerative hierarchical clustering *not* in model building, but in Section 4.1 for feature engineering , so as to obtain a rough estimate of the number clusters we may expect from the data.

2.3 Evaluation Metrics for Model Selection

Given data, model selection is the task of selecting the best statistical model from a set of candidate models.

In our clustering analysis, the main conundrum is determining the number of clusters.

The estimation of the true number of classes has been recognised as “one of the most difficult problems in cluster analysis" [6]. The correct choice of K is often ambiguous - sometimes, choosing the best fitting model is not the wisest choice.

Consider increasing K without any penalty - this will keep reducing the error but the model is less than ideal (e.g. In K-Means, if we have n clusters for n points, each point is its own center, reducing the cost function to 0).

Moreover, given multiple candidate models having similar accuracy, the simplest model is most likely to be the best choice (Occam's razor).

Thus, the optimal choice of K balances accuracy and number of parameters required for the model. This is tackled in 2.3.1 and 2.3.2.

A secondary conundrum is comparing two different clustering models, given that we obtain "ground truth" values for some of the clusters. This is handled in 2.3.3.

2.3.1 Information Criteria

When a statistical model is used to represent the process that generated the data, the representation will almost never be exact. Some information is thus lost by the model. Bayesian Information Criteria (BIC) and Akaike Information Criteria (AIC) [7] deal with estimating this lost information, and calculate the trade-off between the *goodness of fit* of the model and the *complexity* of the model.

2.3.1.1 BIC and AIC

BIC and AIC penalise the total number of parameters. Both are functions of the maximised log-likelihood of the model and the estimated number of parameters - however, BIC penalises model complexity more heavily. We define the AIC and BIC as follows-

$$AIC = \log L(\theta) - P$$

$$BIC = \log L(\theta) - \frac{P}{2}\log(N)$$

where $L(\theta)$ is the maximised log-likelihood of the model θ, P is the number of free parameters and N is the number of data points. For a range of K values, we calculate AIC/BIC, and choose the K for which this criterion is *maximum*. Thus, higher AIC or BIC values indicate better fitting models.

2.3.2 Elbow Method

The elbow method [17] is a popular method to determine the true number of clusters by plotting the cost function for a given clustering algorithm against different K, and observing the point where the graph makes a sharp *elbow* shape, as seen in

Figure 9. More formally, this method looks at the percentage of variance explained as a function of the number of clusters. In essence, the elbow method observes the point of sharpest decrease in the cost function, i.e. the most improvement to the cost function, and estimates that to be the ideal number of clusters. However, the limitation of this method is that the elbow point cannot always be unambiguously identified - sometimes there is no elbow, or sometimes there are several elbows.

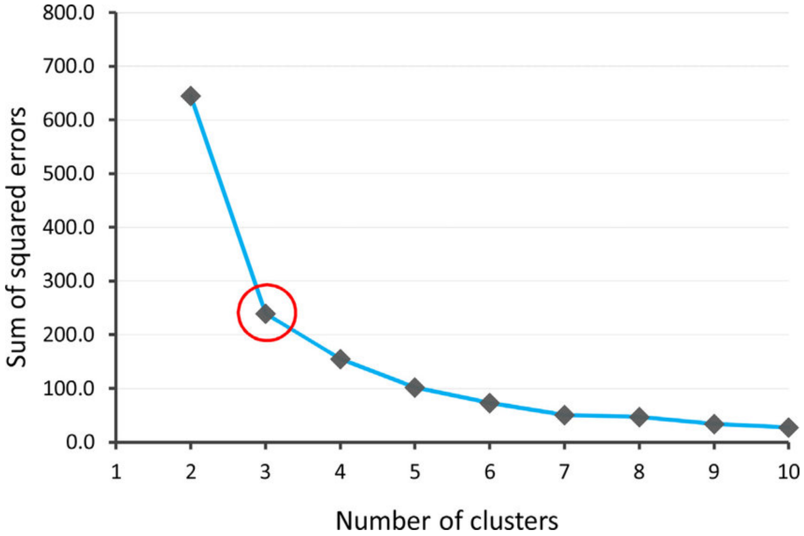


Figure 9: Elbow method for k-means clustering

2.3.3 Misclassification Error (ME) Distance

ME distance is used to calculate the similarity between two different clusterings of the same data. Given the "ground truth" (i.e. true classes of each point), we can compare the clustering obtained by our technique with the ground truth to see how our model compares. ME distance thus represents the well known cost of classification, minimised over all permutations of the labels [19].

It is calculated as follows -

Consider two clusterings $\mathbb{C} = \{C_1, C_2, ..., C_K\}$ and $\mathbb{C}' = \{C'_1, C'_2`..., C'_{K'}\}$. We define the confusion matrix of $\mathbb{C}$ and $\mathbb{C}'$ as a KxK' matrix $M = [m_{kk'}]$, with $m_{kk'} = |\mathbb{C}\neg \cap \mathbb{C}'_{\mathbb{K}'}|$. We consider the case where both models cluster N points, and $K = K'$.

Then the ME distance is defined as

$$d(C, C') = 1 - \frac{1}{N} \max_{\pi \in \Pi_k} \sum_{k=1}^{K} m_{k,\pi(k)}$$

where Π_k contains all permutations of the label K.

We want the ME Distance between our models and the ground truth to be close to 0.

2.3.4 Gene Ontology

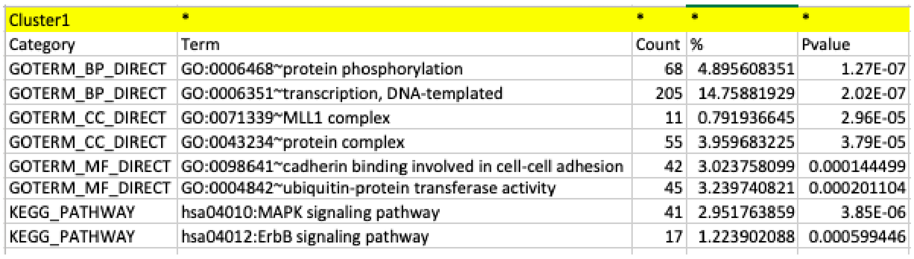
Gene Ontology (GO) is a framework for the model of biology, unifying the representation of gene and gene product attributes across all species. We define concepts and classes used to describe gene functions, and observe the relationships between them.

The ontology covers 4 categories-

1. Biological Process: operations or sets of molecular events with a defined beginning and end, pertinent to the functioning of integrated living units: cells, tissues, organs, and organisms.
2. Cellular Component: the parts of a cell or its extracellular environment.
3. Molecular Function: the elemental activities of a gene product at the molecular level.
4. KEGG Pathway: pathways representing molecular interaction, reaction and relation networks for various biological processes.

We use **DAVID** [15] to perform gene ontology analysis on the clusters we generate.

Each cluster obtained from our model(s) contains genes that share similar biological properties. We perform GO analysis on all these different clusters, and attempt to identify terms that are related to infection - like viral process, translation, DNA repair, etc. If we do find terms related to viral process, we use guilt-by-association to find other significant terms.



Cluster1	*	*	*	*
Category	Term	Count	%	Pvalue
GOTERM_BP_DIRECT	GO:0006468~protein phosphorylation	68	4.895608351	1.27E-07
GOTERM_BP_DIRECT	GO:0006351~transcription, DNA-templated	205	14.75881929	2.02E-07
GOTERM_CC_DIRECT	GO:0071339~MLL1 complex	11	0.791936645	2.96E-05
GOTERM_CC_DIRECT	GO:0043234~protein complex	55	3.959683225	3.79E-05
GOTERM_MF_DIRECT	GO:0098641~cadherin binding involved in cell-cell adhesion	42	3.023758099	0.000144499
GOTERM_MF_DIRECT	GO:0004842~ubiquitin-protein transferase activity	45	3.239740821	0.000201104
KEGG_PATHWAY	hsa04010:MAPK signaling pathway	41	2.951763859	3.85E-06
KEGG_PATHWAY	hsa04012:ErbB signaling pathway	17	1.223902088	0.000599446

Figure 10: Sample GO for a cluster

p-value is the probability of seeing at least x number of genes out of the total n genes in the list annotated to a particular GO term, given the proportion of genes in the whole genome that are annotated to that GO term. In essence, the GO terms shared by the genes in the user's list are compared to the background distribution of the annotation. Thus, the smaller the p-value is, the more significant the particular GO term associated with the group of genes is (i.e. the less likely the observed annotation of the particular GO term to a group of genes occurs by chance) [15].

3 Exploratory Analysis

3.1 Data Description

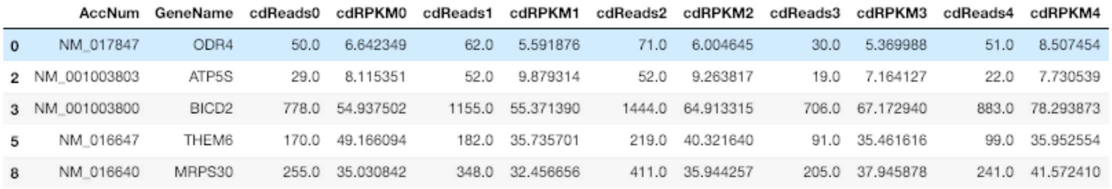
We have 3 datasets consisting of unlabelled human gene expression values, obtained after infection with the EV-A71 virus.

- **RNASeq**: 12,340 samples of RNASeq data at 5 timepoints
- **RPF**: 11,746 samples of RPF data at 5 timepoints.
- **TE**: 11,633 samples of TE data at 5 timepoints.

The attributes are gene expression values x hours post infection (h.p.i), where $x \in \{0, 2, 4, 6, 8\}$.

As described in Section 1.3, RNASeq represents the total amount of genes within the cell, while RPF represents the amount of genes being translated. TE measures the rate at which genes get translated.

3.1.1 Data Preprocessing



	AccNum	GeneName	cdReads0	cdRPKM0	cdReads1	cdRPKM1	cdReads2	cdRPKM2	cdReads3	cdRPKM3	cdReads4	cdRPKM4
0	NM_017847	ODR4	50.0	6.642349	62.0	5.591876	71.0	6.004645	30.0	5.369988	51.0	8.507454
2	NM_001003803	ATP5S	29.0	8.115351	52.0	9.879314	52.0	9.263817	19.0	7.164127	22.0	7.730539
3	NM_001003800	BICD2	778.0	54.937502	1155.0	55.371390	1444.0	64.913315	706.0	67.172940	883.0	78.293873
5	NM_016647	THEM6	170.0	49.166094	182.0	35.735701	219.0	40.321640	91.0	35.461616	99.0	35.952554
8	NM_016640	MRPS30	255.0	35.030842	348.0	32.456656	411.0	35.944257	205.0	37.945878	241.0	41.572410

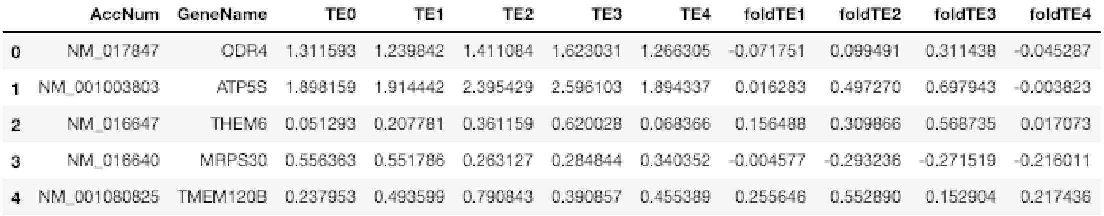
Figure 11: RNASeq Data

cdRPKM values are raw gene expression values; **cdReads** are the number of times a gene was read (as per the biological technique used), while the **AccNum** and **GeneName** represent how to ID the gene.

We first filter the dataset for cdReads ≥ 10 to remove noise; this removes genes with very small reads, likely generated due to experimental errors.

Then, we transform the cdRPKM values to $\log_2$, which is a standard practice in bioinformatics.

We can calculate TE data in Figure 12 by dividing the raw RPF value by the raw RNASeq value, and transforming it to log_2. Alternatively, we can subtract the log_2 RNASeq value from the log_2 RPF value to also obtain TE.

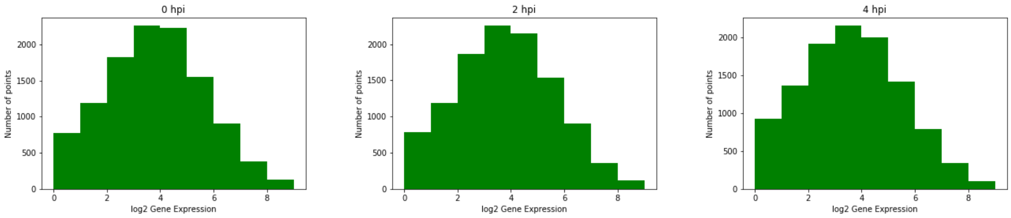


	AccNum	GeneName	TE0	TE1	TE2	TE3	TE4	foldTE1	foldTE2	foldTE3	foldTE4
0	NM_017847	ODR4	1.311593	1.239842	1.411084	1.623031	1.266305	-0.071751	0.099491	0.311438	-0.045287
1	NM_001003803	ATP5S	1.898159	1.914442	2.395429	2.596103	1.894337	0.016283	0.497270	0.697943	-0.003823
2	NM_016647	THEM6	0.051293	0.207781	0.361159	0.620028	0.068366	0.156488	0.309866	0.568735	0.017073
3	NM_016640	MRPS30	0.556363	0.551786	0.263127	0.284844	0.340352	-0.004577	-0.293236	-0.271519	-0.216011
4	NM_001080825	TMEM120B	0.237953	0.493599	0.790843	0.390857	0.455389	0.255646	0.552890	0.152904	0.217436

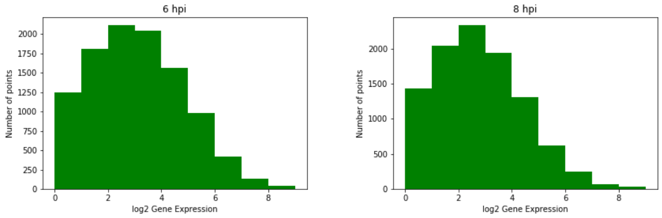
Figure 12: TE Data

3.2 Initial Analysis

For the RNASeq and RPF datasets, we plot individual timepoint log_2 values in Figures 13 and 14.



(a) 0 h.p.i (b) 2 h.p.i (c) 4 h.p.i



(d) 6 h.p.i (e) 8 h.p.i

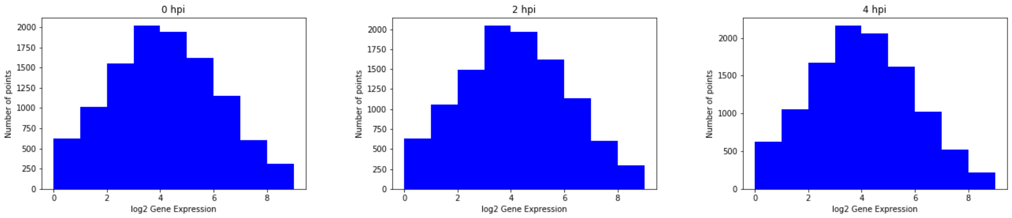
Figure 13: Distribution of log_2 values of RNASeq data

From the plots, we infer that all timepoints follow a Gaussian distribution. we posit that Gaussian Mixture Models and Hidden Markov Models with continuous emissions will be useful in clustering our time-series gene expression values.

For both datasets, we observe that the median keeps shifting to the left for later timepoints. This is due to the infection spreading- with the passage of time, more viral translation leads to an overall decrease in gene expression.

To standardise the medians for all timepoints, we again follow a standard practice in bioinformatics - we median-normalise each timepoint (subtract the median from every value) so that each timepoint has the same median (0).

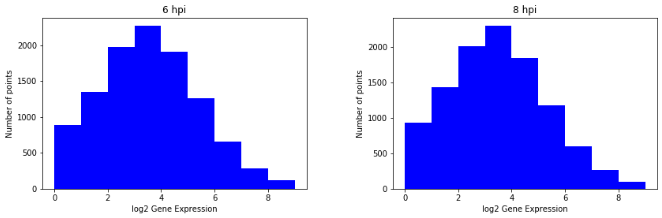
After median-normalisation, the distribution of the maximum and minimum values are generally similar for all time points.



(a) 0 h.p.i

(b) 2 h.p.i

(c) 4 h.p.i



(d) 6 h.p.i

(e) 8 h.p.i

Figure 14: Distribution of log_2 values of RPF data

4 Results

We apply the models from Section 2.1 to the datasets described in Section 3, and attempt to identify those groups of genes that are critical for viral translation.

There are two tasks we need to perform - firstly, identify an optimal K, and secondly, decide the benchmark algorithm (the most optimal model) that we will compare all our other techniques with.

We evaluate the performance of the models through the methods described in Section 2.3 and interpret the significance of the results obtained.

We perform our analysis in a systematic way -

1. A priori, we know that the datasets will not contain more than 20-30 clusters. We restrict our analysis for $2 \leq K \leq 30$.
2. We apply model-based clustering techniques on the complete datasets, and use the evaluation criteria to determine the optimal K.
3. If the results are inconclusive, we implement feature engineering in Section 4.1 and re-run the model-based clustering techniques on this modified data.
4. Attempt to determine K through BIC/AIC.
5. If no optimal value is found, do clustering for specific K (as determined by biologists, as in Figure 15).
6. Interpret results through GO analysis and/or other analyses.

4.1 Feature Engineering

Since this is time-series data, we also quantify the change between successive timepoints.

From the RNASeq and RPF datasets, we find the amount of change between timepoints and add these as features to the original dataset. Formally known as $\log_2$ fold change in bioinformatics, it is calculated as follows -

- Calculate the $\log_2$ of the gene expression value, **cdRPKM**, for each timepoint. Let this be x_i.
- Subtract x_0 (the value at timepoint 0) from each successive timepoint x_i.
- Add this as a new attribute to the RNASeq and RPF data.

For TE data, since the values are already log ratios, we merely subtract each timepoint's value from the first timepoint.

Even with the new features, we might encounter problems due to *noise*. Fortunately, we have a biological approach of dealing with this.

When a cell is infected with a virus, the RNASeq does not change much (as it measures the total content of all genes) while those genes that help in virus production will get translated more, i.e. the RPF of these genes will increase. Since TE measures translational efficiency ($TE = \frac{RPF}{RNASeq}$), the TE of such genes will also keep increasing across timepoints.

We filter out those genes that do not change much at the RNASeq level. We retain genes which have **foldChange** values for each timepoint between -0.5 and 0.5. More formally, if x_i is the foldChange for $1 \leq i \leq 5$, then $-0.5 \leq x_i \leq 0.5$ for each x_i (later on, we reduce this threshold further to [-0.3,0.3] to check whether the GO terms we obtained are indeed important or not).

After this filtering, we are left with **5576 genes** that are highly translated.

As we are interested in genes that are highly translated between timepoints, we apply hierarchical clustering only on the **foldChange** features in the TE dataset. We plot heatmaps for different linkages, and check the general trends to find the number of clusters. In a heatmap, the values of each gene are plotted according to the colour key (bottom left).

The values in the heatmaps are sorted accorded to how they have been clustered.

In our case, we find that *complete* linkage performs the best. We can identify 4-6 distinct clusters from the heatmap, and cut the dendogram accordingly (represented by the green line). The result is shown in Figure 15.

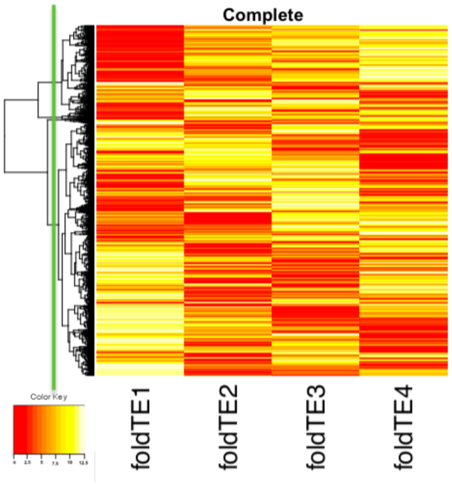


Figure 15: Heatmap with Complete Linkage Hierarchical Clustering for Filtered Data

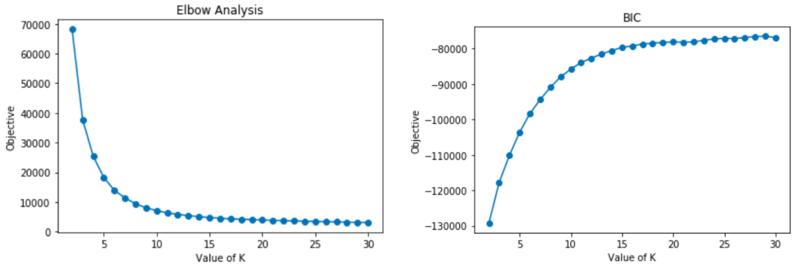
After applying hierarchical clustering and doing heatmap analysis on the filtered data, we infer that K lies between 4 and 6. We use these values of K for clustering for Step 5 of Procedure 4.

4.2 Model-Based Clustering Results

4.2.1 K-Means

We first apply K-Means on all the datasets to determine the optimal K.

We initialise the algorithm using K-Means++ for optimal clusterings, and apply the Elbow method and BIC to evaluate our results. Results for RNASeq are given in Figure 16.



(a) Elbow Method (b) BIC Curve

Figure 16: Cluster Analysis for RNASeq

The full set of graphs for all datasets are given in Appendix A, Section A.1.1. The results are summarised in Table 2. "-" means inconclusive.

Data	**Elbow Method**	**BIC**
RNASeq (5 points)	-	15*
RPF (5 points)	-	-
TE (5 points)	5	17*

Table 2: Optimal K for different datasets, K-Means

*For BIC, if there is no clear peak, we take the first local maxima as K

From the plots, we observe that determining the optimal K is difficult.

There is no clear 'elbow' point, and the BIC keeps increasing with K; thus K-Means is not penalising enough. (Using AIC is futile, as the penalisation would be even lower in this case). The elbow plots and GO analysis do not mutually agree on a K.

Despite the mixed results from the information criteria metrics, we went ahead and performed GO analysis as per the K determined in Table 2. However, the GO results were also found to be inconclusive - there were multiple clusters associated with the *'viral process'* term, and the p-value was quite high (i.e. the terms were not enriched).

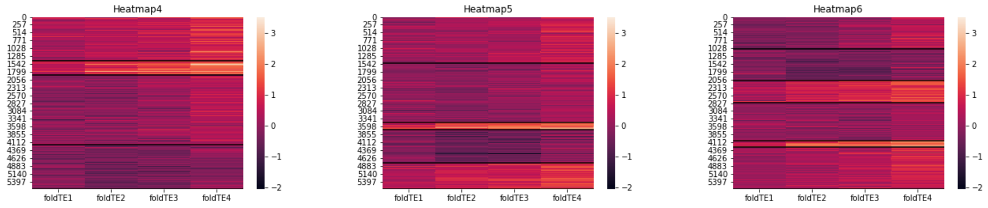
Thus, there is no reliable estimate for K. This is most likely due to 2 reasons -

- K-means cannot capture temporal dependencies in data.
- Data is too noisy.

Both can be solved through feature engineering - we add the changes between the timepoints as features.

We filter the data as described in Section 3, and using the heatmap results, perform K-Means clustering for 4,5 and 6 clusters on the **foldChange** TE data.

We plot heatmaps to observe the general trends of clusters from this heuristic method, and identify those clusters which have an increasing trend (the clusters are plotted top to bottom in the heatmap and demarcated by black lines).



(a) 4 clusters (b) 5 clusters (c) 6 clusters

Figure 17: Heatmaps for K-Means clustering on TE data (top to bottom)

Cluster 2, 3 and 5 in Figures 17a, 17b and 17c have an increasing trend, so we focus on the GO analyses of those clusters (covered in Section 4.4).

This heuristic method gives us GO results (specifically, terms related to GTPases from Section 1.2.4.1) that are relevant to our hypothesis. This is covered in greater detail in Section 4.4.

4.2.2 Gaussian Mixture Models

As seen in the K-Means approach, it is futile to do GO analysis on complete datasets (especially for the noisy RNASeq data, as we will most likely observe no significant trends by merely tracking overall changes in gene expression).

To test this, we still apply GMMs on the complete TE data (which are most likely to show certain trends due to genes being translated). As expected, we have no definitive results. The GO is unclear, and the BIC keeps increasing with K for all covariance settings, offering no clear peak. The plots are in Appendix A, Section A.2.

We decide to apply GMMs on the feature-engineered data from Section 4.1. We use the full covariance matrix and do BIC analysis for $2 \leq K \leq 20$.

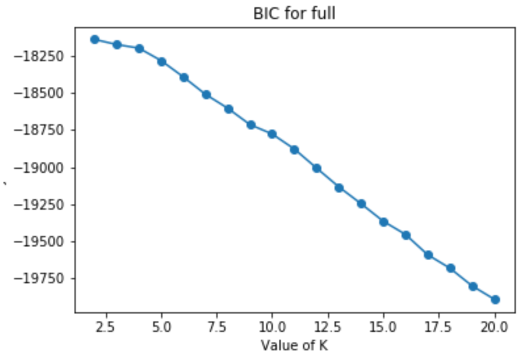


Figure 18: BIC for full covariance matrix, TE data

For filtered TE data, we conclude that the ideal number of clusters are $K = 2$ from Figure 18. We plot the heatmap and general foldChange graph (in Section A.2.2) for this clustering to check the general trend as per the heatmap (clusters are demarcated by the green line).

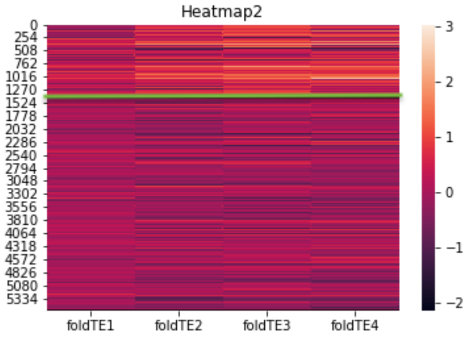


Figure 19: Heatmap for 2 clusters, GMM clustering on TE data

We see in Figure 19 that Cluster 1 has an increasing trend. The GO analysis of this cluster is similar to that obtained through K-Means - we again observe terms related to GTPases.

We still perform GMM-based clustering for $K = 4, 5, 6$ for evaluating GMMs performance against the benchmark algorithm in Section 4.3.

4.2.3 HMM-based Clustering

We speculated that HMM-based clustering might perform better than other algorithms on the complete and unfiltered datasets, as HMMs inherently take the time dependencies into account.

We assume 3 hidden states for simplicity, and apply HMM-based clustering on all the datasets. The plots are in Section A.4.1, while the results are summarised in Table 3.

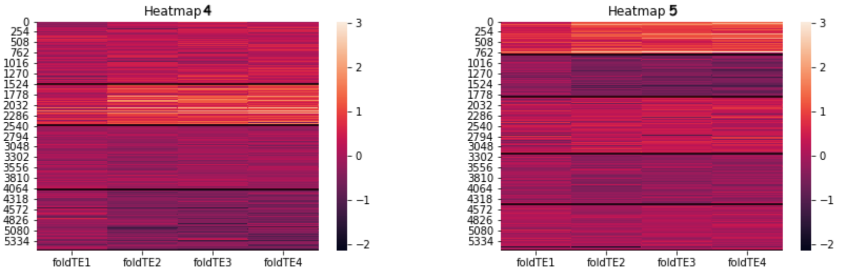
Data	**BIC**
RNASeq (5 points)	13*
RPF (5 points)	8*
TE (5 points)	16*

Table 3: Optimal K for different datasets, HMMs

*For BIC, if there is no clear peak, we take the first local maxima as K

Again, we observe no clear peaks. The GO results for the local peaks are inconclusive, although some of the terminologies in the GO for RPF and TE do include GTPases and other related terms. However, there were no other clear trends.

We can deduce from BIC's inability to determine an optimal K that we may be estimating too many parameters and unnecessarily complicating the models (Occam's Razor).



(a) Heatmap for 4 clusters (b) Heatmap for 5 clusters

Figure 20: HMM Heatmap results on feature-engineered data

We proceed to the next steps of the analysis. We apply HMM-based clustering on the feature-engineered data for $K = 4, 5$ (we skip 6 as HMMs take the longest time to train) and generate heatmaps in Figure 20. We then check the heatmaps for up-regulated clusters, and perform GO analysis on the Clusters 2 and 1 respectively from Figure 20.

The trend graphs of each cluster are given in Section A.4.2.

The GO results from HMMs are comparable to those from GMMs and K-Means, and we again observe GTPases in the gene ontology of the up-regulated clusters.

4.2.4 Bayesian Networks

A limitation of Bayesian networks is that the current implementation is only applicable to discrete random variables (or discrete features).

Although there is some literature that deals with hybrid Bayesian Networks that model data containing both discrete and continuous attributes [8], data with purely continuous variables are difficult to model using Bayesian Networks.

Thus, as a first step, we *discretise* our gene expression data [21].

According to [21], there is no conclusive way of determining K for CL Multinets based clustering, so we skip AIC/BIC analysis here. Instead, we perform clustering using $K = 4, 5, 6$ to determine the structure of the underlying Bayesian network for the complete datasets.

The heatmap results for the TE dataset showed some increasing trends (covered in Section 29; the GO analysis for the complete TE dataset included GTPases in more abundance than HMM-based clustering.

Moreover, the underlying structure for each tree (determined by the clustering, for different K) unfailingly resulted in the same line graph (in Figure 21), with the timepoints in order (timepoint 1, followed by timepoint 2, and so on). This, along with the GO results, shows that Bayesian networks cluster even noisy data with a certain level of accuracy.

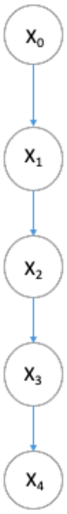
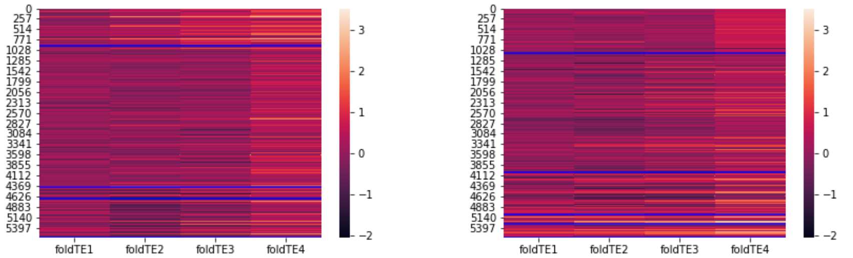


Figure 21: Tree Structure

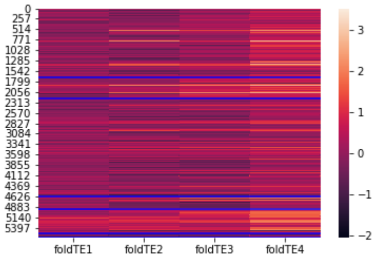
We then apply Bayesian networks to the feature engineered TE dataset. The expression values range from -2 to 4, and we choose 100 bins between -2 and 4, given by $[-2.0, -1.94, ..., 3.94, 4.0]$ and then digitise each observation according to the bins it belongs to.

The heatmaps for 6 clusters are given in Figure 22. The blue lines demarcate the clusters.



(a) Heatmap for 4 clusters

(b) Heatmap for 5 clusters



(c) Heatmap for 6 clusters

Figure 22: CL Multinets Heatmap results on feature-engineered data

Although the heatmaps are not as uniform and do not display concrete trends (as those from K-Means clustering) the GO results are reliable and contain more GTPase-related terms, along with some new terms related to viral translation (e.g. G2/M transition of mitotic cell [10]).

4.3 Performance Evaluation

We compare algorithms against a benchmark for the feature engineered TE dataset (with the foldChange features).

Bayesian networks consider the inherent time-dependency between attributes, resulting in a graph with the timepoints in correct sequence each time.

The GO results are also significant as they capture those genes that are upregulated yet also distinct from each other despite having similar expression values (unlike K-Means, which clusters according to Euclidean distance between absolute attribute values).

We pick Bayesian networks as our benchmark algorithm, and treat the results obtained from it as the *ground truth*.

	Accuracy for Clusters		
Algorithm	**4**	**5**	**6**
K-Means	56.52%	48.35%	41.61%
GMMs	58.40%	53.11%	50.31%
HMMs	68.72%	79.45%	-

Table 4: Performance evaluation for all algorithms*

We use the ME distance metric to compare the other techniques against Bayesian networks for $K = 4, 5, 6$. For ME Distance, the closer it is to 0, the better the clustering is. We calculate the accuracy as 100(1-d_{ME})%.

*compared to Bayesian networks

4.4 Biological Results

Existing literature on picornavirus (which the EV-A71 virus is a type of) suggests that the proteins in the virus interact with certain GTPases and disrupt normal cellular functions (like nuclear transport of other viral and nuclear proteins). [22].

Looking more closely at the collection of heatmaps obtained by all the algorithms, we identify all those clusters that are up-regulated and contain terms containing/related to GTPases in their GO analysis.

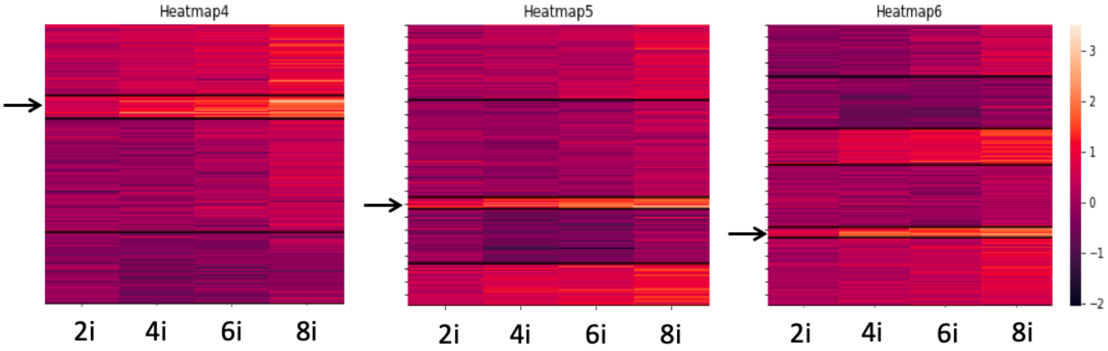


Figure 23: Heatmap from K-Means, RNASeq filter [-0.5, 0.5]

To be more rigorous about the terms we observe, we also reduce the filter to $[-0.3, 0.3]$ and check the GO analysis of the up-regulated clusters. For each cluster, we only consider enriched GO terms (those with low p-values, mentioned in Section 2.3.4).

Throughout our (repeated) experiments, we find that terms related to GTPases from *Biological Processes* (BP) and *Molecular Functions* (MF) keep re-appearing. The terms of interest are -

- Positive regulation of GTPase activity

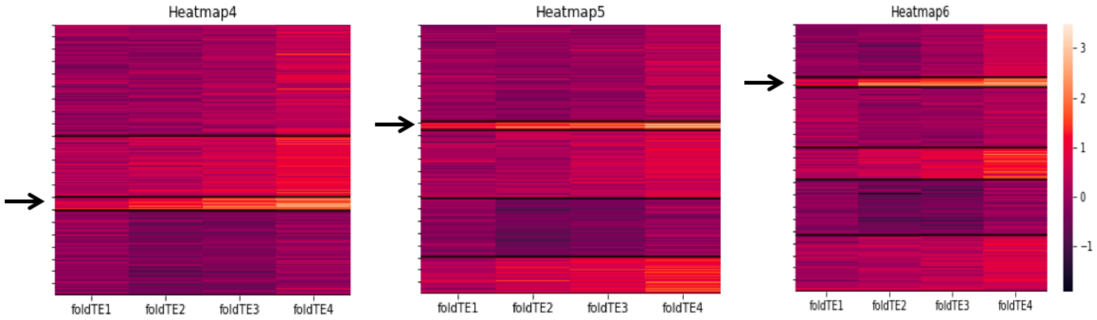


Figure 24: Heatmap from K-Means, RNASeq filter [-0.3, 0.3]

- Activation of GTPase activity
- Regulation of small GTPase mediated signal transduction
- Positive regulation of transcription, DNA templated
- Signal transduction

Through the principle of guilt-by-association mentioned in Section 4, we also identify other enriched terms that are of interest -

- Viral Process
- DNA repair
- Negative regulation of transcription from RNA pol II promoter
- Establishment of cell polarity
- Smoothened signaling pathway
- Positive regulation of dendrite development
- G2/M transition of mitotic cell

Some GO terms (viral process, regulation of small GTPase mediated signal transduction, DNA repair) are observed across multiple clusters in the HMMs and Bayesian Networks analysis. This suggests interactions between the clusters too.

From the GO analysis, we find plenty of evidence from our datasets to support our hypothesis that a certain class/member of GTPase is important for establishing this infection.

5 Conclusion

This thesis explored the application of unsupervised learning techniques on viral genomic data - more specifically, human cells infected with the EV-A71 virus (which causes HFMD). We attempted to determine host factors important to the propagation of this virus, and identified several key biological terms based on our clustering results.

5.1 Summary

As described in Section 1.3.1, we collected RNASeq and RPF gene expression data at different timepoints. In Figure 2, we saw that the space occupied by the virus increases with time. Initial analysis of the complete datasets (RNASeq, RPF, TE) were consistent with this trend. Next, we attempted to determine groups of genes crucial to viral translation. Initial clustering results contained some relevant GO terms (e.g. viral process, DNA repair) that appeared in multiple clusters; however, the trend graphs and heatmaps proved to be inconclusive, suggesting that our datasets were too *noisy*, containing many points insignificant to our analysis.

We then focused on those genes which were highly translated at the RPF level but not at the RNASeq level. Collating data from all our GO analyses, we identified the following terms - Positive regulation of GTPase activity, Activation of GTPase activity, Regulation of small GTPase mediated signal transduction, Positive regulation of transcription, DNA templated and Signal transduction.

These terms imply that some proteins in the class GTPase are definitely important to viral translation, possibly increasing the production of viral genes at key phases.

Following up on these results, we conduct more extensive literature reviews to verify the link between small GTpases and picornavirus infection.

5.1.1 Biological Experiments

We also design experiments to verify whether GTPases are relevant to the EV-A71 virus.

The first experiment tracks changes in GTPase activity throughout infection, focusing more on Rho GTPases. For GTPase protein-specific activity, there are also targeted studies to check for specificity of GAP/GEF, which are related to activation of GTPases.

The second experiment inhibits specific GTPases to check their effect on viral cells. 3 kinds of drugs targeting different GTPases are used -

- Specific Rho drugs e.g. Rhosin or C3 transferase against RhoA/B/C, but not cdc42/Rac1
- Specific cdc42 drugs e.g. Casin, ML141
- Specific Rho kinase inhibitors e.g. Rhodblock series

Thus, the appearance of terms related to viral process in the GO analysis for all methods suggests that our clustering was useful. The abundance of GTPase-related terms also support our hypothesis that they are indeed vital to the propagation of the virus inside the cell.

6 Future Work

6.1 Algorithms

6.1.1 Tuning

We can improve the performance of our current model-based clustering techniques by incorporating additional data or adjusting the hyperparameters for the prior distributions we use for initialisation. We can consider discretising the data for Hidden Markov Model based clustering [28]; conversely, we can consider continuous variables for Bayesian networks [14].

6.1.2 Different Algorithms

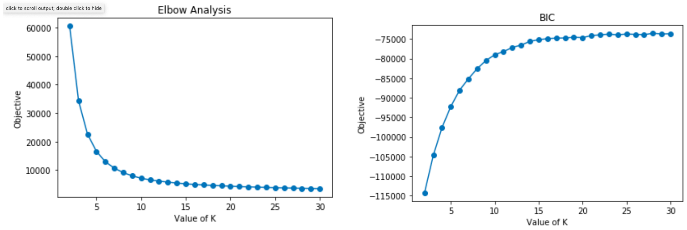
We can consider self-organizing maps (SOMs) for clustering gene expression data too [25]. SOMs impose partial structure on the clusters (in contrast to the rigid structure of hierarchical clustering, the strong prior hypotheses used in Bayesian clustering, and the non-structure of k-means clustering) and facilitate easy visualization and interpretation. SOMs have good computational properties and are easy to implement, reasonably fast, and scalable to large data sets too.

We can also consider soft clustering, or fuzzy clustering; where one gene may belong to more than one cluster. Existing fuzzy-clustering techniques can indeed be modified for short time-series gene expression data [20].

A Appendix

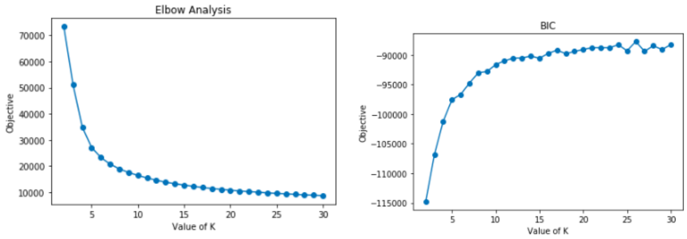
A.1 K-Means

A.1.1 BIC Plots



(a) Elbow Method (b) BIC Curve

Figure 25: Cluster Analysis for RPF

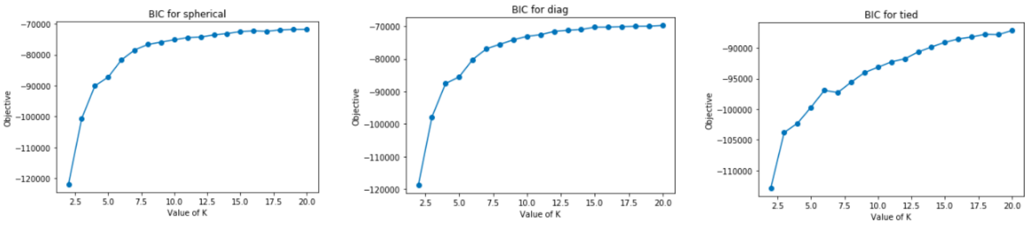


(a) Elbow Method (b) BIC Curve

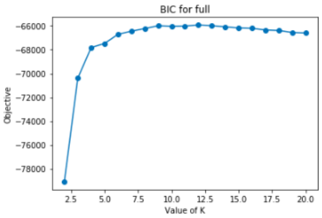
Figure 26: Cluster Analysis for TE

A.2 Gaussian Mixture Models

A.2.1 BIC Plots



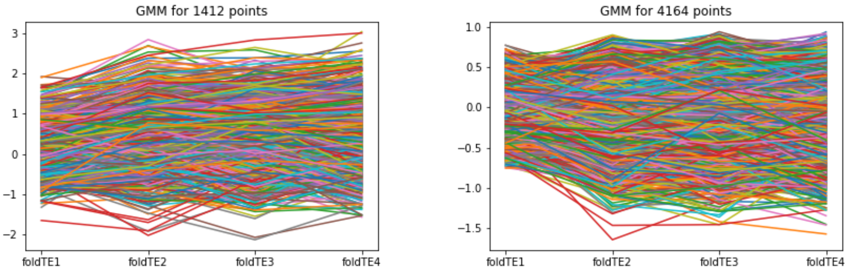
(a) Spherical (b) Diagonal (c) Tied



(d) Full

Figure 27: BIC for different covariance settings in GMMs, TE

A.2.2 Trend Plots



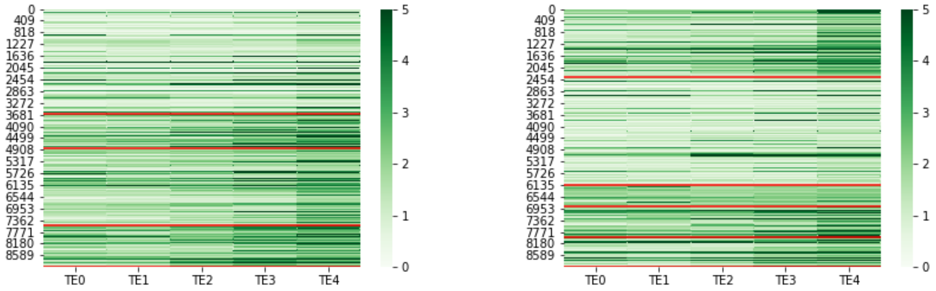
(a) Cluster 1 (b) Cluster 2

Figure 28: Trend plots for foldChange values, 2 clusters, GMM

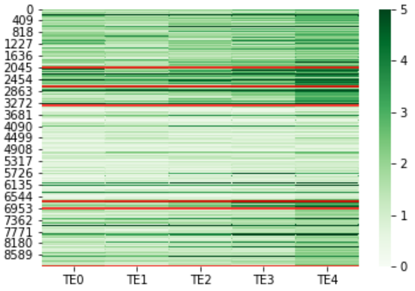
A.3 Bayesian Networks

A.3.1 Heatmaps for Complete Datasets

The clusters are demarcated by red lines.

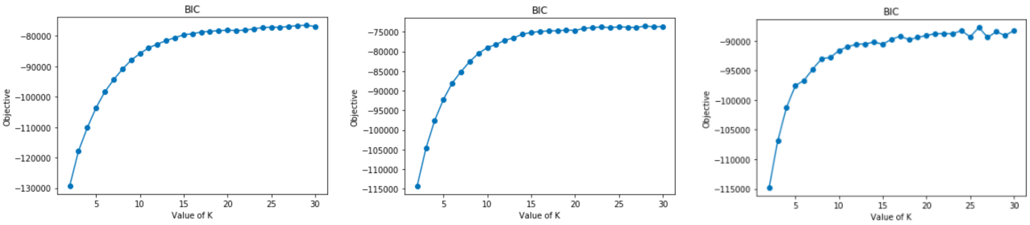


(a) 4 clusters (b) 5 clusters



(c) 6 clusters

Figure 29: Heatmaps for foldChange TE value



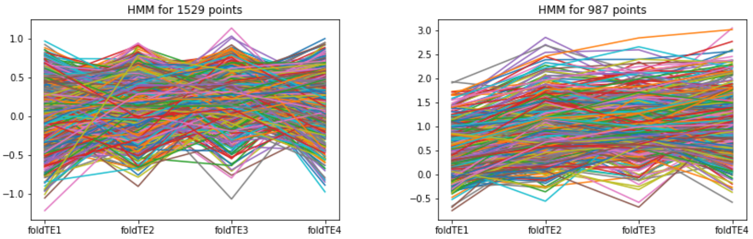
(a) RNASeq data (b) RPF Data (c) TE Data

Figure 30: BIC results for HMM-based clustering

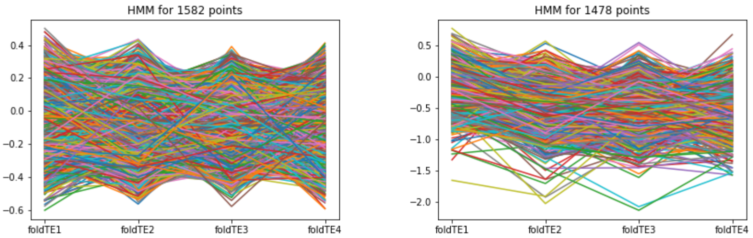
A.4 HMM-based Clustering

A.4.1 BIC Plots

A.4.2 Trend plots

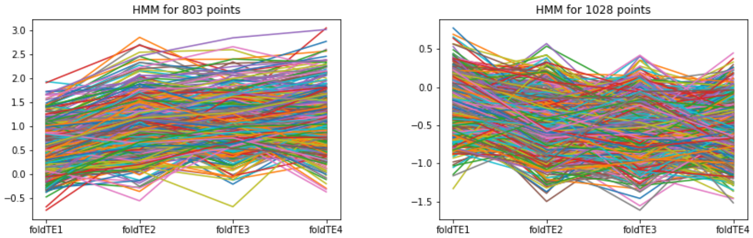


(a) Cluster 1 (b) Cluster 2



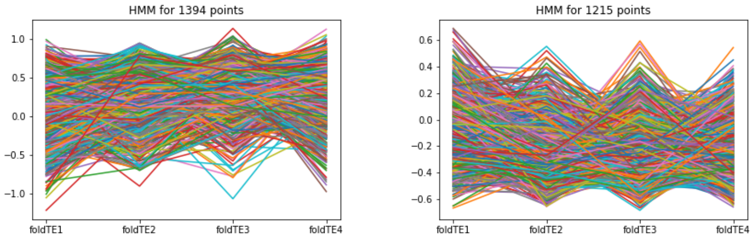
(c) Cluster 3 (d) Cluster 4

Figure 31: Trend plots for foldChange values, 4 clusters, HMM-based clustering



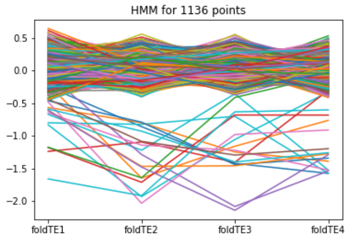
(a) Cluster 1

(b) Cluster 2



(c) Cluster 3

(d) Cluster 4



(e) Cluster 5

Figure 32: Trend plots for foldChange values, 5 clusters, HMM-based clustering

List of Figures

1 Central dogma of Molecular Biology 4
2 EV-A71 (chrE) infection over time* 5
3 EV-A71 life cycle 6
4 Mixture of 3 Gaussians in one dimension 12
5 Different covariance settings for the 2D Iris dataset 15
6 First-order Markov Chain 16
7 Hidden Markov Model 16
8 Hierarchical Clustering Dendogram for Single-Linkage 21
9 Elbow method for k-means clustering 23
10 Sample GO for a cluster 24
11 RNASeq Data 26
12 TE Data 27
13 Distribution of log_2 values of RNASeq data 27
14 Distribution of log_2 values of RPF data 28
15 Heatmap with Complete Linkage Hierarchical Clustering for Filtered Data 30
16 Cluster Analysis for RNASeq 31
17 Heatmaps for K-Means clustering on TE data (top to bottom) . . . 32
18 BIC for full covariance matrix, TE data 33
19 Heatmap for 2 clusters, GMM clustering on TE data 33
20 HMM Heatmap results on feature-engineered data 34
21 Tree Structure 35
22 CL Multinets Heatmap results on feature-engineered data 36
23 Heatmap from K-Means, RNASeq filter [-0.5, 0.5] 37
24 Heatmap from K-Means, RNASeq filter [-0.3, 0.3] 38
25 Cluster Analysis for RPF 41
26 Cluster Analysis for TE 41
27 BIC for different covariance settings in GMMs, TE 41
28 Trend plots for foldChange values, 2 clusters, GMM 42
29 Heatmaps for foldChange TE value 42
30 BIC results for HMM-based clustering 43
31 Trend plots for foldChange values, 4 clusters, HMM-based clustering . 43
32 Trend plots for foldChange values, 5 clusters, HMM-based clustering . 44

List of Tables

1 Covariance Settings for GMM 15
2 Optimal K for different datasets, K-Means 31
3 Optimal K for different datasets, HMMs 34
4 Performance evaluation for all algorithms* 37

References

[1] Virus. `https://www.nature.com/scitable/definition/virus-308`. Accessed: 2019-02-13.

[2] Maria João Amorim. A comprehensive review on the interaction between the host gtpase rab11 and influenza a virus. *Frontiers in cell and developmental biology*, 6:176, 2018.

[3] Li Wei Ang, Benjamin KW Koh, Kwai Peng Chan, Lian Tee Chua, Lyn James, and Kee Tai Goh. Epidemiology and control of hand, foot and mouth disease in singapore. *Ann Acad Med Singapore*, 38(2):106–12, 2009.

[4] David Arthur and Sergei Vassilvitskii. 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, 2007.

[5] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[6] Hans H Bock. Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis*, 23(1):5–28, 1996.

[7] Kenneth P Burnham and David R Anderson. Multimodel inference: understanding aic and bic in model selection. *Sociological methods & research*, 33(2):261–304, 2004.

[8] Barry R Cobb, Rafael Rumí, and Antonio Salmerón. Bayesian network models with discrete and continuous variables. In *Advances in probabilistic graphical models*, pages 81–102. Springer, 2007.

[9] Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561, 1970.

[10] Clare Davy and John Doorbar. G2/m cell cycle arrest in the life cycle of viruses. *Virology*, 368(2):219–226, 2007.

[11] Patrik D'haeseleer. How does gene expression clustering work? *Nature biotechnology*, 23(12):1499, 2005.

[12] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'Er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[13] James Glass and Victor Zue. 6.345 automatic speech recognition, 2003.

[14] Reimar Hofmann and Volker Tresp. Discovering structure in continuous variables using bayesian networks. In *Advances in neural information processing systems*, pages 500–506, 1996.

[15] Da Wei Huang, Brad T Sherman, Qina Tan, Jack R Collins, W Gregory Alvord, Jean Roayaei, Robert Stephens, Michael W Baseler, H Clifford Lane, and Richard A Lempicki. The david gene functional classification tool: a novel biological module-centric algorithm to functionally analyze large gene lists. *Genome biology*, 8(9):R183, 2007.

[16] Daxin Jiang, Chun Tang, and Aidong Zhang. Cluster analysis for gene expression data: a survey. *IEEE Transactions on knowledge and data engineering*, 16(11):1370–1386, 2004.

[17] Trupti M Kodinariya and Prashant R Makwana. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[18] Oded Maimon and Lior Rokach. Data mining and knowledge discovery handbook. 2005.

[19] Marina Meilă. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd international conference on Machine learning*, pages 577–584. ACM, 2005.

[20] Carla S Möller-Levet, Frank Klawonn, Kwang-Hyun Cho, and Olaf Wolkenhauer. Fuzzy clustering of short time-series and unevenly distributed sampling points. In *International Symposium on Intelligent Data Analysis*, pages 330–340. Springer, 2003.

[21] Duc Truong Pham and Gonzalo A Ruz. Unsupervised training of bayesian networks for data clustering. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 465, pages 2927–2948. The Royal Society, 2009.

[22] Frederick W Porter, Yury A Bochkov, Alison J Albee, Christiane Wiese, and Ann C Palmenberg. A picornavirus protein interacts with ran-gtpase and disrupts nucleocytoplasmic transport. *Proceedings of the National Academy of Sciences*, 103(33):12417–12422, 2006.

[23] John Quackenbush. Microarrays–guilt by association. *Science*, 302(5643):240–241, 2003.

[24] Alexander Schliep, Alexander Schönhuth, and Christine Steinhoff. Using hidden markov models to analyze gene expression time course data. *Bioinformatics*, 19(suppl_1):i255–i263, 2003.

[25] Pablo Tamayo, Donna Slonim, Jill Mesirov, Qing Zhu, Sutisak Kitareewan, Ethan Dmitrovsky, Eric S Lander, and Todd R Golub. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences*, 96(6):2907–2912, 1999.

[26] Vincent YF Tan. Ma4270 lecture notes, Jan 2018.

[27] Shao-Hua Wang, Ao Wang, Pan-Pan Liu, Wen-Yan Zhang, Juan Du, Shuang Xu, Guan-Chen Liu, Bai-Song Zheng, Chen Huan, Ke Zhao, et al. Divergent pathogenic properties of circulating coxsackievirus a6 associated with emerging hand, foot, and mouth disease. *Journal of virology*, 92(11):e00303–18, 2018.

[28] Guoqing Zhao and Wei Deng. An hmm-based hierarchical clustering method for gene expression time series data. In *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, pages 219–222. IEEE, 2010.