High Dimensional Inference of Gene Expressions Using a Parametric Trajectory: Developmental Transcriptome of Drosophila melanogaster

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

Master of Technology

IN

Faculty of Engineering

BY

Sameeksha Bhatia

Department of Computational and Data Science
Indian Institute of Science
Bangalore – 560 012 (INDIA)

June, 2025

Declaration of Originality

I, **Sameeksha Bhatia**, with SR No. **06-18-01-10-51-23-1-22824** hereby declare that the material presented in the thesis titled

High Dimensional Inference of Gene Expressions Using a Parametric Trajectory: Developmental Transcriptome of *Drosophila melanogaster*

represents original work carried out by me in the **Department of Computational and Data Science** at **Indian Institute of Science** during the years **2023-2025**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: 17.06.2025 Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

Advisor Name: Advisor Signature

Prof. Murugesan Venkatapathi

DEDICATED TO

My beloved parents

whose unwavering support and encouragement have been my constant source of strength.

Acknowledgements

I would like to express my heartfelt gratitude to Prof. Murugesan Venkatapathi for the invaluable opportunity to work under his guidance. His unwavering support, starting from my second semester through the completion of this project, has been instrumental in shaping this work. He has always been available whenever I needed help, and this thesis would not have been possible without his mentorship.

I am also grateful to Prof. Debnath Pal and Dr. Chirag Jain for their insightful suggestions and constructive feedback throughout the course of this work, especially during the evaluation of my thesis. Their inputs helped significantly in refining and improving the quality of this research.

I extend my deepest thanks to my mother and father for their unconditional love, motivation, and encouragement, which have always inspired me to believe in myself and pursue my goals. I also thank my sister for her constant support and encouragement throughout this journey.

A special thank you goes out to my dearest friends — Siddharth Jain, Diksha Seth, Eleena Gupta, Varsha Rana, Prashant Jadhao, and Parvesh Barak — for being there during tough times, sharing laughter, offering support, and making this journey more enjoyable with their constant presence and positivity.

I would also like to acknowledge all the members of the Computational Statistics and Physics Lab (CSPL) for providing a collaborative and pleasant working environment.

Finally, I am thankful to all the wonderful friends I made during my time at IISc. You made this journey truly memorable, and I will forever cherish the moments we shared together.

Abstract

Gene expression analysis plays a crucial role in understanding cellular functions and regulatory mechanisms. However, the high-dimensional nature of gene expression data, coupled with noise and complex temporal dynamics, may pose significant challenges. Traditional distance metrics such as Euclidean distance, or correlation metrics suited for linear relationships, often fail to capture biologically meaningful relationships where gene expression data exhibit high curvature. We propose methods to measure distances between noisy data points (or states) in high dimensions where the expected temporal evolution of the data is reasonably known. This approach uses a parametric curve given by a large set of polynomials representing the varying levels of expression of each gene with time. The distances both along and orthogonal to this highly curved expected path of evolution are shown to be applicable to gene expression data where the direct Euclidean distance to the known states is largely meaningless for inferences. We used the well established transcriptome of *Drosophila melanogaster* to study the utility of the proposed methods. Our experiments in inverting the time stamps of a given expression show reasonable resolution in the inference. We also show that additional scaling of distances using the polynomial fitting errors improves the (validation) performance within datasets, but it degrades the inference across datasets with varying measurement techniques. Note that the fitting errors largely capture measurement uncertainties in a dataset and cannot be translated to different measurement techniques for improvements in inference. Additionally, we have studied the use of the dynamics of the evolution i.e. the first and second derivatives of the trajectory to improve the robustness of the inferences, and this is a direction of future work.

Acronyms

Table 1: Key Acronyms used in the Report

Acronym	**Expansion**
CUR	Column-U-Row decomposition
GBA	Guilt-by-Association
GEO	Gene Expression Omnibus
ICA	Independent Component Analysis
modENCODE	Model Organism Encyclopedia of DNA Elements
NCBI	National Center for Biotechnology Information
PCA	Principal Component Analysis
RNA-seq	RNA sequencing
MSE	Mean Squared Error
RMSE	Root Mean Square Error
RPKM	Reads Per Kilobase of transcript per Million mapped reads
scRNA-seq	Single-cell RNA sequencing
SVD	Singular Value Decomposition
TDA	Topological Data Analysis
TI	Trajectory Inference
TPM	Transcripts Per Million
WGCNA	Weighted Gene Co-expression Network Analysis

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Understanding gene expression dynamics is fundamental to uncovering the regulatory mechanisms driving biological processes. High-throughput technologies like microarrays and RNA sequencing have generated large-scale gene expression datasets, enabling unprecedented biological insights. However, the high-dimensionality and noise inherent in such data pose significant computational challenges, particularly in extracting biologically meaningful patterns from complex, nonlinear interactions [4].

Traditional approaches such as clustering, network-based models, and dimensionality reduction have been widely used to identify co-expression patterns. While useful, these methods often assume linear relationships or predefined similarity metrics, limiting their ability to capture the intricate dynamics of gene regulation [1].

With the rise of single-cell transcriptomics, trajectory inference (TI) methods have emerged as powerful tools to model dynamic biological processes such as cell differentiation. These methods infer pseudotemporal orderings from static gene expression snapshots, reconstructing cell-state transitions without requiring explicit time-series data [5]. By doing so, TI generalizes clustering into topological maps that capture developmental continua.

To better handle the structural complexity in gene expression data, recent work has also explored topological and geometric frameworks. Topological Data Analysis (TDA), especially persistent homology, allows the detection of subtle global patterns and shape-based features that are often lost in traditional analyses [3]. Complementing this, dynamic modeling approaches—like differential equations—have shown success in describing gene trajectories over developmental time [6].

Time-series transcriptomic studies have further emphasized the importance of capturing coherent temporal patterns across genes. These datasets offer insights into dynamic regulatory programs but introduce challenges in clustering and alignment due to noise, delays, and variable timing across genes [7]. This complexity has motivated the need for frameworks that incorporate the temporal structure of expression data more explicitly.

Finally, curve-fitting methods based on linear and nonlinear regression are frequently employed in biological systems to capture trends from noisy data [8]. Collectively, these approaches signal a shift toward mathematically grounded frameworks that can model biological complexity more faithfully.

1.1 Motivation

The rapid advancements in high-throughput sequencing, microarrays, and mass spectrometry have ushered in the era of 'omics' sciences, enabling large-scale and parallel measurements of biological features. These technologies provide unprecedented opportunities to explore cellular functions and underlying regulatory networks [9]. However, the analysis of such high-dimensional data presents a multitude of computational challenges, necessitating the development of novel and robust analytical techniques [10].
Gene expression data exemplify the "four Vs" of big data:

1. **Volume:** Large-scale generation of gene expression data across time points and conditions.
2. **Velocity:** Increasing demands for real-time or rapid data processing.
3. **Variability:** Heterogeneous origins of biological data, including different tissues, cell types, or experimental conditions.
4. **Veracity:** Issues with noise, missing values, and biological variability that affect data reliability [4].

Specifically, gene expression analysis faces the following computational hurdles:

- **Curse of Dimensionality:** Thousands of genes measured across a relatively small number of samples make pattern recognition difficult with conventional metrics [10].
- **Non-linearity & Complexity:** Gene regulation involves intricate, non-linear interactions across multiple levels of biological hierarchy [4].

- **Causality vs. Correlation:** Observed correlations between gene expressions may not represent causal relationships [10].
- **Noise:** Experimental noise and biological variability can obscure meaningful signals [4].
- **Scalability:** Existing models often fail to scale efficiently with increasing dimensionality [9].

Furthermore, time-series gene expression data adds complexity by embedding temporal dependencies that many traditional methods overlook. This motivates the development of mathematical frameworks that can preserve temporal coherence while modeling the underlying biological structure [7].

Given the complexity and scale of modern biological data, there is a critical need for mathematical frameworks that can effectively handle high-dimensionality while preserving biologically meaningful insights. Traditional approaches—such as vector algebra, clustering, and dimensionality reduction—often oversimplify the nonlinear and intricate nature of gene regulation. Motivated by these challenges, our work introduces a novel mathematical framework that integrates topological and geometric principles to better represent the structure of gene expression data, with an ongoing effort to establish a principled theoretical basis for its biological relevance.

1.2 Related Work

Several strategies have been proposed for the analysis of high-dimensional gene expression data. Classical techniques like vector algebra, k-means and hierarchical clustering group genes [1] with similar profiles, while dimensionality reduction methods such as SVD, PCA [10, 11], and ICA [12] help visualize complex data. Co-expression networks constructed with graph theory [13], such as WGCNA [14], provide modular insights but often lack temporal awareness.

To address this, trajectory inference (TI) methods have gained traction, particularly with the advent of single-cell technologies. These methods infer developmental or differentiation pathways by estimating pseudotime from static gene expression snapshots. Tools like Monocle, Slingshot, and Palantir reconstruct cell-state transitions using clustering, graph traversal, and probabilistic modeling [5], enabling high-resolution insights into developmental trajectories.

Time-series expression studies further demand methods that explicitly model regulatory dynamics over time. Traditional clustering approaches typically ignore temporal information,

leading to biologically implausible groupings. Approaches such as point-wise distance metrics, model-based clustering, and symbolic transformation methods have been proposed to address this, but they face challenges in robustness and integration of biological priors [7].

Topological approaches, particularly persistent homology, have been applied to gene expression data to uncover shape-level features—such as loops and voids—that relate to biological transitions [3]. In parallel, dynamic modeling using differential equations has been used to track gene expression through developmental stages, such as in *Drosophila melanogaster* [6]. These models provide biologically grounded, interpretable representations of gene regulation dynamics.

Despite these advances, most existing methods either lack a principled geometric interpretation or fail to incorporate domain-specific developmental priors. Our work builds upon these foundations by integrating topological, geometric, and trajectory-based reasoning into a unified mathematical framework.

1.3 Contribution

In this study, we propose a mathematically grounded and computationally efficient framework that integrates topological and geometric principles to analyze high-dimensional gene expression data. Specifically, we introduce a method to measure distances between noisy data points in gene expression space, leveraging the known developmental trajectory as a prior as shown in Figure 1.1. This approach combines polynomial curve fitting, high-dimensional geometry, and topological insights to infer the proximity of new states (or samples) to the underlying biological trajectory.

Traditional distance metrics often fail to capture the complex geometry and high curvature inherent in developmental gene expression trajectories. Our framework addresses this by defining distances relative to the modeled trajectory, thereby incorporating geometric and temporal context into the analysis. This helps distinguish biologically meaningful proximity from superficial numerical similarity in high-dimensional space.

Using the developmental transcriptome of *Drosophila melanogaster* as a model system, we demonstrate how encoding the curvature of gene expression dynamics enables more accurate assessment of a sample's relation to developmental stages, offering a new perspective on interpreting transcriptomic variation through a geometric lens.

This method is particularly suited for small datasets with good temporal resolution and biological understanding. We hypothesize that the developmental cycle evolves in a coiled fashion in the high-dimensional gene space. Various metrics can be extracted and analyzed from this trajectory, such as:

- Determining the relative position of a test point t along the developmental trajectory between states i and j.
- Measuring the deviation of an outlier point k from the inferred gene expression curve.
- Estimating the rate of change for a given point t to reach k.

1.4 Outline of the Report

The report is structured as follows:

- Chapter 2. **Literature Survey**
  This chapter presents a comprehensive review of existing literature, focusing on computational and mathematical approaches used for analyzing high-dimensional gene expression data. It also discusses the associated challenges and highlights the gaps that motivate the proposed methodology.
- Chapter 3. **Methodology**
  This chapter describes the proposed methodology in detail, including dataset description, data preprocessing, curve fitting in high-dimensional space, polynomial coefficient extraction, and the computation of topological distances.
- Chapter 4. **Experimental Results**
  This chapter presents the experimental setup, dataset characteristics, evaluation metrics, and the results obtained using the proposed framework.
- Chapter 5. **Conclusion and Future Work**
  The final chapter summarizes the key contributions of the thesis, discusses its limitations, and outlines potential directions for future research.

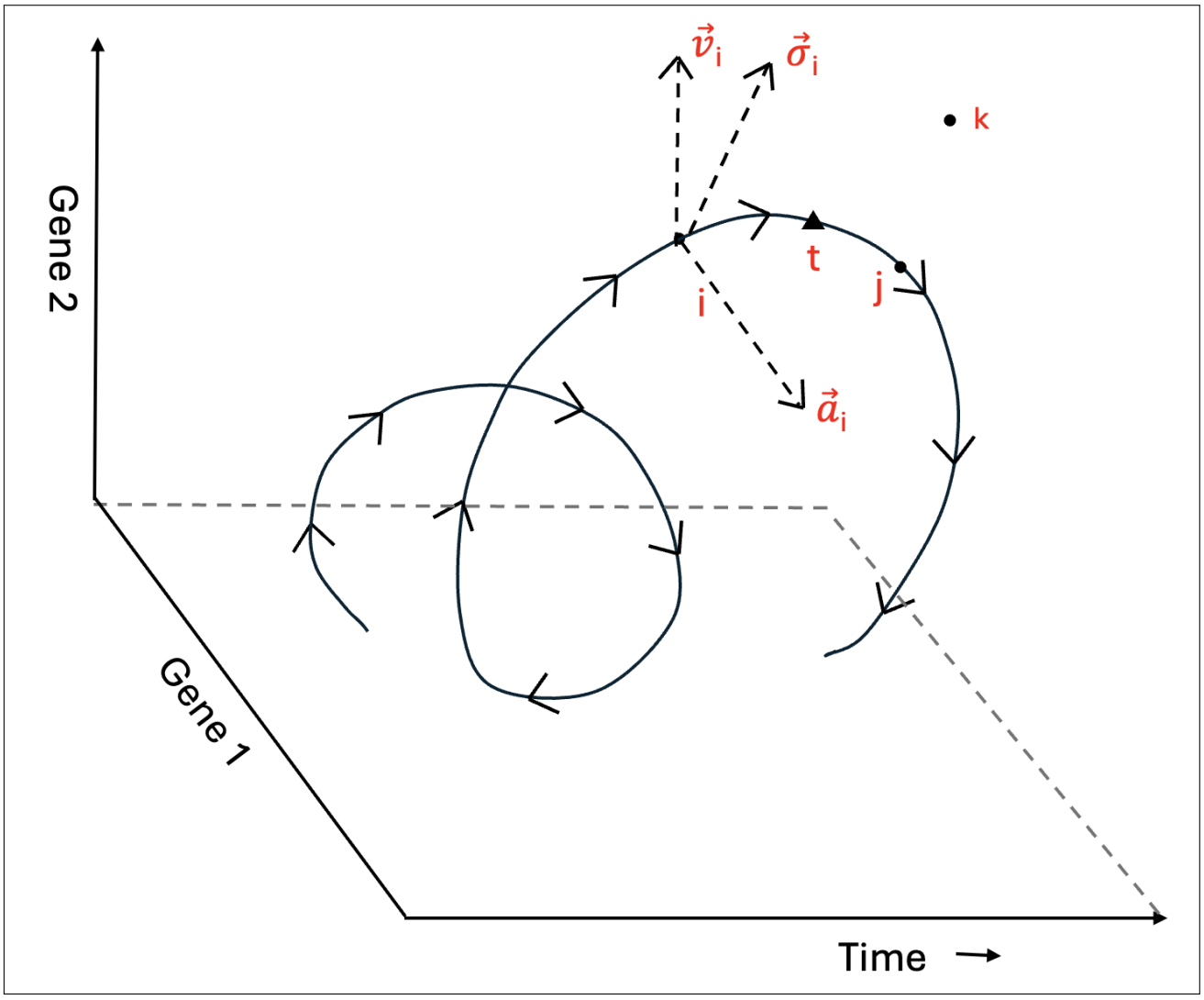

Figure 1.1: Schematic representation of the proposed trajectory-based inferential analysis of gene expression.

Chapter 2

Literature Review

2.1 Biological Data, Time-Series Gene Expression, and Their Challenges

Advancements in high-throughput experimental platforms have revolutionized biological data acquisition, particularly in transcriptomics. Techniques such as DNA microarrays, bulk RNA sequencing (RNA-seq), and more recently, single-cell RNA sequencing (scRNA-seq), allow for genome-wide quantification of gene expression across a variety of conditions and time points. These sequencing technologies have not only expanded the scale of data generation but have also enabled more refined temporal and spatial resolution in transcriptomic profiling [15].

As a result, modern biological datasets are characterized by the "four Vs" of big data: **Volume** (high gene counts and sample sizes), **Velocity** (rapid data generation and analysis), **Variability** (diverse sample sources and biological replicates), and **Veracity** (measurement noise, missing data, and experimental inconsistencies) [4, 9].

Within this landscape, time-series gene expression data—such as profiles measured across developmental stages or in response to perturbations—offer critical insights into the dynamic regulatory mechanisms of biological systems. These datasets enable tracking of transcriptional changes over time and have been instrumental in studying processes like immune response, embryonic development, circadian rhythms, and disease progression [7].

Unlike static datasets, time-series data exhibit temporal structure and dependencies. This calls for analytical techniques that can capture not only the magnitude but also the directional-

ity, smoothness, and progression of expression changes. Identifying coordinated patterns among genes over time can reveal underlying regulatory modules and transcriptional control systems.

Analyzing biological time-series data presents numerous challenges, both computational and biological. One of the primary difficulties lies in the unsupervised nature of clustering methods. These techniques aim to group genes with similar expression profiles, but similarity in the input space does not always imply functional or regulatory similarity. Consequently, the effectiveness of a clustering algorithm should not be judged solely by internal metrics, but by its ability to yield biologically interpretable groupings—such as genes co-regulated by common transcription factors or involved in the same pathway [7]. In the absence of standardized validation frameworks, comparing clustering outcomes across studies remains inconsistent.

A second challenge arises from the high dimensionality and sparse sampling of transcriptomic datasets. Typically, thousands of genes are profiled across only a limited number of samples or time points, leading to a disproportion between features and observations. This imbalance increases the likelihood of spurious correlations and hinders statistical power. While incorporating prior biological knowledge—such as gene ontology or pathway information—can help refine the analysis, it risks biasing the results and constraining the discovery of novel regulatory patterns. Additionally, most traditional clustering methods treat time points as independent, failing to capture the temporal dependencies intrinsic to biological processes. In time-series data, it is not just the magnitude of gene expression that matters, but also the trajectory, direction, and timing of changes. Methods that ignore these dynamics are poorly suited for identifying genes that are co-regulated over time or involved in time-dependent biological responses [7].

Experimental design poses its own set of challenges. Selecting an appropriate sampling rate is critical—undersampling can lead to missed transient events, while oversampling is often impractical due to cost or feasibility [16]. Moreover, synchronization of biological samples, especially in cyclic systems like the cell cycle, tends to degrade over time, complicating downstream analysis. Accurately identifying and adjusting for desynchronization is essential for maintaining biological interpretability.

From a data quality perspective, issues such as missing values, non-uniform time intervals, and biological noise can significantly degrade the reliability of inferences. Traditional interpolation and imputation techniques often struggle under these conditions, particularly when data are sparse or noisy. Furthermore, aligning gene expression profiles across experiments with

different temporal resolutions or phase shifts adds another layer of complexity.

At the pattern recognition level, algorithms like k-means or self-organizing maps assume that each time point is independent, making them ill-suited for modeling temporally correlated gene expression. This limitation reduces their ability to extract meaningful dynamic patterns, especially when dealing with short or noisy time series.

On a broader systems level, inferring gene regulatory networks from time-series data is a nontrivial task. It often requires integrating additional data types—such as chromatin binding, knock-out screens, or protein–DNA interaction datasets—to improve accuracy. While advanced models like dynamic Bayesian networks and spline-based regression offer promise, they are typically constrained to small-scale datasets due to computational complexity and limited sample sizes [16].

Finally, commonly used similarity and distance metrics, such as Euclidean distance or Pearson correlation, assume linearity and uniform variance—assumptions that are frequently violated in high-dimensional, nonlinear biological data. These limitations reduce their effectiveness in capturing the true structure and dynamics of gene expression patterns, especially when genes exhibit time-shifted or nonlinear responses[17].

Despite these challenges, time-series expression datasets remain indispensable for decoding transcriptional dynamics. Their effective analysis calls for computational strategies that are not only scalable and robust but also capable of capturing the temporal and geometric structure inherent in biological data.

2.2 Traditional Methods for Gene Expression Inference

Vector algebra has emerged as a powerful computational framework for analyzing genome-wide gene expression data. Given the high-dimensional nature of these datasets, vector-based methods offer a geometrically intuitive and computationally efficient approach to representing transcription profiles [1].

In this framework, gene expression is represented in a high-dimensional vector space, where each gene corresponds to a coordinate axis, and each experiment or condition is treated as a point (or vector) in that space. Similarities between gene expression profiles can be measured using vector operations such as angles and magnitudes. Notably, the cosine of the angle

between two expression vectors serves as a similarity measure equivalent to Pearson correlation.

Figure 2.1 illustrates this concept: Part **(a)** shows two transcription profiles across three genes. In part **(b)**, experiments are used as axes and genes as data points—an approach that helps identify variation across experiments but suffers from scalability as the number of experiments increases. Part **(c)** reverses the roles, using genes as axes and experiments as vectors. This formulation preserves dimensional consistency and enables more tractable geometric interpretations as the number of genes increases.

Overall, vector algebra provides a mathematically grounded approach for identifying expression similarities and serves as a foundation for clustering, classification, and projection techniques. However, as dimensionality increases, computational complexity and interpretability become limiting factors, motivating the development of more scalable alternatives.

The high-dimensional nature of gene expression data poses challenges in extracting meaningful patterns due to noise, redundancy, and computational complexity. Singular Value Decomposition (SVD) has emerged as a powerful dimensionality reduction technique for processing and modeling genome-wide expression data[10].

SVD transforms the original gene expression matrix into a lower-dimensional space composed of eigengenes and eigenarrays, which are unique orthonormal representations of genes and samples, respectively[10]. SVD decomposes a given matrix $A \in \mathbb{R}^{m \times n}$ into three matrices:

$$A = U \Sigma V^T \tag{2.1}$$

where:

- $U \in \mathbb{R}^{m \times m}$ is an orthogonal matrix containing the left singular vectors,
- $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix containing the singular values in decreasing order, and
- $V \in \mathbb{R}^{n \times n}$ is an orthogonal matrix containing the right singular vectors.

This decomposition helps in:
Singular Value Decomposition (SVD) provides several advantages in processing gene expression data. By filtering out eigengenes that primarily capture experimental noise or artifacts, SVD enhances data quality and reliability, making the analysis more robust. Additionally, it facilitates feature extraction by identifying dominant eigengenes, which allows for a better biological

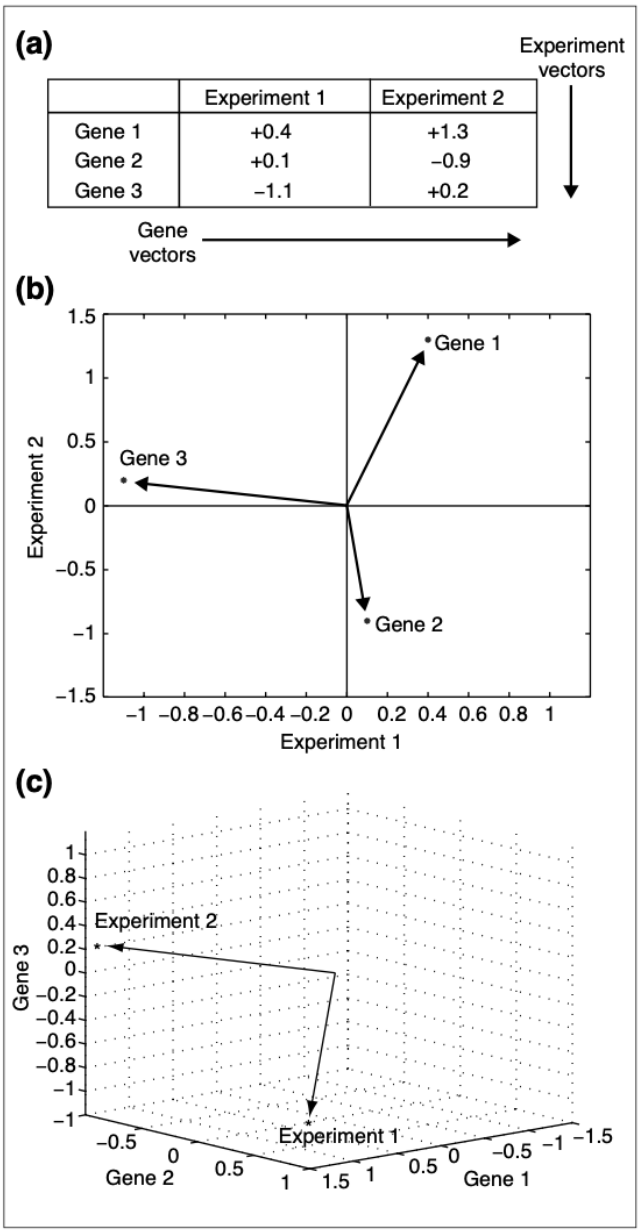

Figure 2.1: Vector-based representation of gene expression data [1].

interpretation of gene regulatory mechanisms. Moreover, SVD contributes to data compression by reducing the number of dimensions, thereby preserving essential biological variation while eliminating redundancy, which is crucial for handling the large-scale genomic datasets efficiently.

One key advantage of SVD is its ability to classify genes and arrays based on their expression dynamics rather than just overall similarity. Sorting the data by eigengenes provides a global view of gene regulation and cellular states, enabling insights into biological pathways and disease mechanisms.

While Singular Value Decomposition (SVD) is widely used for reducing the dimensionality of gene expression data, its reliance on linear combinations of all data points results in singular vectors that are often difficult to interpret biologically as singular vectors may not always correspond to real biological profiles, potentially leading to artificial or misleading outcomes[1].

In contrast CUR matrix decomposition provides a low-rank approximation that is explicitly expressed in terms of a small number of actual columns and rows from the original dataset [18]. CUR decomposition represents a matrix A as:

$$A \approx CUR \tag{2.2}$$

where:

- C consists of a subset of actual columns from A,
- R consists of a subset of actual rows from A, and
- U is a small matrix that captures interactions between selected rows and columns.

Unlike SVD, which produces abstract singular vectors, CUR retains actual data elements, making it inherently more interpretable and biologically meaningful. The selection of columns and rows is guided by statistical leverage scores, which identify features that exert a disproportionately large influence on the best low-rank fit [18].

Despite their advantages, both SVD and CUR decompositions have inherent limitations when applied to high-dimensional biological data. One major concern is the loss of information, as dimensionality reduction, while removing redundancy, may also discard weak but biologically relevant signals. Additionally, scalability issues arise since genome-scale datasets demand significant computational resources for decomposition and reconstruction. Another limitation is

the linear approximation assumption in SVD, which may fail to capture the complex nonlinear regulatory networks governing gene expression. In the case of CUR decomposition, its effectiveness depends on the method used to select columns and rows, making it susceptible to selection bias, where poor selection strategies can lead to suboptimal representations.

Clustering methods are widely used for identifying co-expressed gene groups in gene expression data. Traditional approaches like hierarchical clustering, k-means clustering, and self-organizing maps (SOMs) categorize genes into distinct clusters based on similarity metrics. However, these methods often assume that each gene belongs to only one cluster, which is a limitation, as many genes participate in multiple biological pathways. To address this, biclustering methods have been developed to identify subsets of genes that co-express across specific subsets of conditions, allowing for overlapping clusters and more biologically meaningful results[12].Clusters formed by such methods may also lack biological relevance, which reduces interpretability. Additionally, grouping data points solely by co-location in vector space can oversimplify the diversity of biological functions, potentially missing subtle yet crucial distinctions.

Co-expression networks are widely used to explore gene function, regulatory interactions, and disease associations by analyzing coordinated expression patterns across multiple samples. These networks are constructed by calculating pairwise correlations between gene expression profiles; genes with similar expression trends are connected as nodes, with edges representing the strength of their co-expression relationships.

As illustrated in Figure 2.2, the construction of co-expression networks begins with a correlation matrix capturing all gene-gene relationships. Network visualization and clustering techniques are then employed to identify distinct modules—groups of genes that exhibit strong internal co-expression. These modules can be analyzed to uncover regulatory hubs, assess pathway enrichment, and identify candidate genes through a guilt-by-association (GBA) strategy [2]. Differential co-expression analysis further enables the detection of modules that respond differently across biological conditions, offering insights into context-specific regulatory dynamics.

One of the main advantages of co-expression networks lies in their ability to identify functional gene modules—groups of co-regulated genes that likely participate in related biological processes. Unlike simple clustering, this framework can highlight context-dependent relationships and previously uncharacterized genes that may play crucial roles in specific pathways.

However, co-expression networks have limitations. They typically infer correlation, not cau-

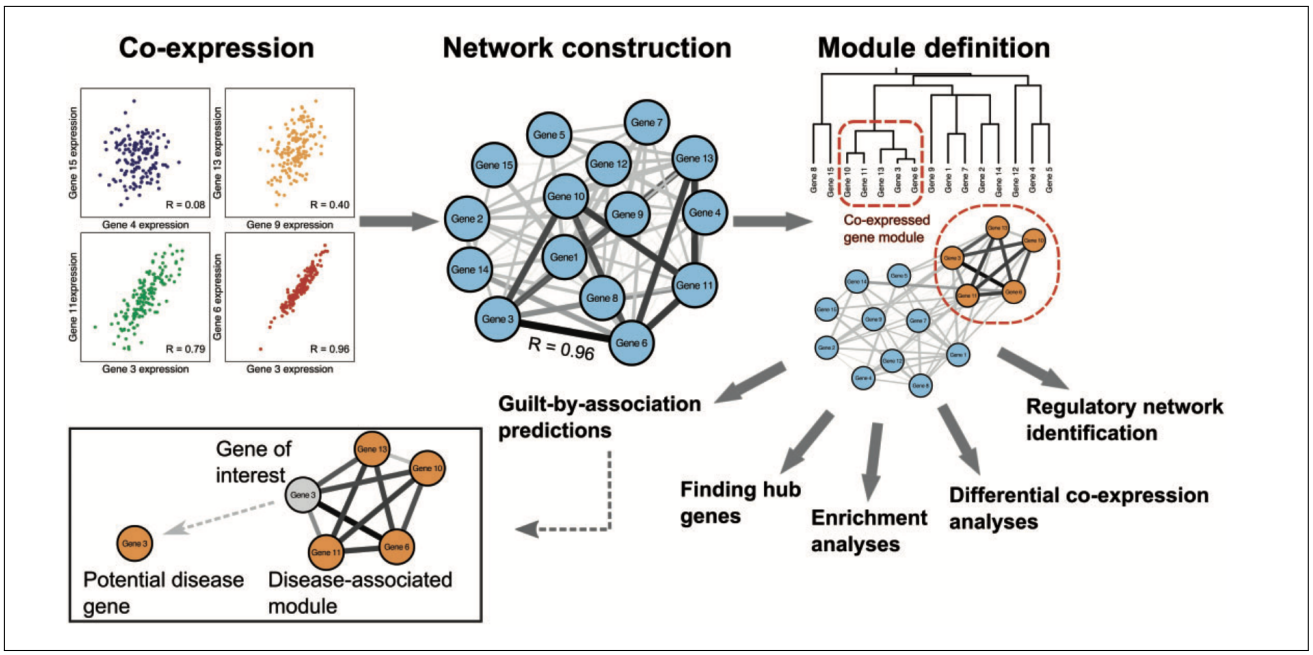

Figure 2.2: Workflow of co-expression network analysis [2]. Pairwise gene correlations are computed, clustered into modules, and analyzed for regulatory significance or functional enrichment.

sation—making it difficult to distinguish regulatory genes from those they regulate. Moreover, most implementations assume static co-expression, failing to capture dynamic changes over time, such as those observed in developmental or stimulus-responsive transcriptomes.

2.3 Trajectory Based Gene Inference Methods

Topological Data Analysis (TDA) is emerging as a powerful technique for analyzing high-dimensional biological data, including gene expression profiles. In this study, TDA was employed to improve phenotype prediction, particularly in distinguishing healthy individuals from those with Parkinson's disease[3].

The approach involves transforming gene expression data into topological signatures using persistent homology. Instead of directly applying machine learning models, gene expression matrices were first converted into weighted point clouds. These point clouds were then processed using TDA techniques to extract persistent homology features as shown in Fig. 2.3, which capture the intrinsic shape of the data. By integrating these topological summaries into machine learning models like support vector machines (SVM), random forests, and neural networks, the study demonstrated improved disease classification compared to traditional

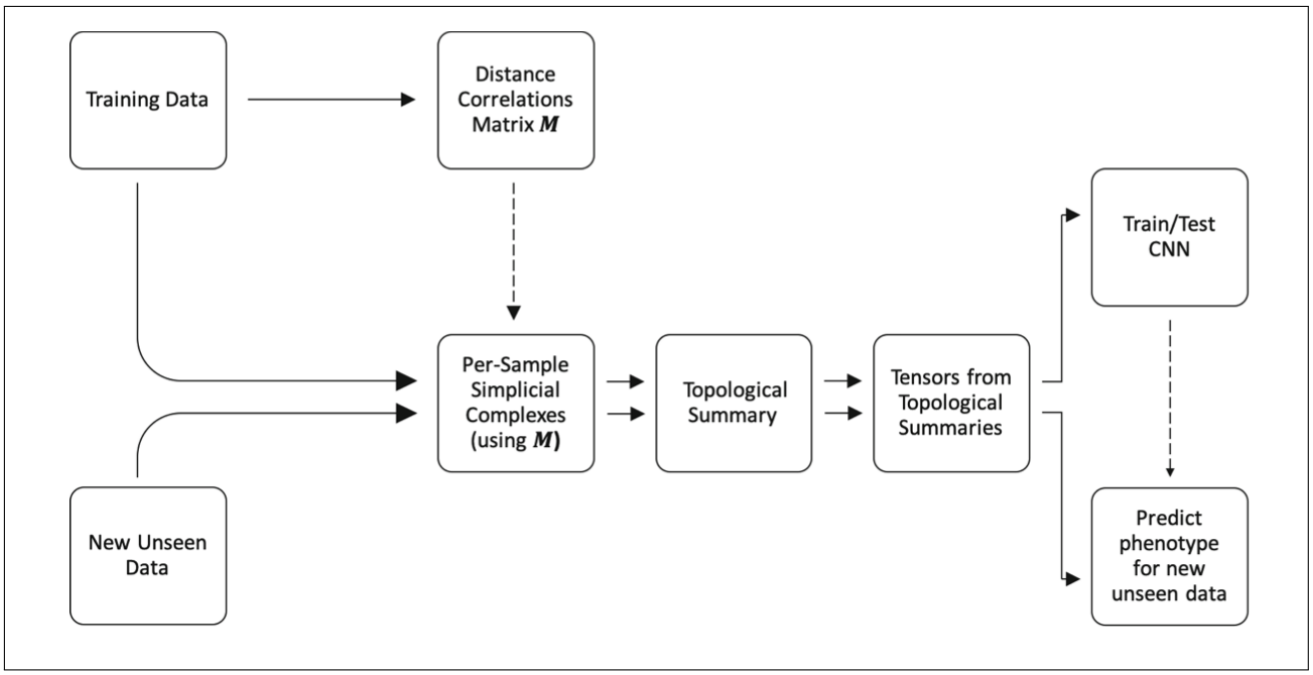

Figure 2.3: Workflow of Topological Data Analysis (TDA)[3]

methods[3].
A key finding was that distance correlation—which captures both linear and nonlinear associations between gene expressions—provided a more informative structure than standard correlation measures. This highlights the ability of TDA to uncover subtle patterns that may be missed by conventional statistical techniques. Additionally, persistence landscapes were used to generate feature representations that enhanced model performance in phenotype classification[3].

Overall, this study confirms that TDA can effectively enhance phenotype prediction by capturing the complex, high-dimensional relationships inherent in gene expression data. The methodology is applicable not only to Parkinson's disease, but can be extended to other biological domains. Curve fitting has long been employed in biological and agricultural sciences to model processes such as growth, environmental responses, and biochemical kinetics. These models help translate complex biological phenomena into mathematical representations by identifying underlying patterns in experimental data. In this framework, linear and nonlinear regression techniques are used to fit parametric functions to observed measurements, enabling predictions, characterization of system dynamics, and evaluation of rate changes over time. By adjusting model parameters to minimize the discrepancy between predicted and actual observations—typically via least squares optimization—researchers can simulate realistic system

behaviors under varying conditions. The interpretability and versatility of these fitted models make them especially valuable for understanding dynamic systems, even when data are noisy or sparse [8]. This foundational principle continues to inform newer approaches to modeling gene expression dynamics across developmental trajectories and cellular differentiation paths.

Traditional dimensionality reduction methods assume gene expression data lie in flat Euclidean space. However, Zhou and Sharpee [19] showed that while local gene expression patterns may appear Euclidean, their global structure is better captured by hyperbolic geometry. Analyzing datasets across various tissues and species, they found that gene expression forms hierarchical structures, with the depth of hierarchy reflected in the radius of the hyperbolic space. Embryonic cells exhibited shallow hierarchies, while specialized brain cells displayed deeper, more complex structures. As more genes were considered, hyperbolic signatures became more apparent. These findings suggest that gene expression is organized in a low-dimensional, tree-like space, and incorporating hyperbolic models allows for a more accurate representation of biological relationships and regulatory hierarchies.

Haye et al. (2009) proposed a top-down framework to model the temporal evolution of Drosophila gene expression using DNA microarray time series [6]. They clustered over 4000 genes into 17 groups with similar temporal profiles and applied linear differential equations to model interactions among these clusters. Remarkably, the system reproduced experimental data with high precision, and parameter reduction validated the hypothesis of sparse regulatory networks—each cluster connected to only a few others. In a follow-up study, Haye et al. (2012) extended this framework using robust nonlinear differential models [20]. These models incorporated biologically motivated constraints such as robustness and temporal stability. While linear models reproduced the data efficiently, they failed to generalize over longer time frames. In contrast, nonlinear models, particularly those with exponential terms, achieved accurate fits, robustness to perturbations, and biological realism. Together, these works highlight the importance of dynamical systems approaches for modeling developmental gene expression.

Trajectory inference (TI) methods have evolved into a central tool for decoding cellular dynamics from single-cell omics data. Deconinck et al. [5] reviewed the rapid expansion of this field, highlighting a shift from early clustering and graph-based approaches toward probabilistic models that incorporate uncertainties and multi-modal data. These advancements allow the integration of RNA velocity, time-series measurements, and epigenomic features, enabling more accurate and context-aware trajectory reconstruction. Additionally, TI has expanded

beyond cell ordering to include downstream analyses like differential expression testing, trajectory alignment across conditions, and dynamic gene regulatory network (GRN) inference. The study also emphasized challenges such as benchmarking, trajectory validation, and the circularity of using the same data for both trajectory and statistical inference. Moving forward, the field is expected to embrace multimodal integration and uncertainty modeling, offering a more comprehensive view of cellular state transitions in development, disease, and regeneration.

Building upon these recent advances, our work aims to develop a novel, geometry-aware approach for analyzing time-series transcriptomic data. By leveraging the structure of high-dimensional gene expression trajectories, we seek to improve biological interpretability and capture temporal progression more effectively. Our methodology is designed to address current challenges in scalability, trajectory resolution, and regulatory inference, particularly in complex developmental datasets such as those of Drosophila melanogaster. We discuss this in more detail in Chapter 3.

Chapter 3

Methodology

3.1 Dataset Selection and Description

In this study, we develop a framework to analyze gene expression distances in high-dimensional space by incorporating topological and geometric insights. Traditional methods, such as Euclidean distance, often fail to capture meaningful relationships in biological datasets, particularly when gene expression trajectories exhibit high curvature and complex temporal patterns. To address this, we utilize the developmental transcriptome of *Drosophila melanogaster* as a model system to better understand the structure of gene expression data and propose an improved method for measuring distances between noisy states.

Our methodology begins with selecting a high-resolution, unbiased dataset from the modENCODE project, which provides a detailed view of gene expression across developmental stages. We then demonstrate that gene expression trajectories in high-dimensional space follow a curved, non-Euclidean structure, making conventional distance measures inadequate. To extract biologically meaningful relationships, we construct co-expression networks and analyze the distances between stage-specific networks to capture the curvature of gene expression changes.

Drosophila melanogaster is a key non-mammalian model in biological research, contributing to major discoveries such as the identification of chromosomes as carriers of genetic information and the role of genes in development. It also shares a significant proportion of its genetic content with humans, making it a valuable translational model for studying human development [21].

A critical dataset for this study is the developmental transcriptome of *Drosophila melanogaster*, comprehensively profiled by Graveley et al. as part of the modENCODE project [21]. This

dataset provides a high-resolution temporal map of gene expression across 30 distinct developmental stages, including: 12 embryonic stages (0–24h embryo at 2-hour intervals), 6 larval stages, 6 pupal stages, and 3 male and 3 female adult stages.

To generate this dataset, high-throughput sequencing technologies were employed, including: Illumina Genome Analyzer IIx (75-bp reads), Applied Biosystems SOLiD platform (50-bp reads), and Roche 454 platform (250-bp reads). Gene expression levels were normalized using Reads Per Kilobase of transcript per Million mapped reads (RPKM) to ensure comparability across samples.

This dataset was chosen due to its fine-grained temporal resolution, enabling a detailed investigation of high-dimensional gene expression trajectories. Furthermore, its unbiased gene selection allows for a more fundamental approach to addressing challenges related to biological curvature and high dimensionality in gene expression data. To begin the analysis, non-coding genes were removed, resulting in a final list of approximately 13,639 genes.

To further refine the analysis, we apply Principal Component Analysis (PCA) and study the projections of developmental stages onto the first two principal components, identifying significant patterns in gene expression variation. Finally, we develop a high-dimensional smooth curve fitting approach to accurately model the developmental cycle of *Drosophila melanogaster*. This fitted curve serves as a benchmark for measuring distances from new test data points in the same high-dimensional space, enabling a more precise analysis of temporal gene expression evolution.

Two test datasets were also selected to validate stage prediction using the fitted curve. Since the expression values were reported in different units (RPKM), they were converted to TPM for consistency and comparability across sequencing platforms. The conversion was performed using the following formulas:

RPKM (Reads Per Kilobase of transcript per Million mapped reads): RPKM normalizes read counts by both the length of the gene and the total number of reads in the experiment. It is computed as:

$$\text{RPKM}_i = \frac{C_i}{L_i \times \frac{N}{10^6}} = \frac{C_i \times 10^9}{L_i \times N}$$

where:

- C_i = Number of reads mapped to gene i
- L_i = Length of gene i in base pairs
- N = Total number of mapped reads in the experiment

TPM (Transcripts Per Million): TPM also normalizes for gene length and sequencing depth but differs from RPKM in the order of operations, making TPM values more comparable across samples. It is calculated as:

$$\text{TPM}_i = \frac{\frac{C_i}{L_i}}{\sum_{j=1}^{N} \frac{C_j}{L_j}} \times 10^6$$

Alternatively, TPM can be derived directly from RPKM values:

$$\text{TPM}_i = \frac{\text{RPKM}_i}{\sum_{j=1}^{N} \text{RPKM}_j} \times 10^6$$

where the denominator sums over all genes in the sample.
The two test datasets used in this study were:

1. **Becker et al. (2018)** [22]: This dataset comprises a combined transcriptome and proteome time-series collected at 14 distinct stages of *Drosophila* embryonic development. Although a moderate correlation was observed between mRNA expression and protein levels ($\rho = 0.54$), mathematical modeling of translation and degradation processes was able to explain 84% of the protein expression dynamics based on mRNA measurements. This enabled the categorization of proteins into four regulatory groups. Additionally, the study provided a detailed analysis of genes regulated post-transcriptionally, including insights into the role of the RNA-binding protein Hrb98DE (GSE121160).

2. **Daines et al. (2011)** [23]: This study generated a broad transcriptomic profile using RNA-seq data from 10 different developmental stages, although with a lower temporal resolution. It included 142.2 million uniquely mapped paired-end reads (64–100 bp), sequenced using the Illumina GA II platform. The sequencing yielded a depth of 3563× and successfully covered more than 95% of FlyBase-annotated genes, along with 90% of known splice junctions. The analysis also led to updates in 30% of FlyBase gene models, incorporating novel exons, alternative splicing patterns, and extended untranslated regions. In total, 319 new transcripts were identified, and alternative splicing was detected in 31% of genes, surpassing earlier estimates (GSE24324).

RNA-seq offers several advantages over microarrays, such as increased sensitivity, wider dynamic range, and the ability to detect novel isoforms and splicing events. These features made the datasets ideal candidates for comparison in our curve-fitting and distance inference framework. These datasets were used to calculate distances from multiple time points along the fitted high-dimensional curve, and the shortest distance was selected as the inferred developmental time. Different distance metrics were also employed, as described in subsequent sections.

3.2 Curvature of the Data

Euclidean Distances

The gene expression data for each of the 30 developmental stages was represented as a set of 30 vectors, each of dimension 13639×1. The Euclidean distance between each pair of stages i and j was computed as:

$$d_{ij} = \sqrt{\sum_{k=1}^{13639} (x_{ik} - x_{jk})^2} \tag{3.1}$$

where x_{ik} and x_{jk} denote the expression levels of gene k in stages i and j, respectively.

To ensure comparability across stages, these distances were normalized with respect to the L_2-norm of the first stage ($i = 1$) as follows:

$$d_{ij}^{\text{norm}} = \frac{d_{ij}}{\|x_1\|_2} \tag{3.2}$$

where

$$\|x_1\|_2 = \sqrt{\sum_{k=1}^{13639} x_{1k}^2} \tag{3.3}$$

represents the Euclidean norm of the first stage's expression vector. This normalization ensures that all distances are expressed relative to the initial stage, allowing for a consistent comparison across developmental transitions.

Next, the gene expression vectors were further normalized to a range of [0,1] to eliminate scale differences across genes. This was achieved by dividing each gene's expression level by its maximum value across all stages:

$$x_{ik}^{\text{scaled}} = \frac{x_{ik}}{\max_i x_{ik}} \tag{3.4}$$

where x_{ik}^{scaled} represents the normalized expression of gene k at stage i. After this transformation, the Euclidean distances were recalculated using the normalized vectors, following the same procedure as before:

$$d_{ij}^{\text{scaled}} = \frac{\|x_i^{\text{scaled}} - x_j^{\text{scaled}}\|_2}{\|x_1^{\text{scaled}}\|_2} \tag{3.5}$$

This additional normalization step ensures that distances are robust to differences in gene expression magnitude and focus solely on relative changes across developmental stages.

Co-expression Analysis

For each developmental stage, a co-expression matrix was generated using the outer product of the corresponding normalised gene expression vector. Given an expression vector x_i for stage i, the co-expression matrix was computed as:

$$M_i = x_i x_i^T \tag{3.6}$$

where M_i represents the co-expression matrix for stage i. To quantify differences between developmental stages, the Frobenius norm of the difference between co-expression matrices of each stage was computed:

$$D_{ij} = \|M_i - M_j\|_F \tag{3.7}$$

where D_{ij} denotes the Frobenius norm of the difference between the co-expression matrices of stages i and j.

To ensure comparability across stages, these differences were normalized with respect to the first stage ($i = 1$) using:

$$D_{ij}^{\text{norm}} = \frac{D_{ij}}{\|M_1\|_F} \tag{3.8}$$

This normalization allows for a consistent measure of co-expression changes over time, relative to the initial developmental stage.

Dimensionality reduction

To analyze patterns in gene expression across developmental stages, Principal Component Analysis (PCA) was applied to the covariance matrix of the gene expression data. Given the expression matrix A with columns representing gene expression vectors across stages, the covariance matrix was computed as:

$$C = AA^T - \mu\mu^T \tag{3.9}$$

where: A is the 13639×30 matrix containing gene expression vectors for all 30 developmental stages and μ is the mean expression vector, computed as:

$$\mu = \frac{1}{30}\sum_{i=1}^{30} A_i \tag{3.10}$$

PCA was performed on C to extract its principal components, and the first two principal components were selected. The 30 expression vectors were then projected onto these two components to obtain a low-dimensional representation:

$$Z = P_2^T A \tag{3.11}$$

where: P_2 is the matrix containing the first two principal components and Z is the projected data matrix in the 2D principal component space.

Finally, the projected points were visualized to identify any significant patterns in the developmental trajectory. This step allows for the detection of underlying structure in the gene expression data and provides insights into the temporal progression of gene regulation.

3.3 High-Dimensional Smooth Curve Fitting

To model the developmental trajectory in a 13,639-dimensional space, we fitted 13,639 independent polynomial curves—each corresponding to a single gene's expression trajectory over developmental time. The degree of each polynomial was allowed to vary from 5 to 8, or until the root mean square error (RMSE) dropped below 0.08, i.e., allowing a maximum of 8% error, whichever occurred first. The coefficients obtained from these polynomial fits define the shape of the curve for each gene.

Algorithm 1 Adaptive Polynomial Curve Fitting for Gene Expression Trajectories

Input: Gene expression matrix $\mathbf{X}$ with genes as rows and stages as columns, stage-to-time mapping T, initial degree d_0, maximum degree $d_{\max}$, RMSE threshold $\epsilon = 0.08$, stages to omit S_{omit}
Initialize list `all_coefficients` to store coefficients and errors for all polynomial fits
Initialize list `best_coefficients` to store the best polynomial fit per gene
for each gene g_i in dataset **do**
    Remove stages S_{omit} from g_i's expression vector
    Let $y \leftarrow$ expression values at retained stages
    Let $t \leftarrow$ corresponding time points from T
    Initialize $d \leftarrow d_0$, $rmse \leftarrow \infty$, $best\_rmse \leftarrow \infty$
    while $rmse > \epsilon$ and $d \leq d_{\max}$ **do**
        Fit polynomial $p_d(t)$ of degree d to (t, y)
        Compute $\hat{y} = p_d(t)$
        Compute mse $= \frac{1}{n}\sum(y - \hat{y})^2$, rmse $= \sqrt{\text{mse}}$
        Store degree-d coefficients and RMSE in `all_coefficients`
        if $rmse < best\_rmse$ **then**
            Save current p_d coefficients as `best_fit`
            Update $best\_rmse \leftarrow rmse$
        end if
        $d \leftarrow d + 1$
    end while
    Append `best_fit` for g_i to `best_coefficients`
end for
Output: `all_coefficients` and `best_coefficients` saved as CSVs

The fitting process was implemented as outlined in Algorithm 1, which adaptively selects the optimal polynomial degree for each gene based on the RMSE constraint. The high-dimensional developmental trajectory is then represented as:

$$\mathbf{D}(t) = [x_1(t), x_2(t), \ldots, x_{13639}(t)] \tag{3.12}$$

where $\mathbf{D}(t)$ denotes the trajectory in 13,639-dimensional space, and $x_k(t)$ represents the polynomial function fitted for gene k over developmental time t.

This high-dimensional curve is expected to be highly curved or coiled, capturing the intricate gene expression dynamics across the developmental stages of *Drosophila melanogaster*. It serves as a reference trajectory for comparing gene expression states and for measuring distances from new data points in the same high-dimensional space.

The full developmental cycle dataset of *Drosophila melanogaster* was also used to validate the robustness of the fitted curve by selectively omitting certain stages during curve fitting. Specifically, the stages `mE_mRNA_em4-6hr`, `mE_mRNA_em12-14hr`, `mE_mRNA_em20-22hr`, `mE_mRNA_L2`, `mE_mRNA_L3_PS7-9`, and `mE_mRNA_P9-10` were excluded from the fitting process. These stages were then used as test points to evaluate how accurately their developmental time could be recovered (i.e., inverted) using the fitted curve as a reference.

A similar procedure was applied to the embryogenesis test dataset, where the time points `03h`, `12h`, and `18h` were omitted during the curve fitting and subsequently predicted using the constructed trajectory.

These experiments were designed to validate the capability of the fitted high-dimensional curve to accurately represent the developmental trajectory and to infer missing or unknown time points based on minimal distances in the gene expression space.

3.4 Stage Inversion

After fitting high-dimensional gene expression trajectories, we aimed to infer the developmental stage of unlabelled or intermediate samples. This was performed on both the Full Cycle and Embryogenesis datasets. Additionally, we evaluated the feasibility of cross-dataset stage inference by using the Full Cycle trajectory as a reference for assigning developmental stages to samples in the Embryogenesis dataset and vice versa.

To facilitate this, we first constructed a finely sampled version of the trajectory by interpolating additional points between each pair of known timepoints. Specifically, for each segment between adjacent original stages, we introduced 10 evenly spaced intermediate points along the fitted curve and calculated the expression at each interpolated timepoint as shown in Algorithm 2 using the coefficients calculated using Algorithm 1. This produced a denser and smoother representation of the trajectory, enabling more precise distance-based mapping of test samples.

Algorithm 2 Interpolated Timepoint Expression Calculation

Input: Mapping of developmental stages to time in hours T_{map}, dividing factor d (default = 10), maximum degree $d_{\max}$, coefficient matrix $\mathbf{C}$ from Algorithm 1
Output: Dense timepoint expression matrix $\mathbf{E}$
Initialize empty list `timepoints_list`
Let $S \leftarrow$ list of stages sorted by T_{map}
for each consecutive stage pair (s_i, s_{i+1}) in S **do**
    $t_{\text{start}} \leftarrow T_{\text{map}}[s_i]$
    $t_{\text{end}} \leftarrow T_{\text{map}}[s_{i+1}]$
    Generate d evenly spaced timepoints between t_{start} and t_{end} using:
        $t_{\text{points}} \leftarrow \texttt{linspace}(t_{\text{start}}, t_{\text{end}}, d, \text{endpoint=False})$
    Round each timepoint to 3 decimal places and append to `timepoints_list`
end for
Power Matrix Computation:
Let $\mathbf{P}$ be a matrix with rows indexed by timepoints and columns representing x^0 to $x^{d_{\max}}$
for each timepoint t in `timepoints_list` **do**
    Construct row vector: $[t^0, t^1, \ldots, t^{d_{\max}}]$
    Append row to matrix $\mathbf{P}$
end for
Expression Computation:
Let $\mathbf{C}$ be the coefficient matrix (genes $\times$ degrees), with columns x^0 to $x^{d_{\max}}$
Transpose $\mathbf{C}$ to get $\mathbf{C}^\top$ of shape (degrees $\times$ genes)
Compute expression matrix: $\mathbf{E} \leftarrow \mathbf{P} \cdot \mathbf{C}^\top$
Save $\mathbf{E}$ as CSV file
Save timepoint-wise expression vectors by transposing $\mathbf{E}$ and saving each row separately in the designated timepoint vector folder
Return: $\mathbf{E}$

Let $\hat{\mathbf{p}}_k$ denote an interpolated point on the curve and $\hat{\mathbf{p}}_0$ represent a test sample whose stage is to be inferred. We used the following two distance-based methods to determine the point on the curve that is closest to the test sample, and thereby assign it a corresponding stage.

Distance without Removing Projections

In the simplest case, we defined the relative vector between the test sample and each interpolated point on the curve as:

$$\hat{\mathbf{R}} = \hat{\mathbf{p}}_k - \hat{\mathbf{p}}_0$$

We then computed the Euclidean norm of $\hat{\mathbf{R}}$ for all interpolated points and selected the one with the smallest norm as the closest match. The corresponding timepoint of this interpolated point was assigned to the test sample as described in Algorithm 3

Algorithm 3 Computation of Distance and Scaled Distance Matrices

Input: Coefficient matrix C, expression vector folder F_e, timepoint vector folder F_t, stage list S, stage-hour map H, timepoint list T, output subfolder f
Filter stages: $S \leftarrow \{s \in S \mid s \in H\}$
Initialize matrices $D \in \mathbb{R}^{|S|\times|T|}$ and $D_s \in \mathbb{R}^{|S|\times|T|}$
Replace zero entries in $C[\texttt{Best\_RMSE}]$ with 10^{-8}
Compute weights: $w_i \leftarrow \frac{1}{\texttt{Best\_RMSE}_i} \quad \forall i$
Compute normalization factor: $Z \leftarrow \sqrt{n}/\sqrt{\sum_i w_i^2}$
for each stage index i and stage name s in S **do**
    Load expression vector $x \leftarrow \texttt{Read}(F_e/\texttt{stage\_}(i+1).csv)$
    for each timepoint index j and time t in T **do**
        Load timepoint vector $y \leftarrow \texttt{Read}(F_t/\texttt{timepoint\_}t.csv)$
        Compute distance: $D[i,j] \leftarrow \|x - y\|_2$
        Compute scaled distance: $D_s[i,j] \leftarrow \frac{\|(x-y)\odot w\|_2}{Z}$
    end for
end for
Save D and D_s to CSVs
Return: D, D_s

Local Clustering Refinement: However, this method can be sensitive to local curvature and non-uniform point density along the trajectory, potentially resulting in inconsistent stage estimates. To improve robustness, we implemented a local clustering strategy. For each test sample, we selected the 10 closest interpolated points based on initial Euclidean distance. We then computed their mean timepoint and standard deviation. Points lying more than two standard deviations away from the mean were iteratively removed, and the statistics were recomputed at each step. This process continued until the set stabilized, and the final mean of

the pruned set was taken as the inferred developmental stage as described in 4. This helped suppress the influence of outlier projections and improved the continuity of stage assignments.

Algorithm 4 Local Clustering Refinement for Stage Estimation

Input: Test sample x, interpolated timepoints $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ with associated expression vectors $\{e(t_i)\}$, number of neighbors $k = 10$
Compute Euclidean distances $d(x, e(t_i))$ for all $t_i \in \mathcal{T}$
Let $\mathcal{N}_x \leftarrow$ timepoints of k closest neighbors to x based on $d(x, e(t_i))$
Extract time values $\{t_j\}$ from $\mathcal{N}_x$
Initialize $\mu \leftarrow \text{mean}(\{t_j\})$, $\sigma \leftarrow \text{std}(\{t_j\})$
repeat
    Prune set: $\mathcal{N}_x \leftarrow \{t_j \in \mathcal{N}_x \mid |t_j - \mu| \leq 2\sigma\}$
    Update: $\mu \leftarrow \text{mean}(\mathcal{N}_x)$, $\sigma \leftarrow \text{std}(\mathcal{N}_x)$
until no changes in $\mathcal{N}_x$
Output: Refined developmental stage estimate $\hat{t}_x \leftarrow \mu$

Distance with Removing Projections

To further account for the geometric structure of the trajectory, we extended the distance calculation by explicitly removing the components of the relative vector that aligned with the tangent and curvature directions of the trajectory as given in Algorithm 6 along with the evolution dynamics thresholding described in Section 3.4. This approach helped focus on deviations orthogonal to the local flow of the trajectory, which are more indicative of biological divergence rather than natural temporal progression.

Formally, the adjusted relative vector was computed as:

$$\hat{\mathbf{R}} = (\hat{\mathbf{p}}_k - \hat{\mathbf{p}}_0) \cdot \left(\mathbf{I} - \frac{1}{\|\hat{\mathbf{u}}\|^2} \hat{\mathbf{u}}\hat{\mathbf{u}}^\top - \frac{1}{\|\hat{\mathbf{a}}\|^2} \hat{\mathbf{a}}\hat{\mathbf{a}}^\top \right)$$

where $\hat{\mathbf{u}} = \frac{d\hat{\mathbf{p}}_k}{dt}$ is the tangent vector representing the first derivative of the curve, and $\hat{\mathbf{a}} = \frac{d\hat{\mathbf{u}}}{dt}$ is the acceleration vector representing the second derivative. The derivatives were calculated at each timepoint using Algorithm 5 and have been further used in Algorithm 6. This formulation effectively projects the relative vector onto the space orthogonal to the trajectory's direction and curvature at that point, discounting variations that are natural to time progression.

Algorithm 5 Computation of First and Second Derivative Coefficients and Values

Input: Polynomial coefficient matrix C (columns 0–4 contain metadata), timepoints list $T = [t_1, t_2, \ldots, t_n]$, folder name f, normalization flag, unit label
for each gene row r in C **do**
    Extract metadata $m \leftarrow r[0{:}5]$, coefficients $c \leftarrow r[5:]$
    Form polynomial $P(x) \leftarrow \texttt{Polynomial}(c)$
    Compute first derivative $P'(x) \leftarrow P.\texttt{deriv}(1)$
    Compute second derivative $P''(x) \leftarrow P.\texttt{deriv}(2)$
    Store $[m, \text{coefficients of } P'(x)]$ in first derivative matrix
    Store $[m, \text{coefficients of } P''(x)]$ in second derivative matrix
end for
Convert both matrices to DataFrames with dynamic column names, fill missing values with 0, and save as CSV files in folder f
Convert timepoints T to NumPy array $T_{\text{array}} \leftarrow \texttt{np.array}(T)$
Evaluate first derivatives: For each coefficient vector c in first derivative DataFrame, compute $\texttt{Polynomial}(c)(T_{\text{array}})$ and stack into matrix V_1
Evaluate second derivatives: For each coefficient vector c in second derivative DataFrame, compute $\texttt{Polynomial}(c)(T_{\text{array}})$ and stack into matrix V_2
Concatenate metadata with V_1 and V_2, then save as CSV files in folder f
Return: DataFrames of first and second derivative coefficients and their evaluated values. Also, derivatives are saved for each timepoint as vectors in CSV.

Algorithm 6 Computation of Distance and Scaled Distance Matrices with Derivative Projection and Thresholding

Input: Coefficient matrix C, expression vector folder F_e, timepoint vector folder F_t, first derivative folder F_v, second derivative folder F_a, uncertainty dataframe U, stage list S, stage-hour map H, timepoint list T, output subfolder f
Filter stages: $S \leftarrow \{s \in S \mid s \in H\}$
Initialize matrices $D, D_s \in \mathbb{R}^{|S| \times |T|}$ {Distance and Scaled Distance}
Replace zero entries in $C[\texttt{Best\_RMSE}]$ with 10^{-8}
Compute weights: $w_i \leftarrow \frac{1}{\texttt{Best\_RMSE}_i}$
Compute normalization factor: $Z \leftarrow \sqrt{(n)}/\sqrt{\sum_i w_i^2}$ where n denotes the total number of genes
for each stage index i and stage name s in S **do**
    Load expression vector $x \leftarrow \texttt{Read}(F_e/\texttt{stage\_}(i+1).csv)$
    for each timepoint index j and time t in T **do**
        Load timepoint vector $y \leftarrow \texttt{Read}(F_t/\texttt{timepoint\_}t.csv)$
        Load first derivative vector $v \leftarrow \texttt{Read}(F_v/\texttt{timepoint\_}t.csv)$
        Load second derivative vector $a \leftarrow \texttt{Read}(F_a/\texttt{timepoint\_}t.csv)$
        $\delta \leftarrow x - y$
        $\|v\|^2 \leftarrow v \cdot v, \quad \|a\|^2 \leftarrow a \cdot a$
        $\hat{r}_v \leftarrow \frac{(v \cdot \delta)}{\|v\|^2} \cdot v \quad \text{if } \|v\|^2 \neq 0$
        $\hat{r}_a \leftarrow \frac{(a \cdot \delta)}{\|a\|^2} \cdot a \quad \text{if } \|a\|^2 \neq 0$
        $r_{\text{hat}} \leftarrow \hat{r}_v + \hat{r}_a$
        Residual vector: $r \leftarrow \delta - r_{\text{hat}}$
        Compute $\Delta t \leftarrow t - t_{\text{prev}}$
        Compute uncertainty threshold $s \leftarrow \left\| v \cdot \Delta t + \frac{1}{2} a \cdot \Delta t^2 \right\|_2$
        $r' \leftarrow \max(0, \|r_{\text{hat}}\| - s)$
        Distance: $D[i,j] \leftarrow \sqrt{\|r\|^2 + r'^2}$
        Scaled Distance: $D_s[i,j] \leftarrow \sqrt{\left(\|r \odot w\|/Z\right)^2 + r'^2}$
    end for
end for
Save D and D_s to CSVs
Return: D, D_s

As before, the clustering-based refinement was applied to these distances as well, ensuring stable and noise-resilient assignments.

Rescaling Distance Based on Curve-Fitting Errors

The fitting process for gene expression trajectories introduces gene-specific errors due to biological discontinuities and technical noise. To account for these, we implemented a rescaling of the relative distances based on the reliability of each gene's fit.

Let x_i be the mean squared curve fitting error (MSE) for gene i, across all timepoints. We computed the root mean squared error (RMSE) to obtain expression-level error magnitudes:

$$z_i = \sqrt{x_i}$$

Step 1: Weighting the Distance Vector. We rescaled each component R_i of the relative vector $\hat{\mathbf{R}}$ by the inverse RMSE of the corresponding gene:

$$\hat{R}'_i = \frac{R_i}{z_i} = w_i R_i, \quad \text{where} \quad w_i = \frac{1}{z_i}$$

This step reduced the influence of poorly fitted genes on the overall distance computation.

Step 2: Normalizing the Weighted Distance. To ensure comparability across genes and samples, the weighted vector was normalized by a factor that preserved the ℓ_2 magnitude relative to the number of genes:

$$\hat{\mathbf{d}} = \frac{\sqrt{n}}{\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}} \cdot \hat{\mathbf{R}}'$$

Here, n denotes the total number of genes. The final rescaled distance was then given by:

$$d = \|\hat{\mathbf{d}}\|_2$$

This formulation ensured that contributions from each gene were proportionally adjusted based on their individual curve-fitting accuracy. By incorporating gene-specific fitting errors into the distance computation, the stage inference became more robust to noise and variability in expression measurements. This rescaling strategy was uniformly applied to both distance estimation methods — with and without the removal of directional projections — to maintain consistency across analyses.

Thresholding for Clock Uncertainty in Evolution

Biological systems may experience variability in internal timing, leading to sample-specific phase shifts in developmental processes. These shifts, while biologically meaningful, may distort distance-based stage estimation. To mitigate this, we implemented a thresholding strategy to ignore minor temporal shifts within an acceptable uncertainty margin.

We began by isolating the component of the relative vector that lay in the 2D subspace defined by the tangent and acceleration directions at $\hat{\mathbf{p}}_k$:

$$\hat{\mathbf{r}} = (\hat{\mathbf{p}}_k - \hat{\mathbf{p}}_0) \cdot \left(\frac{1}{\|\hat{\mathbf{u}}\|^2} \hat{\mathbf{u}}\hat{\mathbf{u}}^\top + \frac{1}{\|\hat{\mathbf{a}}\|^2} \hat{\mathbf{a}}\hat{\mathbf{a}}^\top \right)$$

This component represents deviations along the natural progression of the trajectory and is less informative for detecting genuine off-trajectory movement.

We then defined a temporal uncertainty threshold based on local motion along the trajectory:

$$s = \left\| \hat{\mathbf{u}}\, \Delta t_k + \frac{1}{2} \hat{\mathbf{a}}\, \Delta t_k^2 \right\|_2$$

where $\Delta t_k = t_k - t_{k-1}$ is the local time interval between interpolated points.

Let $\hat{\mathbf{R}}$ denote the orthogonal component of the relative vector (already rescaled as above), and $\hat{\mathbf{r}}$ the projection within the tangent–acceleration plane. If the magnitude of $\hat{\mathbf{r}}$ lies within the threshold s, we treat it as negligible and zero it out. Otherwise, we subtract s and retain the residual:

$$r' = \max\{0, (\|\hat{\mathbf{r}}\|_2 - s)\}$$

Final Corrected Distance. The complete corrected distance from $\hat{\mathbf{p}}_0$ to $\hat{\mathbf{p}}_k$ was then computed as:

$$\text{Distance} = \sqrt{d^2 + r'^2}$$

This formulation combines error-weighted orthogonal deviations with a thresholded projection residual to yield a biologically informed and noise-tolerant estimate of developmental proximity.

Chapter 4

Experimental Results

The high-throughput sequencing data used in this study was obtained from FlyBase, accessible at https://flybase.org, and processed using Python 3.11.7 along with libraries such as `pandas`, `numpy`, `matplotlib`, and others.
The two test datasets were downloaded from the NCBI Gene Expression Omnibus (GEO) database, available at https://www.ncbi.nlm.nih.gov/geo/, with accession IDs: GSE121160 and GSE24324. The preprocessing pipeline was designed to extract only the developmental transcriptome, retaining coding genes while excluding non-coding genes.
All experiments and analyses were carried out in accordance with the methodology described in Chapter 3.

4.1 Curvature of the Data

Euclidean Distance

First, the Euclidean distance between developmental stages was computed using unnormalized gene expression vectors. To visualize the variation in distances across stages, a heatmap was generated, as shown in Fig. 4.3. It is observed that the distances between stages remain relatively uniform, rather than following a smooth progression. This suggests that every stage is approximately equidistant from every other stage, rather than forming a continuous trajectory in gene expression space.

Additionally, the distance curve in Fig. 4.1 and Fig. 4.2 further supports this observation. The lack of a gradual increase or decrease in distances across stages indicates that developmental transitions do not follow a strictly linear or smooth trajectory. Instead, the trajectory of development appears to be inherently curved in the high-dimensional space of gene expression.

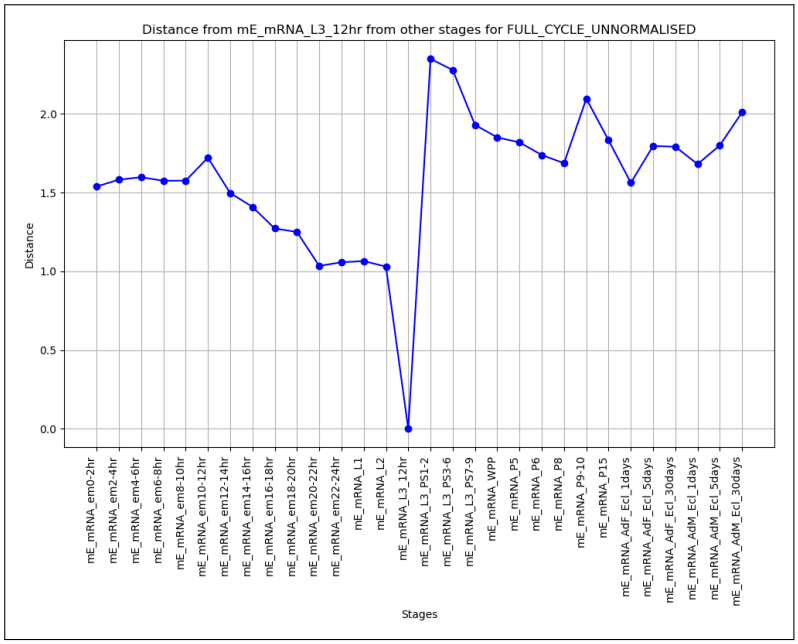

Figure 4.1: Distance of a selected stage(15) from other developmental stages using unnormalized gene expression vectors.

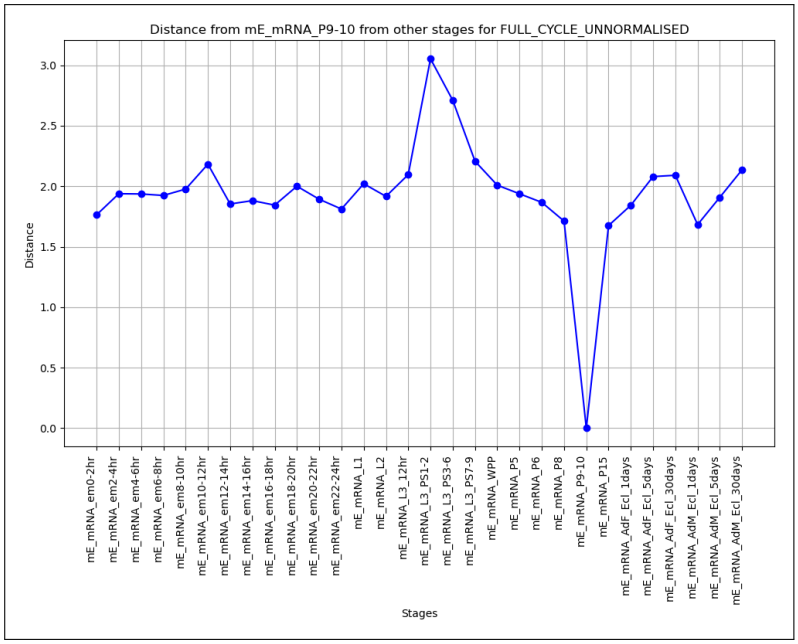

Figure 4.2: Distance of a selected stage(23) from other developmental stages using unnormalized gene expression vectors.

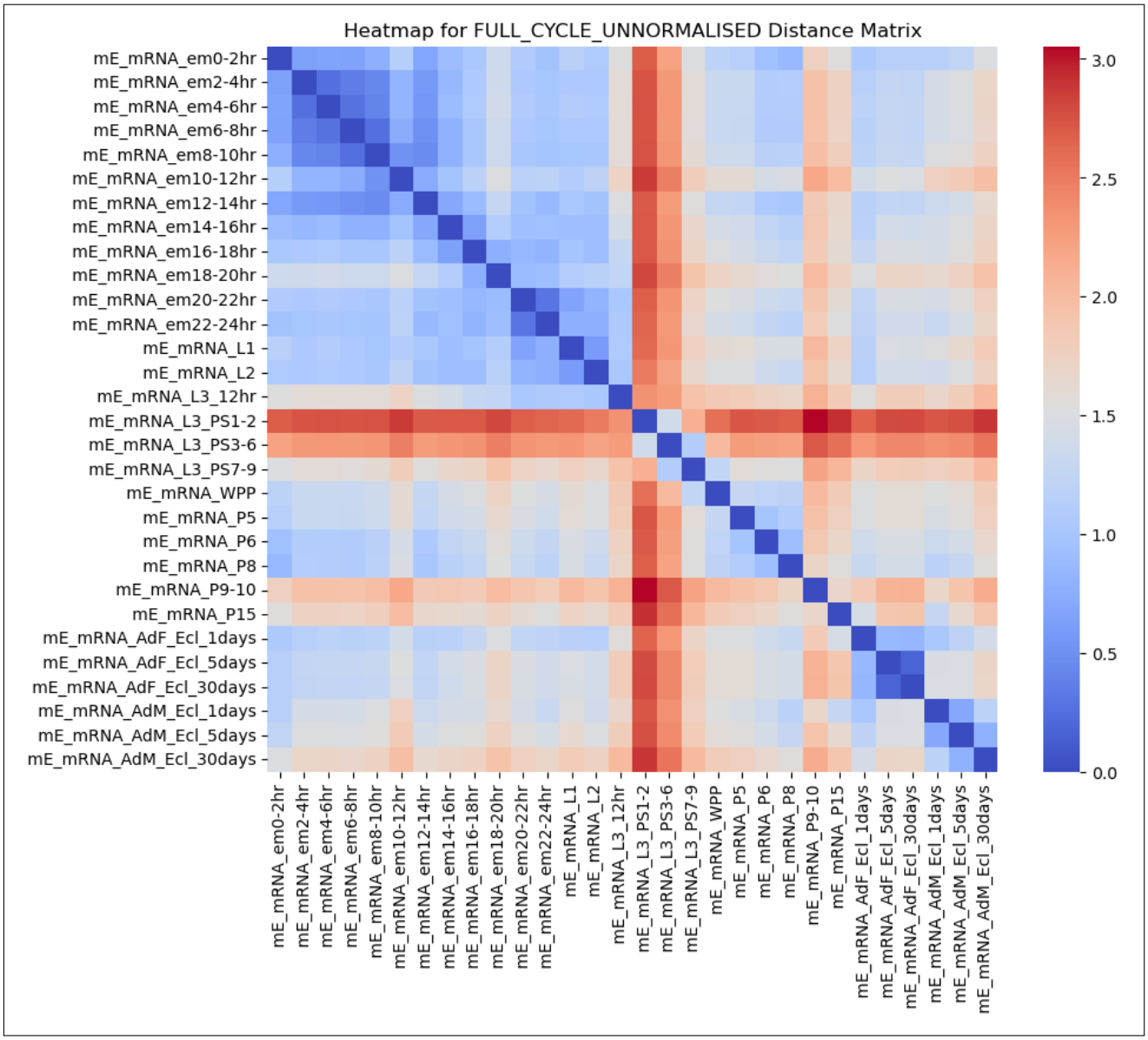

Figure 4.3: Heatmap of Euclidean distances between developmental stages using unnormalized gene expression vectors.

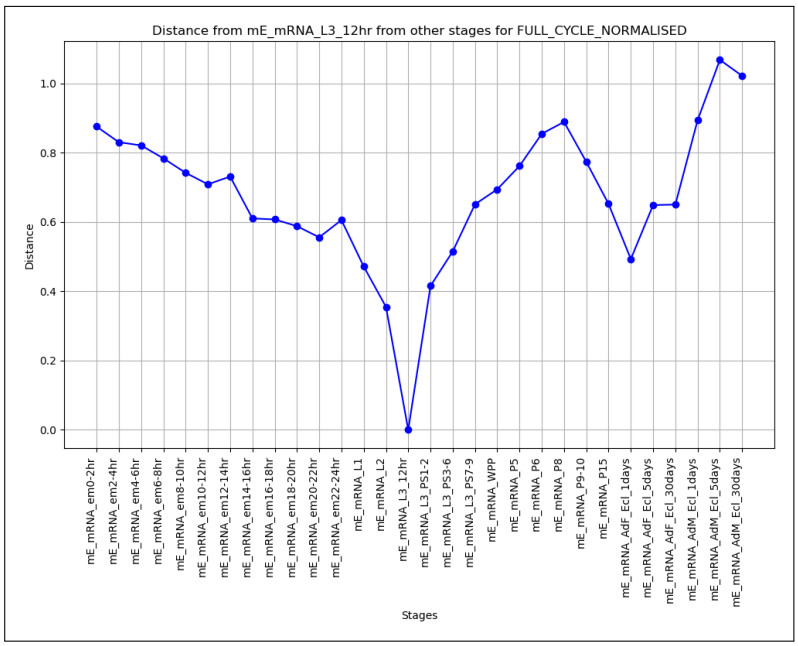

Figure 4.4: Distance of a selected stage(15) from other developmental stages using normalized gene expression vectors.

To further analyze the structure of developmental transitions, we used normalized gene expression vectors, as shown in Fig. 4.6. Normalization reduces scale-dependent variations while preserving the overall structure of stage-wise distances. The heatmap reveals that early embryonic stages (0-12 hr) still show relatively uniform distances, but later developmental transitions—particularly those associated with larval-to-adult transitions—exhibit larger variations.

Figures 4.4 and 4.5 reinforce that while some nearby stages remain relatively close, transitions between major developmental phases show sharp discontinuities. This suggests that gene expression dynamics do not follow a strictly linear trajectory but instead exhibit phase shifts at critical stages.

Co-expression Analysis

To further investigate the structure of developmental transitions, we computed the outer product of normalized gene expression vectors. This analysis helps in understanding co-expression patterns across developmental stages. As shown in Fig. 4.8, the results are largely consistent with those obtained from the Euclidean distance matrices of normalized vectors, reinforcing the presence of curvature in gene expression space.

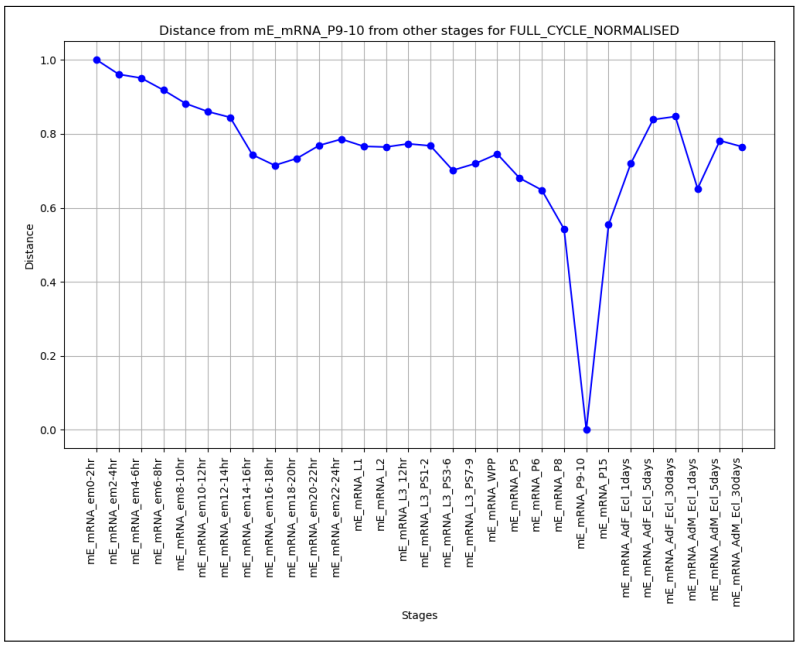

Figure 4.5: Distance of a selected stage(23) from other developmental stages using normalized gene expression vectors.

The co-expression distance curve (Fig. 4.7) again highlights that while some stages exhibit lower distances, major developmental transitions show discontinuities. This suggests that gene interactions undergo nonlinear changes over time, which cannot be fully captured by simple linear distance metrics.

Dimensionality Reduction

Principal Component Analysis (PCA) was applied to reduce the dimensionality of the gene expression dataset while preserving as much variance as possible. The goal was to project the high-dimensional data onto a lower-dimensional space for better visualization and analysis.

Fig.4.9 presents the PCA projection of different developmental stages in the maximum variance 2D space. Each point corresponds to a stage in the dataset, with colors indicating the stage number. The distribution of stages in the projected space does not exhibit clear clustering or interpretable patterns. The PCA projection shows a widely dispersed pattern with no discernible clusters corresponding to biological stages suggesting that the first two principal components may not fully capture a structured separation of developmental stages.

Possible reasons for this include the presence of nonlinear relationships that PCA cannot cap-

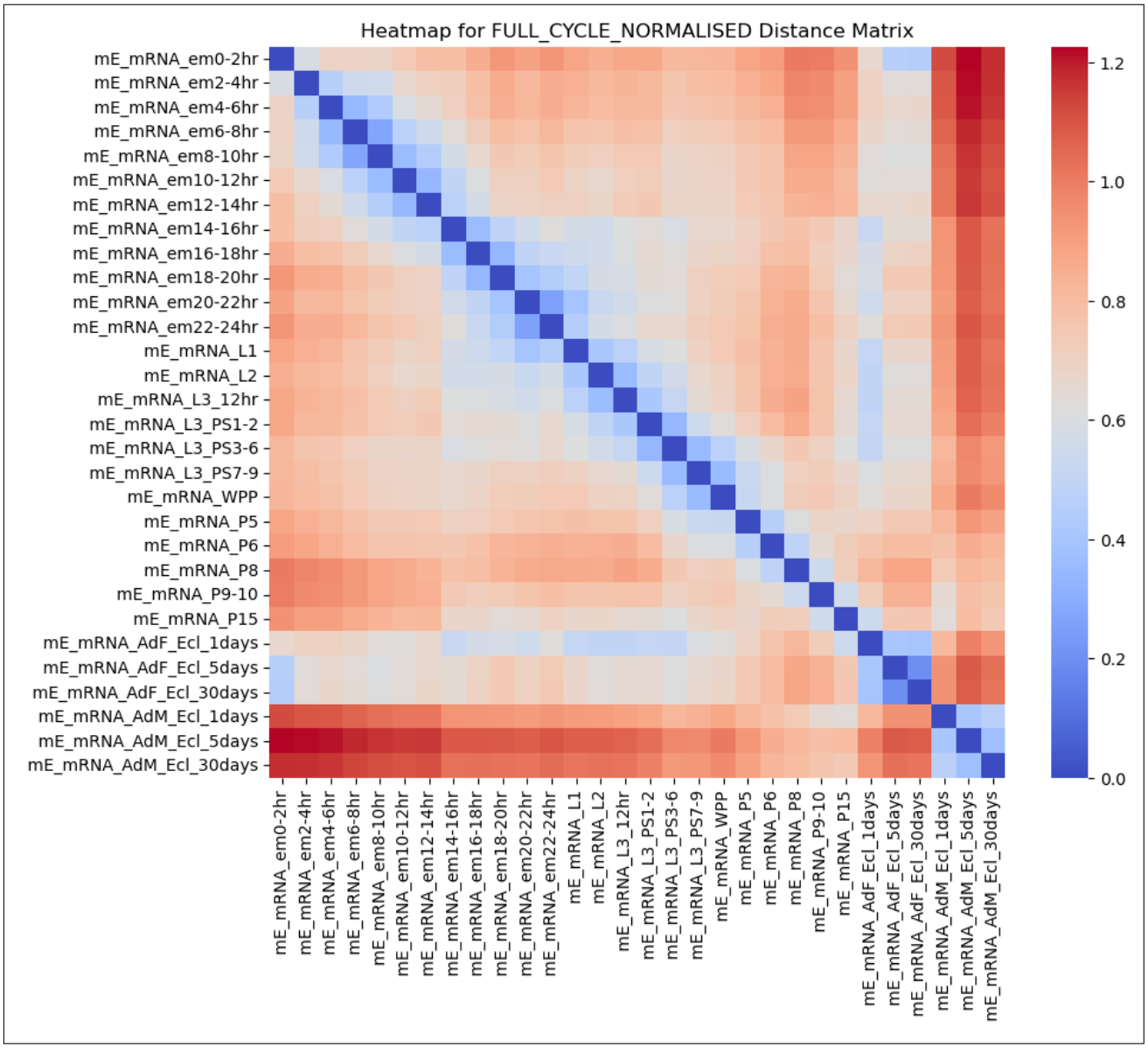

Figure 4.6: Heatmap of Euclidean distances between developmental stages using normalized gene expression vectors.

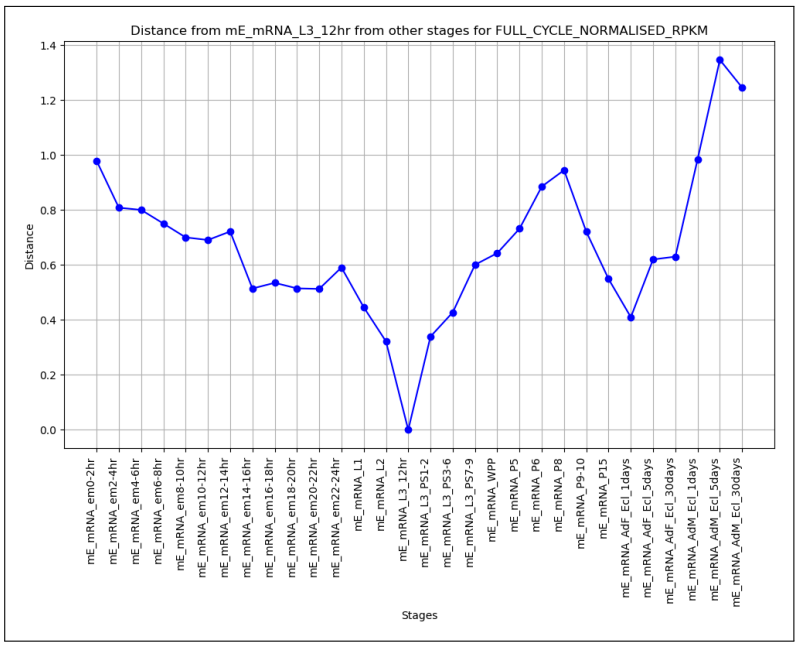

Figure 4.7: Distance of a selected stage(15) from other developmental stages using co-expression matrices.

ture, key biological variations residing in higher-order components, or inherent variability in gene expression across stages. Despite its linear nature, PCA provides a useful first approximation for visualizing global structure and variance in the dataset.

To determine the optimal number of principal components to retain, we analyzed the explained variance ratio, as shown in Fig.4.10. The first principal component captures approximately 60% of the variance, while subsequent components contribute decreasing amounts. The scree plot guided the selection of an appropriate number of principal components that balance variance retention with dimensionality reduction.

All the above experiments suggest a highly curved trajectory of the developmental cycle of *Drosophila melanogaster* in a high-dimensional space. This highlights the need for new trajectory-based methods that account for such curvature. Both test datasets were also evaluated and exhibited similar patterns, revealing highly curved trajectories, as detailed in Appendix.

4.2 High-Dimensional Smooth Curve Fitting

To model the developmental trajectory in a 13,639-dimensional space, we fitted 13,639 independent polynomial curves—each corresponding to a single gene's expression trajectory over developmental time as shown in Fig. 4.11. The degree of each polynomial was allowed to vary from 5 to 8, or until the root mean square error (RMSE) dropped below 0.08, which corresponds

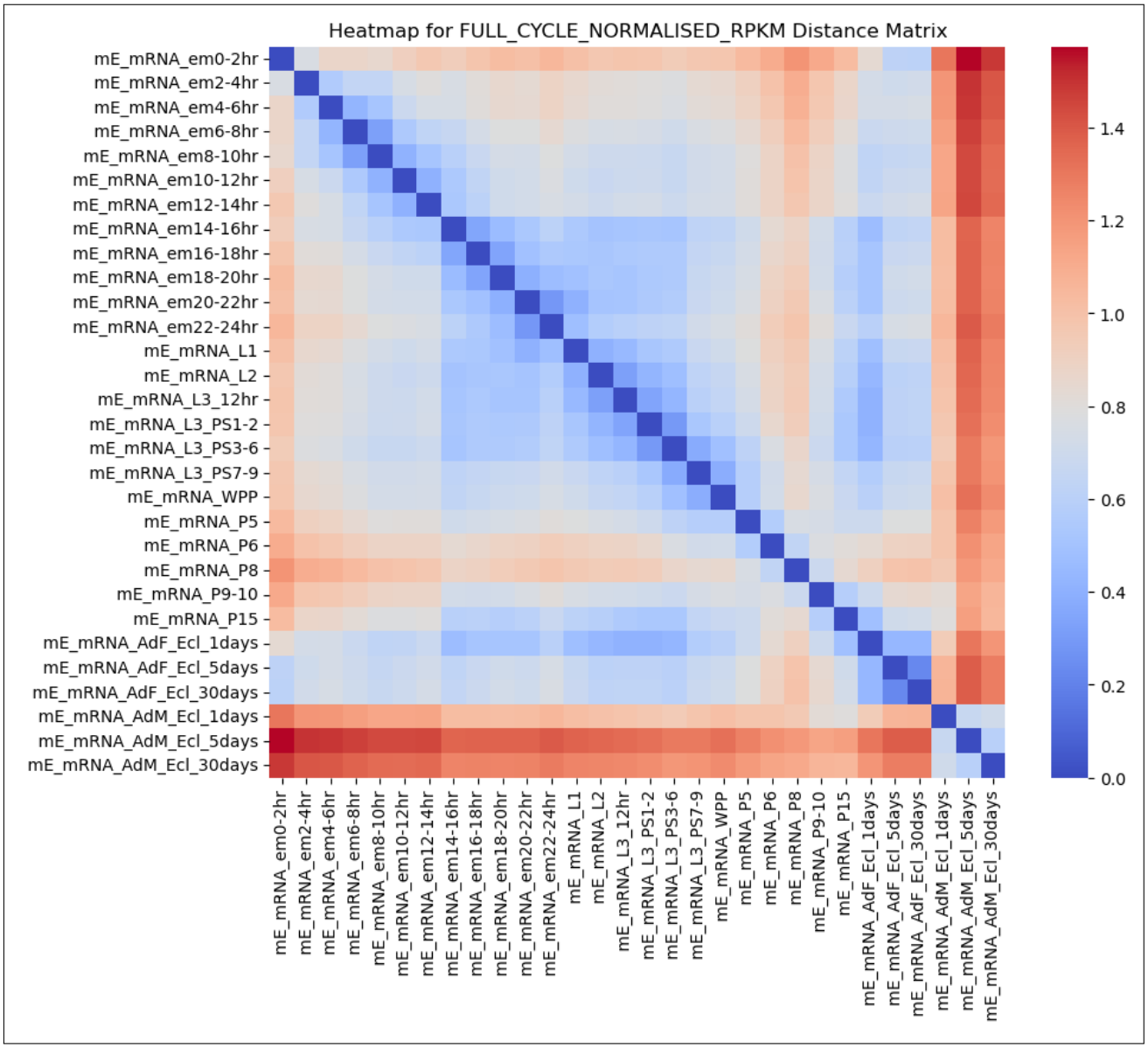

Figure 4.8: Heatmap of co-expression patterns obtained from the outer product of normalized gene expression vectors.

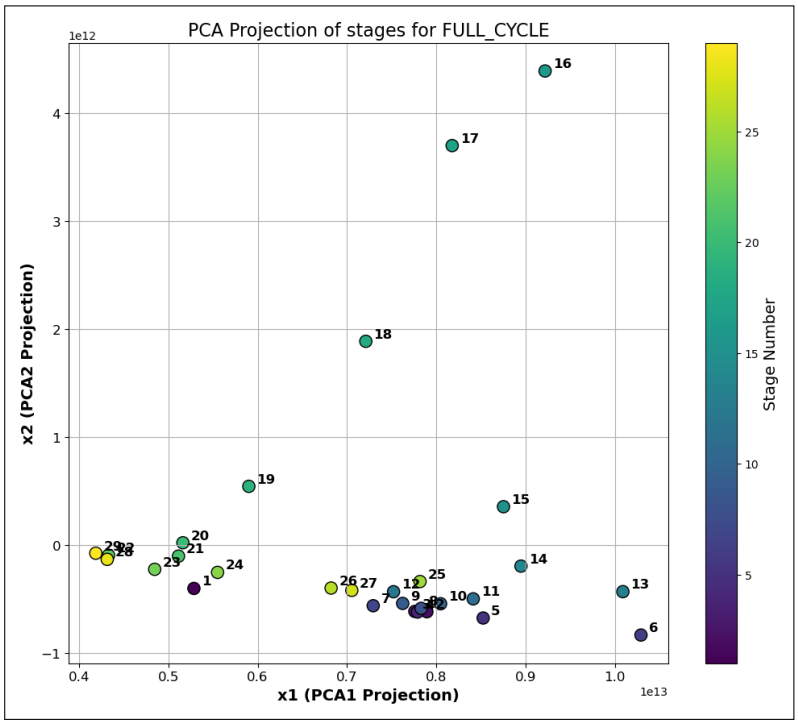

Figure 4.9: PCA projection of different developmental stages for the full cycle dataset in the maximum variance 2D space.

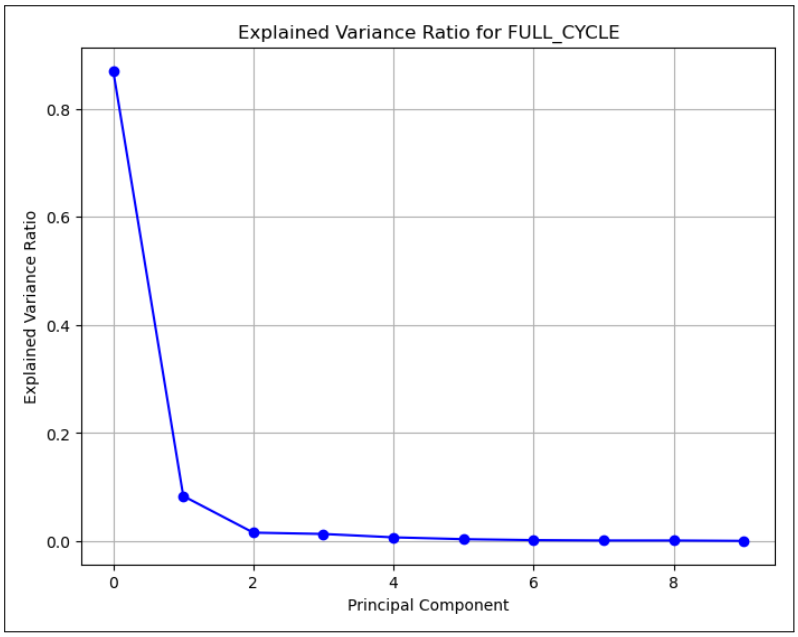

Figure 4.10: Explained variance ratio of principal components in PCA for the full cycle dataset.

to a maximum allowable deviation of 8% from actual expression levels, whichever occurred first. The coefficients obtained from these polynomial fits define the shape of the curve for each gene.

The high-dimensional developmental trajectory is then represented as:

$$\mathbf{D}(t) = [x_1(t), x_2(t), \ldots, x_{13639}(t)] \tag{4.1}$$

where $\mathbf{D}(t)$ denotes the trajectory in 13,639-dimensional space, and $x_k(t)$ represents the polynomial function fitted for gene k over developmental time t.

This high-dimensional curve is expected to be highly curved or coiled, capturing the intricate gene expression dynamics across the developmental stages of *Drosophila melanogaster*. It serves as a reference trajectory for comparing gene expression states and for measuring distances from new data points in the same high-dimensional space.

The full developmental cycle dataset of *Drosophila melanogaster* was also used to validate the robustness of the fitted curve by selectively omitting certain stages during curve fitting as shown in Fig. 4.12. Specifically, the stages `mE_mRNA_em4-6hr`, `mE_mRNA_em12-14hr`, `mE_mRNA_em20-22hr`, `mE_mRNA_L2`,`mE_mRNA_L3_PS7-9`, and `mE_mRNA_P9-10` were excluded from the fitting process. These stages were then used as test points to evaluate how accurately their developmental time could be recovered (i.e., inverted) using the fitted curve as a reference. This simulates how the model generalizes to unseen developmental stages and tests its ability to infer temporal positioning from expression data alone.
A similar procedure was applied to the embryogenesis test dataset, where the time points `03h`, `12h`, and `18h` were omitted during the curve fitting as shown in Fig. 4.13 and subsequently predicted using the constructed trajectory.
These experiments were designed to validate the capability of the fitted high-dimensional curve to accurately represent the developmental trajectory and to infer missing or unknown time points based on minimal distances in the gene expression space.

4.3 Stage Inversion Results

This section presents the evaluation of stage inversion performance across different datasets and inversion techniques. Mean Percentage Error (MPE) was used as the primary metric to assess the accuracy of inferred timepoints under four methods:

The eight evaluated methods fall into four core categories:

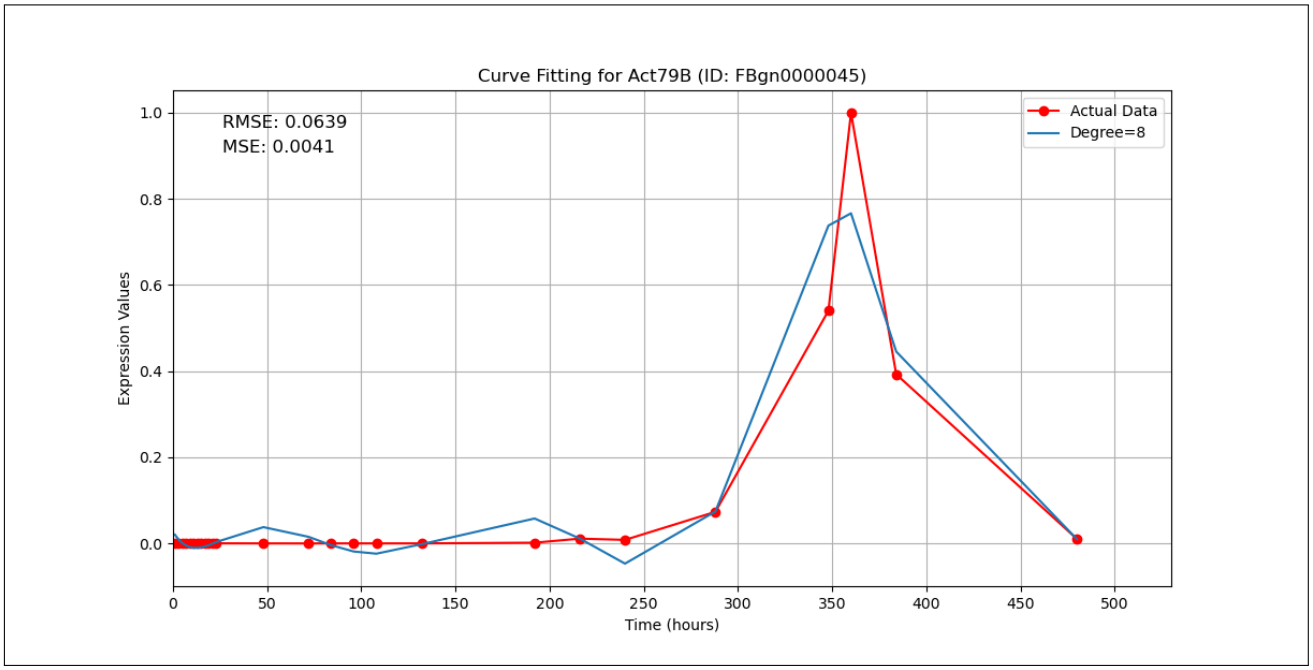

Figure 4.11: Polynomial curve fitting for gene *Argk1* (FBgn0000045) for full cycle dataset by omitting certain stages

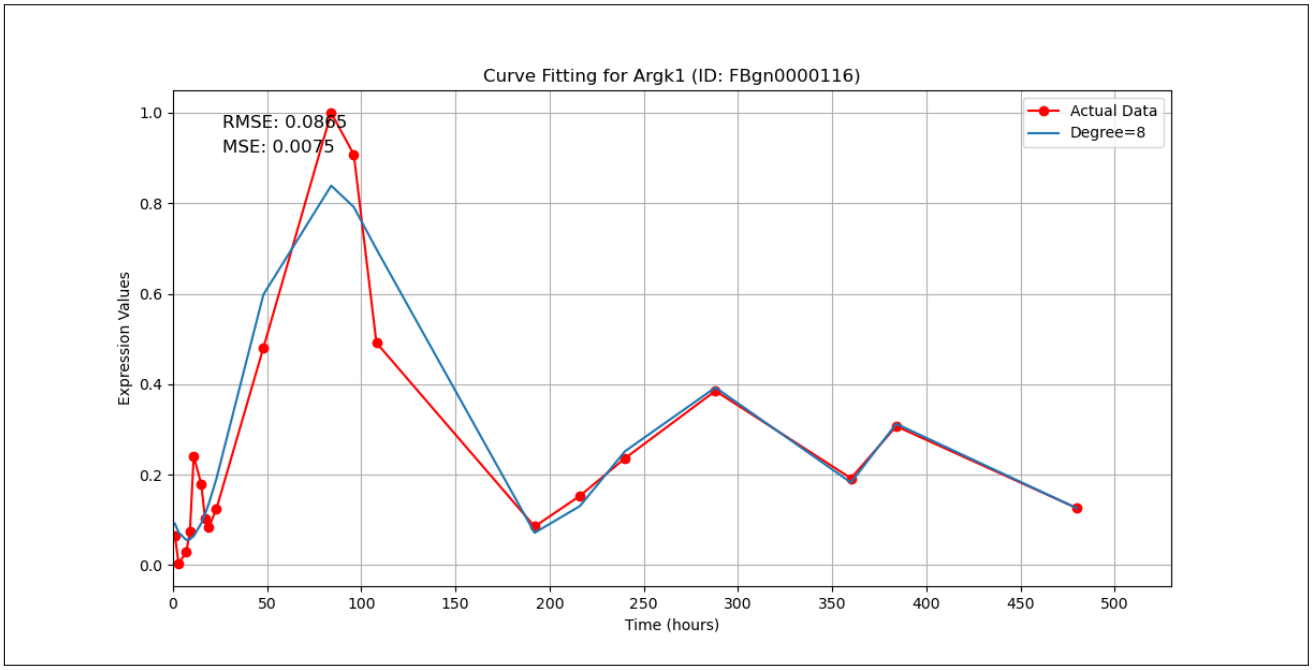

Figure 4.12: Polynomial curve fitting for gene *Argk1* (FBgn00000116) for full cycle dataset by omitting certain stages

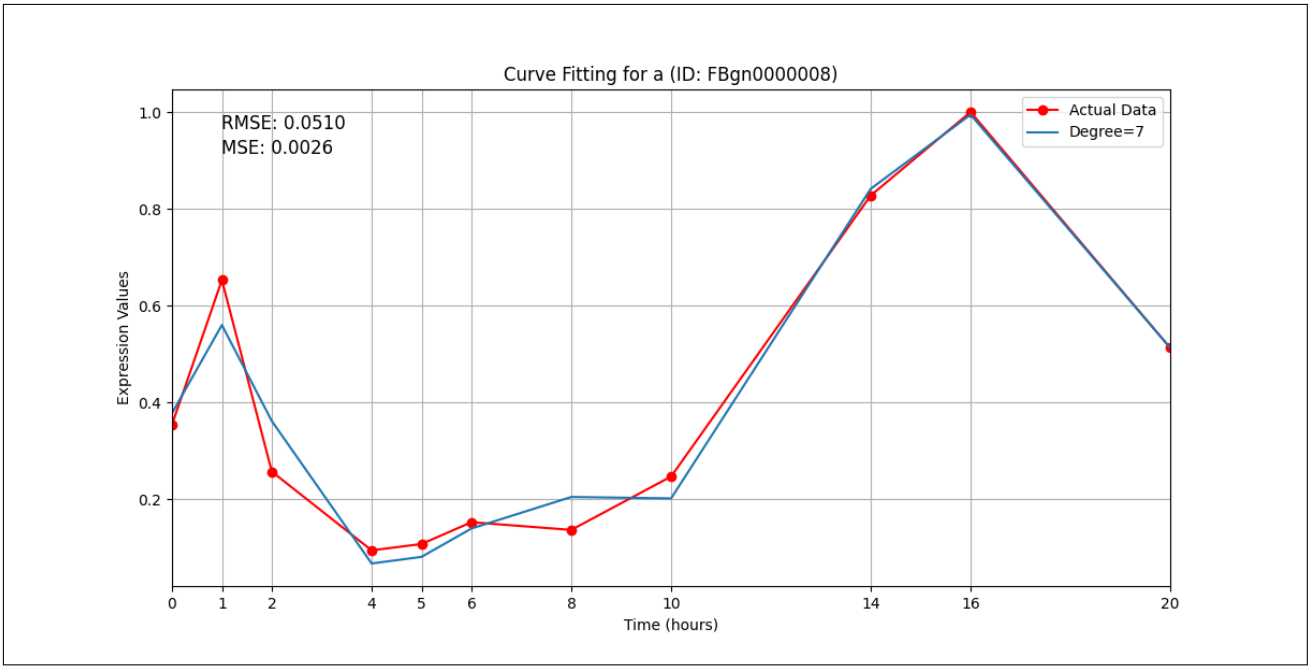

Figure 4.13: Polynomial curve fitting for gene *a* (FBgn0000008) for embryogenesis dataset by omitting certain stages

1. Shortest Distance from Curve
2. Clustered Distance from Curve
3. Shortest Distance with Projection
4. Clustered Distance with Projection

Each of the above was further tested in both unscaled and RMSE-scaled variants.

Full Cycle: Missing Stage Inversion

In the case of missing stages within the Full Cycle dataset (Fig. 4.14), RMSE-scaled methods—particularly the unclustered variant—achieved the lowest mean percentage error (MPE). This demonstrates the advantage of incorporating gene-wise variability in distance computations. While clustering is typically effective in reducing noise, it slightly increased errors in this case, especially when combined with RMSE scaling—possibly due to amplified variance after rescaling. Interestingly, removing derivative components before distance computation did not significantly increase error, indicating robustness of the core signal. However, methods combining projection and RMSE scaling showed very high relative error and are thus omitted from the plot.

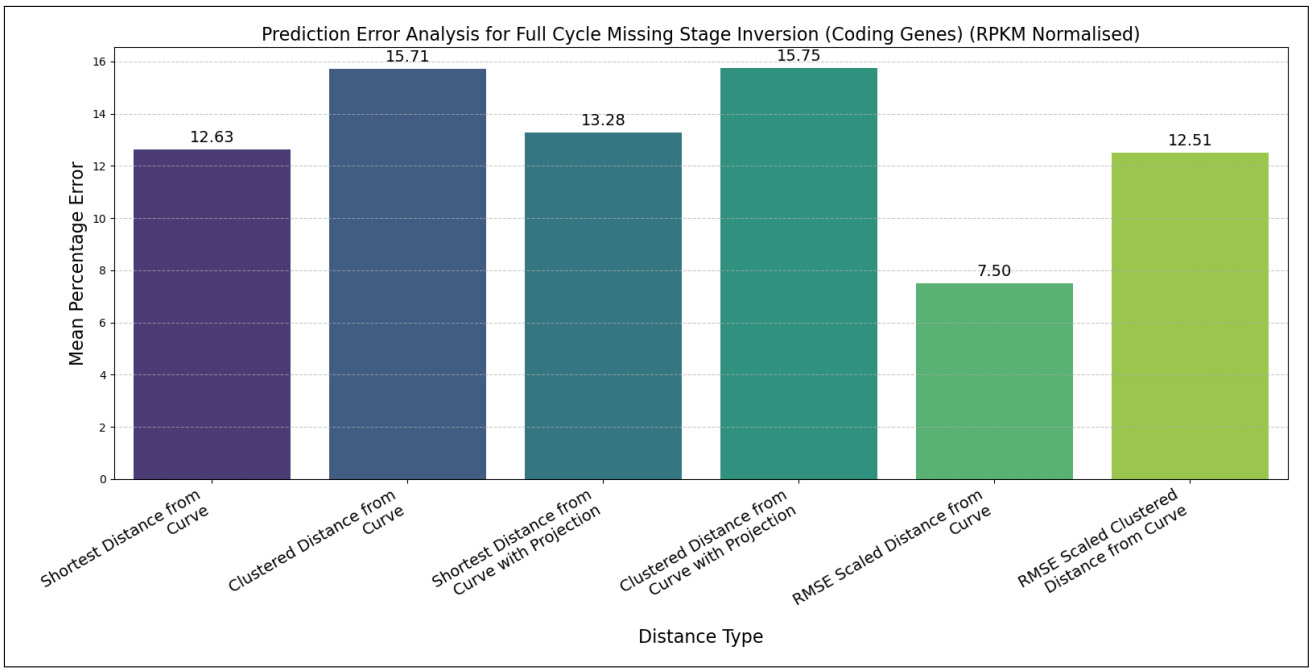

Figure 4.14: Mean Percentage Error comparison across inversion techniques for missing stages in the Full Cycle dataset.

Table 4.1 presents the predicted timepoints using the best-performing method—RMSE-scaled shortest distance—and compares them to actual stage time mappings. **Note:** The following stages were excluded from the curve fitting process and are highlighted in light gray: `mE_mRNA_em4-6hr`, `mE_mRNA_em12-14hr`, `mE_mRNA_em20-22hr`, `mE_mRNA_L2`, `mE_mRNA_L3_PS7-9`, and `mE_mRNA_P9-10`.

Embryo: Missing Stage Inversion

In the case of missing stages within the Embryogenesis dataset (Fig. 4.15), RMSE-scaled methods—particularly the unclustered variant—again achieved the lowest mean percentage error (MPE). This reinforces the advantage of incorporating gene-wise variability when computing distances in high-dimensional expression space. Notably, clustering without scaling resulted in significantly higher error, suggesting that while curvature alignment is important, it must be tempered by normalization to avoid being dominated by genes with larger variances.

As in the Full Cycle scenario, removing derivative components prior to inversion did not substantially degrade performance, indicating the robustness of the core expression signal. Also, as in the Full Cycle dataset, projection-based methods combined with RMSE scaling consistently underperformed and were therefore excluded from the comparison.

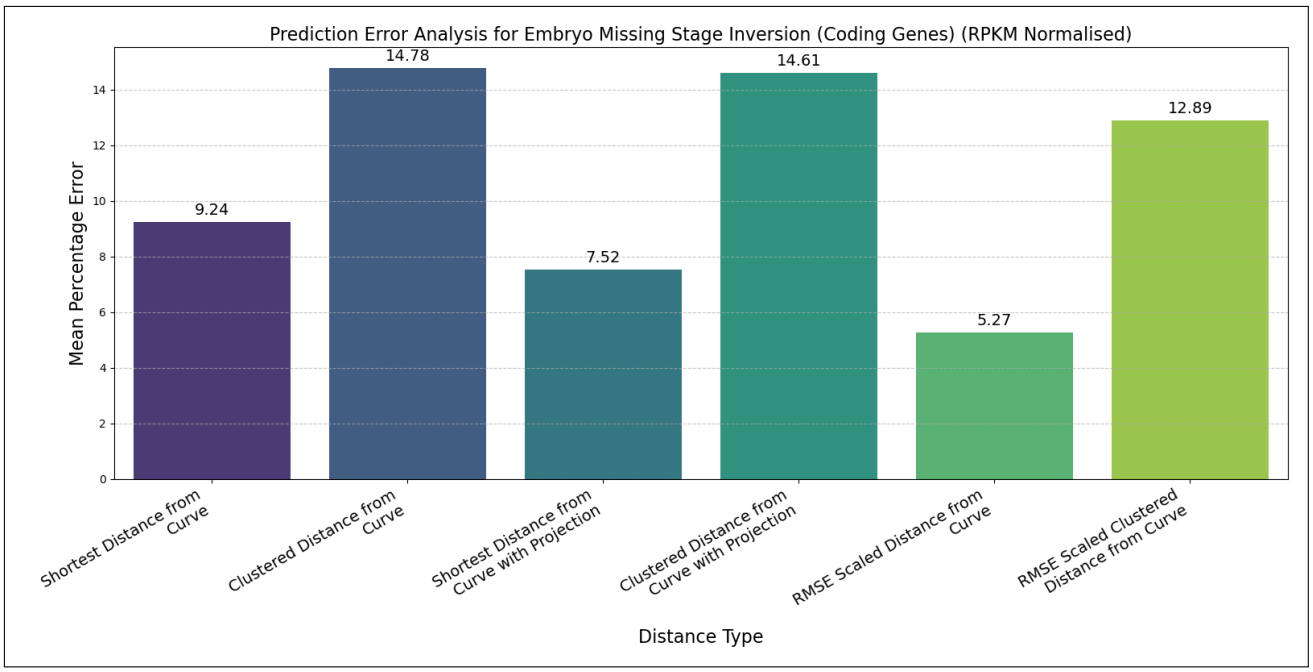

Figure 4.15: Mean Percentage Error comparison for missing stage inversion in the Embryogenesis dataset.

Table 4.2 presents the predicted timepoints using the best-performing method—RMSE-scaled shortest distance—and compares them to actual stage time mappings. **Note:** The following stages were excluded from the curve fitting process and are highlighted in light gray: `03h`, `12h`, and `18h`.

Embryo–Full Cycle Inter-Dataset Inversion

To evaluate cross-dataset temporal alignment, we performed inter-dataset inversion by using Embryogenesis as the query and Full Cycle as the reference dataset. A pronounced contrast in performance was observed across different distance metrics, particularly highlighting the limitations of RMSE-scaled distances. These scaled metrics consistently underperformed, revealing that gene-wise RMSE normalization fails to generalize effectively when applied across datasets originating from distinct sequencing platforms or experimental conditions. The normalization introduces disproportionate weighting of gene expression dimensions, leading to reduced inversion accuracy and poor temporal mapping.

In contrast, the unscaled-clustered distance metric yielded notably better results. By retaining the original expression scale while incorporating gene-wise structural relationships, this approach achieved more accurate and robust alignment between datasets. This indicates that

preserving absolute expression levels, rather than enforcing uniform scaling, is critical for maintaining biological signal consistency across heterogeneous data sources.

To assess the influence of RNA-seq quantification schemes, we conducted inversion analysis using both TPM and RPKM normalization. As shown in Figures 4.16 and 4.17, the mean percentage error (MPE) profiles under both normalization methods were highly comparable, with only minor differences observed. This consistency is further reflected in the predicted timepoints listed in Tables 4.3 and 4.4, where the inversion trajectories closely follow the expected developmental timeline across stages. For example, under TPM normalization (Table 4.3), the predicted time for the 10h stage is 8.0, while for RPKM (Table 4.4), it is 8.6, both showing tight agreement with the actual value. These results suggest that inter-dataset inversion performance is largely robust to the choice of normalization strategy and that biologically meaningful mappings can be preserved across commonly used quantification approaches.

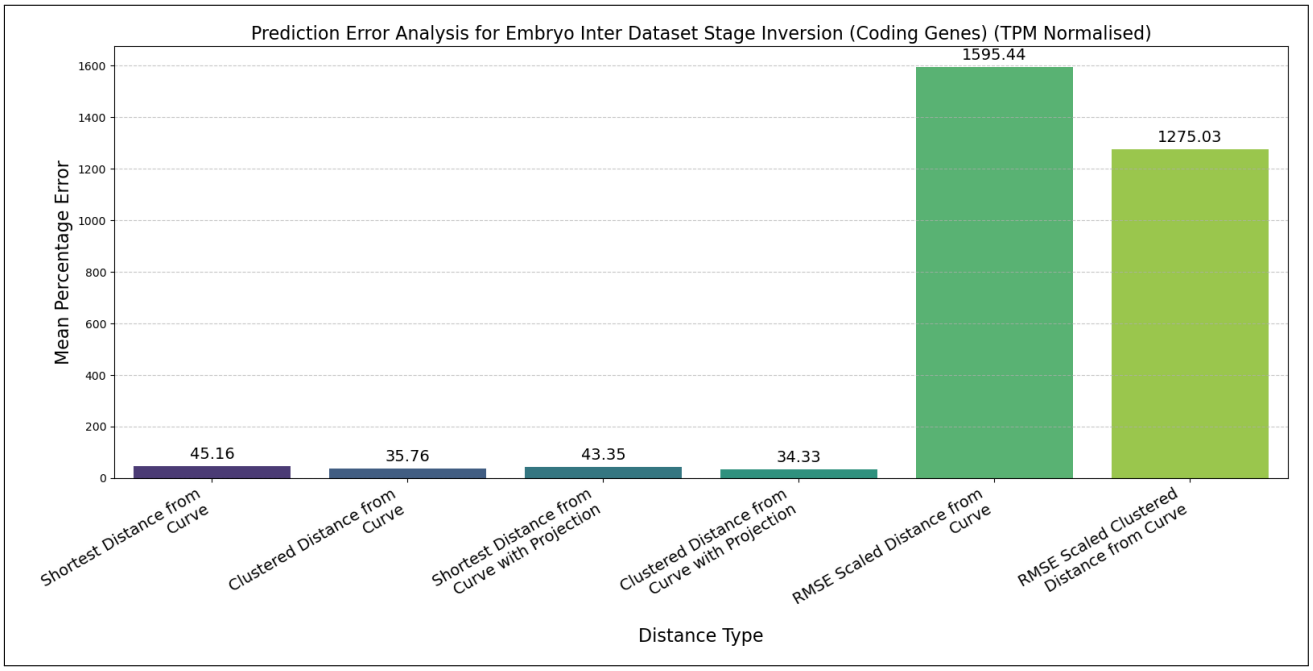

Figure 4.16: Mean Percentage Error comparison for inter-dataset inversion from Embryogenesis to Full Cycle reference (TPM).

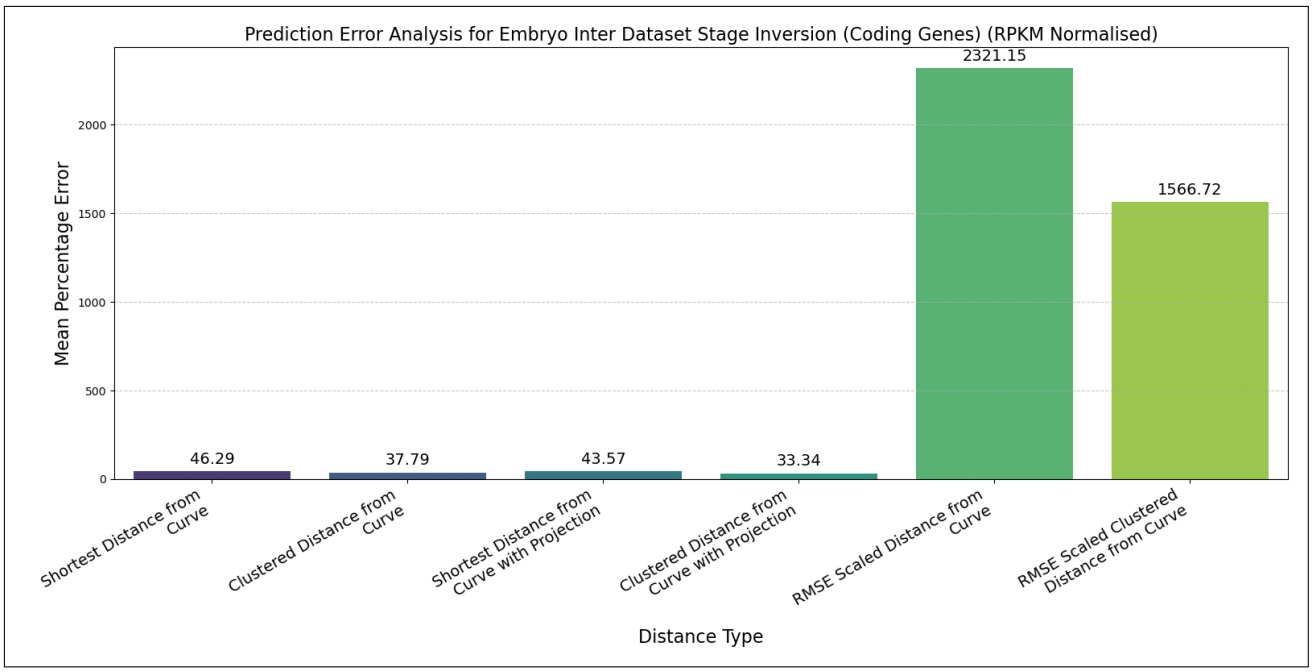

Figure 4.17: Mean Percentage Error comparison for inter-dataset inversion from Embryogenesis to Full Cycle reference (RPKM).

GSE24324–Full Cycle Inter-Dataset Inversion

For the GSE24324 dataset (Fig. 4.18), which shares sequencing equipment and preprocessing protocols with the Full Cycle dataset, all inversion methods yield comparable performance. This consistency indicates that when experimental conditions are matched, normalization using gene-wise RMSE does not introduce significant distortion. In such settings, all distance metrics—whether normalized or unnormalized—perform similarly, as evidenced by the comparable Mean Percentage Error (MPE) trends in both TPM (Fig. 4.18) and RPKM (Fig. 4.19) scales. The timepoint mappings derived from minimum distance values for each developmental stage are presented in Tables 4.5 and 4.6.

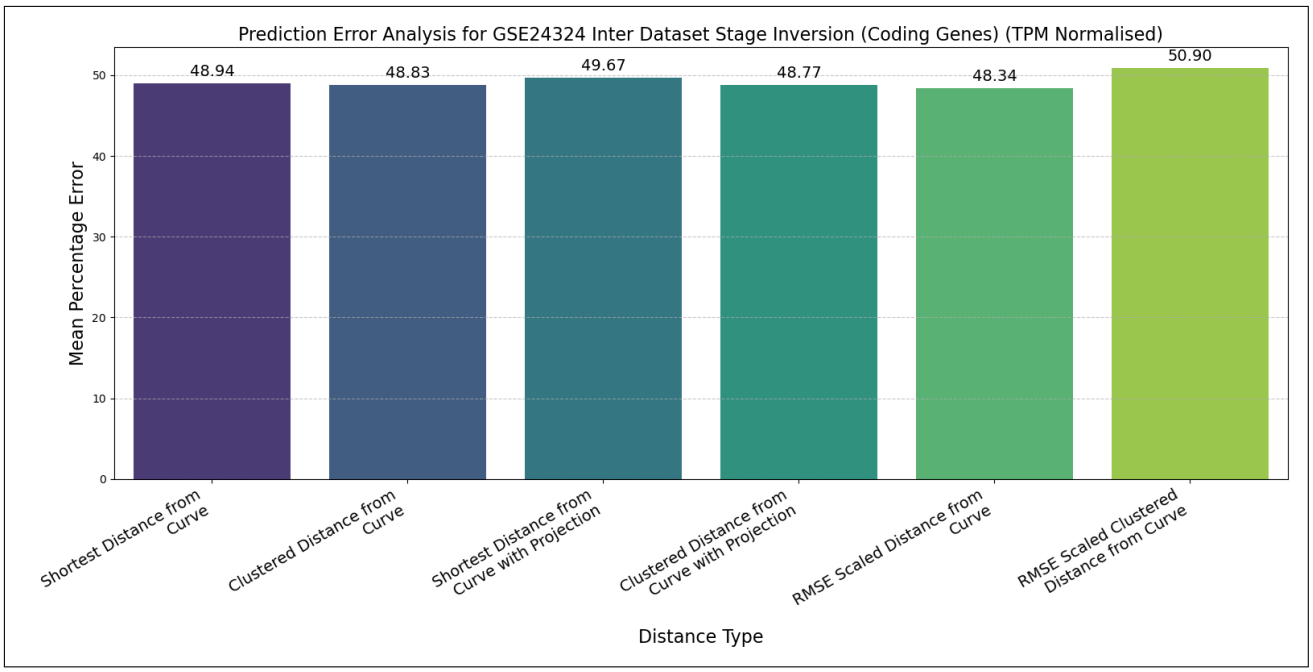

Figure 4.18: Mean Percentage Error comparison for inter-dataset inversion from GSE24324 to Full Cycle reference (TPM).

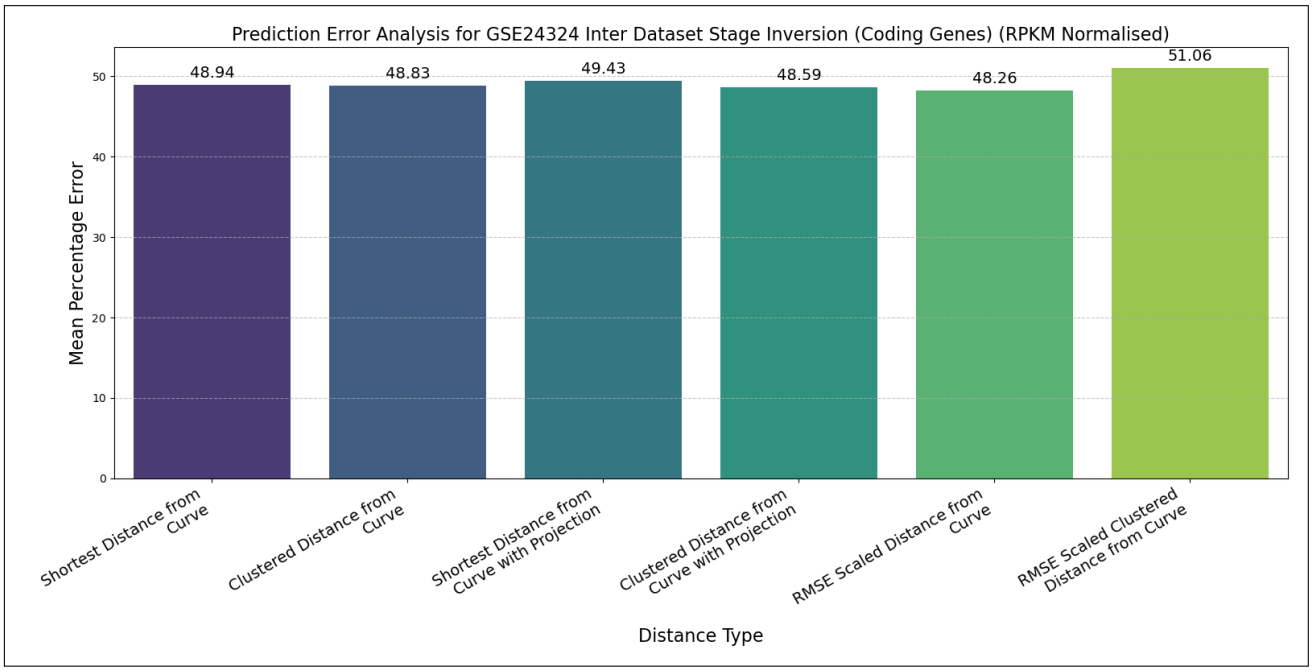

Figure 4.19: Mean Percentage Error comparison for inter-dataset inversion from GSE24324 to Full Cycle reference (RPKM).

Stage	Actual Time Mapping	Predicted Timepoint (RMSE Scaled Distance)
mE_mRNA_em0-2hr	1	1.0
mE_mRNA_em2-4hr	3	3.0
mE_mRNA_em4-6hr	5	5.2
mE_mRNA_em6-8hr	7	6.8
mE_mRNA_em8-10hr	9	7.8
mE_mRNA_em10-12hr	11	10.4
mE_mRNA_em12-14hr	13	11.8
mE_mRNA_em14-16hr	15	15.4
mE_mRNA_em16-18hr	17	18.4
mE_mRNA_em18-20hr	19	23.0
mE_mRNA_em20-22hr	21	25.5
mE_mRNA_em22-24hr	23	25.5
mE_mRNA_L1	48	50.4
mE_mRNA_L2	72	75.6
mE_mRNA_L3_12hr	84	80.4
mE_mRNA_L3_PS1-2	96	98.4
mE_mRNA_L3_PS3-6	108	110.4
mE_mRNA_L3_PS7-9	132	186.0
mE_mRNA_WPP	192	186.0
mE_mRNA_P5	216	213.6
mE_mRNA_P6	240	240.0
mE_mRNA_P8	288	283.2
mE_mRNA_P9-10	348	318.0
mE_mRNA_P15	360	355.2
mE_mRNA_AdF_Ecl_1days	384	384.0
mE_mRNA_AdF_Ecl_5days	480	384.0

Table 4.1: Actual vs. predicted timepoints using RMSE-scaled shortest distance for Intra-Stage Inversion (Full Cycle Dataset). Stages excluded from curve fitting are shaded.

Stage	Actual Time Mapping	Predicted Timepoint (RMSE Scaled Distance)
00h	0	0.0
01h	1	0.9
02h	2	2.0
03h	3	3.6
04h	4	4.2
05h	5	5.0
06h	6	6.0
08h	8	7.8
10h	10	10.0
12h	12	10.0
14h	14	14.0
16h	16	16.0
18h	18	15.6
20h	20	19.8

Table 4.2: Actual vs. predicted timepoints using RMSE-scaled shortest distance for Intra-Stage Inversion (Embryogenesis Dataset). Stages excluded from curve fitting are shaded.

Stage	Actual Time Mapping	Predicted Timepoint (Unscaled Clustered Distance)
00h	0	3.2
01h	1	2.8
02h	2	3.8
03h	3	4.8
04h	4	5.0
05h	5	5.4
06h	6	6.2
08h	8	6.6
10h	10	8.0
12h	12	9.2
14h	14	12.6
16h	16	15.8
18h	18	15.4
20h	20	22.4

Table 4.3: Predicted timepoints using Unscaled Clustered Distance for Embryo to Full Cycle inter-dataset inversion (TPM).

Stage	Actual Time Mapping	Predicted Timepoint (Unscaled Clustered Distance)
00h	0	3.2
01h	1	2.6
02h	2	3.8
03h	3	5.2
04h	4	5.4
05h	5	5.8
06h	6	6.6
08h	8	6.6
10h	10	8.6
12h	12	10.2
14h	14	13.4
16h	16	16.8
18h	18	16.0
20h	20	28.0

Table 4.4: Predicted timepoints using Unscaled Clustered Distance for Embryo to Full Cycle inter-dataset inversion (RPKM).

Stage	Actual Time Mapping	Predicted Timepoint (RMSE Scaled Distance)
E2-4hr	4	1.6
E2-16hr	9	5.6
E2-16hr100	9	18.6
E2-16hr_avg	9	11.0
E14-16hr	16	20.2
Larva	72	74.4
L3i	96	132.0
L3i100	96	108.0
L3i_avg	96	120.0
Pupa1	144	342.0
P3d	168	273.6
Pupa2	186	348.0
MA3d	240	336.0
FA3d	240	384.0
A17d	408	381.6

Table 4.5: Predicted timepoints using RMSE Scaled Distance for GSE24324 to Full Cycle inter-dataset inversion (TPM).

Stage	Actual Time Mapping	Predicted Timepoint (RMSE Scaled Distance)
E2-4hr	4	1.4
E2-16hr	9	5.6
E2-16hr100	9	18.2
E2-16hr_avg	9	10.8
E14-16hr	16	19.8
Larva	72	75.6
L3i	96	132.0
L3i100	96	110.4
L3i_avg	96	120.0
Pupa1	144	336.0
P3d	168	278.4
Pupa2	186	348.0
MA3d	240	336.0
FA3d	240	384.0
A17d	408	381.6

Table 4.6: Predicted timepoints using RMSE Scaled Distance for GSE24324 to Full Cycle inter-dataset inversion (RPKM).

Chapter 5

Conclusion and Future Work

Gene expression trajectories, particularly during the developmental stages of *Drosophila melanogaster*, exhibit high curvature in high-dimensional space. As a result, traditional similarity measures like Euclidean distance between states become ineffective and often misleading. In this work, we addressed this issue by leveraging known temporal trajectories of gene expression and analyzing distances with respect to the expected parametric curve fitted for each gene.

Conclusion

Our experiments demonstrate several key findings:

- **Intra-dataset inversion (Full Cycle, Embryo)**: When predicting developmental time stamps within the same dataset, additional scaling of distances using the polynomial fitting errors improves the (validation) performance within datasets, but it degrades the inference across datasets with varying measurement techniques. This is likely because the measurement errors potentially captured by the deviation from the smooth trajectory, are consistently captured only within the same experimental setting.

- **Inter-dataset inversion (Embryo → Full Cycle)**: When attempting inversion across datasets originating from different sequencing platforms, normalization based on the polynomial fitting errors proved to ineffective. The underlying gene-wise error distributions in the measurement of expressions vary considerably with techniques and instruments used and hence cannot be translated across datasets.

- **Inter-dataset inversion (GSE24324 → Full Cycle)**: In this case, the use of similar sequencing technologies resulted in reasonably consistent error distributions across

datasets. Consequently, both normalized and un-normalized distances performed comparably.

- **Derivative-based distance with projection removal and thresholding**: To better capture biologically relevant deviations from the expected gene expression trajectory, we removed components of the relative vector along the directions of velocity and acceleration (first and second derivatives) of the trajectory. This ensured that the calculated distances significantly ignore deviations due to any variations in the biological clocks of the process. This method demonstrated performance comparable to simpler metrics, while offering a more nuanced and interpretable approach, and holds promise for further refinement. However, when combined with the RMSE-scaling approach, its performance declined, for reasons that remain unclear and warrant further investigation.

- **Expression normalization units (RPKM vs TPM)**: The choice of expression normalization—RPKM versus TPM—did not produce significant differences in results for our inversion tasks. This suggests that the proposed methods are robust to such preprocessing choices, at least within the datasets (validations) considered.

These observations underscore the importance of dataset compatibility, normalization techniques, and the geometric nature of gene expression trajectories in choosing appropriate distance metrics for developmental stage prediction. This work needs further improvements, experimentation, and broader tests of inference across datasets and conditions.

Future Work

The work presented here opens up multiple avenues for further exploration and refinement. Below are a few specific directions along which the work can be extended.

1. **Improved distance methodologies**: More robust distance metrics can be developed that better incorporate local curvature and dynamics of evolution, noise characteristics, and other biological constraints.

2. **Dimensionality reduction and gene selection**: Clustering genes based on co-expression patterns, functional similarity, or temporal synchrony may help reduce noise and improve trajectory modeling. Alternatively, gene selection can be guided by statistical properties — such as selecting genes with higher order moments across time points, even without prior knowledge of the biological significance of a gene.

3. **Dataset expansion**: Incorporating more transcriptome datasets from different experimental conditions and sequencing platforms will test the generalizability of the proposed methods.

4. **Cross-organism analysis**: Applying this framework to developmental data from other model organisms may reveal conserved or divergent expression dynamics.

5. **Functional gene classification**: Clustering genes based on known biological functions or pathways could allow for pathway-specific trajectory modeling and analysis.

6. **Abnormality detection**: Extending the method to detect anomalies or disease-related deviations from expected trajectories could have significant applications in biomedical research.

7. **Understanding RMSE-scaling interactions**: Investigating why RMSE-scaling degrades performance when combined with derivative-based distance could shed light on interactions between normalization and trajectory geometry.

Appendix

Curvature of the Embryogenesis Dataset

To validate the curved nature of developmental gene expression trajectories in the Embryogenesis dataset (GSE121160), we repeated the same curvature analyses used for the Full Cycle dataset.

Euclidean Distance (Unnormalized)

We first computed the Euclidean distances between embryonic stages using raw, unnormalized gene expression vectors. As shown in Fig. 5.1, the distance profiles appear mostly flat, with only minor variations in distances from selected stages to others. The corresponding heatmap (Fig. 5.2) confirms this trend—distances across developmental time are largely uniform and do not exhibit a clear temporal progression.

Euclidean Distance (Normalized)

We repeated the analysis using normalized gene expression vectors. As seen in Fig. 5.3, the distance curves continue to show relatively flat behavior. However, slight differentiation appears toward later stages, reflecting minor shifts in expression trends.

The heatmap in Fig. 5.4 shows more structured variation in distances, especially between early and late embryonic stages. Nevertheless, the overall trend remains consistent with the hypothesis that the trajectory is highly curved in gene expression space rather than linearly separable.

Co-expression Analysis

To explore stage-wise co-expression dynamics, we computed the outer product of normalized gene expression vectors across all embryonic stages. This yielded co-expression matrices that reflect gene–gene interactions at each stage.

Fig. 5.5 illustrates the co-expression distance curve from a selected stage. The curve shows

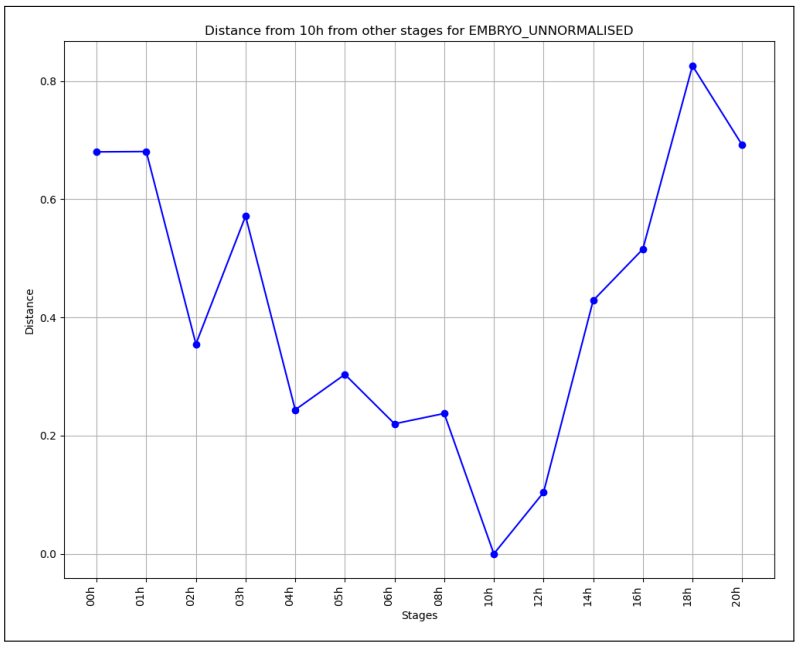

Figure 5.1: Distance of a selected stage (9) from other embryonic stages using unnormalized gene expression vectors.

that while neighboring stages maintain strong similarity, discontinuities appear at certain transitions, indicating nonlinear expression shifts.

The heatmap in Fig. 5.6 visualizes global co-expression patterns and further supports the curved nature of the transcriptomic trajectory, as abrupt changes in co-expression similarity align with known developmental transitions.

Dimensionality Reduction

Principal Component Analysis (PCA) was applied to project the high-dimensional gene expression data into a 2D plane to assess global structure and variance.

As shown in Fig. 5.7, the PCA plot for embryonic stages lacks distinct clustering, with stages scattered broadly. This dispersion supports the hypothesis that developmental transitions are not linearly separable and instead follow a complex, curved trajectory.

To quantify the variance captured by each component, the explained variance ratio was plotted in Fig. 5.8. The first component explains a large portion of the variance, but subsequent components rapidly decline, indicating that meaningful biological variation may be spread across

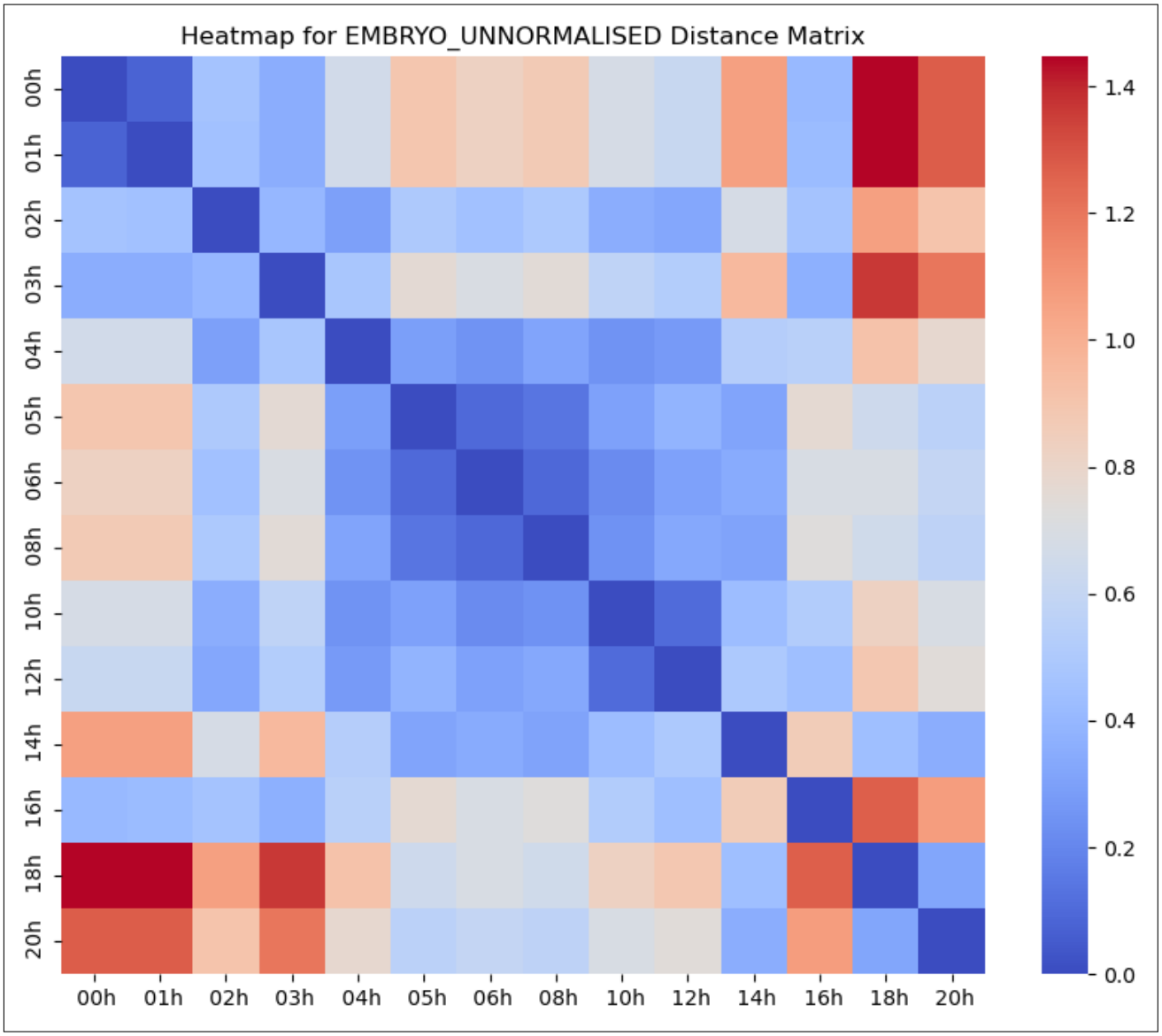

Figure 5.2: Heatmap of Euclidean distances between embryonic stages using unnormalized gene expression vectors.

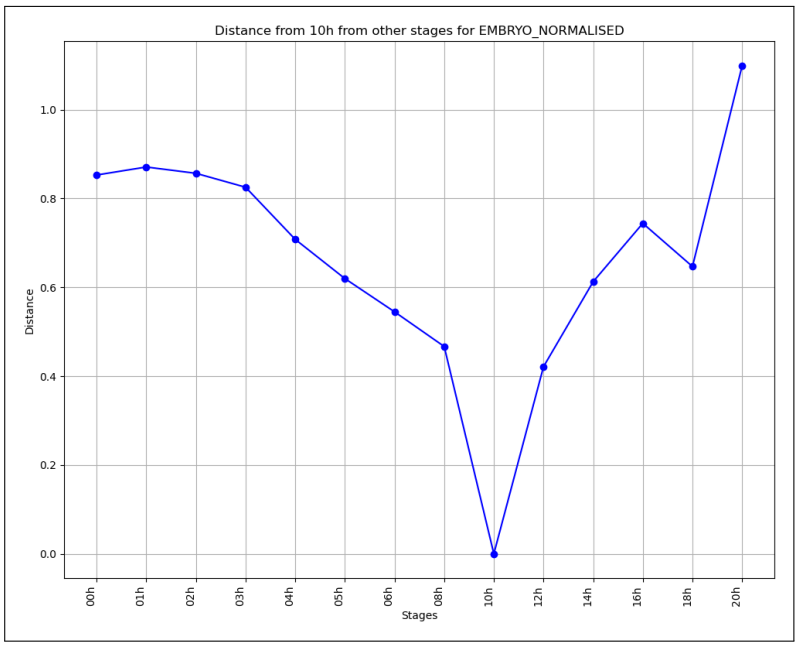

Figure 5.3: Distance of a selected stage (9) from other embryonic stages using normalized gene expression vectors.

many dimensions.

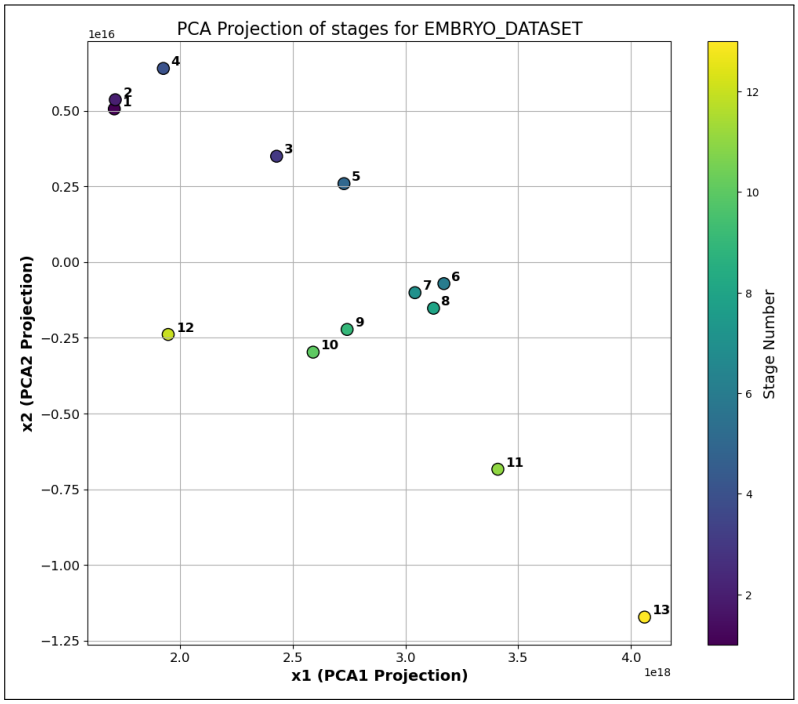

Figure 5.7: PCA projection of embryonic stages in the maximum variance 2D space for the Embryogenesis dataset.

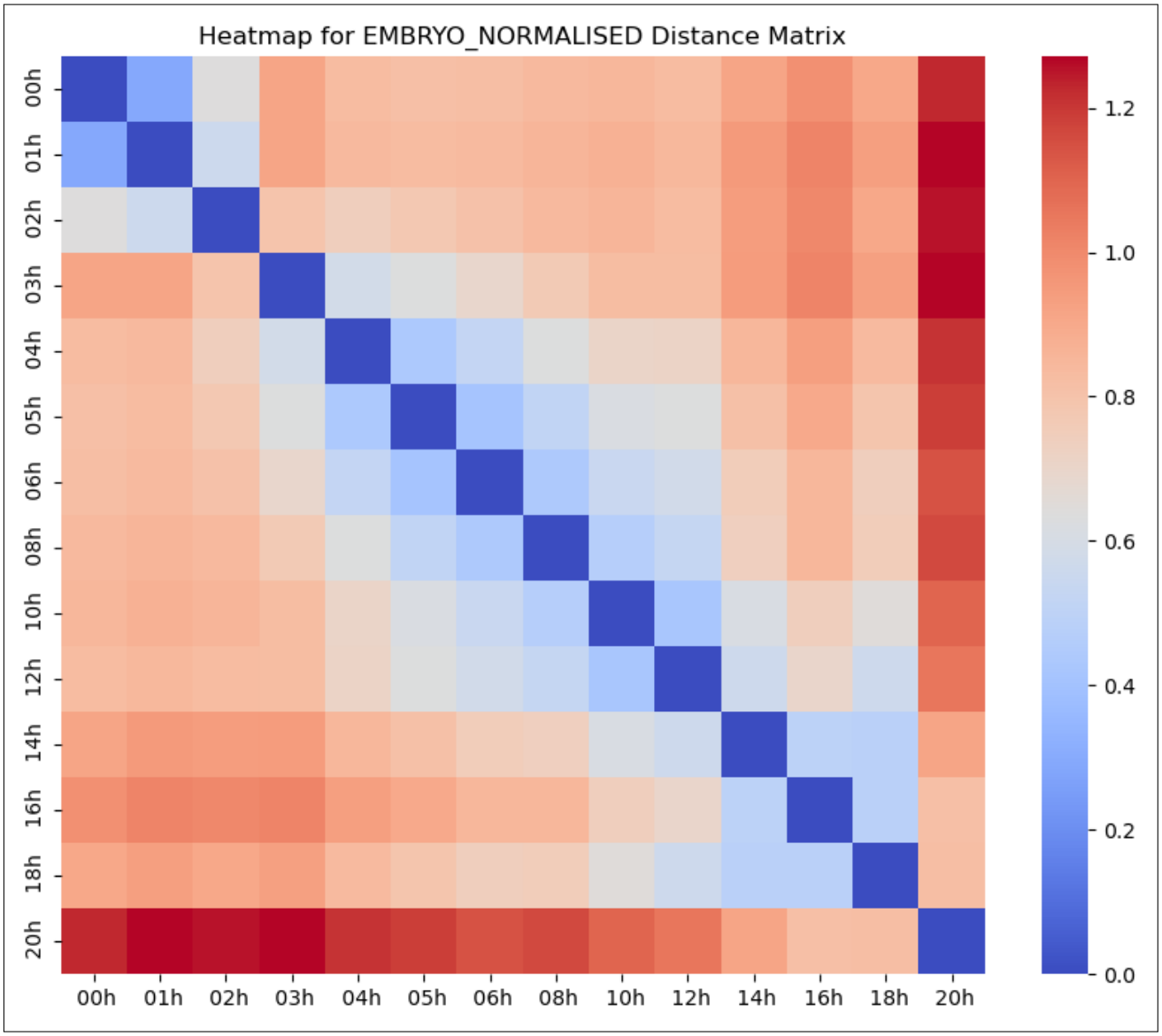

Figure 5.4: Heatmap of Euclidean distances between embryonic stages using normalized gene expression vectors.

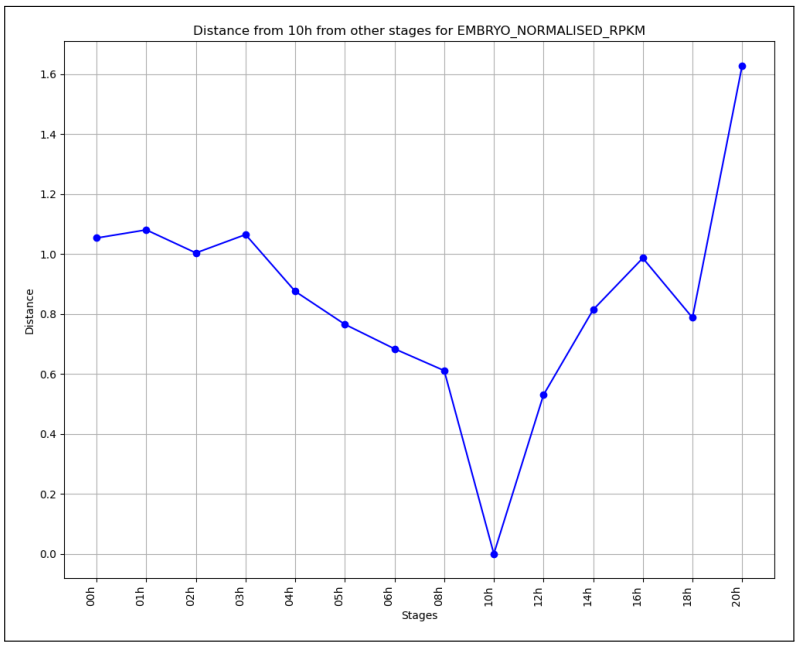

Figure 5.5: Distance of a selected embryonic stage (9) from other stages using co-expression matrices.

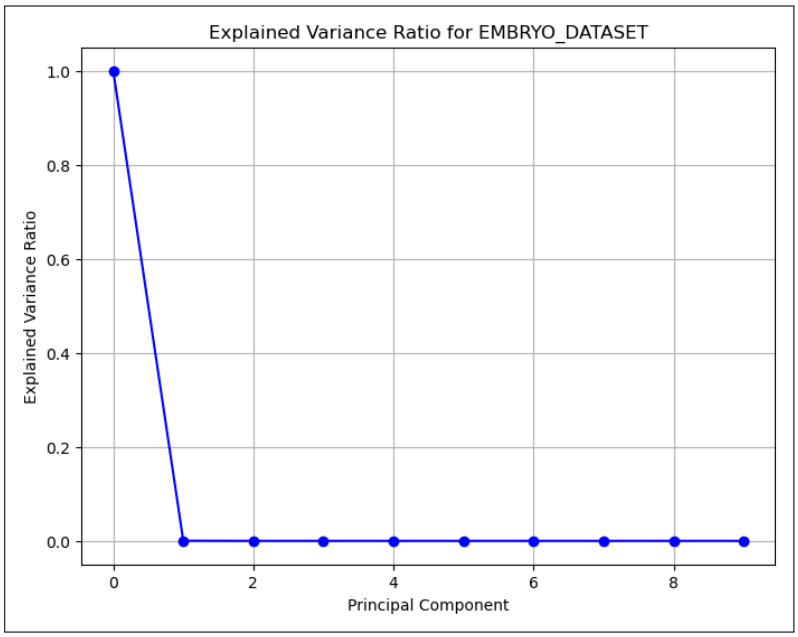

Figure 5.8: Explained variance ratio (EVR) of principal components from PCA on the Embryogenesis dataset.

Together, these findings provide strong evidence that the embryonic developmental trajectory of *Drosophila melanogaster* is similarly curved in gene expression space, justifying the use of

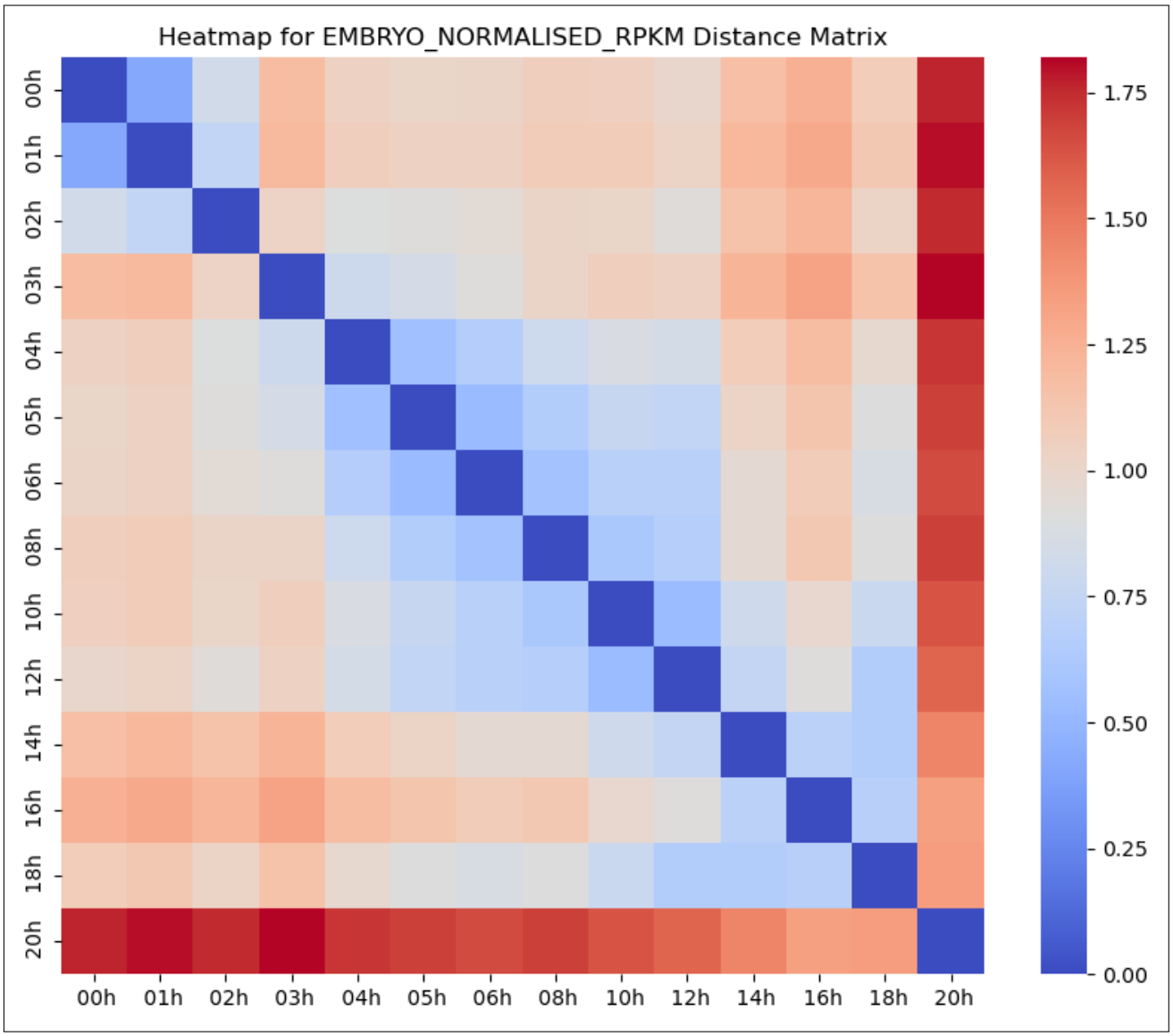

Figure 5.6: Heatmap of co-expression patterns from the outer product of normalized gene expression vectors in the Embryogenesis dataset.

trajectory-based inference methods as proposed in this work.

Curvature of the GSE24324 Dataset

To further validate the generalizability of our curvature-based trajectory hypothesis, we applied the same suite of analyses to the GSE24324 dataset, which represents another independent developmental transcriptomic profile of *Drosophila melanogaster*. These evaluations assess whether the geometric characteristics observed in the Full Cycle and Embryogenesis datasets persist in other developmental datasets.

Euclidean Distance (Unnormalized)

The Euclidean distances between developmental stages were first computed using unnormalized gene expression vectors. The resulting curves (e.g., Fig. 5.9) show that the distances remain relatively uniform across stages, lacking clear trends of temporal progression.

This is further confirmed in the heatmap shown in Fig. 5.10, where distances between stages are broadly consistent, without exhibiting gradients or continuity. This uniformity in raw distance patterns again points to the inadequacy of linear distance measures in capturing temporal relationships within gene expression space.

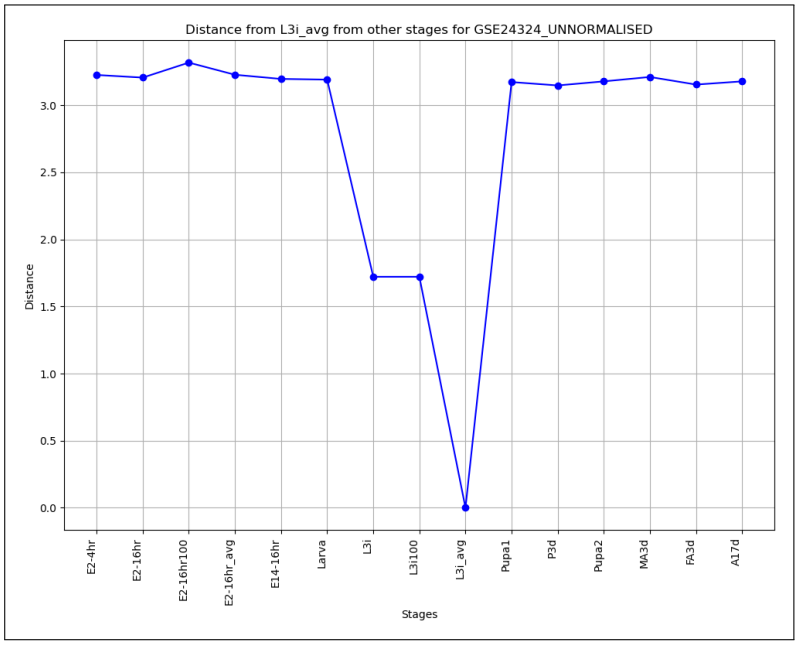

Figure 5.9: Euclidean distances from a selected stage (9) to other stages using unnormalized gene expression vectors in the GSE24324 dataset.

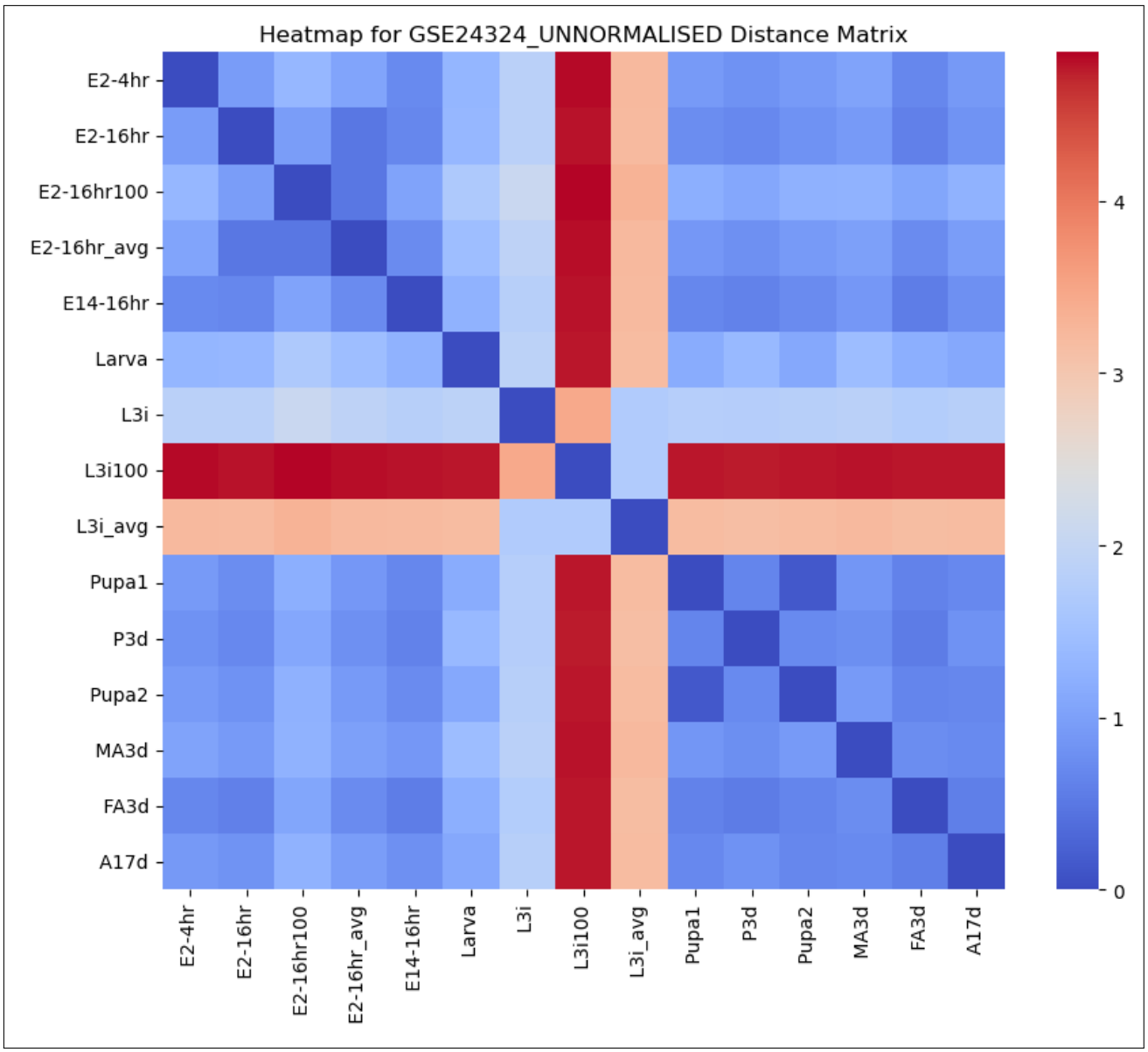

Figure 5.10: Heatmap of Euclidean distances using unnormalized gene expression profiles in the GSE24324 dataset.

Euclidean Distance (Normalized)

Upon normalization of expression vectors, we observed more distinct trends, particularly in later stages where transcriptional shifts are biologically more pronounced. Fig. 5.11 illustrates the stage-wise distances from a representative sample.

While early stages still cluster closely, the heatmap in Fig. 5.12 begins to show variation across

time, indicating that certain phases of development involve sharper transitions. These transitions, however, still deviate from linear patterns—implying a curved embedding in gene space.

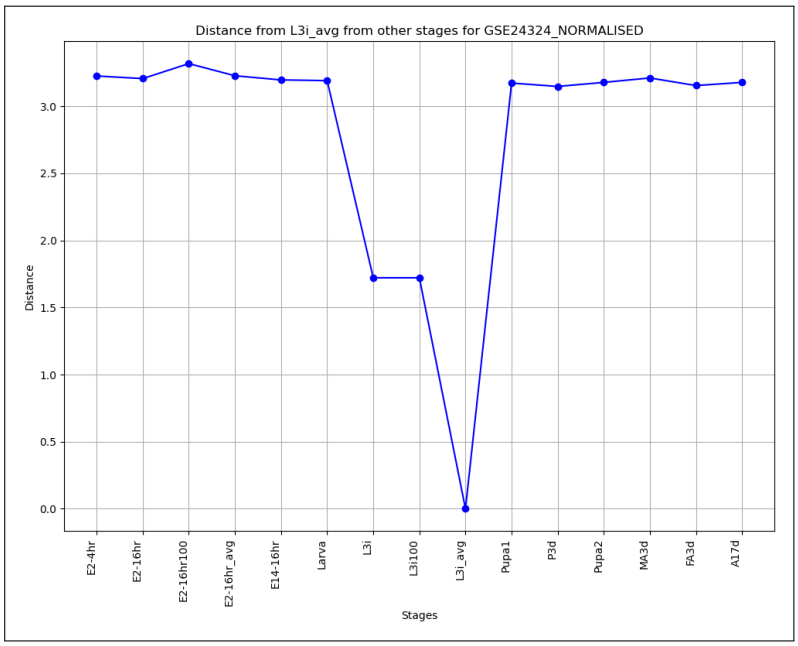

Figure 5.11: Distance from a selected stage (9) to other stages using normalized gene expression vectors.

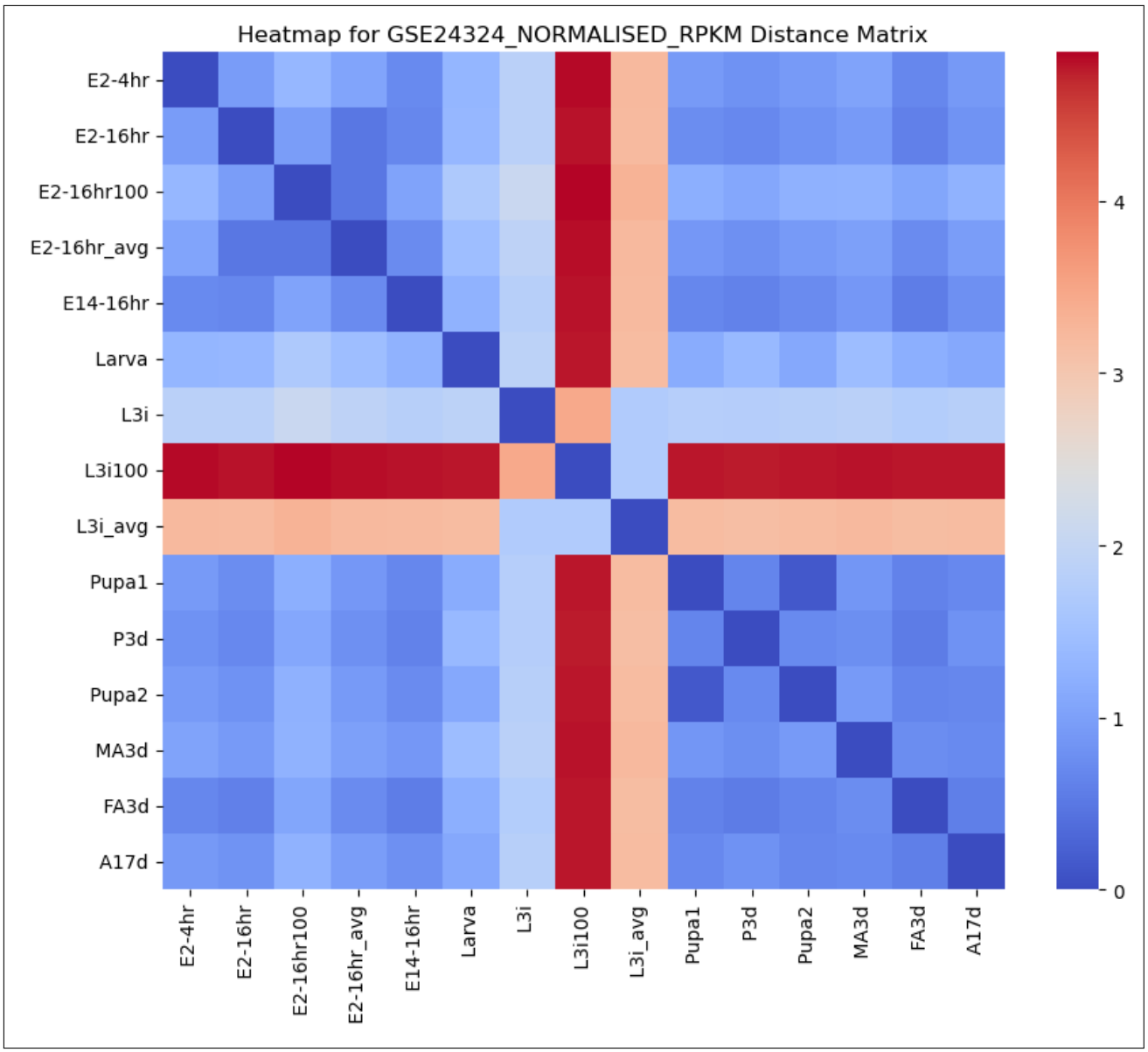

Figure 5.12: Heatmap of normalized Euclidean distances across stages in the GSE24324 dataset.

Co-expression Analysis

To probe deeper into gene–gene regulatory coordination, we computed co-expression matrices by evaluating the outer product of normalized expression vectors. This analysis helps identify transitions in transcriptional coordination that aren't easily captured by direct distance measures.

As shown in Fig. 5.13, co-expression distances show irregular but noticeable deviations, with

specific stages showing jumps in gene interaction patterns. The heatmap in Fig. 5.14 displays such variations, confirming phase-wise nonlinearity in gene coordination as development progresses.

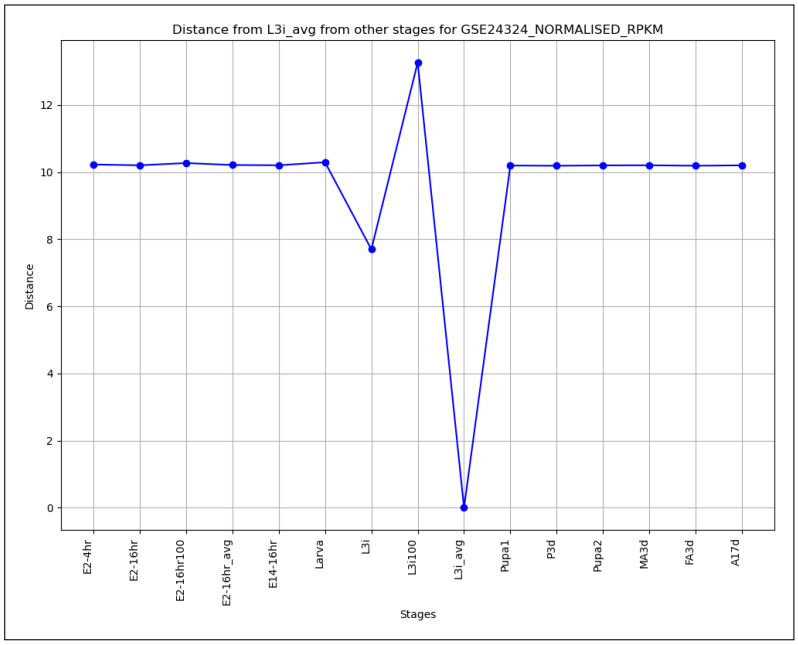

Figure 5.13: Co-expression distance curve from a selected stage (9) to other stages in the GSE24324 dataset.

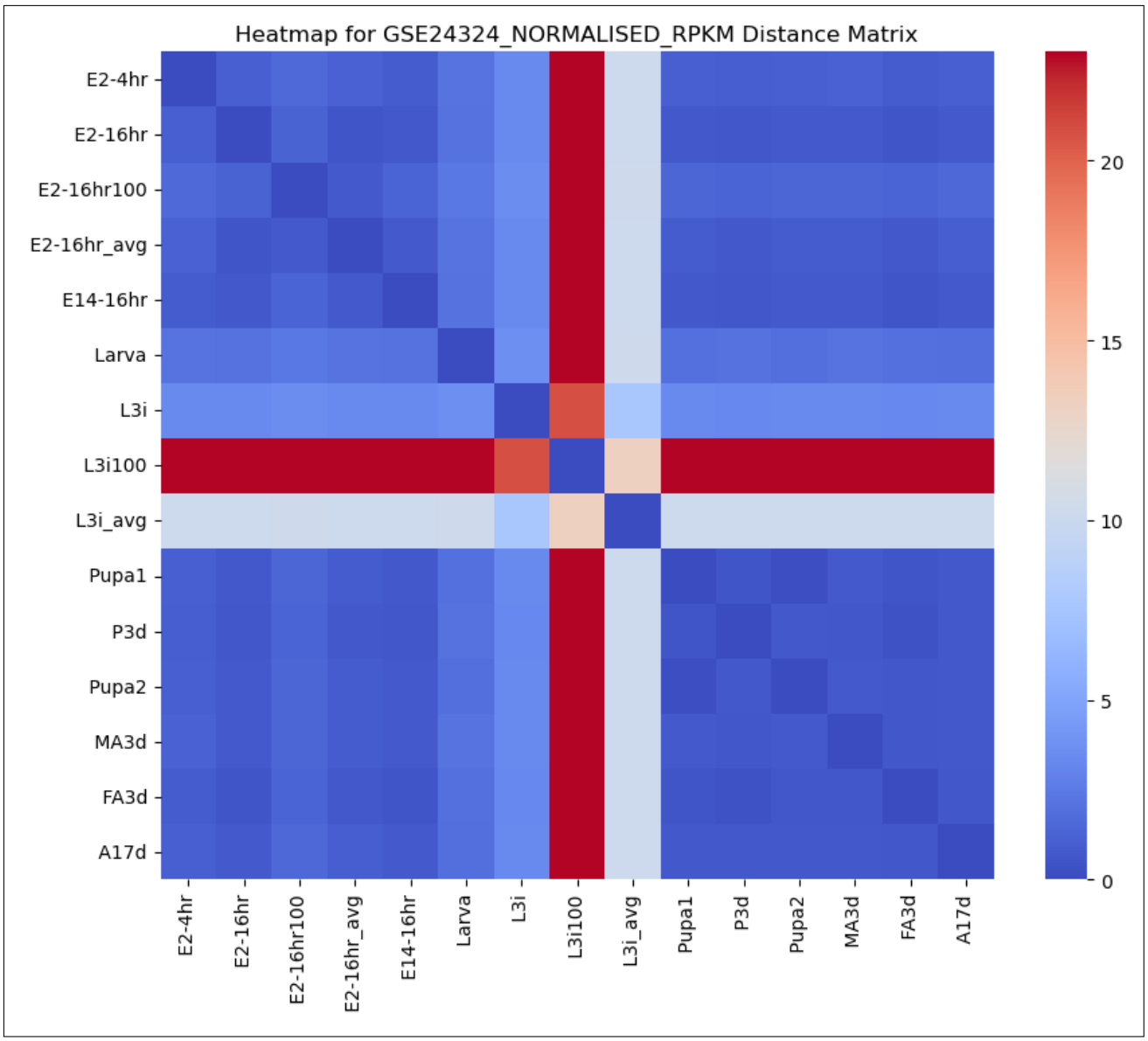

Figure 5.14: Heatmap of co-expression similarity across stages in the GSE24324 dataset.

Dimensionality Reduction

We also conducted PCA to visualize the structure of developmental transitions in reduced dimensions. As seen in Fig. 5.15, the 2D projection reveals a dispersed trajectory without distinct stage clusters, reinforcing the nonlinearity of transitions.

Fig. 5.16 shows the explained variance ratio for each principal component. While the first few components retain most of the variance, the rapid decay of variance suggests that sig-

nificant information is encoded in higher dimensions—further affirming the high-dimensional complexity of developmental gene expression.

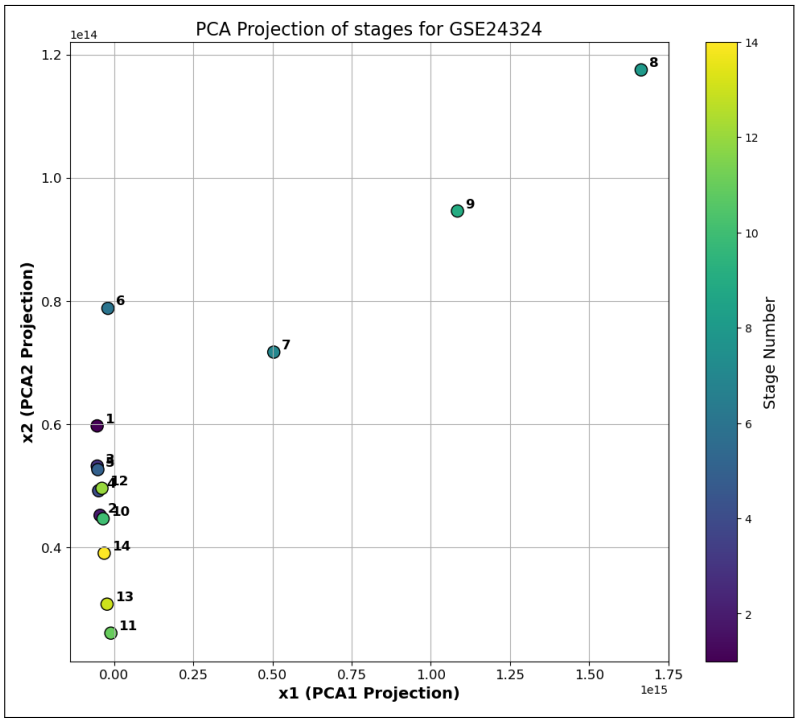

Figure 5.15: PCA projection of developmental stages in the GSE24324 dataset (2D space).

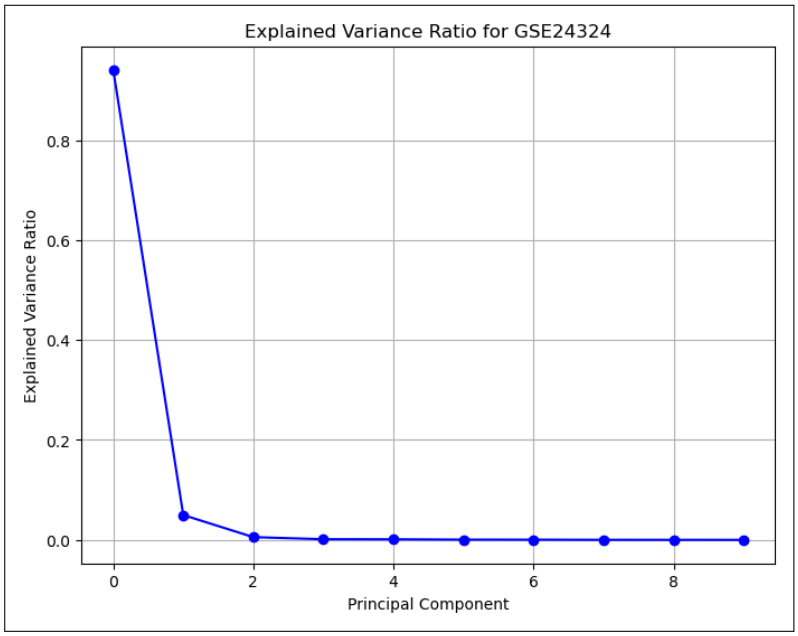

Figure 5.16: Explained variance ratio of principal components from PCA on GSE24324.

In summary, the GSE24324 dataset mirrors the curvature patterns found in the Full Cycle and Embryogenesis datasets. These consistent findings across three distinct datasets solidify our claim that developmental gene expression evolves along a complex, nonlinear trajectory in high-dimensional space.

Bibliography

[1] P. J. Park, S. L. Schreiber, and i. G. Kuruvilla, "Vector Algebra in the Analysis of Genome-Wide Expression Data Citation Terms of Use Share Your Story Vector algebra in the analysis of genome-wide expression data," tech. rep., 2002. vi, 1, 3, 9, 11, 12

[2] S. van Dam, U. Võsa, A. van der Graaf, L. Franke, and J. P. de Magalhães, "Gene co-expression analysis for functional classification and gene-disease predictions," *Briefings in Bioinformatics*, vol. 19, no. 4, pp. 575–592, 2018. vi, 13, 14

[3] S. Mandal, A. Guzmán-Sáenz, N. Haiminen, S. Basu, and L. Parida, "A Topological Data Analysis Approach on Predicting Phenotypes from Gene Expression Data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12099 LNBI, pp. 178–187, Springer, 2020. vi, 1, 4, 14, 15

[4] Y. Li and L. Chen, "Big biological data: Challenges and opportunities," *Genomics, Proteomics and Bioinformatics*, vol. 12, pp. 187–189, 10 2014. 1, 2, 3, 7

[5] L. Deconinck, R. Cannoodt, W. Saelens, B. Deplancke, and Y. Saeys, "Recent advances in trajectory inference from single-cell omics data," *Current Opinion in Systems Biology*, vol. 27, p. 100344, 2021. 1, 3, 16

[6] A. Haye, Y. Dehouck, J. M. Kwasigroch, P. Bogaerts, and M. Rooman, "Modeling the temporal evolution of the drosophila gene expression from dna microarray time series," *Physical Biology*, vol. 6, no. 1, p. 016004, 2009. 1, 4, 16

[7] I. P. Androulakis, E. G. Yang, R. R. Almon, and W. J. Jusko, "Analysis of time-series gene expression data: Methods, challenges, and opportunities," *Annual Review of Biomedical Engineering*, vol. 9, pp. 205–228, 2007. 2, 3, 4, 7, 8

[8] J. B. Cunha, "Curve fitting: Fitting functions to agricultural and biological data," in *4th World Congress on Computers in Agriculture*, pp. 618–620, ASABE, 2006. 2, 16

[9] T. G. Brooks, N. F. Lahens, A. Mrčela, and G. R. Grant, "Challenges and best practices in omics benchmarking," 5 2024. 2, 3, 7

[10] O. Alter, P. O. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," tech. rep. 2, 3, 10

[11] S. Allanki, M. Dixit, P. Thangaraj, and N. K. Sinha, "Analysis and modelling of septic shock microarray data using Singular Value Decomposition," *Journal of Biomedical Informatics*, vol. 70, pp. 77–84, 6 2017. 3

[12] J. M. Engreitz, B. J. Daigle, J. J. Marshall, and R. B. Altman, "Independent component analysis: Mining microarray data for fundamental human gene expression modules," *Journal of Biomedical Informatics*, vol. 43, pp. 932–944, 12 2010. 3, 13

[13] G. A. Pavlopoulos, M. Secrier, C. N. Moschopoulos, T. G. Soldatos, S. Kossida, J. Aerts, R. Schneider, and P. G. Bagos, "Using graph theory to analyze biological networks," 2011. 3

[14] P. Langfelder and S. Horvath, "WGCNA: An R package for weighted correlation network analysis," *BMC Bioinformatics*, vol. 9, 12 2008. 3

[15] R. Stark, M. Grzelak, and J. Hadfield, "Rna sequencing: the teenage years," *Nature Reviews Genetics*, vol. 20, no. 11, pp. 631–656, 2019. 7

[16] Z. Bar-Joseph, "Analyzing time series gene expression data," *Bioinformatics*, vol. 20, no. 16, pp. 2493–2503, 2004. 8, 9

[17] S. Xia, Z. Xiong, Y. Luo, Weixu, and G. Zhang, "Effectiveness of the Euclidean distance in high dimensional spaces," *Optik*, vol. 126, pp. 5614–5619, 12 2015. 9

[18] M. W. Mahoney and P. Drineas, "CUR matrix decompositions for improved data analysis," 2009. 12

[19] Y. Zhou and T. O. Sharpee, "Hyperbolic geometry of gene expression," *iScience*, vol. 24, no. 3, p. 102225, 2021. 16

[20] A. Haye, J. Albert, and M. Rooman, "Robust non-linear differential equation models of gene expression evolution across drosophila development," *BMC Research Notes*, vol. 5, p. 46, 2012. 16

[21] B. R. Graveley, A. N. Brooks, J. W. Carlson, M. O. Duff, J. M. Landolin, L. Yang, C. G. Artieri, M. J. Van Baren, N. Boley, B. W. Booth, J. B. Brown, L. Cherbas, C. A. Davis, A. Dobin, R. Li, W. Lin, J. H. Malone, N. R. Mattiuzzo, D. Miller, D. Sturgill, B. B. Tuch, C. Zaleski, D. Zhang, M. Blanchette, S. Dudoit, B. Eads, R. E. Green, A. Hammonds, L. Jiang, P. Kapranov, L. Langton, N. Perrimon, J. E. Sandler, K. H. Wan, A. Willingham, Y. Zhang, Y. Zou, J. Andrews, P. J. Bickel, S. E. Brenner, M. R. Brent, P. Cherbas, T. R. Gingeras, R. A. Hoskins, T. C. Kaufman, B. Oliver, and S. E. Celniker, "The developmental transcriptome of Drosophila melanogaster," *Nature*, vol. 471, pp. 473–479, 3 2011. 18

[22] K. Becker, C. Hirsch, U. Bönisch, J. Hörnig, P. Strauch, K. Mohr, H. Urlaub, A. Wodarz, U. Rüther, S. Legewie, *et al.*, "Quantifying post-transcriptional regulation in the development of drosophila melanogaster," *Nature Communications*, vol. 9, no. 1, p. 4970, 2018. 20

[23] B. Daines, H. Wang, L. Wang, Y. Li, Y. Han, D. Emmert, W. Gelbart, X. Wang, W. Li, R. A. Gibbs, *et al.*, "The drosophila melanogaster transcriptome by paired-end rna sequencing," *Genome research*, vol. 21, no. 2, pp. 315–324, 2011. 20