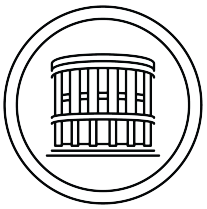
COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS



SIMILARITY OF WILLIAM SHAKESPEARE'S PLAYS BASED ON MATHEMATICAL CHARACTERIZATION OF THEIR CHARACTERS' INTERACTIONS

Master's Thesis

2026 **Bc. Yuliia POLOVYNKA**

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

SIMILARITY OF WILLIAM SHAKESPEARE'S PLAYS BASED ON MATHEMATICAL CHARACTERIZATION OF THEIR CHARACTERS' INTERACTIONS

Master's Thesis

Study programme:	Mathematics of Economy, Finance and Modeling
Field of Study:	Mathematics
Department:	Department of Applied Mathematics and Statistics
Supervisor:	doc. RNDr. Beáta Stehlíková PhD.

Bratislava 2026 **Bc. Yuliia Polovynka**





Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Bc. Yuliia Polovynka
Study programme:	Mathematical Economics, Finance and Modelling (Single degree study, master II. deg., full time form)
Field of Study:	Mathematics
Type of Thesis:	Diploma Thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Similarity of William Shakespeare's plays based on mathematical characterization of their characters' interactions

Annotation: A play can be represented as a social network, where characters are connected by an edge if they appear in the same scene. It is also possible to consider a weighted network, where the weights represent the number of such co-occurrences. Data in this format are available for a large number of plays on the DraCor project website. A comparison of the networks requires a suitable approach which allows for the varying number of characters in different plays. There is also a possibility to consider other available information on characters' interactions, available at DraCor, and to use it for comparison of the plays. This thesis examines several methodological approaches to address these issues, applied to the plays of William Shakespeare. The results are used for clustering and identifying the most similar plays.

Aim:
1. Select and apply a suitable method for network comparison based on the review article (Tantardini et al., 2019).
2. Propose other approaches for the comparison of W. Shakespeare's plays using the DraCor project data.
3. Compare the results obtained using different methods and interpret them.

Literature:
1. Fischer, F. et al. (2019). Programmable Corpora: Introducing DraCor, an Infrastructure for the Research on European Drama. In Proceedings of DH2019: "Complexities", Utrecht University
2. Tantardini, M., Ieva, F., Tajoli, L., & Piccardi, C. (2019). Comparing methods for comparing networks. Scientific reports, 9(1), 17557.

Supervisor:	doc. RNDr. Beáta Stehlíková, PhD.
Department:	FMFI.KAMŠ - Department of Applied Mathematics and Statistics
Head of department:	doc. Mgr. Igor Melicherčík, PhD.
Assigned:	10.01.2025
Approved:	15.01.2025

prof. RNDr. Daniel Ševčovič, DrSc.
Guarantor of Study Programme





Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

....

Student Supervisor





Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Yuliia Polovynka
Študijný program: ekonomicko-finančná matematika a modelovanie (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Similarity of William Shakespeare's plays based on mathematical characterization of their characters' interactions
Podobnosť divadelných hier Williama Shakespeara na základe matematickej charakterizácie interakcií ich postáv

Anotácia: Divadelná hra sa dá reprezentovať ako sociálna sieť, kde sú postavy prepojené hranou, ak sa vyskytujú v rovnakej scéne. Môžeme uvažovať aj váženú sieť, kde váhy reprezentujú počet týchto spoločných výskytov. Dáta v tomto formáte sú k dispozícii pre veľké množstvo hier na stránke projektu DraCor. Porovnanie sietí vyžaduje použitie metód, ktoré umožňujú pracovať s rôznym počtom postáv v jednotlivých hrách. Je tiež možné použiť iné dáta o interakciách postáv, ktoré sú dostupné v rámci projektu DraCor. Diplomová práca skúma niekoľko metodologických prístupov k riešeniu týchto otázok a aplikuje ich na hry Williama Shakespeara. Získané výsledky sú využité na zhlukovanie a identifikáciu najpodobnejších hier.

Cieľ: 1. Vybrať a aplikovať vhodnú metódu na porovnanie sietí na základe prehľadového článku (Tantardini et al., 2019).
2. Navrhnúť iné prístupy k porovnávaniu hier W. Shakespeara pomocou dát z projektu DraCor.
3. Porovnať výsledky získané rôznymi metódami a interpretovať ich.

Literatúra:
1. Fischer, F. et al. (2019). Programmable Corpora: Introducing DraCor, an Infrastructure for the Research on European Drama. In Proceedings of DH2019: "Complexities", Utrecht University
2. Tantardini, M., Ieva, F., Tajoli, L., & Piccardi, C. (2019). Comparing methods for comparing networks. Scientific reports, 9(1), 17557.

Vedúci: doc. RNDr. Beáta Stehlíková, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: doc. Mgr. Igor Melicherčík, PhD.

Dátum zadania: 10.01.2025

Dátum schválenia: 15.01.2025

prof. RNDr. Daniel Ševčovič, DrSc.
garant študijného programu





Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

....

študent vedúci práce

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Doc. RNDr. Beáta Stehlíková, PhD., for her guidance, valuable advice and continuous support throughout the process of writing this thesis. Her insights and suggestions greatly helped me to write this work.

I would also like to thank my family and close friends for their patience, encouragement and motivation during the preparation of this thesis.

Affidavit

I declare that the diploma thesis on the topic "Similarity of William Shakespeare's plays based on mathematical characterization of their characters' interactions", including all its appendices and images, I wrote independently, using the literature listed in the attached references and artificial intelligence tools. I declare that I have used artificial intelligence tools in accordance with the relevant legal regulations, academic rights and freedoms, ethical and moral principles while maintaining academic integrity and that their use is appropriately marked in my thesis.

Bratislava, 29.4.2026

.....

Bc. Yuliia Polovynka

Abstrakt

V našej práci skúmame divadelné hry Williama Shakespeara z matematického hľadiska. V časti analýzy reprezentujeme každú hru ako sociálnu sieť. Takéto siete poskytujú kompaktné štrukturálne zhrnutie: zachytávajú, koľko postáv sa v hre vyskytuje a aké vzťahy medzi nimi existujú. Zároveň používame časovo založené reprezentácie interakcií postáv medzi scénami vo forme špeciálnych časových radov, nazývaných beat chart, aby sme preskúmali, ako sa tieto vzťahy v priebehu hry vyvíjajú. Naším cieľom je použiť rôzne metódy analýzy dát na pochopenie dynamiky a štruktúry hier. Porovnaním a klastrovaním hier identifikujeme skupiny s podobnou štruktúrou a interpretujeme výsledky v širšom literárnom kontexte.

Kľúčové slová: sociálne siete, časové údaje, Shakespeare, kvantitatívna literárna analýza, zhlukovanie

Abstract

In our thesis, we explore William Shakespeare's plays from a mathematical point of view. In part of our analysis, we represent each play as a social network. Such networks provide a compact structural summary: they reflect how many characters appear in the play and how they are related to each other. We also use time-based representations of character interactions between the scenes in form of special time series called beat charts, to examine how these relationships evolve throughout the play. Our goal is to apply various data analysis methods to better understand how such patterns capture the dynamics and structure of the plays. By comparing and clustering the plays, we identify groups with similar structure and interpret the results in a broader literary context.

Keywords: social networks, temporal data, Shakespeare, quantitative literary analysis, clustering

Contents

Introduction **8**

1 Shakespeare's Plays in the Context of Data **10**

1.1 Overview of Shakespeare's Plays 10

1.2 Data Description: Social Networks and Beat Charts 10

2 Social Network Analysis **13**

2.1 Theoretical Background on Social Networks and Graph Theory 13

2.2 Network Laplacian Spectral Descriptor 14

3 Analysis of Temporal Data **23**

3.1 Earth Mover's Distance 23

3.2 Dynamic Time Warping Method for Time Series of Different Length 33

3.3 Roughness and Spectral Entropy 39

3.4 Archetypal Analysis 45

4 Clustering and Comparison: Investigation of Groups of Similar Plays **53**

4.1 Adjusted Rand Index (ARI) 53

4.2 Comparison Across Methods 53

4.3 Community Detection On The Co-occurrence Graph 60

Conclusions **66**

Bibliography **69**

Introduction

Mathematical and quantitative approaches have long been applied to the study of literature, including the works of William Shakespeare. For instance, in [4] author uses statistical techniques to address authorship uncertainty. Other researchers apply data-driven methods to study how Shakespeare's writing style evolves across different periods of his life, for example, in [43]. There are also interdisciplinary studies combining quantitative analysis with psychophysical measurements, for instance as an eye-tracking investigation of how readers react to Shakespeare's sonnets [49]. These examples illustrate the growing interest in connecting mathematics, data science and literary studies to uncover patterns that are not easy to be discovered with traditional qualitative approaches, including stylistic and rhetorical studying, critical discussions and reviews in literary scholarship etc. – approaches which are mainly based on litarary experts' judgments and do not involve any mathematical analysis.

While previously mentioned studies are focused on linguistic style, authorship or textual features, there are works which explore a different perspective - the relationships between characters, and how they develop throughout each play, which can be studied from networks of characters or temporal representations of Shakespeare's drama. For example, in [40] authors analyze social networks for ten Shakespeare's plays and investigate properties such as group size and connectivity, comparing them with what is known about naturally occurring human social groups.

Similarly, our thesis uses character networks and temporal representations to study Shakespeare's plays. The main objective of this thesis is to apply various data analysis methods to a corpus of Shakespeare's plays and study their social network structure and time series patterns.

In this thesis we first introduce the selected thirty seven plays and describe their main characteristics, including their period of origin, genre and general facts.

Next, we describe the data used in our analysis, which consist of social networks of characters, downloaded from corpus DramaCorpora (https://dracor.org/shake), and temporal data of interactions between characters in the form of *beat charts*, which we calculated from character information by scenes provided in Shakespare's Words (https://www.shakespeareswords.com/Public/TheatreCompanion/CharactersInScenes.aspx).

In these social networks, nodes represent characters and edges indicate coappearance within the same scene, with edge weights reflecting the number of such cooccurrences, while each beat chart represents a play as a time series over its scenes, where each value summarizes which character pairs interact in a given scene, allowing us to track how these interactions change from one scene to the next. This representation makes it possible to study both the plays' structure from the social networks and dynamicity from the beat charts.

We then apply multiple data analysis techniques to our data and provide a brief description of each method. The approaches include network-based analysis and time-series-based analysis of character interactions. We apply Network Laplacian Spectral Descriptor for social networks, as well as compute several beat chart features, such as roughness and spectral entropy, and further explore archetypal analysis of beat charts, which have so far not been used in Shakespeare studies.

Finally, we perform clustering and comparison of the results in order to identify groups of plays with similar characteristics, which are subsequently interpreted in a broader literary context. All computations, codes and data used in this thesis are available on GitHub at https://github.com/YuliiaPolovynka/Master-Thesis/.

The motivation for this research is to detect relationships that are difficult to notice through traditional literary methods. This approach also helps us compare plays in a new way and bring new interesting insights about Shakespeare's plays.

In summary, this thesis contributes to the developing field of quantitative literary analysis by combining social network analysis, temporal data and clustering methods to study Shakespeare's plays.

1 Shakespeare's Plays in the Context of Data

1.1 Overview of Shakespeare's Plays

In this study we analyze a corpus of thirty seven William Shakespeare's plays. These plays were selected because of their availability in the DraCor database and include all Shakespeare's plays except for *Edward III* and *The Two Noble Kinsmen*.

According to the traditional classification of Shakespeare's works based on the First Folio, the plays are divided into three main genres: comedies, tragedies and histories [31].

The comedies in our corpus include: *All's Well That Ends Well*, *As You Like It*, *The Comedy of Errors*, *Love's Labour's Lost*, *Measure for Measure*, *The Merchant of Venice*, *The Merry Wives of Windsor*, *A Midsummer Night's Dream*, *Much Ado About Nothing*, *The Taming of the Shrew*, *The Tempest*, *Twelfth Night*, *The Two Gentlemen of Verona* and *The Winter's Tale*.

The tragedies include: *Antony and Cleopatra*, *Coriolanus*, *Cymbeline*, *Hamlet*, *Julius Caesar*, *King Lear*, *Macbeth*, *Othello*, *Romeo and Juliet*, *Timon of Athens*, *Titus Andronicus* and *Troilus and Cressida*.

The histories include: *Henry IV Part I*, *Henry IV Part II*, *Henry V*, *Henry VI Part I*, *Henry VI Part II*, *Henry VI Part III*, *Henry VIII*, *King John*, *Richard II* and *Richard III*.

The remaining play is *Pericles* and was not included in the First Folio, but is often classified as tragicomedy.

For additional interpretation of the results, we also consider metadata about the plays. The approximate chronology of composition is taken from the Royal Shakespeare Company timeline [32]. We also consider the geographical settings of the plays, such as Greek, English, Roman and Italian, based on information from the Morehead State University Shakespeare database [22].

1.2 Data Description: Social Networks and Beat Charts

In this subsection, we provide a general overview of the data used in our analysis. We describe how Shakespeare's plays are represented as social networks of characters taken from corpus DraCor and as time series known as beat charts, which capture the dynamics

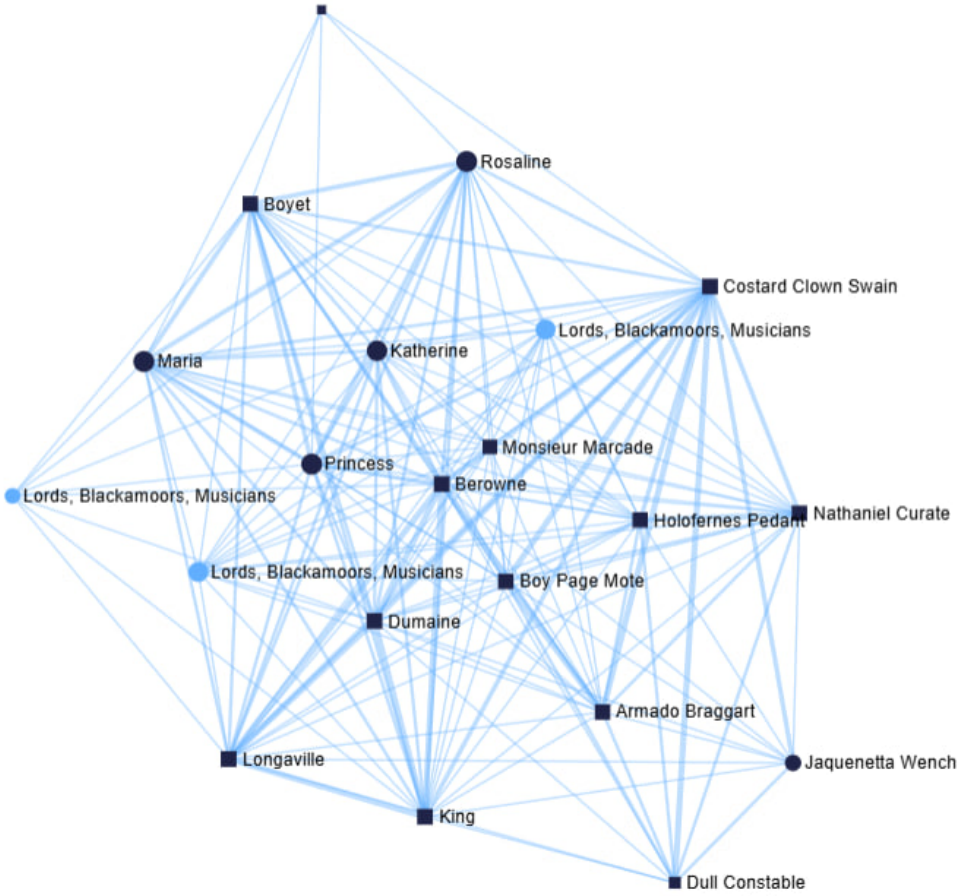


Figure 1: Example of a character social network for *Love's Labour's Lost*. Source: DraCor, https://dracor.org/shake/loves-labors-lost.

of character interactions between the scenes.

Figure 1 shows an example of a character social network for *Love's Labour's Lost*. Each node represents a character and an edge indicates that the two characters appear together in at least one scene. Edge thickness reflects the number of co-occurrences across scenes so that thicker edges correspond to more frequent appearances together. Naturally, different plays yield networks of varying sizes, with different numbers of nodes and edges. This property is important when choosing a method for network comparison.

Figure 2 illustrates an example of a beat chart for *Two Gentlemen of Verona*. Beat charts provide a view of how characters interact across the scenes. We quantify how much the interaction changes from scene t to $t{+}1$ using the segment-change rate

$$r_t = \frac{|S_t \,\Delta\, S_{t+1}|}{|S_t \cup S_{t+1}|},$$

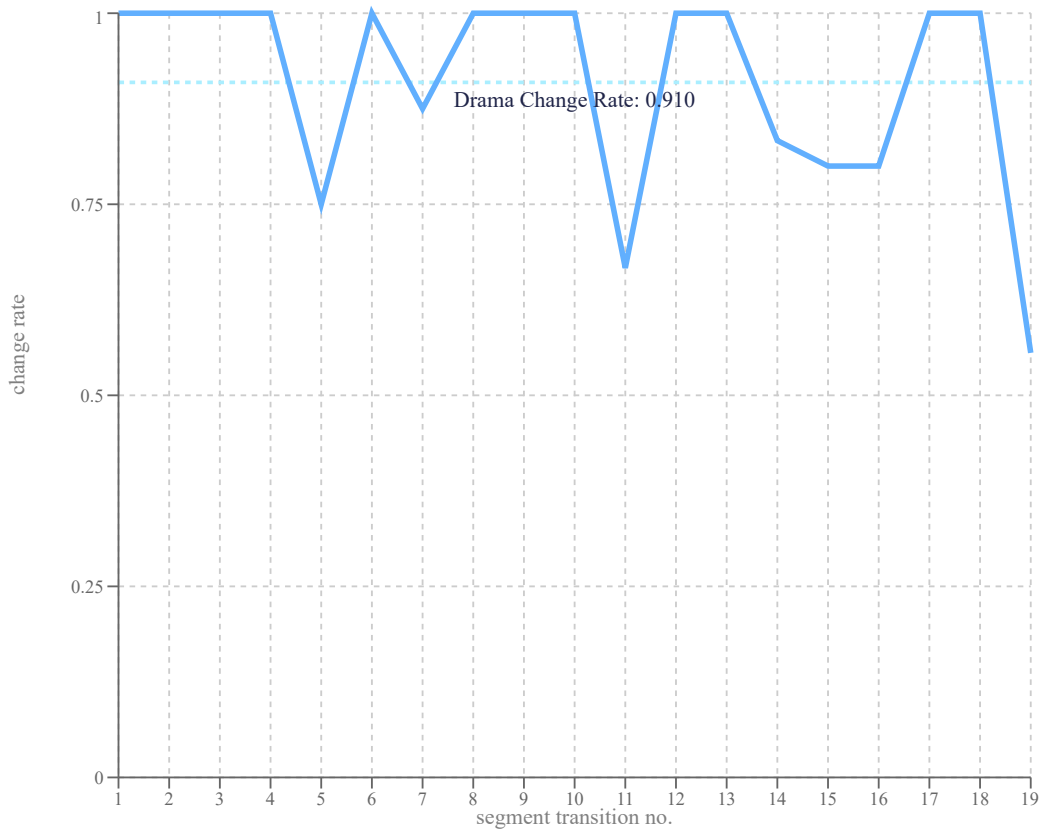


Figure 2: Example of a beat chart for *Two Gentlemen of Verona*. Source: DraCor, https://dracor.org/shake/two-gentlemen-of-verona.

where S_t denotes the set of interacting character pairs observed in scene t, Δ is the symmetric difference and T is the number of scenes. The drama-change rate (DCR) [13] is then defined as the average change across all scene transitions:

$$\mathrm{DCR} \;=\; \frac{1}{T-1}\sum_{t=1}^{T-1} r_t.$$

2 Social Network Analysis

2.1 Theoretical Background on Social Networks and Graph Theory

Since our analysis is based on character social networks, we briefly introduce several basic concepts from graph theory.

A graph is usually denoted by $G = (V, E)$, where V is the set of vertices (or nodes) and E is the set of edges connecting pairs of vertices. In our case, nodes represent characters in a play and edges represent interactions between them.

More precisely, we work with weighted undirected graphs. The graph is undirected because the relation of co-occurrence in a scene is symmetric: if character i appears with character j, then character j also appears with character i. The graph is weighted because each edge has a weight that shows how many times two characters appear together in the same scene.

Such a graph can be represented by *an adjacency matrix* $A = (A_{ij})$. For a weighted undirected graph, the matrix entry A_{ij} is equal to the weight of the edge between vertices i and j, and $A_{ij} = 0$ if there is no edge between them. Since the graph is undirected, the adjacency matrix is symmetric, that is, $A_{ij} = A_{ji}$.

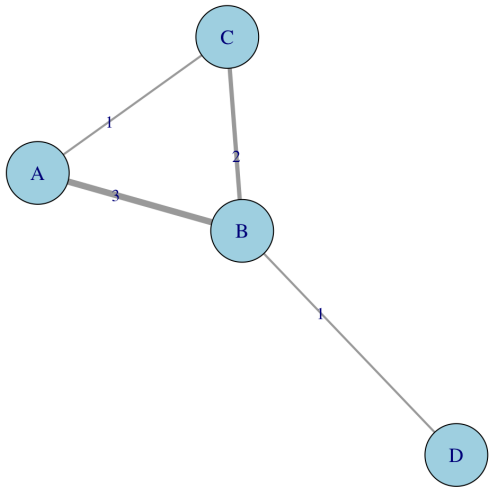


Figure 3: Example of a weighted undirected graph.

Figure 3 shows a simple example of a weighted undirected graph together with its adjacency matrix. For example, the edge between vertices A and B has weight 3, which means that these two vertices are connected three times. The adjacency matrix for this example looks as follows

$$A = \begin{pmatrix} 0 & 3 & 1 & 0 \\ 3 & 0 & 2 & 1 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

.

In addition to the adjacency matrix, it is also useful to introduce *the degree of a vertex*. The degree of a vertex i, denoted by $\deg(i)$, is defined as the sum of the weights of all edges connected to this vertex and can be computed as

$$\deg(i) = \sum_j A_{ij}.$$

The degree reflects how strongly a given vertex is connected to the rest of the network. In our context, it corresponds to the total number of interactions of a given character with other characters. *The degree matrix* D is therefore a diagonal matrix containing the degrees of all vertices:

$$D = \mathrm{diag}(\deg(1), \deg(2), \ldots, \deg(n)).$$

More information on graph theory can be found in [24].

2.2 Network Laplacian Spectral Descriptor

In modern data analysis, various systems can be represented as graphs, where the nodes stand for entities and the edges correspond to relations or interactions between them. This idea is used in different fields, for example in social network analysis, biology, economics, chemistry etc.

Comparing graphs helps to understand how similar two networks are, and spectral methods are one of the powerful tools for graph analysis. These methods use the eigenvalues and eigenvectors of matrices related to the graph's structure. One of the modern approaches based on this idea is called the *Network Laplacian Spectral Descriptor*

(NetLSD). It gives a compact way to describe the shape of a graph using specific spectral properties of normalized Laplacian matrix [44].

In this chapter, we cover the theoretical background of NetLSD, including key concepts from spectral graph theory, Laplacian-based node signatures and the construction of graph-level descriptors [44, 17].

Definition 2.1 (Graph Laplacian [44])**.** *Let $G = (V, E)$ be an undirected weighted graph with adjacency matrix A and degree matrix D. The (unnormalized) graph Laplacian is defined as*

$$L = D - A.$$

Graph Laplacian matrix is symmetric and positive semi-definite, and thus all its eigenvalues are real and non-negative. The multiplicity of the zero eigenvalue equals the number of connected components in the graph [5].

Definition 2.2 (Normalized Graph Laplacian [44])**.** *For graphs with varying node degrees, the normalized Laplacian is defined as*

$$\mathcal{L} = I - D^{-1/2} A D^{-1/2}.$$

The normalized Laplacian is either symmetric matrix. The eigenvalues of the normalized Laplacian $\mathcal{L}$ always lie in the interval $[0, 2]$. These eigenvalues encode important structural properties of the graph. In particular, the second smallest eigenvalue λ_2, also called the Fiedler value, is known as the algebraic connectivity of the graph: it equals zero if and only if the graph is disconnected, and larger values indicate a higher degree of connectivity between nodes [19, 5].

Furthermore, the spectrum of the Laplacian matrix is invariant under relabeling of vertices. That is, if P is a permutation matrix, then $P^\top L P$ corresponds to a relabeled version of L that has the same eigenvalues [15].

This invariance is fundamental for spectral methods, as it allows comparison of graphs without requiring a node correspondence. This idea is used in various methods, one of which is NetLSD [44].

Laplacian-Based Node Signatures and the Heat Kernel

One way to describe the role of a node in a graph is through its spectral signature. A widely used approach for this is called the *Laplacian Family Signatures* (LFS), which was introduced in [17].

Definition 2.3 (Laplacian Family Signature [17])**.** *Let $\{(\lambda_k, \phi_k)\}_{k=1}^n$ be the eigenvalues and eigenvectors of the normalized graph Laplacian $\mathcal{L}$. Then the signature of node $i \in V$ at time t is defined as*

$$s_i(t) = \sum_{k=1}^{n} h(t; \lambda_k) \cdot \phi_k(i)^2,$$

where $h(t; \lambda_k)$ is a function (called a kernel) that works with the spectrum.

A common choice for h is the *heat kernel*, which is defined as:

$$h(t; \lambda_k) = e^{-t\lambda_k}.$$

It originates from the solution of the heat diffusion equation on a graph, which provides the interpretation of $s_i(t)$ as the amount of heat remaining at node i at time t [41, 5].

This idea is used in several modern methods, including NetLSD, where we combine all node signatures into one compact signature for the whole graph [44].

In [17], the authors also show that this kind of node signature is stable — meaning that if we slightly change the graph, the signatures either do not change much. This is important when working with real data that may have noise or small errors.

Theorem 2.4 (Stability of Laplacian Family Signatures [17])**.** *Let A, A' be the adjacency matrices of two graphs and $\mathcal{L}, \mathcal{L}'$ the corresponding normalized Laplacians. Let n be the number of nodes and $\lambda_1 < \cdots < \lambda_k$ be the first k distinct eigenvalues of $\mathcal{L}$. Let $s_i(t)$ and $s_i'(t)$ denote the LFSs of node i. Assume $\lambda_{j+1} - \lambda_j \geq \delta$ for all j, $\|A - A'\|_F \leq \frac{\epsilon}{\sqrt{n+1}} < \delta$, and $h(\cdot\,;\cdot) \in C^2(\mathbb{R}_+^2)$.*

If $k = n$ (i.e., non-repeating eigenvalues), then:

$$|s_i'(t) - s_i(t)| \leq C_0(\delta, t)\, \epsilon,$$

where $C_0(\delta, t)$ is a constant independent of ϵ.

If $k < n$ (i.e., some eigenvalues repeat), then:

$$|s_i(t) - s_i'(t)| \leq C_1(t) \left(\frac{\delta}{\delta - \epsilon} - 1 \right) + C_2(t)\epsilon,$$

where $C_1(t)$ and $C_2(t)$ are constants depending only on t.

This result provides a formal guarantee that small changes in the graph structure lead to small changes in the node signatures, which is essential for ensuring robustness in applications.

The NetLSD Descriptor

In further part of our thesis we describe the approach introduced in [44]. While node-level signatures provide detailed local information about individual vertices, they are not well suited for comparing entire graphs, especially when the graphs are large. The NetLSD method addresses this by aggregating spectral information into a compact descriptor that represents the entire graph.

Instead of relying directly on the Laplacian spectrum, NetLSD considers an associated diffusion process over the graph. Specifically, it uses the *heat equation*, which describes how heat spreads across the graph structure. The continuous-time heat equation associated with the normalized Laplacian $\mathcal{L}$ is

$$\frac{\partial u(t)}{\partial t} = -\mathcal{L}u(t), \quad u(0) = u_0,$$

where $u(t)$ is a vector of heat values at all nodes at time t, and u_0 is the initial heat distribution. The solution of this system is given by

$$u(t) = e^{-t\mathcal{L}}u_0,$$

where $H_t = e^{-t\mathcal{L}}$ is called the *heat kernel matrix*. It does not depend on the specific initial condition u_0; rather, it is the fundamental solution of the heat equation, so that $u(t) = H_t u_0$ for any initial heat distribution u_0. In particular, if one starts from a unit heat source at node j, the entry $H_t(i, j)$ gives the amount of heat present at node i at time t.

The heat kernel can also be expressed using the eigendecomposition of $\mathcal{L}$:

$$H_t = \Phi e^{-t\Lambda}\Phi^\top,$$

where Λ is the diagonal matrix of eigenvalues λ_j, and Φ is the matrix of corresponding eigenvectors ϕ_j.

Since H_t is an $n \times n$ matrix, it is not practical to compare graphs directly using the full heat kernel, especially when graphs have different sizes. Instead, NetLSD uses the

heat trace, defined as the trace of H_t,

$$h_t = \mathrm{tr}(H_t) = \sum_{j=1}^{n} e^{-t\lambda_j}.$$

The trace depends only on the eigenvalues of the Laplacian and therefore provides a compact, permutation-invariant summary of the diffusion process that can be used for graph comparison.

Definition 2.5 (Heat Trace Signature [44])**.** *Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of the normalized Laplacian matrix $\mathcal{L}$. The* heat trace signature *of a graph G is defined as a vector of heat traces at a sequence of time scales:*

$$h(G) = \{h_t\}_{t>0} = [h_{t_1}, \ldots, h_{t_T}],$$

where each $h_t = \sum_{j=1}^{n} e^{-t\lambda_j}$ is computed at time t.

The time points $\{t_1, \ldots, t_T\}$ are chosen on a logarithmic scale. Using the spectral decomposition $H_t = \Phi e^{-t\Lambda}\Phi^\top$, one obtains the short-time expansion

$$H_t = I - t\mathcal{L} + O(t^2) \qquad (t \to 0).$$

The Laplacian $\mathcal{L}$ contains only local information: its diagonal entries are node degrees and its off–diagonal entries encode the weights of edges between directly connected nodes. Therefore the first–order term $-t\mathcal{L}$ reflects how heat immediately flows between neighboring vertices, so the trace $h_t = \mathrm{tr}(H_t)$ is determined by *local connectivity* such as node degrees and the presence of short cycles.

In the opposite limit $t \to \infty$, all factors $e^{-t\lambda_k}$ with $k > 2$ vanish, and

$$H_t \approx I - e^{-t\lambda_2}\,\phi_2\phi_2^\top.$$

Here λ_2 is the Fiedler eigenvalue and its eigenvector ϕ_2 describe the slowest non-trivial diffusion mode. This slowest diffusion mode describes how heat spreads between large groups of vertices and reflects the *overall connectivity* of the graph. For example, it can reveal whether the graph contains a weakly connected region that separates two major parts of the network. Therefore small values of t emphasize local structure, while large values of t reveal the global structure of the graph. A detailed mathematical derivation of these asymptotic behaviors can be found in [44].

To illustrate the heat trace signature in practice, we compute the function $h_t = \mathrm{tr}(H_t)$ for each play in our dataset. Figure 4 shows the resulting heat trace curves for all graphs. Each curve corresponds to one play and represents how the diffusion process evolves over time. We can see that each function converges to 1 or 2, depending on whether the graph consists of one or two connected components.

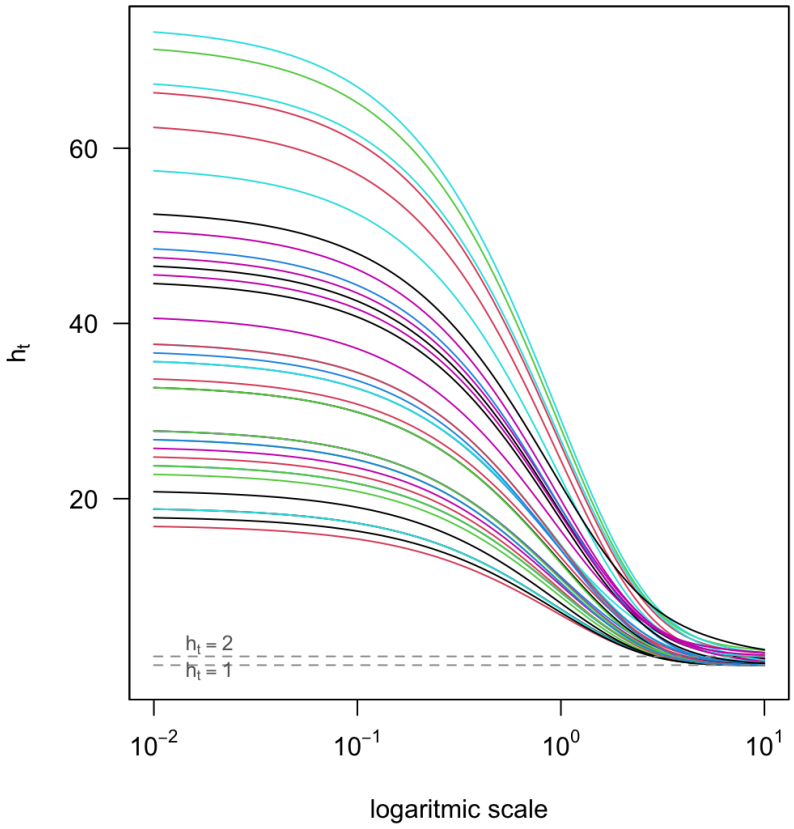


Figure 4: Heat trace signatures h_t for all plays. The time parameter t is shown on a logarithmic scale.

In their work, the authors highlight main important properties of the heat trace signature $h(G)$ that make it suitable for practical applications [44].

Let G be an undirected graph with normalized Laplacian $\mathcal{L}$ and heat trace signature $h(G) = \{\mathrm{tr}(e^{-t\mathcal{L}})\}_{t>0}$. Then $h(G)$ satisfies the following properties:

1. **Permutation invariance.** $h(G)$ depends only on the spectrum of $\mathcal{L}$ and is therefore unchanged under any relabeling of the vertices.

2. **Scale adaptivity.** Varying the time parameter t reveals structural information from local neighborhoods ($t \to 0$) to global graph organization ($t \to \infty$).

3. **Size invariance.** Although the raw trace depends on the number of nodes, normalization using reference graphs (empty or complete) allows meaningful comparison of graphs of different sizes.

Application of NetLSD in Network Analysis of Shakespeare's Plays

In the practical part of this thesis, we employed the Network Laplacian Spectral Descriptor method to characterize the structural properties of social networks derived from the texts of Shakespeare's plays. Each network corresponds to a play, where nodes represent characters and weighted edges indicate the frequency of their interactions.

For each graph we computed the normalized Laplacian matrix and then evaluated heat trace signature over 100 logarithmically spaced time points $t \in [0.01, 10]$, yielding a 100-dimensional vector $h(G)$ that encodes the diffusion dynamics of the graph. This representation captures both local and global structural features [44].

To quantify structural similarity between plays, we computed the pairwise *Euclidean distances* between all NetLSD vectors. Specifically, for two graphs G_1 and G_2 with corresponding heat signature vectors $h(G_1)$ and $h(G_2)$, the distance is given by:

$$\mathrm{dist}(G_1, G_2) = \|h(G_1) - h(G_2)\|_2 = \sqrt{\sum_{i=1}^{T} (h_{G_1}(t_i) - h_{G_2}(t_i))^2},$$

where $T = 100$ is the number of time points.

This computation resulted in a 37×37 distance matrix that summarizes the pairwise structural dissimilarities between the plays. Based on this matrix, we applied hierarchical agglomerative clustering using four standard linkage metods: *single*, *complete*, *average*, and *Ward.D2*. For each method, a dendrogram was created, and the number of clusters k can be selected heuristically based on dendrogram's branches lengths, however we put $k = 4$ for this and all further methods for easier comparison of results.

Figure 5 shows the dendrogram obtained with the Ward.D2 linkage. The cluster assignment for the Ward.D2 linkage with $k = 4$ is shown in Table 1.

Cluster 1 is the biggest cluster of sixteen plays, dominated by comedies. It contains the majority of Shakespeare's comedy plays, including *Much Ado About Nothing*, *As You*

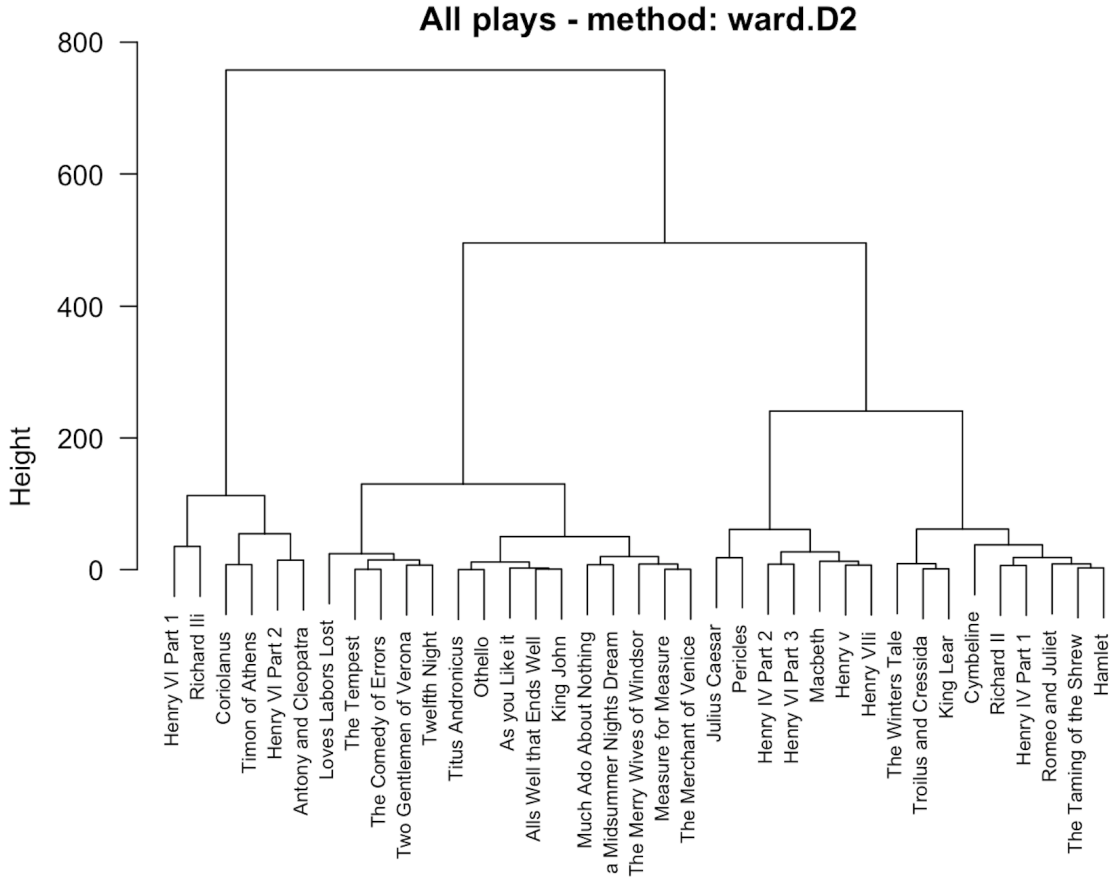


Figure 5: Hierarchical clustering dendrogram obtained using the Ward.D2 linkage.

Like It, *Twelfth Night*, *A Midsummer Night's Dream* and *The Comedy of Errors*. Ten of these plays are set in Italian cities. A few non-comedic plays, such as *Othello* and *Titus Andronicus* also appear in this cluster.

Cluster 2 is a mixed cluster of nine plays, yet dominated with tragedies, including *Romeo and Juliet*, *Hamlet*, *King Lear*, *Cymbeline* and *Troilus and Cressida*.

Cluster 3 is largely composed of history plays, namely *Henry IV Henry VI Part 2*, *Henry V*, *Henry VI Part 3* and *Henry VIII*. In addition, it contains the tragedies *Julius Caesar* and *Macbeth*, together with *Pericles*. This cluster appears to group plays with strong historical themes.

Cluster 4 includes both history plays and tragedies, specifically *Henry VI Part 1*, *Henry VI Part 2* and *Richard III*, together with the tragedies *Coriolanus*, *Timon of Athens* and *Antony and Cleopatra*.

Overall, the clustering partly reflects the traditional genre classification. Comedies tend to form a distinct cluster, while history plays and tragedies are distributed across

several clusters.

Table 1: Plays grouped into $k = 4$ clusters obtained by hierarchical clustering with Ward.D2 linkage.

Cluster	**Plays**
1	The Tempest, Two Gentlemen of Verona, The Merry Wives of Windsor, Measure for Measure, The Comedy of Errors, Much Ado About Nothing, Love's Labour's Lost, A Midsummer Night's Dream, The Merchant of Venice, As You Like It, All's Well That Ends Well, Twelfth Night, King John, Titus Andronicus, Othello.
2	The Taming of the Shrew, The Winter's Tale, Richard II, Henry IV Part 1, Troilus and Cressida, Romeo and Juliet, Hamlet, King Lear, Cymbeline.
3	Henry IV Part 2, Henry V, Henry VI Part 3, Henry VIII, Julius Caesar, Macbeth, Pericles.
4	Henry VI Part 1, Henry VI Part 2, Richard III, Coriolanus, Timon of Athens, Antony and Cleopatra.

3 Analysis of Temporal Data

In this section we aim to explore different methods to analyze the beat charts of Shakespeare's plays. In [13], authors use mean and standard deviation as main metrics for comparing beat charts. In our thesis we consider a range of methods and metrics for beat charts' comparison. These include Earth Mover's Distance and Dynamic Time Warping, as well as measures such as roughness and entropy. Furthermore, we perform Archetypal and Archetypoid analysis to identify representative profiles in our data.

3.1 Earth Mover's Distance

A central problem in data analysis is how to quantify the dissimilarity between probability distributions.Intuitively, the Earth Mover's Distance can be viewed as the minimal "work" needed to move piles of soil in order to reshape one distribution into the other, where the cost is proportional to the amount of mass moved times the distance of transportation. Classical norms such as ℓ_1 or ℓ_2 compare distributions bin by bin and therefore ignore the geometry of the underlying space. The *Earth Mover's Distance* (EMD), also known as the 1-Wasserstein distance, addresses this limitation by formulating the problem as one of optimal transport: the goal is to find the minimal cost required to transform one distribution into another [33].

The Optimal Transport Problem

The mathematical foundation of EMD is the theory of optimal transport, introduced by Monge (1781) and reformulated by Kantorovich (1942). A *transport plan* is a probability measure π on the product space $\mathcal{X} \times \mathcal{X}$ whose first marginal equals μ and second marginal equals ν, that is,

$$\pi(A \times \mathcal{X}) = \mu(A), \qquad \pi(\mathcal{X} \times B) = \nu(B)$$

for all measurable sets $A, B \subseteq \mathcal{X}$. The set of all such couplings is denoted by $\Pi(\mu, \nu)$. The optimal transport cost is then defined as

$$C(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} d(x, y)\, d\pi(x, y),$$

where $\Pi(\mu, \nu)$ denotes the set of all transport plans [46].

Definition 3.1 (Optimal transport cost)**.** *A* transport plan *is a probability measure π on the product space $\mathcal{X} \times \mathcal{X}$ whose first marginal equals μ and second marginal equals ν, that is,*

$$\pi(A \times \mathcal{X}) = \mu(A), \qquad \pi(\mathcal{X} \times B) = \nu(B)$$

for all measurable sets $A, B \subseteq \mathcal{X}$. The set of all such couplings is denoted by $\Pi(\mu, \nu)$. Given a distance function $d(x, y)$ on $\mathcal{X}$ (typically the Euclidean distance), the optimal transport cost *between μ and ν is defined as*

$$C(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} d(x, y)\, d\pi(x, y),$$

where $\Pi(\mu, \nu)$ denotes the set of all admissible transport plans [46].

Definition 3.2 (Wasserstein distance [46])**.** *Let $(\mathcal{X}, d)$ be a metric space, and let μ, ν be two probability measures on $\mathcal{X}$. For $p \geq 1$, the* Wasserstein distance of order p *is defined as*

$$W_p(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} d(x, y)^p\, d\pi(x, y) \right)^{1/p}.$$

In particular, for $p = 1$ we obtain the Earth Mover's Distance:

$$EMD(\mu, \nu) = W_1(\mu, \nu).$$

It can be shown that the p-Wasserstein distance satisfies the axioms of a metric on the space of probability measures:

Theorem 3.3 (Properties of the Wasserstein distance [46])**.** *Let $(\mathcal{X}, d)$ be a metric space, and let μ, ν, σ be probability measures on $\mathcal{X}$. For $p \geq 1$, the p-Wasserstein distance $W_p(\cdot, \cdot)$ satisfies the following properties:*

1. ***Non-negativity:***

$$W_p(\mu, \nu) \geq 0.$$

2. ***Identity of indiscernibles:***

$$W_p(\mu, \nu) = 0 \quad \iff \quad \mu = \nu.$$

3. ***Symmetry:***

$$W_p(\mu, \nu) = W_p(\nu, \mu).$$

4. ***Triangle inequality:*** *For any third measure σ,*

$$W_p(\mu, \nu) \leq W_p(\mu, \sigma) + W_p(\sigma, \nu).$$

Consequently, W_p defines a metric on the space of probability measures with finite p-th moment.

Since the Wasserstein distance W_p satisfies all axioms of a metric, the Earth Mover's Distance W_1 provides a well-defined notion of distance between probability distributions. This is not the case for many commonly used divergence measures. For example, the Kullback–Leibler divergence is not symmetric ($D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$ in general) and can even be infinite if the support of P is not contained in the support of Q [26].

From the definition of the 1-Wasserstein distance [Definition 3.2], the transport cost depends directly on the ground metric $d(x, y)$. Moving mass over larger distances therefore increases the total cost proportionally to $d(x, y)$. Due to taking geometry into account, the Earth Mover's Distance is particularly useful in applications such as computer vision, shape analysis amd machine learning [33, 38].

Discrete Distributions

In many applications, probability measures are available only in discrete form, for example as histograms. Formally, this discrete setting is exactly the 1-Wasserstein distance between two empirical probability distributions as introduced in Definition 3.2 when the underlying measures are empirical measures supported on finite sets [6].

Let $x = (X, w)$ and $y = (Y, u)$, where $X = \{x_i\}_{i=1}^m$ and $Y = \{y_j\}_{j=1}^n$ are support points, and $w_i, u_j \geq 0$ are probabilities satisfying $\sum_i w_i = \sum_j u_j = 1$.

A transport plan is represented by a nonnegative *flow matrix* $F = (f_{ij})$, where each entry f_{ij} specifies the amount of probability mass moved from x_i to y_j. The constraints

$$\sum_{j=1}^{n} f_{ij} = w_i, \qquad \sum_{i=1}^{m} f_{ij} = u_j, \qquad f_{ij} \geq 0$$

ensure that all outgoing flow from each x_i equals its weight w_i and all incoming flow to each y_j equals its weight u_j.

Given a distance $d(x_i, y_j)$ (for instance, the absolute difference $|x_i - y_j|$ or Euclidean distance), the discrete Earth Mover's Distance is defined as the minimum total transport

cost:

$$EMD(x,y) = \min_{F \in \mathcal{F}(x,y)} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}\, d(x_i, y_j),$$

where $\mathcal{F}(x,y)$ denotes the set of all feasible transport plans.

Example

Now let us consider two vectors of different lengths.

$$X = \{x_1, x_2\} = \{1, 3\}, \qquad Y = \{y_1, y_2, y_3\} = \{1, 2, 3\}$$

with probability measures

$$w = (\mu_1, \mu_2) = \left(\tfrac{1}{2}, \tfrac{1}{2}\right), \qquad u = (\nu_1, \nu_2, \nu_3) = \left(\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3}\right),$$

so that $\sum_i \mu_i = \sum_j \nu_j = 1$.

We use the ground distance

$$d(x_i, y_j) = |x_i - y_j|.$$

The corresponding cost matrix $D = (d_{ij})$ is

$$D = \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 1 & 0 \end{pmatrix}.$$

A transport plan is represented by a nonnegative matrix

$$F = (f_{ij}) = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \end{pmatrix}, \qquad f_{ij} \geq 0,$$

where f_{ij} is the amount of mass moved from x_i to y_j. The constraints satisfy that the total mass at the input and at the output stays the same.

$$f_{11} + f_{12} + f_{13} = \mu_1 = \tfrac{1}{2}, \tag{1}$$

$$f_{21} + f_{22} + f_{23} = \mu_2 = \tfrac{1}{2}, \tag{2}$$

$$f_{11} + f_{21} = \nu_1 = \tfrac{1}{3}, \tag{3}$$

$$f_{12} + f_{22} = \nu_2 = \tfrac{1}{3}, \tag{4}$$

$$f_{13} + f_{23} = \nu_3 = \tfrac{1}{3}. \tag{5}$$

The total transport cost associated with a plan F is

$$\begin{aligned} C(F) &= \sum_{i=1}^{2}\sum_{j=1}^{3} f_{ij}\, d_{ij} \\ &= 0 \cdot f_{11} + 1 \cdot f_{12} + 2 \cdot f_{13} + 2 \cdot f_{21} + 1 \cdot f_{22} + 0 \cdot f_{23} \\ &= f_{12} + 2f_{13} + 2f_{21} + f_{22}. \end{aligned}$$

The discrete Earth Mover's Distance between $x = (X, w)$ and $y = (Y, u)$ is obtained by minimizing $C(F)$ over all F satisfying (1)–(5) and $f_{ij} \geq 0$.

Solution 1: Intuitive

Here is how we can solve this linear programming problem analytically in intuitive way, due to the fact that we have five equations with six unknown parameters. From (3),

$$f_{11} + f_{21} = \tfrac{1}{3}.$$

Since $d_{21} = 2$, we choose $f_{21} = 0$, which implies

$$f_{11} = \tfrac{1}{3}.$$

Substituting into (1) gives

$$f_{12} + f_{13} = \tfrac{1}{2} - \tfrac{1}{3} = \tfrac{1}{6}. \tag{6}$$

Similarly, from (5),

$$f_{13} + f_{23} = \tfrac{1}{3}.$$

Since $d_{23} = 0$ and $d_{13} = 2$, it is optimal to choose

$$f_{13} = 0, \qquad f_{23} = \tfrac{1}{3}.$$

Then (2) yields

$$f_{22} = \tfrac{1}{2} - f_{21} - f_{23} = \tfrac{1}{2} - 0 - \tfrac{1}{3} = \tfrac{1}{6}.$$

Using (6) and $f_{13} = 0$ we obtain

$$f_{12} = \tfrac{1}{6}.$$

Finally, (4) is satisfied automatically:

$$f_{12} + f_{22} = \tfrac{1}{6} + \tfrac{1}{6} = \tfrac{1}{3}.$$

Therefore the optimal transport plan is

$$F^* = \begin{pmatrix} \frac{1}{3} & \frac{1}{6} & 0 \\ 0 & \frac{1}{6} & \frac{1}{3} \end{pmatrix}.$$

The corresponding Earth Mover's Distance between x and y, is

$$\mathrm{EMD}(x, y) = C(F^*) = 1 \cdot \frac{1}{6} + 1 \cdot \frac{1}{6} = \frac{1}{3}.$$

Solution 2: Numerical optimization

As stated above, this solution is intuitive, yet good for understanding. Solving this task as a classical linear programming problem yields the same optimal flows and EMD. We consider the transport variables in the vector form

$$(f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}),$$

and minimize the total transport cost subject to the constraints (1)–(5).

Using the cost matrix D the problem can be solved using standard numerical optimization. In our case, we used Python implementation based on linear programming.

The optimal transport plan is obtained as

$$F = \begin{pmatrix} 0.3333 & 0.1667 & 0 \\ 0 & 0.1667 & 0.3333 \end{pmatrix},$$

and the resulting Earth Mover's Distance is

$$\mathrm{EMD} = 0.3333.$$

Complete script used for the LP solution can be found in [27].

Solution 3: Graphical

The problem can be also written as a linear programming task with unknown variables f_{22} and f_{23}. The remaining flows are then expressed as follows

$$f_{11} = f_{22} + f_{23} - \frac{1}{6}, \qquad f_{12} = \frac{1}{3} - f_{22}, \qquad f_{13} = \frac{1}{3} - f_{23}, \qquad f_{21} = \frac{1}{2} - f_{22} - f_{23},$$

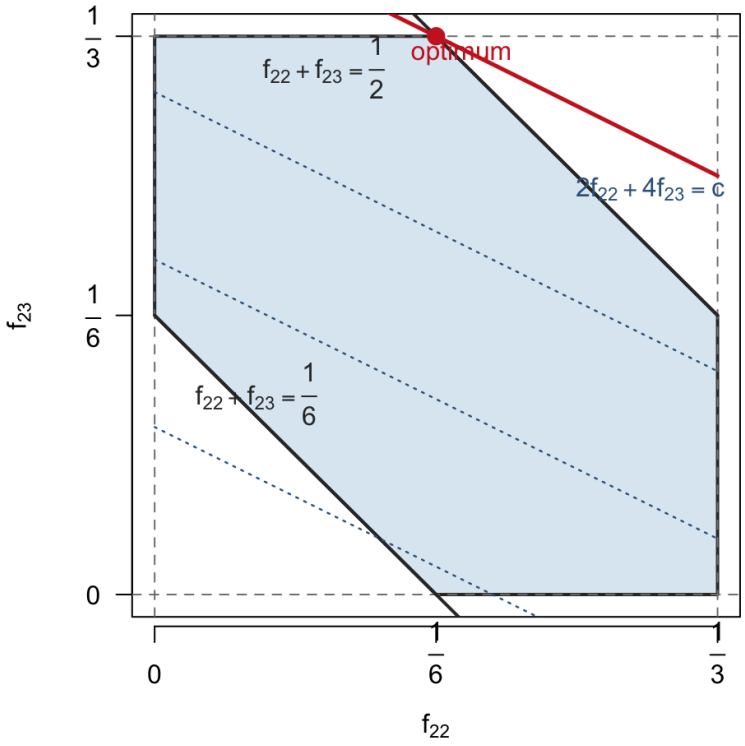


Figure 6: Graphical solution of the linear programming formulation of the EMD example

and the task is

$$\begin{aligned} \max \quad & 2f_{22} + 4f_{23} \\ \text{s.t.} \quad & \frac{1}{2} - f_{22} - f_{23} \geq 0, \\ & -\frac{1}{6} + f_{22} + f_{23} \geq 0, \\ & 0 \leq f_{22} \leq \frac{1}{3}, \\ & 0 \leq f_{23} \leq \frac{1}{3}. \end{aligned}$$

This is a problem with two variables, thus it can be solved graphically. The feasible region and the optimum point $(f_{22}, f_{23}) = (\frac{1}{6}, \frac{1}{3})$ are shown in Figure 6.

Substituting these optimal values into the flow variables yields

$$(f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}) = \left(\frac{1}{3},\ \frac{1}{6},\ 0,\ 0,\ \frac{1}{6},\ \frac{1}{3}\right),$$

and the resulting distance

$$\text{EMD} = \frac{1}{3}.$$

Application of EMD to Beat Charts of Shakespeare Plays

In addition to its theoretical properties, the Earth Mover's Distance was applied in this thesis to compare beat charts of Shakespeare's plays.

Since beat charts are discrete time series of varying length, each beat chart was interpreted as a histogram-like distribution, and the Earth Mover's Distance was used to compute pairwise dissimilarities between plays. This choice was motivated by the fact that EMD accounts for the geometry of the underlying space and provides a measure of similarity even when the beat charts are not equal in length.

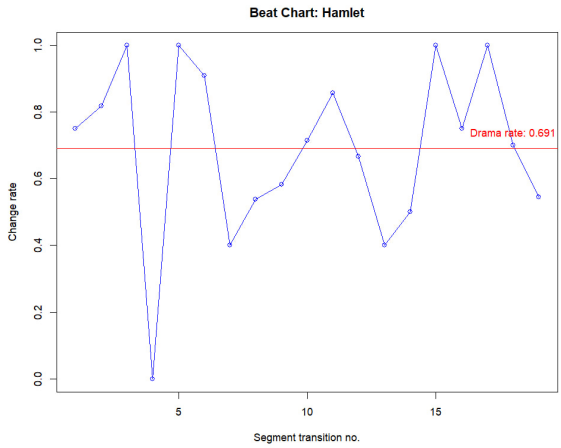


Figure 7: Beat chart of *Hamlet*. The x-axis shows the segment transition number, and the y-axis shows the change rate between consecutive segments.

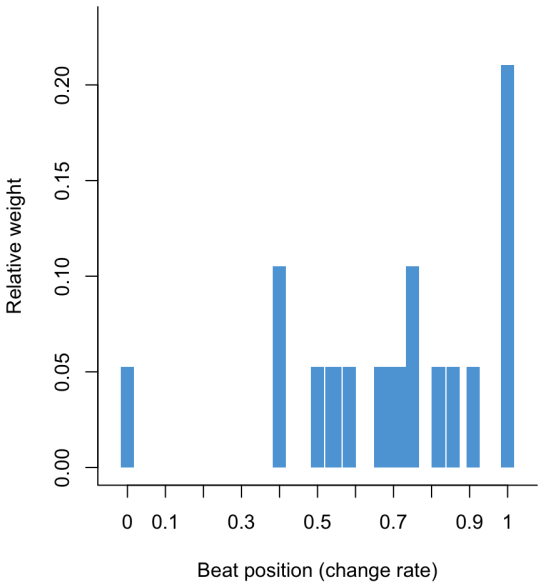


Figure 8: Histogram representation of the beat chart on the left. The x-axis shows beat positions (change rate values), and the y-axis shows their relative weights.

The computation was implemented in R in two equivalent ways. First, EMD was calculated via a *Kantorovich formulation* as a linear programming problem solved by convex optimization (using the package `CVXR`). Second method is applying a standard EMD function from the `emdist` package directly to the data. The use of EMD in this setting follows the same principle as in image retrieval and shape analysis, where distributions with possibly different supports are compared by evaluating the minimal cost of transforming one

into the other [33, 38]. Here, the distributions correspond to the beat chart values of Shakespeare's plays.

The Figure 7 illustrates a beat chart as a time series of change rates, while Figure 8 shows its its normalized histogram representation, where the x-axis corresponds to change-rate values (beat positions) and the y-axis to their relative frequencies. This histogram serves as the input probability distribution used in the EMD computation.

Pairwise EMD values were then computed between all plays, resulting in a distance matrix. This matrix was further used as an input for hierarchical clustering. The obtained clustering results are summarized in Table 2. The obtained dendrogram is illustrated in Figure 9.

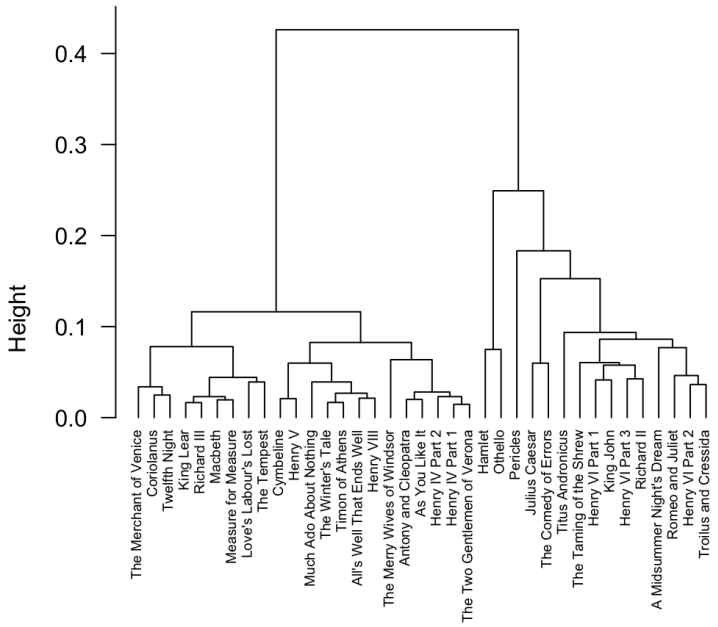


Figure 9: Hierarchical clustering dendrogram based on EMD distances.

Cluster 1 contains a mixture of comedies, tragedies and history plays, including *A Midsummer Night's Dream*, *Romeo and Juliet*, *Titus Andronicus* and several history plays such as *Henry VI Parts 1-3* and *Richard II*. It is interesting to notice, that this cluster contains many plays from Shakespeare's early period.

Cluster 2 forms the largest group of twenty one plays, including *Much Ado About Nothing*, *As You Like It*, *Macbeth*, *King Lear*, *Coriolanus* and *The Tempest*. This cluster includes plays from all three genres.

Cluster 3 consists only of two plays, *Hamlet* and *Othello*. Both are major tragedies, written in the beginning of the 17th century (1600 and 1604 respectively).

Finally, Cluster 4 contains only *Pericles*, indicating that its beat chart differs substantially from all other plays. This can be explained by the fact that the Earth Mover's Distance captures differences in the values of the beat charts, and *Pericles'* beat chart has high segment change rate values due to the structure of the play, where the main character travels between different locations, which leads to changes in the set of interacting characters. As a result, *Pericles* forms a separate cluster. *Pericles* is also widely considered to be a collaborative play, with scholars suggesting that the first acts were written by George Wilkins and the later ones by Shakespeare [18].

Table 2: Plays grouped into $k = 4$ clusters obtained using EMD method.

Cluster	**Plays**
1	A Midsummer Night's Dream, Henry VI Part 1, Henry VI Part 2, Henry VI Part 3, Julius Caesar, King John, Richard II, Romeo and Juliet, The Comedy of Errors, The Taming of the Shrew, Titus Andronicus, Troilus and Cressida.
2	All's Well That Ends Well, Antony and Cleopatra, As You Like It, Coriolanus, Cymbeline, Henry IV Part 1, Henry IV Part 2, Henry V, Henry VIII, King Lear, Love's Labour's Lost, Macbeth, Measure for Measure, Much Ado About Nothing, Richard III, The Merchant of Venice, The Merry Wives of Windsor, The Tempest, The Two Gentlemen of Verona, The Winter's Tale, Timon of Athens, Twelfth Night.
3	Hamlet, Othello.
4	Pericles.

3.2 Dynamic Time Warping Method for Time Series of Different Length

An important task in time series analysis is to quantify the similarity between sequences that vary in length. Classical distances such as the Euclidean norm compare time series point by point and therefore are more effective for series which are equal in length. DTW addresses this problem by warping the sequences in a non-linear way, so that they can be aligned even if they evolve at different speeds. Originally, DTW was introduced in the context of automatic speech recognition, where it was used to compare spoken words with varying speed or intonation [34]. Since then, the algorithm has found wide application in diverse domains such as handwriting and signature verification, gesture and motion recognition, medical signal analysis and financial time series [2, 37].

Mathematical Formulation of DTW

In their recent work, Bringmann et al. [2] provide an appealing intuitive representation of Dynamic Time Warping. Imagine a dog walking along one curve $X = (x_1, \ldots, x_n)$ and its owner walking along another curve $Y = (y_1, \ldots, y_m)$. Both start at the beginning of their respective curves, that is, at (x_1, y_1), and must eventually reach the end point (x_n, y_m). At each step, either the dog, the owner, or both may move forward by one position, but they are not allowed to move backwards. This restriction guarantees a consistent forward traversal of the two sequences. The *DTW distance* is then defined as the minimal total distance that the dog and the owner must cover while walking along their curves. An examplanary illustration of the DTW alignment between two sequences is shown in Figure 10.

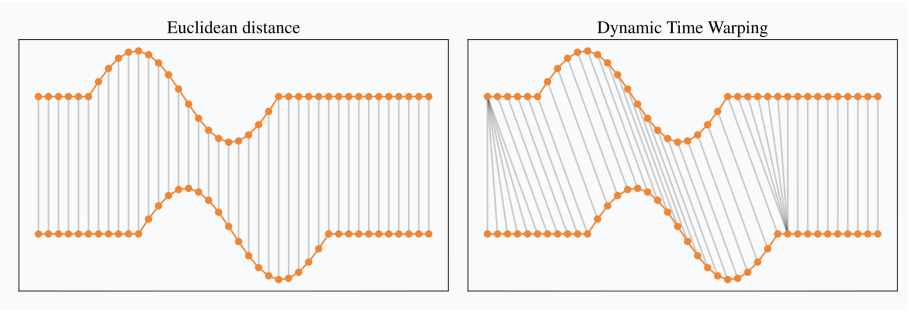


Figure 10: Illustration of Dynamic Time Warping alignment between two time series. Source: R. Tavenard, https://rtavenar.github.io/blog/dtw.html.

Formally, let $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_M)$ be two time series of lengths $N, M \in \mathbb{N}$, respectively. We assume that all elements take values from a feature space $\mathcal{F}$. To compare two elements $x \in \mathcal{F}$ and $y \in \mathcal{F}$, we define a local distance measure

$$c(x, y) : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0},$$

where $c(x, y)$ is typically the squared Euclidean distance or another valid metric. Intuitively, the value of $c(x, y)$ is small if x and y are similar, and large otherwise. Evaluating this measure for all pairs (i, j) yields the cost matrix $C \in \mathbb{R}^{N \times M}$ with entries $C(i, j) = c(x_i, y_j)$.

To formalize the alignment process, we introduce the notion of a *warping path*, which specifies how elements of the two sequences are matched under certain constraints.

Definition 3.4 (Warping Path [23]). *Let $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_M)$ be two time series of lengths $N, M \in \mathbb{N}$, respectively. A* warping path *is a sequence $p = (p_1, \ldots, p_L)$ with $p_\ell = (n_\ell, m_\ell) \in [1 : N] \times [1 : M]$ for $\ell \in \{1, \ldots, L\}$, where $L \geq \max(N, M)$, satisfying:*

1. ***Boundary condition:*** *$p_1 = (1, 1)$ and $p_L = (N, M)$.*
2. ***Monotonicity:*** *$n_{\ell+1} \geq n_\ell$ and $m_{\ell+1} \geq m_\ell$ for all $\ell \in \{1, \ldots, L-1\}$.*
3. ***Continuity (step size condition):*** *$(n_{\ell+1} - n_\ell,\, m_{\ell+1} - m_\ell) \in \{(1, 0), (0, 1), (1, 1)\}$ for all $\ell \in \{1, \ldots, L-1\}$.*

The cost of a warping path p is then defined as

$$C_p(X, Y) = \sum_{\ell=1}^{L} c(x_{n_\ell}, y_{m_\ell}).$$

The constraints on a warping path have natural interpretations: the boundary condition enforces that the first and last elements of X and Y are aligned to each other; the monotonicity condition ensures faithful ordering, i.e., if an element in X precedes another, the same order holds in Y; finally, the continuity condition (step size condition) expresses that no elements in X or Y are skipped, and all pairs (n_ℓ, m_ℓ) are pairwise distinct [23].

To quantify the similarity between two time series, we do not consider the cost of a single warping path, but rather the minimal cost over all admissible paths. This leads to the following definition.

Definition 3.5 (Dynamic Time Warping Distance [37]). *Let $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_M)$ be two time series. An* optimal warping path *p^* between X and Y is the warping path that minimizes the total cost among all warping paths. The Dynamic Time Warping distance between X and Y is then defined as*

$$DTW(X,Y) := C_{p^*}(X,Y) = \min\{C_p(X,Y) \mid p \text{ is an } (N,M)\text{-warping path}\},$$

where the minimum is taken over all warping paths p that satisfy the boundary, monotonicity, and continuity constraints.

This formal definition can be directly related to the dog-and-owner analogy. A warping path shows how the two sequences are traversed together, while the DTW distance measures the smallest possible total difference during this traversal.

Dynamic Programming Formulation

The optimal warping path can be computed efficiently using dynamic programming. Let $D(i,j)$ denote the minimal cumulative cost of aligning the two sequences $(x_1, \ldots, x_i)$ and $(y_1, \ldots, y_j)$. Formally, $D(i,j)$ is defined as

$$D(i,j) = c(x_i, y_j) + \min\big\{D(i-1,j),\ D(i,j-1),\ D(i-1,j-1)\big\},$$

with the initialization

$$D(1,1) = c(x_1, y_1),$$

and boundary conditions

$$D(i,0) = D(0,j) = \infty \quad \text{for all } i,j > 0.$$

The value $D(i,j)$ represents the minimal cost of any admissible warping path that ends at (i,j). Consequently, the DTW distance between X and Y is given by

$$DTW(X,Y) = D(N,M),$$

where N and M are the lengths of the two time series.

The recursive definition above naturally leads to a dynamic programming algorithm for computing the DTW distance. The algorithm iteratively fills the accumulated cost matrix $D \in \mathbb{R}^{N \times M}$ according to the recursion [34, 37].

Properties and Extensions of Dynamic Time Warping

Throughout this section we assume a symmetric, non-negative cost function $c : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0}$ with $c(x, x) = 0$ and $c(x, y) = c(y, x)$ (e.g., $c(x, y) = |x - y|$). Under these assumptions, we now establish several basic properties of the DTW distance: we show non-negativity and symmetry, and provide simple counterexamples showing that identity of indiscernibles and the triangle inequality do not hold in general [23, 37].

Proposition 3.6 (Non-negativity and symmetry)**.** *For any two time series $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_M)$, the DTW distance satisfies:*

1. ***Non-negativity:*** $\mathrm{DTW}(X, Y) \geq 0$.

2. ***Symmetry:*** $\mathrm{DTW}(X, Y) = \mathrm{DTW}(Y, X)$.

Proof. (1) Every admissible warping path p has cost $C_p(X, Y) = \sum_{\ell=1}^{L} c(x_{n_\ell}, y_{m_\ell}) \geq 0$ because $c \geq 0$. Taking the minimum over all paths yields $\mathrm{DTW}(X, Y) \geq 0$.

(2) Let $p = ((n_1, m_1), \ldots, (n_L, m_L))$ be any admissible path for (X, Y). Then $\tilde{p} = ((m_1, n_1), \ldots, (m_L, n_L))$ is an admissible path for (Y, X). By symmetry of c, $C_{\tilde{p}}(Y, X) = \sum_{\ell=1}^{L} c(y_{m_\ell}, x_{n_\ell}) = \sum_{\ell=1}^{L} c(x_{n_\ell}, y_{m_\ell}) = C_p(X, Y)$. After minimizing on both sides we obtain $\mathrm{DTW}(X, Y) = \mathrm{DTW}(Y, X)$. □

Proposition 3.7 (Failure of identity of indiscernibles)**.** *There exist distinct time series $X \neq Y$ with $\mathrm{DTW}(X, Y) = 0$.*

Proof. Consider $X = (0, 1)$ and $Y = (0, 0, 1)$ with $c(x, y) = |x - y|$. The path $p = ((1, 1), (1, 2), (2, 3))$ is admissible and aligns equal values at each step, hence $C_p(X, Y) = 0 + 0 + 0 = 0$. Therefore $\mathrm{DTW}(X, Y) \leq 0$, and by Lemma 3.6(1), $\mathrm{DTW}(X, Y) = 0$ although $X \neq Y$. □

Proposition 3.8 (Failure of triangle inequality)**.** *In general, DTW does not satisfy the triangle inequality; that is, there exist X, Y, Z with*

$$\mathrm{DTW}(X, Z) \; > \; \mathrm{DTW}(X, Y) + \mathrm{DTW}(Y, Z).$$

Proof. Let $c(x, y) = |x - y|$, $X = (0)$, $Y = (1)$, and $Z = (1, 1)$. Then $\mathrm{DTW}(X, Y) = |0 - 1| = 1$ and $\mathrm{DTW}(Y, Z) = 0$ (align the single 1 in Y with either copy in Z and duplicate the match). For $\mathrm{DTW}(X, Z)$ any admissible path must match 0 with both 1's, giving cost $|0-1|+|0-1| = 2$. Hence $2 > 1+0$, which violates the triangle inequality. □

Because identity of indiscernibles and the triangle inequality fail in general, DTW is not a metric on time series. Nevertheless, it is widely used in practice due to its ability to capture elastic similarity under timing differences [23, 37].

Remark: Several modifications of DTW have been proposed to improve efficiency or robustness: global path constraints such as the Sakoe–Chiba band [34], multiscale and approximate algorithms like FastDTW [35], and extensions to multivariate time series [23]. A detailed overview can be either found in [37, 2].

Application of DTW to the Beat Charts of Shakespeare's Plays

In our setting, the beat charts naturally have different lengths, and DTW is therefore an appropriate dissimilarity measure because it permits nonlinear time alignments.

Let $\mathcal{P} = \{P_1, \ldots, P_{37}\}$ denote the set of plays and let $X^{(i)} = (x_1^{(i)}, \ldots, x_{N_i}^{(i)})$ be the beat chart of play P_i, where $N_i \in \mathbb{N}$ is its length. We computed pairwise DTW distances between all beat charts in R using the `dtw` package. Concretely, for each pair (i, j) we applied the `dtw()` function from the `dtw` package on $X^{(i)}$ and $X^{(j)}$ and recorded the optimal alignment cost as

$$D_{ij} \;=\; \mathrm{DTW}\big(X^{(i)}, X^{(j)}\big), \qquad i, j \in \{1, \ldots, 37\}.$$

This yields a symmetric distance matrix $D \in \mathbb{R}^{37\times 37}$ with zeros on the diagonal, i.e., $D_{ii} = 0$ for all i, and $D_{ij} = D_{ji}$.

The resulting DTW distance matrix was then used as input to agglomerative hierarchical clustering. We fixed the number of clusters to $k = 4$ by cutting the dendrogram at the level that produces four groups. The obtained dendrogram is shown on Figure 11.

The clustering results obtained using DTW-based distances are summarized in Table 3.

Similarly to the EMD clustering, the DTW results contain two large mixed clusters: one consisting of thirteen plays and another of twenty one plays. Both clusters include plays from all three genres and from different periods of Shakespeare's career.

Cluster 3 contains only *Henry VI Part 1.* This play appears as an "outlier" in context of the DTW distance, which suggests that its beat chart differs significantly from those of the other plays.

Finally, Cluster 4 consists solely of *Pericles.* Similarly to the EMD clustering, this play forms a separate cluster, again indicating that this play is substantially different from the rest of the corpus.

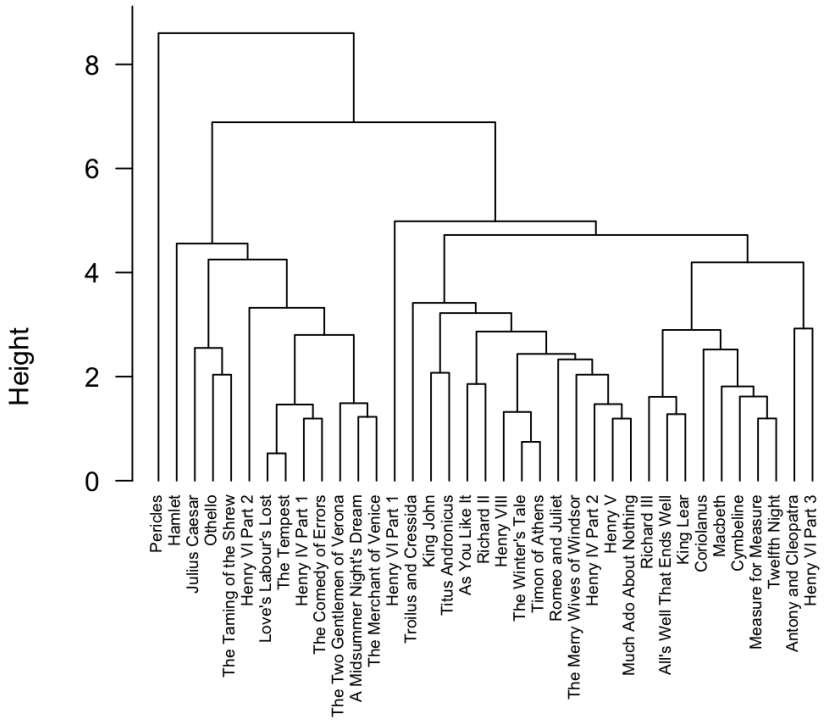


Figure 11: Hierarchical clustering dendrogram based on DTW distances.

Table 3: Plays grouped into $k = 4$ clusters obtained using DTW.

Cluster	**Plays**
1	A Midsummer Night's Dream, Hamlet, Henry IV Part 1, Henry VI Part 2, Julius Caesar, Love's Labour's Lost, Othello, The Comedy of Errors, The Merchant of Venice, The Taming of the Shrew, The Tempest, The Two Gentlemen of Verona.
2	All's Well That Ends Well, Antony and Cleopatra, As You Like It, Coriolanus, Cymbeline, Henry IV Part 2, Henry V, Henry VI Part 3, Henry VIII, King John, King Lear, Macbeth, Measure for Measure, Much Ado About Nothing, Richard II, Richard III, Romeo and Juliet, The Merry Wives of Windsor, The Winter's Tale, Timon of Athens, Titus Andronicus, Troilus and Cressida, Twelfth Night.
3	Henry VI Part 1.
4	Pericles.

3.3 Roughness and Spectral Entropy

In this subsection, we describe the indicators *roughness* and *spectral entropy*, which are used to characterize the time series. We begin with the definition of roughness.

Intuitively, roughness quantifies how strongly a curve fluctuates over its domain. Roughness is defined as the integral of squared second derivative of a function. This metric measures the degree of curvature and therefore the variability of the function [30].

Definition 3.9 (Roughness [30])**.** *Let $x : [a, b] \to \mathbb{R}$ be a twice continuously differentiable function. The roughness of x, denoted by $r(x)$, is defined as*

$$r(x) = \int_a^b (x''(t))^2 \, dt. \tag{7}$$

The second derivative $x''(t)$ represents the local curvature of the function. A linear function has zero second derivative and therefore zero roughness. Functions with large curvature on some subintervals produce larger values of $r(x)$. Roughness is also referred to as the total curvature of a function, due to the fact that it is represented as an integral over the whole function domain. The definition can either be generalized to higher-order derivatives (see Chapter 5 in [30]).

In R, the total curvature can be computed using the function `roughness()` from the `seewave` package. In many applications, as well as in our case, we do not observe continuous functions but discrete time series. This function takes a numeric vector as input and returns a single numeric value as output, which is the roughness value of the input vector, which is calculated by central numeric approximation of the second derivative [42].

In our case, the `roughness()` function was applied to all 37 beat charts. Each beat chart was converted to a common length of 100 points. This was done by rescaling the original series to the interval [0,1] and computing values at 100 equally spaced points using the R function `approx()`, which makes linear interpolation. The resulting series of length 100 were then used for computing the roughness values. In our setting, higher roughness values are associated with frequent changes between scenes, while lower roughness values indicate smoother transitions. The resulting roughness values were then used as input for k-means clustering with $k = 4$. Figure 12 visualizes the clustering results obtained from the roughness values.

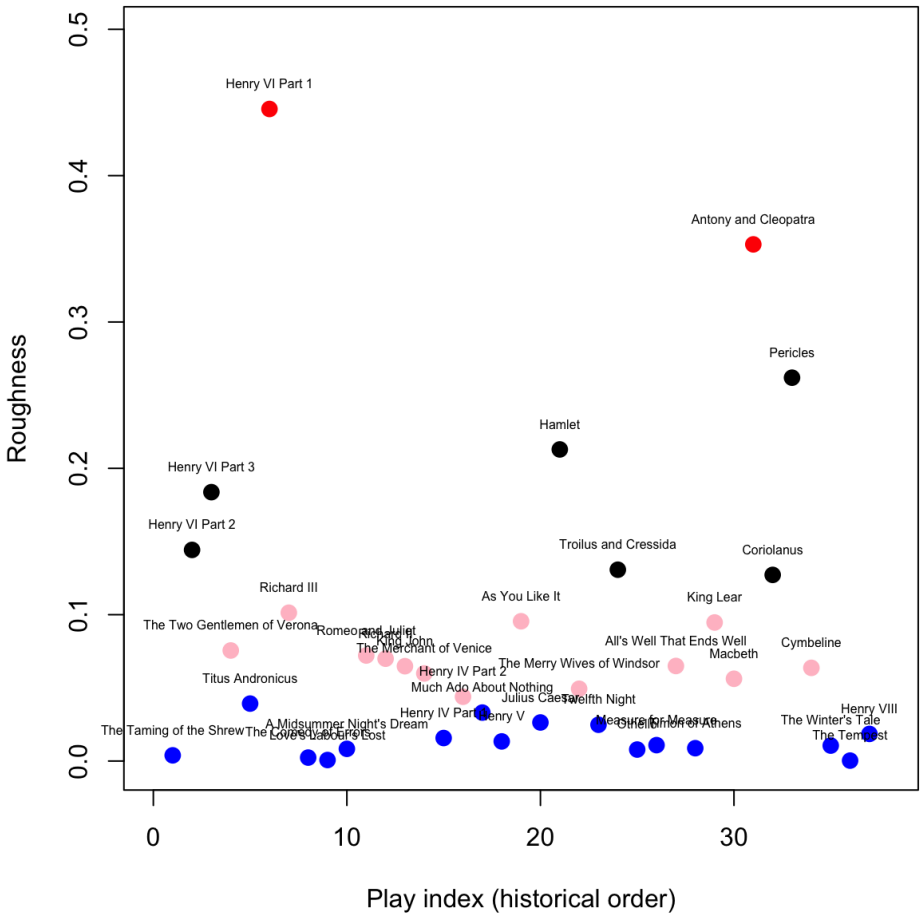


Figure 12: k-means clustering ($k = 4$) based on roughness values.

Now let us describe the spectral entropy. Spectral entropy measures how the energy of a signal is distributed across its frequency components. It is based on the concept of Shannon entropy applied to the spectrum [20].

Let $x(t)$ be a discrete time series and let P_i denote the power at frequency i, obtained from the Fourier transformation of the signal. Let N be the total number of frequency bins.

We first normalize the power spectrum so that it can be interpreted as a probability distribution. We define

$$p(i) = \frac{P_i}{\sum_{j=1}^{N} P_j}, \quad i = 1, \ldots, N, \sum_{i=1}^{N} p(i) = 1.$$

where $\mathbf{p} = (p_1, \ldots, p_N)$ is the probability mass function of the normalized power spectrum.

The non-normalized spectral entropy [20], also reffered to as the Shannon entropy, is defined as

$$H = -\sum_{i=1}^{N} p(i) \log_2 p(i). \tag{8}$$

To make the entropy comparable across spectra with different numbers of frequency bins, it is normalized by dividing by $\log_2(N)$ which is the maximum possible entropy. The normalized spectral entropy is therefore defined as

$$S = -\frac{\sum_{i=1}^{N} p(i) \log_2 p(i)}{\log_2(N)}, 0 \leq S \leq 1. \tag{9}$$

According to the documentation of the `seewave` package [42] , a value close to 0 corresponds to a very regular spectrum (for example, a pure tone with a single dominant peak), whereas a value close to 1 corresponds to a noisy signal.

In R spectral entropy is computed using the function `sh()` from the `seewave` package, which returns the normalized Shannon spectral entropy scaled between 0 and 1.

Similarly to the roughness measure, we computed the spectral entropy for all 37 beat charts. The obtained spectral entropy values were then used as input for k-means clustering with $k = 4$.

The resulting clustering based on spectral entropy values is shown in Figure 13.

In addition, we decided to cluster the plays based on both roughness and spectral entropy. For this purpose, we applied k-means clustering in the two-dimensional space defined by these two indicators. The resulting clustering can be observed in Figure 14.

Tables 4– 6 summarize the resulting k-means cluster assignments based on roughness alone, spectral entropy alone and their combined two-dimensional representation, respectively.

In case of two-dimensional clustering based on roughness and spectral entropy, Cluster 1 contains nine plays and is dominated by comedies, including *Love's Labour's Lost*, *Measure for Measure*, *The Comedy of Errors*, *The Taming of the Shrew*, *The Tempest* and *The Winter's Tale*. It also includes two history plays (*Henry IV Part 1* and *Henry V*) and the tragedy *Timon of Athens*.

Cluster 2 is the largest group of fifteen plays and contains plays from all three genres, including tragedies such as *Hamlet* and *Romeo and Juliet*, comedies such as *As You Like*

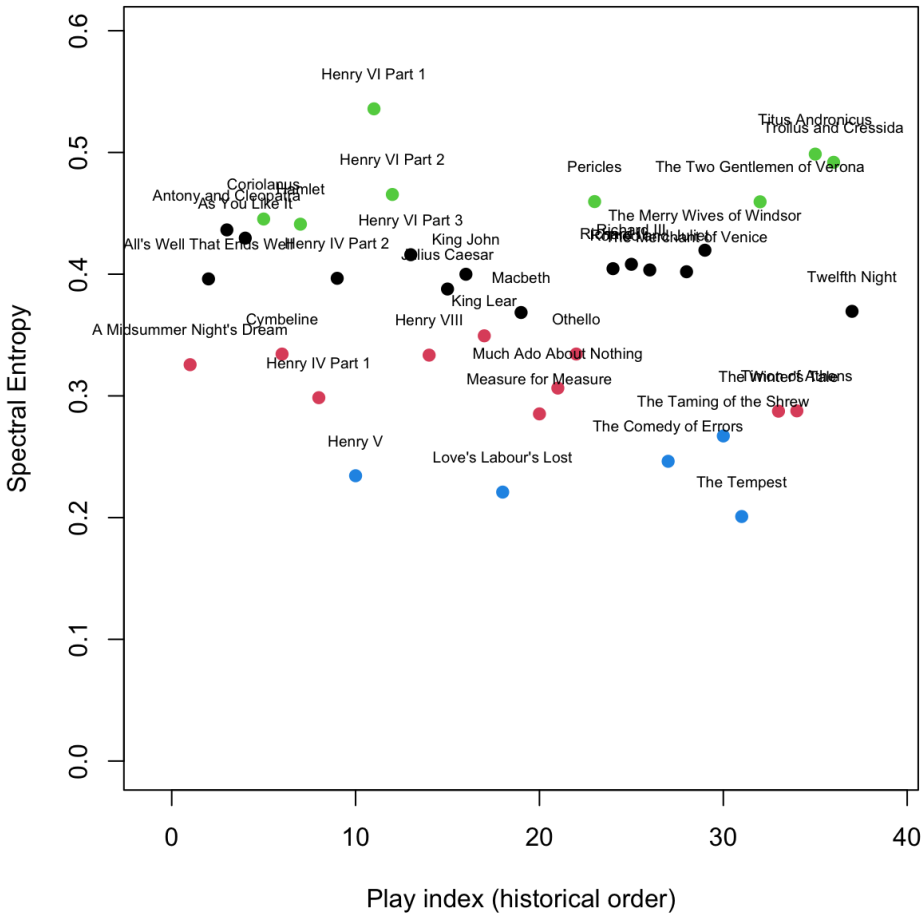


Figure 13: k-means clustering ($k = 4$) based on spectral entropy values.

It and *The Merry Wives of Windsor*, and several history plays including *Henry VI Parts 2–3* and *Richard II–III*.

Cluster 3 also contains a mixture of genres but includes several major tragedies such as *King Lear*, *Macbeth*, *Othello* and *Julius Caesar*, together with comedies like *A Midsummer Night's Dream* and *Twelfth Night*.

Finally, Cluster 4 contains only three plays: *Antony and Cleopatra*, *Pericles*, and *Henry VI Part 1*. One can observe that similarly as in DTW method, where *Pericles* and *Henry VI Part 1* were separated from the main clusters, these two plays belong together in this method as well.

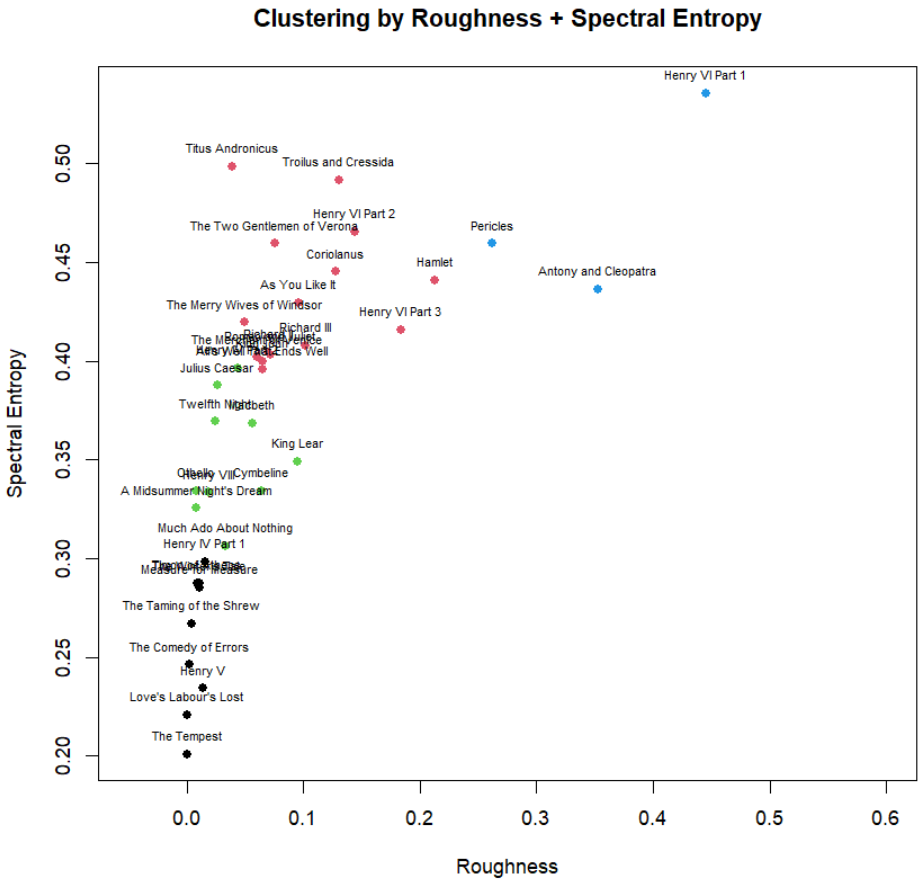


Figure 14: k-means clustering ($k = 4$) in the two-dimensional space defined by roughness and spectral entropy.

Table 4: Plays grouped into $k = 4$ clusters obtained by k-means clustering based on roughness values.

Cluster	**Plays**
1	A Midsummer Night's Dream, Henry IV Part 1, Henry V, Henry VIII, Julius Caesar, Love's Labour's Lost, Measure for Measure, Much Ado About Nothing, Othello, Romeo and Juliet, The Comedy of Errors, The Taming of the Shrew, The Tempest, The Winter's Tale, Timon of Athens, Twelfth Night.
2	Coriolanus, Hamlet, Henry VI Part 2, Henry VI Part 3, Pericles, Titus Andronicus, Troilus and Cressida.
3	All's Well That Ends Well, As You Like It, Cymbeline, Henry IV Part 2, King John, King Lear, Macbeth, Richard II, Richard III, The Merchant of Venice, The Merry Wives of Windsor, The Two Gentlemen of Verona.
4	Antony and Cleopatra, Henry VI Part 1.

Table 5: Plays grouped into $k = 4$ clusters obtained by k-means clustering based on spectral entropy values.

Cluster	**Plays**
1	All's Well That Ends Well, Antony and Cleopatra, As You Like It, Henry IV Part 2, Henry VI Part 3, Julius Caesar, King John, Macbeth, The Merchant of Venice, The Merry Wives of Windsor, Richard II, Richard III, Romeo and Juliet, Twelfth Night.
2	A Midsummer Night's Dream, Cymbeline, Henry IV Part 1, Henry VIII, King Lear, Measure for Measure, Much Ado About Nothing, Othello, The Winter's Tale, Timon of Athens.
3	Coriolanus, Hamlet, Henry VI Part 1, Henry VI Part 2, Pericles, The Two Gentlemen of Verona, Troilus and Cressida, Titus Andronicus.
4	Henry V, Love's Labour's Lost, The Comedy of Errors, The Taming of the Shrew, The Tempest.

Table 6: Plays grouped into $k = 4$ clusters obtained by k-means clustering in the two-dimensional space defined by roughness and spectral entropy.

Cluster	**Plays**
1	Henry IV Part 1, Henry V, Love's Labour's Lost, Measure for Measure, The Comedy of Errors, The Taming of the Shrew, The Tempest, The Winter's Tale, Timon of Athens.
2	All's Well That Ends Well, As You Like It, Coriolanus, Hamlet, Henry VI Part 2, Henry VI Part 3, King John, Richard II, Richard III, Romeo and Juliet, The Merchant of Venice, The Merry Wives of Windsor, The Two Gentlemen of Verona, Titus Andronicus, Troilus and Cressida.
3	A Midsummer Night's Dream, Cymbeline, Henry IV Part 2, Henry VIII, Julius Caesar, King Lear, Macbeth, Much Ado About Nothing, Othello, Twelfth Night.
4	Antony and Cleopatra, Pericles, Henry VI Part 1.

3.4 Archetypal Analysis

Archetypal analysis (AA), introduced by Cutler and Breiman [8], aims to describe a dataset by a small number of representative extreme profiles, called archetypes. Each observation is written as a convex combination of these archetypes, and each archetype is itself a convex combination of observed data points. Archetypoid analysis (ADA) is a modification in which the extremal profiles are required to be actual observations, called archetypoids [48].

Since the original formulation, AA and ADA have been extended to functional data [10] and to ordinal data [12], among other settings. They have been applied in many fields, for example to texture segmentation in image analysis [3], in the context of statistical data analysis [48], economics and finance [39], as well as in other machine-learning problems [21].

Mathematical Formulation for AA algorithm

Let $x_1, \dots, x_n \in \mathbb{R}^m$ be n m-dimensional data points, and let $X \in \mathbb{R}^{n \times m}$ denote the data matrix whose i-th row is $x_i^\top$. Let $K \geq 1$ be a number of archetypes. Archetypal analysis seeks K vectors $z_1, \dots, z_K \in \mathbb{R}^m$ (the archetypes) such that

1. each archetype z_k is a convex combination of the data points;
2. each observation x_i is approximated by a convex combination of the archetypes.

Formally, archetypes can be formulated as

$$z_k = \sum_{j=1}^{n} \beta_{kj} x_j, \qquad \beta_{kj} \geq 0, \quad \sum_{j=1}^{n} \beta_{kj} = 1, \quad k = 1, \dots, K, \tag{10}$$

as proposed in [8]. For fixed archetypes $z_1, \dots, z_K$, each observation is approximated by

$$\hat{x}_i = \sum_{k=1}^{K} \alpha_{ik} z_k, \qquad \alpha_{ik} \geq 0, \quad \sum_{k=1}^{K} \alpha_{ik} = 1, \quad i = 1, \dots, n, \tag{11}$$

that is, $\hat{x}_i$ is constrained to lie in the convex hull of the archetypes.

The archetypes and the coefficients α_{ik} and β_{kj} are determined by minimising the residual sum of squares

$$\mathrm{RSS}_K(\alpha, \beta) = \sum_{i=1}^{n} \left\| x_i - \sum_{k=1}^{K} \alpha_{ik} z_k \right\|^2 = \sum_{i=1}^{n} \left\| x_i - \sum_{k=1}^{K} \alpha_{ik} \sum_{j=1}^{n} \beta_{kj} x_j \right\|^2 \tag{12}$$

subject to the constraints (10)–(11) [8].

In other words, AA algorithm returns three matrices. First, the K archetypes $z_1, \ldots, z_K$ are the rows of a $K \times m$ matrix Z. Second, a matrix $A = (\alpha_{ik}) \in \mathbb{R}^{n \times K}$ contains the coefficients that approximate each observation x_i as a convex combination of the archetypes ($\hat{x}_i = \sum_{k=1}^{K} \alpha_{ik} z_k$). Third, a matrix $B = (\beta_{kj}) \in \mathbb{R}^{K \times n}$ contains the coefficients that define each archetype as a convex combination of the data points [8].

The constraints can be written as

$$A \geq 0, \quad A\mathbf{1}_K = \mathbf{1}_n, \qquad B \geq 0, \quad B\mathbf{1}_n = \mathbf{1}_K,$$

where $\mathbf{1}_r$ denotes the r-dimensional column vector of ones. The K archetypes form the rows of the matrix

$$Z = BX \in \mathbb{R}^{K \times m},$$

and the approximation of X is given by

$$\hat{X} = AZ = ABX.$$

Geometric interpretation and basic properties

Let $\mathrm{co}(S)$ denote the convex hull of a set $S \subset \mathbb{R}^m$. Because each archetype z_k is a convex combination of the data points ((10)), we have

$$z_k \in \mathrm{co}\{x_1, \ldots, x_n\}, \qquad k = 1, \ldots, K.$$

Similarly, each reconstructed point $\hat{x}_i$ is a convex combination of the archetypes ((11)), so

$$\hat{x}_i \in \mathrm{co}\{z_1, \ldots, z_K\}, \qquad i = 1, \ldots, n.$$

In other words, archetypal analysis approximates the cloud of data points by a convex hull $\mathrm{co}\{z_1, \ldots, z_K\}$ spanned by a small number of "corner" patterns (the archetypes) [8]. The coefficients α_{ik} show how strongly each data point x_i is related to each archetype.

Cutler and Breiman studied where the optimal archetypes are located. Let $C = \mathrm{co}\{x_1, \ldots, x_n\}$ be the convex hull of the data, and let S be the set of points lying on the boundary of C. In [8] they prove that:

- For $K = 1$, the unique archetype is the sample mean $\bar{x}$, and RSS_1 is equal to the total sum of squares of the data.

- For $1 < K < |S|$, there exists an optimal solution with all archetypes $z_1, \ldots, z_K$ lying on the boundary of C.

- For $K = |S|$, the boundary points S can be chosen as archetypes and then the residual sum of squares can be made equal to zero.

Mørup and Hansen [21] describe AA as estimating the *principal convex hull* of the data: a low-dimensional convex polytope that best fits the data in least-squares sense. They also show that the AA solution is in general unique up to a permutation of the components.

Algorithms for archetypal analysis

The optimisation problem (12) can be solved by a nonlinear least squares algorithm, but this may be impractical except for very small porblems. Instead, Cutler and Breiman [8] suggest an alternating constrained least squares (CLS) algorithm. The idea is to alternate between updating the coefficients α_{ik} for a fixed set of archetypes and updating the archetypes (equivalently, the coefficients β_{kj}) for a fixed set of α_{ik}.

The algorithm contains of the following steps:

1. Initialising B.

2. For fixed B, updating A by solving n independent convex least squares problems, one for each observation x_i:

$$\min_{\alpha_i} \left\| x_i - \sum_{k=1}^{K} \alpha_{ik} z_k \right\|^2 \quad \text{subject to} \quad \alpha_{ik} \geq 0, \ \sum_{k=1}^{K} \alpha_{ik} = 1.$$

3. For fixed A, updating B by solving K convex least squares problems, one for each archetype, under the convexity constraints on β_{kj}.

4. Iterating steps 2–3 until the reduction in RSS_K is a sufficiently small value.

Each step reduces the residual sum of squares, and each subproblem is a convex quadratic program with linear constraints.

Later work has developed faster algoriths for AA, for example projected–gradient algorithms that update the matrices A and B more efficiently, as well as kernel version of AA has also been proposed in [21].

The number of archetypes K is not fixed by the model and has to be optimally chosen. A common practical rule is the elbow criterion: we compute the residual sum of squares RSS for several values of K, plot RSS against K and then choose the smallest K where the curve clearly starts to decrease much more slowly [48]. In R the `archetypes` package is used for archetypal analysis [11].

Archetypoid analysis

Archetypoid analysis (ADA) is a variant of AA in which the extremal profiles are the actual data points or vectors. The reconstructed observations are still convex combinations of the archetypoid, but each archetypoid is the initial data point(vector) itself.

Formally, the representation of x_i as a mixture of the K archetypoids is identical to (11). The difference lies in z_k. For ADA the formulation is:

$$z_k = \sum_{j=1}^{n} \beta_{kj} x_j, \qquad \beta_{kj} \in \{0, 1\}, \quad \sum_{j=1}^{n} \beta_{kj} = 1, \quad k = 1, \ldots, K, \tag{13}$$

so that for each k there is a unique index j_k such that $\beta_{kj_k} = 1$ and $z_k = x_{j_k}$, so each archetypoid is a conctete data point. The optimisation problem becomes

$$\min_{\alpha,\beta} \ \mathrm{RSS}_K(\alpha, \beta) = \sum_{i=1}^{n} \left\| x_i - \sum_{k=1}^{K} \alpha_{ik} z_k \right\|^2 \quad \text{subject to (11) and (13).} \tag{14}$$

This is a mixed-integer optimisation problem: α is continuous, while β is discrete with values 0 or 1. As in AA, the coefficients α_{ik} can be interpreted as weights of archetypoids in the representation of x_i [48].

Vinué, Epifanio and Alemany [48] proposed a two-stage algorithm (BUILD and SWAP) for ADA based on ideas from Paartitioning Around Medoids (PAM). In the *BUILD* phase an initial set of K archetypoids is constructed; in the *SWAP* phase the initial set is improved by trying swaps between selected and non-selected observations and keeping only those swaps that decrease the residual sum of squares. This approach has been implemented in R package `Anthropometry`.

In contrast to AA, archetypoids do not need to lie exactly on the boundary of the convex hull of the data. However, they still play the role of extreme profiles: they are selected observations that, when combined convexly, approximate all data points well [48].

Implementation on the Shakespeare's Plays

In our analysis we applied both AA and ADA to the $n = 37$ Shakespeare's plays represented by rescaled beat charts, each with length $m = 100$.

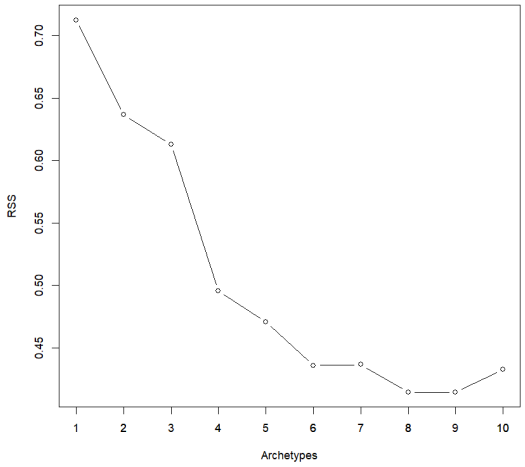
Table 7: Plays grouped into $k = 4$ clusters obtained from the archetype-based representation of rescaled beat charts.

Cluster	Plays
1	Hamlet, Othello, Troilus and Cressida.
2	A Midsummer Night's Dream, All's Well That Ends Well, Coriolanus, Henry VI Part 1, Macbeth, Romeo and Juliet, The Comedy of Errors, The Taming of the Shrew, The Merry Wives of Windsor, Titus Andronicus.
3	Antony and Cleopatra, As You Like It, Cymbeline, Henry IV Part 1, Henry IV Part 2, Henry V, Henry VI Part 2, Henry VI Part 3, Henry VIII, Julius Caesar, King Lear, Love's Labour's Lost, Measure for Measure, Much Ado About Nothing, Richard II, Richard III, The Merchant of Venice, The Tempest, The Two Gentlemen of Verona, The Winter's Tale, Timon of Athens, Twelfth Night.
4	King John, Pericles.

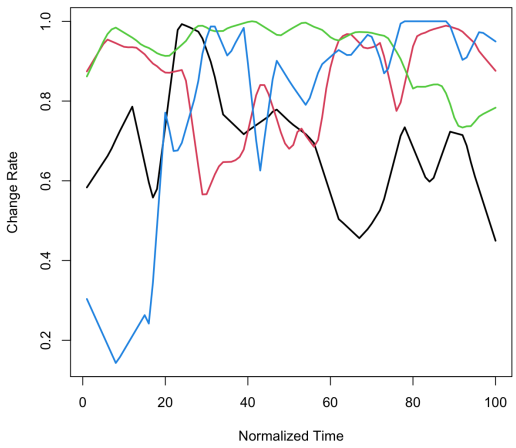
For archetypal analysis we used the R package `archetypes` [11]. We applied the function `stepArchetypes()` to the matrix of rescaled beat charts with $K = 1, \ldots, 10$ and 5 repetitions for each K. We used the function `screeplot()` to plot the residual sum of squares RSS for different values of K and choose $K = 4$ (see Figure 15a). The corresponding model was extracted with `bestModel()`, the 4×100 matrix of archetype curves was obtained with `parameters()`, and the 37×4 matrix of coefficients α_{ik} describing how each play is expressed as a convex combination of the four archetypes was obtained with `coef()`. The shape of the four archetypes obtained from the rescaled beat charts is shown in Figure **??**. The archetypes are dominated by plays with the highest corresponding coefficients, namely Othello, The Merry Wives of Windsor, Henry V and Pericles. The assignment of plays to the four archetype-based clusters is showm in Table 7.

Cluster 1 contains three plays: *Hamlet*, *Othello* and *Troilus and Cressida*. All three are tragedies. *Hamlet* and *Othello* again appear together in a small cluster, similarly to EMD - based method.

Cluster 2 contains ten plays and represents a mixture of all genres, yet is dominated with five comedies. Examples include comedies such as *A Midsummer Night's Dream*, *The Comedy of Errors* and *The Merry Wives of Windsor*, as well as tragedies such as *Macbeth*, *Romeo and Juliet* and *Coriolanus*. The cluster also includes one history play



(a) Scree plot for the AA method



(b) Four archetypes obtained by AA model with $K = 4$

Figure 15: Results of archetypal analysis

Henry VI Part 1.

Cluster 3 is the largest group and contains the majority of the plays. It includes works from all three genres, such as the comedies *Much Ado About Nothing*, *Twelfth Night* and *As You Like It*, the tragedies *King Lear*, *Julius Caesar* and *Antony and Cleopatra*, as well as several history plays including *Henry IV Parts 1-2*, *Henry V* and *Richard II–III*.

Finally, Cluster 4 consists of two plays: *King John* and *Pericles*, with *Pericles* being distinct across many methods.

For archetypoid analysis we used the R package `Anthropometry` [47]. First, we computed archetypes on the same matrix of rescaled beat charts X using `stepArchetypesRawData()` with $K = 1, \ldots, 4$ and 5 repetitions. Then we ran `stepArchetypoids()` with four archetypoids and the nearest–neighbour option using these archetypes as input (`ArchObj`). This produced four archetypoids, that is, four plays selected as representative extreme profiles; their indices were taken from the `cases` component. The corresponding matrix of coefficients α_{ik} relating each play to the four archetypoids was taken from the `alphas` component. As in the AA case, we assigned each play to the dominant archetypoid by the maximal coefficient of α and used these assignments to define play groups.

The beat charts of four archetypoid plays which are *Titus Andronicus*, *Timon of Athens*, *The Comedy of Errors* and *Pericles*, are displayed in Figure 16. The clustering

Table 8: Plays grouped into $k = 4$ clusters obtained from the archetypoid method.

Cluster	Plays
1	Titus Andronicus.
2	All's Well That Ends Well, Antony and Cleopatra, As You Like It, Coriolanus, Cymbeline, Henry IV Part 1, Henry IV Part 2, Henry V, Henry VIII, King Lear, Love's Labour's Lost, Macbeth, Measure for Measure, Much Ado About Nothing, Richard III, Romeo and Juliet, The Merchant of Venice, The Merry Wives of Windsor, The Tempest, The Two Gentlemen of Verona, The Winter's Tale, Timon of Athens, Troilus and Cressida, Twelfth Night.
3	A Midsummer Night's Dream, Hamlet, Henry VI Part 1, Henry VI Part 2, Henry VI Part 3, Julius Caesar, King John, Othello, Richard II, The Comedy of Errors, The Taming of the Shrew.
4	Pericles.

results obtained from ADA analysis can be found in Table 8.

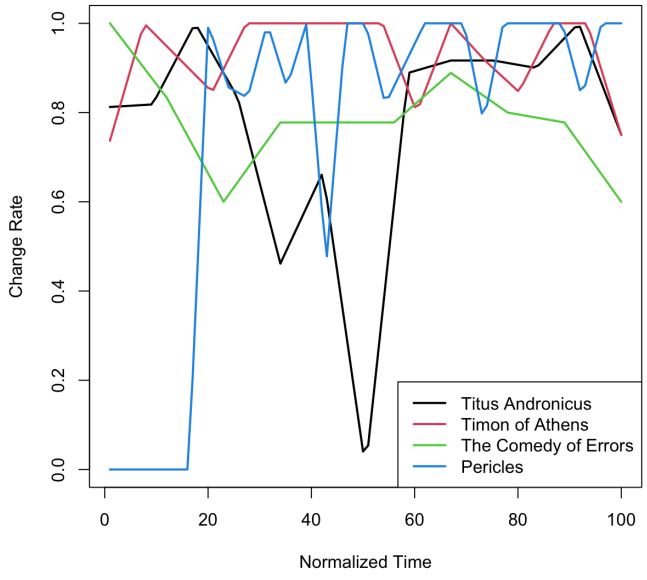


Figure 16: Beat charts of the four archetypoid plays selected by the ADA model with $K = 4$.

Cluster 1 contains only *Titus Andronicus*, indicating that this play forms a distinct outlier in the archetypoid-based representation. Titus Andronicus has sometimes been considered a collaborative play of Shakespeare's early years, as well as previosly mentioned

Pericles, possibly involving George Peele [18].

Cluster 2 is the largest group and contains twenty four plays from all three genres, including comedies such as *As You Like It*, *Much Ado About Nothing* and *Twelfth Night*, tragedies such as *Macbeth*, *King Lear* and *Antony and Cleopatra*, and history plays such as *Henry IV Parts 1-2*, *Henry V* and *Richard III*. This cluster represents the main mixed group.

Cluster 3 also contains a mixture of genres but includes a stronger history component, with five plays such as *Henry VI Parts 1-3*, *King John* and *Richard II*. It also contains the tragedies *Hamlet*, *Julius Caesar* and *Othello*, together with the comedies *A Midsummer Night's Dream*, *The Comedy of Errors* and *The Taming of the Shrew*.

Finally, Cluster 4 again consists only of *Pericles*.

4 Clustering and Comparison: Investigation of Groups of Similar Plays

4.1 Adjusted Rand Index (ARI)

In order to evaluate the consistency between the clustering results obtained from different methods, we computed pairwise *Adjusted Rand Index (ARI)* values. This index measures the agreement between two partitions, corrected for chance, and is widely used to compare clustering solutions that may have a different number of clusters.

The Adjusted Rand Index (ARI) is a corrected-for-chance version of the Rand Index, named after William M. Rand [29]. It measures the similarity between two partitions $\mathcal{U}$ and $\mathcal{V}$ of a set of n elements. Let:

$$a = \text{number of pairs assigned to the same cluster in both } \mathcal{U} \text{ and } \mathcal{V},$$

$$b = \text{number of pairs assigned to different clusters in both } \mathcal{U} \text{ and } \mathcal{V}.$$

Then the classical Rand Index (RI) is defined as:

$$RI = \frac{a+b}{\binom{n}{2}}.$$

The ARI adjusts this measure by accounting for random chance:

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

where $E[RI]$ is the expected value of RI for random labelings. The index ranges from -1 to 1, with 1 indicating perfect agreement, 0 corresponding to random similarity and negative values representing systematic disagreement.

While in this work we focus primarily on ARI as the main similarity measure, a similar metric *Normalized Mutual Information (NMI)* was also computed for comparison. The two measures showed consistent results across all methods, differing only slightly in absolute values.

4.2 Comparison Across Methods

We compared nine clustering approaches derived from different features:

- NetLSD with Ward's linkage (denoted as netlsd_wardD2 in subsequent figure)
- Statistics-based clustering (statistics): k-means clustering based on the mean and standard deviation of the beat charts
- Dynamic Time Warping (dtw)
- Archetypal analysis (archetype)
- Roughness combined with entropy (roughness_entropy4)
- Spectral entropy (spectral_entropy4)
- Earth Mover's Distance (emd)
- Roughness-based similarity (roughness4)
- Archetypoid method (archetypoid)

The resulting pairwise Adjusted Rand Index matrix (Figure 17) reveals the degree of agreement between the clustering results obtained by these methods. The highest similarity was observed among archetypoid, statistics-based and EMD-based clusterings. In contrast, methods based on NetLSD, spectral entropy and Dynamic Time Warping yielded noticeably lower agreement with the rest.

Co-occurrence Analysis of Plays

To gain a deeper understanding of the relationships between Shakespeare's plays across different clustering approaches, we constructed a so called *co-occurrence matrix* with diagonal entries equal to 0. Each off-diagonal cell in this matrix shows how many times a pair of plays appeared in the same cluster across all nine methods mentioned previously. In other words, a high co-occurrence value indicates that two plays were repeatedly grouped together regardless of the specific method.

Formally, for each clustering method m, pairs of plays (i, j) were examined, and the matrix entry C_{ij} was incremented by one whenever both plays belonged to the same cluster under method m. The resulting symmetric $n \times n$ matrix thus encodes the frequency of pairwise assignments across methods.

The co-occurrence matrix was then represented as a weighted undirected graph, where vertices correspond to plays and edge weights reflect the number of co-occurrences. This

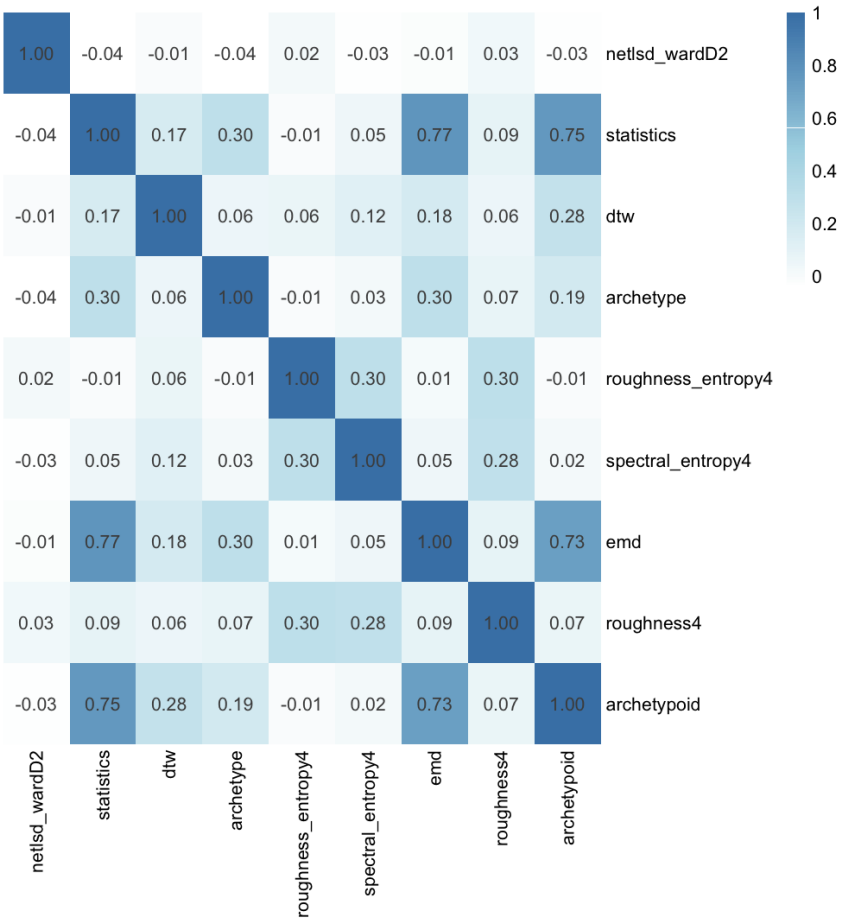
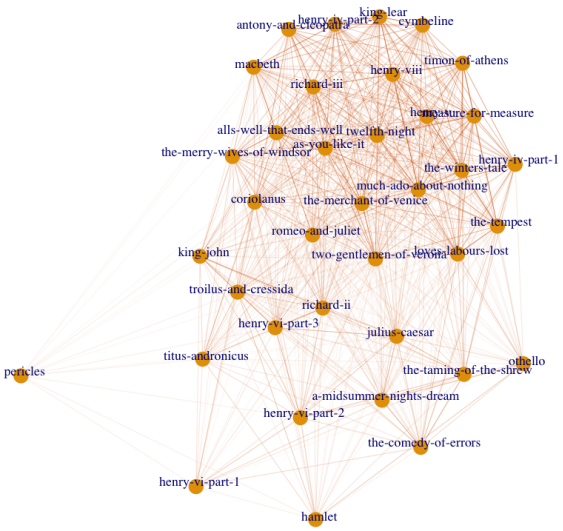


Figure 17: Pairwise Adjusted Rand Index (ARI) between clustering methods.

graph representation made it possible to examine how stable the relationships between plays were across different clustering approaches. To do this, we gradually increased the edge-weight threshold and removed all edges with weights below the chosen value. In this way, only the connections that appeared frequently across methods were retained, highlighting the most consistent groups (see Figure 18).

We focused on the threshold values of 6, 7 and 8, which represent strict criteria for pairwise consistency. We also looked at the threshold value of 9, aiming to find plays which were always assigned together in the same cluster within all the methods.

At a threshold of 6, the resulting graph contained a single large connected component of 32 plays and 124 edges, with an overall density of approximately 0.25. When the threshold was increased to 7, the network began to fragment into several components. The largest of these contained 21 plays and 54 edges (with density 0.26) and two smaller groups of three plays appeared, each with density 0.67. This level marked a transition from general



(a) Full weighted graph



(b) Graph after thresholding ($t = 8$)

Figure 18: Thresholding of the co-occurrence graph: edges with weights below the chosen threshold are removed.

to more specific relationships among plays, capturing subsets that consistently clustered together across a majority of methods.

At the threshold of 8, only the most stable and recurrent groups remained. The network reduced to a few small components, with two main components consisting of 7 and 6 plays, respectively, together with three pairs (two-play components with density equal to 1.0).

For each threshold, we analyzed the resulting graphs using the *igraph* package in R [**?**]. In every case, we identified the connected components — groups of plays that remained linked after weaker connections were removed — and the maximal cliques, which are subsets of vertices of a graph such that every two distinct vertices in these graphs are adjacent. To describe these structures more clearly, we recorded for each component its size, number of edges and network density, as well as the plays that remained isolated (see Table 9).

Before examining the specific plays forming the detected components and cliques, we refer to the metadata described in Section 1.1. This metadata included three attributes: the *genre* (comedy, tragedy or histories), the *period* of composition (the beginning of the decade in which the play was written) and the *geographic setting* of the play.

Since our analysis mainly focuses on the dynamic characteristics of plays and their temporal evolution, the genre and period are expected to be particularly relevant. Genre captures differences in narrative rhythm, while the period reflects the stylistic development of Shakespeare's writing over time.

In the detailed analysis of individual plays, we focus primarily on the higher thresholds ($t = 8$ and $t = 9$), which reveal the most stable and closely connected groups of plays.

At the strongest threshold ($t = 9$), the network contained only two persistent pairs, both linking plays that are close in genre and setting. The first pair *Cymbeline* and *King Lear* includes two tragedies set in England. The second dyad, *The Tempest* and *Love's Labour's Lost*, is two comedies set in Italy.

At threshold $t = 8$, the co-occurrence graph splits into five connected components, while a large number of plays become isolated.

Three components consist of two plays each: *Cymbeline* and *King Lear* (both tragedies set in England); *Henry IV Part 2* and *Macbeth* (history and tragedy in English settings);

Table 9: Summary of connected components detected at different co-occurrence thresholds.

Threshold	Size	Edges	Density	Isolated plays
6	32	124	0.25	*Hamlet*, *Henry VI Part 1*, *Pericles*, *Titus Andronicus*, *Troilus and Cressida*
7	21	54	0.26	
	3	2	0.67	*Coriolanus*, *Hamlet*, *Henry VI Part 1*, *Henry VI Part 2*, *Julius Caesar*, *Othello*, *Pericles*, *Romeo and Juliet*, *Titus Andronicus*, *Troilus and Cressida*
	3	2	0.67	
8	7	7	0.33	
	6	5	0.33	*A Midsummer Night's Dream*, *Antony and Cleopatra*, *Coriolanus*, *Hamlet*, *Henry V*, *Henry VI Part 1*, *Henry VI Part 2*, *Henry VI Part 3*, *Julius Caesar*, *King John*, *Othello*, *Pericles*, *Richard II*, *Romeo and Juliet*, *The Comedy of Errors*, *The Taming of the Shrew*, *Titus Andronicus*, *Troilus and Cressida*
	2	1	1.00	
	2	1	1.00	
	2	1	1.00	
9	2	1	1.00	All except *Cymbeline*, *King Lear*, *Love's Labour's Lost*, *The Tempest*
	2	1	1.00	

and *Love's Labour's Lost* and *The Tempest* (both comedies with Italian settings). Notably, two of these pairs, *Cymbeline* with *King Lear* and *Love's Labour's Lost* with *The Tempest* remain even at the strongest threshold ($t = 9$).

The remaining components consist of six and seven plays respectively. The first group (6 plays) includes *All's Well That Ends Well*, *As You Like It*, *Richard III*, *The Merchant of Venice*, *The Merry Wives of Windsor* and *The Two Gentlemen of Verona*. This is a comedy dominated group with one history.

The second group (7 plays) consists of *Cymbeline*, *Henry IV Part 1*, *Henry VIII*, *King Lear*, *Measure for Measure*, *Much Ado About Nothing*, *The Winter's Tale*, *Timon of Athens* and *Twelfth Night*. This component includes a mix of comedies, histories and tragedies. Within this component, a maximal clique of size three can be identified, containing plays *The Winter's Tale*, *Measure for Measure* and *Timon of Athens*.

Table 10: Groups of plays detected at co-occurrence thresholds $t = 9$ and $t = 8$.

Threshold t	**Groups / Connected Plays**
9	1. *Cymbeline – King Lear*
	2. *The Tempest – Love's Labour's Lost*
8	1. Connected component (7 plays): *Henry IV Part 1*, *Henry VIII*, *Measure for Measure*, *Much Ado About Nothing*, *The Winter's Tale*, *Timon of Athens*, *Twelfth Night*.
	2. Connected component (6 plays): *All's Well That Ends Well*, *As You Like It*, *Richard III*, *The Merchant of Venice*, *The Merry Wives of Windsor*, *The Two Gentlemen of Verona*.
	3. Connected component (2 plays): *Henry IV Part 2*, *Macbeth*.
	4. Connected component (2 plays): *Love's Labour's Lost*, *The Tempest*.
	5. Connected component (2 plays): *Cymbeline*, *King Lear*.
	6. Maximal clique: *The Winter's Tale*, *Measure for Measure*, *Timon of Athens*.

Outlier Plays in the Co-occurrence Network

To identify potential outliers, we examined the total co-occurrence counts for each play by summing the corresponding rows of the co-occurrence matrix. Three plays—*Pericles*, *Henry VI Part 1* and *Hamlet*—showed significantly lower co-occurrence values compared to the rest of the corpus. None of these plays appeared in any connected component or clique even at lower thresholds, indicating that they consistently remained isolated across clustering methods.

The case of *Pericles* aligns with previous observations about its irregular dramatic structure. According to Fischer *et al.* (2017), *Pericles* displays one of the highest values of drama change rate across the Shakespeare corpus, reflecting its highly episodic composition [13]. The story takes place in many different locations, as the main character travels across several regions, which leads to frequent changes of characters and scenes throughout the play. These features help explain why it might show low co-occurrence

counts and is excluded from the stable clusters. The play is also considered collaborative with George Wilkins being co-author [18].

Similarly, *Henry VI, Part 1* has often been considered a collaborative work rather than a solo composition. According to the Folger Shakespeare Library's authorship appendix, since the late 18th century editors have debated Shakespeare's single authorship and proposed possible contributions from contemporaries such as Nashe or Greene [14].

Finally, *Hamlet* is structurally distinct within the plays. It is the longest of all Shakespeare's plays, containing over 30000 words [25]. The play also contains a famous "play within a play", called *The Mousetrap*, where the characters stage a performance to expose the murder of Hamlet's father. Such metatheatrical structure was uncommon in the drama of that period. In addition, *Hamlet* contains one of the largest numbers of monologues and soliloquies among Shakespeare's plays [9]. Still these general facts cannot be the reason of *Hamlet's* being an outlier. The Mousetrap is a part of a single scene and cannot have a big influence on the whole beat chart's dynamicity. We also highlight that the play was not an outlier in individual methods, but in context of the Co-occurrence analysis.

4.3 Community Detection On The Co-occurrence Graph

Optimal Modularity Community Detection

Now, we apply the optimal modularity community detection method to identify communities in the co-occurrence graph constructed from the co-occurrence matrix.

The goal of this approach is to find a partition of the graph into groups of nodes (communities) such that connections within groups are stronger than between different groups. To evaluate the quality of a partition, the modularity is used.

Modularity is a quality measure for graph clusterings that compares the actual number of edges within communities to the expected number of such edges in a random graph with the same degree distribution.

Formally, given a graph $G = (V, E)$ with m edges and a clustering $\mathcal{C}$, the modularity $Q(\mathcal{C})$ is defined as

$$Q(\mathcal{C}) = \sum_{C \in \mathcal{C}} \left[\frac{|E(C)|}{m} - \left(\frac{\sum_{v \in C} \deg(v)}{2m} \right)^2 \right],$$

where $|E(C)|$ denotes the number of edges inside cluster C and $\deg(v)$ is the degree of vertex v [1].

The optimal modularity algorithm searches for a partition of the graph that maximizes the modularity value. This problem can be formulated as an integer linear problem and solved exactly for small graphs with up to two hundred nodes. More precisely, for a graph $G = (V, E)$ with $|V| = n$, let $X_{uv} \in \{0, 1\}$ be binary variables for each pair of vertices $u, v \in V$. The variable X_{uv} shows whether vertices u and v are in the same community ($X_{uv} = 1$) or not ($X_{uv} = 0$).

Variables X_{uv} must satisfy the following constarints:

- **Reflexivity:** $\forall u \in V : \ X_{uu} = 1$,
- **Symmetry:** $\forall u, v \in V : \ X_{uv} = X_{vu}$,
- **Transitivity:** $\forall u, v, w \in V :$

$$\begin{cases} X_{uv} + X_{vw} - 2X_{uw} \leq 1, \\ X_{uw} + X_{wv} - 2X_{uv} \leq 1, \\ X_{vw} + X_{uw} - 2X_{uv} \leq 1. \end{cases}$$

Under this formulation, the modularity maximization problem can be written as

$$\max \frac{1}{2m} \sum_{u,v \in V} \left(E_{uv} - \frac{\deg(u) \deg(v)}{2m} \right) X_{uv},$$

where $E_{uv} = 1$ if $(u, v) \in E$ and $E_{uv} = 0$ otherwise [1].

In our implementation, we applied this method in R using the `cluster_optimal()` function from the `igraph` package. The summary of the two detected communities can be found in Table 11 and is visualized on Figure 19.

Walktrap Community Detection

In addition to the optimal modularity approach, we also apply another community detection algorithm based on random walks.

Mathematically, this method is based on a stochastic process. Let G be a graph with adjacency matrix A, where $A_{ij} = 1$ if vertices i and j are connected and $A_{ij} = 0$ otherwise.

Table 11: Plays grouped into 2 communities obtained using the optimal modularity algorithm on the co-occurrence graph.

Community	**Plays**
1	A Midsummer Night's Dream, Hamlet, Henry VI, Part 1, Henry VI, Part 2, Henry VI, Part 3, Julius Caesar, King John, Othello, Pericles, Richard II, Romeo and Juliet, The Comedy of Errors, The Taming of the Shrew, Titus Andronicus, Troilus and Cressida, The Merry Wives of Windsor.
2	All's Well That Ends Well, Antony and Cleopatra, As You Like It, Coriolanus, Cymbeline, Henry IV, Part 1, Henry IV, Part 2, Henry V, Henry VIII, King Lear, Love's Labour's Lost, Macbeth, Measure for Measure, Much Ado About Nothing, Richard III, The Merchant of Venice, The Tempest, The Winter's Tale, Timon of Athens, Twelfth Night, Two Gentlemen of Verona.

The degree of a vertex i is defined as

$$d(i) = \sum_j A_{ij}.$$

A random walk on the graph is a stochastic process in which, at each step a walker located at vertex i moves to one of its neighboring vertices. The transition probability from vertex i to vertex j is given by

$$P_{ij} = \frac{A_{ij}}{d(i)}.$$

Therefore, the movement of the walker is described by the transition matrix P. The probability of reaching vertex j from vertex i after a walk of length t is given by the (i, j)-th element of the matrix power P^t, denoted by

$$P^t_{ij}.$$

The main intuition behind the Walktrap algorithm is that short random walks tend to stay inside the same community, because vertices within a community are more densely connected. Therefore, vertices are considered similar if the probability distributions of short random walks starting from these vertices are similar. At each step, the closest communities are merged using Ward's method, resulting in a hierarchical structure (dendrogram) [28].

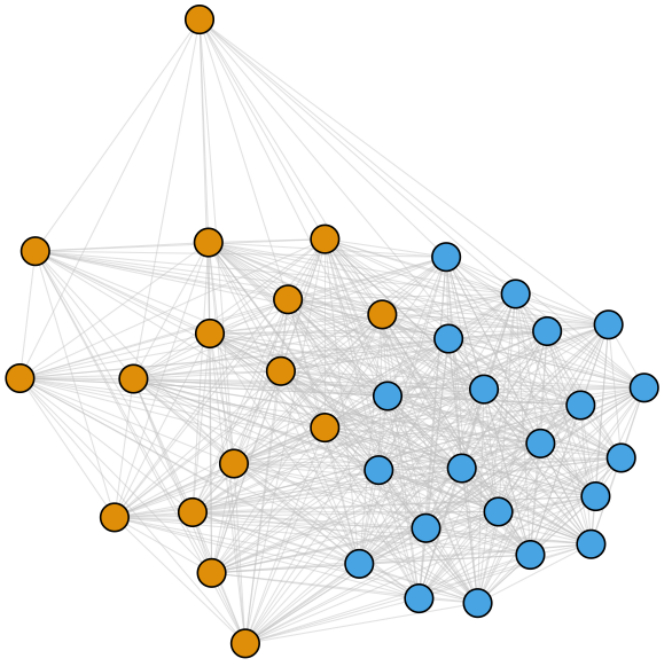


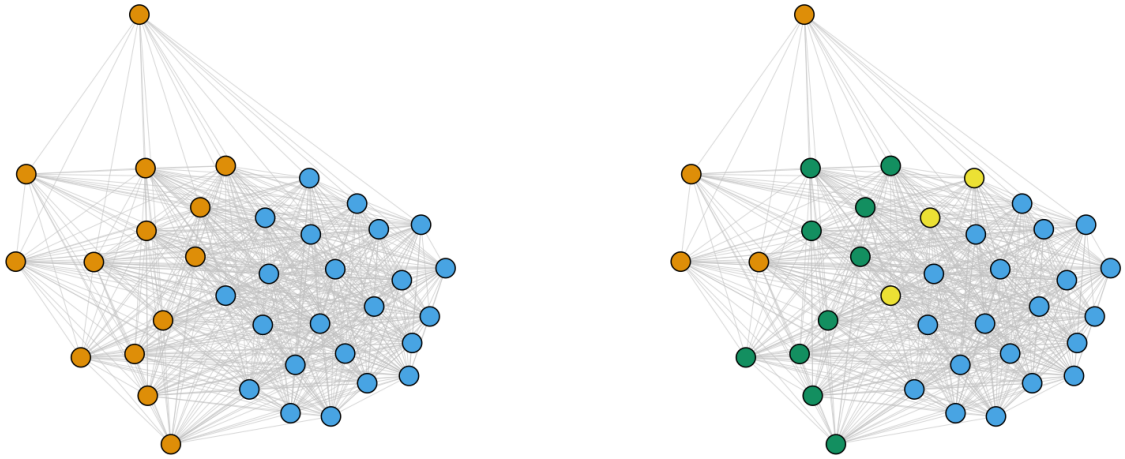
Figure 19: Vizualization of detected communities by the optimal modularity algorithm on the co-occurrence graph.

In this work, the method was implemented in R using the `cluster_walktrap()` function from the `igraph` package.

In contrast to the optimal modularity approach, where the number of communities is determined automatically, an advantage of the Walktrap algorithm is that it produces a dendrogram, so we can choose the desired number of clusters, using the `cut_at()` function.

In our analysis, the Walktrap algorithm identified a partition into two communities as optimal, similarly to community detection based on optimal modularity. However, to compare the results with other methods, we also considered a cut at the level of $k = 4$ communities. The visualization of these communities is shown on Figure 20.

First, we computed the Adjusted Rand Index (ARI) between the Walktrap community



(a) $k = 2$ **(b)** $k = 4$

Figure 20: Communities detected by the Walktrap algorithm on the co-occurrence graph.

partitions and the optimal modularity partition (see Table 12).

The results show that high similarity was achieved for the optimal Walktrap partition with $k = 2$, indicating a strong agreement with the optimal modularity clustering. In contrast, a very low similarity is observed for the Walktrap with $k = 4$.

Walktrap partition	**ARI with optimal modularity**
optimal ($k = 2$)	0.790
$k = 4$	0.163

Table 12: Adjusted Rand Index between Walktrap partitions and optimal modularity clustering.

We further examined the similarity between the Walktrap partition with $k = 4$ and the individual clustering methods used in our analysis. For this purpose we again computed the pairwise Adjusted Rand Index. The values can be found in Table 13.

The results show that there is moderate agreement between the Walktrap communities and several methods, namely the EMD, the statistics-based method and the archetypoid

Method	**ARI (with Walktrap $k = 4$)**
EMD	0.734
Statistics	0.716
Archetypoid	0.618
Archetype	0.370
DTW	0.134
Roughness	0.146
Spectral Entropy	0.077
Roughness + Entropy	0.012
NetLSD	0.008

Table 13: Adjusted Rand Index between Walktrap ($k = 4$) and clustering methods.

and archetypal approaches.

On the other hand, lower agreement was observed for DTW and roughness clusterings. The NetLSD clustering and the combined roughness and entropy method, as well as spectral entropy clustering show almost no agreement with the Walktrap partition.

It is also worth mentioning that Walktrap partition with $k = 4$ identified a community containing *Hamlet*, *Pericles*, *Henry VI Part 1* and *Henry VI Part 2*, where the first three plays correspond to outliers in the context of the co-occurrence analysis.

Table 14: Plays grouped into $k = 4$ communities obtained using the Walktrap algorithm on the co-occurrence graph.

Community	**Plays**
1	Hamlet, Henry VI Part 1, Henry VI Part 2, Pericles.
2	All's Well That Ends Well, Antony and Cleopatra, As You Like It, Cymbeline, Henry IV Part 1, Henry IV Part 2, Henry V, Henry VIII, King Lear, Love's Labour's Lost, Macbeth, Measure for Measure, Much Ado About Nothing, Richard III, The Merchant of Venice, The Tempest, The Winter's Tale, Timon of Athens, Twelfth Night, Two Gentlemen of Verona.
3	A Midsummer Night's Dream, Henry VI Part 3, Julius Caesar, King John, Othello, Richard II, The Comedy of Errors, The Taming of the Shrew, Titus Andronicus, Troilus and Cressida.
4	Coriolanus, Romeo and Juliet, The Merry Wives of Windsor.

Conclusions

In this thesis, we investigated William Shakespeare's plays using methods from quantitative data analysis, includiong social network and temporal data analysis. The main objective was to explore similarities between plays by representing them as character interaction networks and time series derived from beat charts. By applying different analytical approaches, we aimed to reveal patterns in the structure and dynamics of Shakespeare's plays.

First, we represented each play as a weighted undirected graph where nodes correspond to charactes and edges represent their co-occurrence within scenes. Using this representation, we applied the Network Laplacian Spectral Descriptor, which captures structural properties of graphs through the spectral characteristics of the Laplacian matrix. The resulting heat trace signatures provided a compact representation of each play's network structure. Based on the Euclidean distances between these signatures, hierarchical clustering revealed several groups of plays that partially correspond to traditional genre classifications, with comedies forming a distinct cluster.

In the second part of the thesis, we focused on the temporal dynamics of character interactions. Each play was represented as a time series called a beat chart, which describes how interactions between characters change across the scenes. Several methods were applied to analyze these time series. The Earth Mover's Distance allowed us to compare distributions derived from beat charts, as well as Dynamic Time Warping was used to align time series of different lengths and capture their similarities. In addition, we computed two descriptive indicators of the beat charts: roughness, which measures variability and total curvature of the series, and spectral entropy, which quantifies the distribution of energy across frequency components. Finally, archetypal and archetypoid analyses were used to identify representative profiles among the plays.

The clustering results obtained from these different approaches revealed several interesting observations. While the specific composition of clusters differed across methods, some consistent patterns can be noticed accros multiple methods. In particular, several plays repeatedly appeared as outliers or formed small separate clusters. For example, *Pericles* was frequently isolated from the rest of the corpus, suggesting that its structural or temporal characteristics differ substantially from other Shakespeare plays. This observa-

tion is is consistent with previous observations about its irregular dramatic structure, as in the story Pericles travels to different places, which causes frequent changes in charachters, and unusually high drama change rate with value 0.972 within the Shakespeare corpus [13]. *Pericles* is widely cosidered to be a collaborative play, with scholars suggesting that the first acts were written by George Wilkins and the later ones by Shakespeare [18].

Similarly, some methods highlighted *Henry VI Part I* as structurally distinct play. *Henry VI Part 1* is often considered a collaborative play, possibly written with other playwrights such as Nashe or Greene, which may explain some structural differences from the rest of the plays [14].

Hamlet also appears as an outlier in context of co-occuremce analysis. It is the longest play in the corpus, with more than 30000 words [25]. The play also contains the well-known "play within a play" (*The Mousetrap*) and many monologues and soliloquies [9]. However, while these are well-known facts, they cannot provide explanation to *Hamlet's* being an outlier, because The Mousetrap is a part of one scene in the play and therefore it does not have a big impact on the drama change rate of the beat chart.

At the same time, the clustering results also demonstrated that different analytical methods emphasize different aspects of the data. Network-based approaches focus on the structure of character relationships, whereas time-series-based approaches capture the dynamics of interactions across the narrative. Consequently, the clusters produced by individual methods do not fully coincide. Especially clustering from NetLSD method highly differs from methods applied to the beat charts, which could be exected, as these methods analyze the plays from different perspectives. This result should not be seen as a problem or a weakness of the methods. It simply shows that literary works are complex and can be studied from different sides.

Overall, the results show that mathematical and computational methods can help us see new patterns in literary texts. These methods cannnot replace literary studies, but they offer another way to look at the relationships between plays. By combining social network analysis, time series analysis and clustering methods, this thesis shows how mathematical tools can be used to study literary works.

Future research could extend this analysis in several directions. One possible extension is to focus on collaboration. In particular, one could take plays written by authors who

are suspected to have collaborated with Shakespeare, construct their character networks and beat charts in the same way as in this thesis, and compare them directly with Shakespeare's plays. This would make it possible to see whether collaborative plays are more similar to the works of these authors than to Shakespeare's plays written on his own.

Another idea is to study individual plays, especially those that are collaborative or behave as outliers in more detailed way. For example, it would be interesting to analyze different parts of a play separately, with applying similar analysis not on the whole beat charts but on their parts. This could help reveal structural differences within the individual plays.

Bibliography

[1] Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., Wagner, D. On Modularity Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2007. DOI: https://doi.org/10.1109/TKDE.2007.190689.

[2] Bringmann, K., Fischer, N., van der Hoog, I., Kipouridis, E., Kociumaka, T. and Rotenberg, E. Dynamic Dynamic Time Warping. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 208–242, 2024. DOI: https://doi.org/10.1137/1.9781611977912.10.

[3] Cabero, I. and Epifanio, I. Archetypal analysis: an alternative to clustering for unsupervised texture segmentation. *Image Analysis & Stereology*, Vol. 38: pp. 151–160, 2019. DOI: https://doi.org/10.5566/ias.2052.

[4] Chambers, P. Employing Mathematics to Identify the Real Shakespeare. *The Oxfordian*, Vol. 24, pp. 111–130, 2022.

[5] Chung, F. R. K. *Spectral Graph Theory*. American Mathematical Society, 1997.

[6] Cohen, S. The Earth Mover's Distance. Technical Report STAN-CS-TN-99-1620, Stanford University, 1999.

[7] Csardi, G. and Nepusz, T. The igraph software package for complex network research. *InterJournal, Complex Systems*, pp. 1695, 2005.

[8] Cutler, A. and Breiman, L. Archetypal analysis. *Technometrics*, Vol. 36(4): pp. 338–347, 1994. DOI: https://doi.org/10.1080/00401706.1994.10485840.

[9] De Grazia, M. and Wells, S. The New Cambridge Companion to Shakespeare. Cambridge University Press, Cambridge, 2nd edition, 2010.

[10] Epifanio, I. Functional archetype and archetypoid analysis. *Computational Statistics & Data Analysis*, Vol. 104: pp. 24–34, 2016. DOI: https://doi.org/10.1016/j.csda.2016.06.007

[11] Eugster, M. J. A. and Leisch, F. From Spider-Man to Hero: Archetypal analysis in R. *Journal of Statistical Software*, Vol. 30(8): pp. 1–23, 2009.

[12] Fernández, D., Epifanio, I. and McMillan, L. F. Archetypal analysis for ordinal data. *Information Sciences*, Vol. 579: pp. 281–292, 2021. DOI: https://doi.org/10.1016/j.ins.2021.07.095.

[13] Fischer, F., Göbel, M., Kampkaspar, D., Kittel, C. and Trilcke, P. *Network Dynamics, Plot Analysis: Approaching the Progressive Structuration of Literary Texts.* In DH, 2017.

[14] Folger Shakespeare Library. Appendix: Authorship of Henry VI, Part 1. Available at https://www.folger.edu/explore/shakespeares-works/henry-vi-part-1/appendix-authorship-of-henry-vi-part-1.

[15] Glickenstein, D. Math 443/543 Graph Theory Notes 5: Graphs as matrices, spectral graph theory, and PageRank. Lecture notes, University of Arizona, November 3, 2014.

[16] Hiscock, A. Pericles, Prince of Tyre and the appetite for narrative. In Power, A. J. and Loughnane, R. (eds.), *Late Shakespeare, 1608–1613*, pp. 16–35. Cambridge University Press, 2012. DOI: https://doi.org/10.1017/CBO9781139060189.002.

[17] Hu, N., Rustamov, R. M. and Guibas, L. J. Stable and Informative Spectral Signatures for Graph Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2313–2320, 2014. DOI: https://doi.org/10.48550/arXiv.1304.1572.

[18] Kinney, A. F. *The Oxford Handbook of Shakespeare.* Oxford University Press, 2011. DOI: https://doi.org/10.1093/oxfordhb/9780199566105.001.0001.

[19] von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, Vol. 17(4): pp. 395–416, 2007. DOI: https://doi.org/10.1007/s11222-007-9033-z.

[20] Misra, H., Ikbal, S., Bourlard, H. and Hermansky, H. Spectral entropy based feature for robust ASR. In *Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol.1, 2004. DOI: https://doi.org/10.1109/ICASSP.2004.1325955.

[21] Mørup, M. and Hansen, L. K. Archetypal analysis for machine learning. In *2010 IEEE International Workshop on Machine Learning for Signal Processing*, 2010. DOI: https://doi.org/10.1109/MLSP.2010.5589222.

[22] Morehead State University. *Alphabetical & Historical Lists of Shakespeare's Plays*. Available at https://research.moreheadstate.edu/shakespeare. Accessed: March 2026.

[23] Müller, M. Dynamic time warping. In *Information Retrieval for Music and Motion*, pp. 69–84. Springer, Berlin, 2007. DOI: https://doi.org/10.1007/978-3-540-74048-3_4.

[24] Newman, M. *Networks: An Introduction*. Oxford University Press, Oxford, 2010. DOI: https://doi.org/10.1093/acprof:oso/9780199206650.001.0001.

[25] Open Source Shakespeare. *Shakespeare's plays, listed by number of words*. Available at https://www.opensourceshakespeare.org/views/plays/plays_numwords.php. Accessed: March 2026.

[26] Panaretos, V. M. and Zemel, Y. Statistical aspects of Wasserstein distances. *Annual Review of Statistics and Its Application*, Vol.6: pp. 405–431, 2019.

[27] Polovynka, Y. Master Thesis Codes Available at: https://github.com/YuliiaPolovynka/Master-Thesis. Accessed: May 2026.

[28] Pons, P. and Latapy, M. Computing Communities in Large Networks Using Random Walks. *Lecture Notes in Computer Science*, 3733, 284–293, 2005.

[29] Rand, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, Vol. 66(336), pp. 846–850, 1971. DOI: https://doi.org/10.2307/2284239.

[30] Ramsay, J. O. and Silverman, B. W. *Functional Data Analysis*. Second Edition. Springer, New York, 2005. DOI: https://doi.org/10.1002/0470013192.bsa239.

[31] Royal Shakespeare Company. *Shakespeare's Plays: Tragedies, Comedies, and Histories*. Available at https://www.rsc.org.uk/shakespeares-plays/tragedies-comedies-histories. Accessed: March 2026.

[32] Royal Shakespeare Company. *Timeline of Shakespeare's plays*. Available at https://www.rsc.org.uk/shakespeares-plays/histories-timeline/timeline. Accessed: March 2026.

[33] Rubner, Y., Tomasi, C. and Guibas, L. J. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, Vol. 40: pp. 99–121, 2000. DOI: https://doi.org/10.1023/A:1026543900054.

[34] Sakoe, H. and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 26(1): pp. 43–49, 1978. DOI: https://doi.org/10.1109/TASSP.1978.1163055.

[35] Salvador, S. and Chan, P. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, Vol. 11(5): pp. 561–580, 2007.

[36] Santambrogio, F. *Optimal Transport for Applied Mathematicians*. Birkhäuser Cham, 2015. DOI: https://doi.org/10.1007/978-3-319-20828-2.

[37] Senin, P. Dynamic Time Warping Algorithm Review. *Information and Computer Science Department, University of Hawaii*, 2008.

[38] Solomon, J., Rustamov, R., Guibas, L. and Butscher, A. Earth mover's distances on discrete surfaces. *ACM Transactions on Graphics*, Vol. 33(4): pp. 1–12, 2014. DOI: https://doi.org/10.1145/2601097.2601175.

[39] Stehlíková, B. Archetypal analysis of interest rates in selected European Union countries. In *Proceedings of International Relations 2023: Current Issues of World Economy and Politics*, pp. 453–462, 2023.

[40] Stiller, J., Nettle, D. and Dunbar, R. The small world of Shakespeare's plays. *Human Nature*, Vol. 14: pp. 397–408, 2003. DOI: https://doi.org/10.1007/s12110-003-1013-1.

[41] Sun, J., Ovsjanikov, M. and Guibas, L. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, Vol. 28(5): pp. 1383–1392, 2009. DOI: https://doi.org/10.1111/j.1467-8659.2009.01515.x.

[42] Sueur, J., Aubin, T. and Simonis, C. Seewave: a free modular tool for sound analysis and synthesis. *Bioacoustics*, Vol. 18: pp. 213–226, 2008. DOI: https://doi.org/10.1080/09524622.2008.9753600.

[43] Swisher, C. and Shamir, L. A data science and machine learning approach to continuous analysis of Shakespeare's plays. *Journal of Data Mining & Digital Humanities*, Article 10829, 2023. DOI: https://doi.org/10.46298/jdmdh.10829.

[44] Tsitsulin, A., Mottin, D., Karras, P., Bronstein, A. and Müller, E. NetLSD: Hearing the Shape of a Graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2347–2356, 2018. DOI: https://doi.org/10.1145/3219819.3219991.

[45] Villani, C. *Topics in Optimal Transportation*. Graduate Studies in Mathematics, Vol. 58, 2003.

[46] Villani, C. *Optimal Transport: Old and New*. Springer, Berlin, Heidelberg, 2009. DOI: https://doi.org/10.1007/978-3-540-71050-9.

[47] Vinué, G. Anthropometry: An R package for analysis of anthropometric data. *Journal of Statistical Software*, Vol. 77(6): pp. 1–39, 2017. DOI: https://doi.org/10.18637/jss.v077.i06.

[48] Vinué, G., Epifanio, I. and Alemany, S. Archetypoids: A new approach to define representative archetypal data. *Computational Statistics & Data Analysis*, Vol. 87: pp. 102–115, 2015. DOI: https://doi.org/10.1016/j.csda.2015.01.018.

[49] Xue, S., Lüdtke, J., Sylvester, T. and Jacobs, A. Reading Shakespeare sonnets: Combining quantitative narrative analysis and predictive modeling – an eye tracking study. *Journal of Eye Movement Research*, Vol. 12(5):2, pp. 1–16, 2019. DOI: https://doi.org/10.16910/jemr.12.5.2.