

---

University of Washington  
**Working Group on Model-Based Clustering**

32nd Summer Session

Université Côte d'Azur  
Nice, July 21–25, 2025



Organized by

Luca Scrucca, Charles Bouveyron, Gilles Celeux, Bettina Grün,  
Brendan Murphy, Rebecca Nugent, Adrian Raftery, Cinzia Viroli,  
Pierre-Alexandre Mattei, Marco Corneli, and Vincent Vandewalle

**WORKING GROUP BOOK**

---

**DO NOT CITE OR DISTRIBUTE ANY UNPUBLISHED MATERIAL**

## Contents

<b>Program</b>	<b>3</b>
<b>Elena Erosheva</b>	<b>4</b>
<b>Andrea Cappozzo</b>	<b>5</b>
<b>Florence Forbes</b>	<b>48</b>
<b>Cinzia Viroli</b>	<b>132</b>
<b>Claire Gormley</b>	<b>176</b>
<b>Bettina Grun</b>	<b>239</b>
<b>Brendan Murphy</b>	<b>256</b>
<b>Marco Corneli</b>	<b>272</b>
<b>Vincent Vandewalle</b>	<b>284</b>
<b>Adrian Raftery</b>	<b>285</b>
<b>Michael Fop</b>	<b>286</b>
<b>Christian Hennig</b>	<b>343</b>
<b>Charles Bouveyron</b>	<b>384</b>
<b>Luca Scrucca</b>	<b>409</b>
<b>Pierre Mattei</b>	<b>429</b>
<b>Naisyin Wang</b>	<b>430</b>
<b>Daniel Sewell</b>	<b>431</b>
<b>Riccardo Rastelli</b>	<b>469</b>
<b>Poster session</b>	<b>507</b>
<b>Software session</b>	<b>539</b>

## Program

Day (Chair)	Time	Speaker	Title
Monday (C. Bouveyron)	09:00-10:20	Elena Erosheva University of Washington	Bayesian rank-clustering
	10:40-12:00	Andrea Cappozzo Università Cattolica Milano	Lifetime data, frailties, and random covariates: model-based clustering for COVID-19 heart failure patients
Tuesday (A. Raftery)	09:00-10:20	Florence Forbes INRIA Grenoble Rhone-Alpes	Incremental inference of high dimensional elliptical mixtures from large data volumes: Application to magnetic resonance fingerprinting
	10:40-12:00	Poster Flash Session	
	14:00-16:00	Poster session	
Wednesday (B. Grün)	9:00-10:20	Cinzia Viroli Università di Bologna	Some contributions to microclustering: a frequentist perspective
	10:40-10:55	Claire Gormley University College Dublin	Integrated differential analysis of multi-omics data using a joint mixture model: idiffomix
	11:00-11:15	Bettina Grun WU Vienna University	Model-based clustering of spherical data
	11:20-11:35	Brendan Murphy University College Dublin	Model-based clustering and variable selection for multivariate count data
	11:40-11:55	Marco Cornelis Université Côte d'Azur	A deep dynamic latent block model for co-clustering of zero-inflated data matrices
	14:00-16:00	Software session	
Thursday (C. Viroli)	09:00-10:20	Vincent Vandewalle Université Côte d'Azur	Multiple partition clustering
	10:40-10:55	Adrian Raftery University of Washington	John H. Wolfe (1933-2024): the inventor of model-based clustering
	11:00-11:15	Michael Fop University College Dublin	Latent space co-clustering for multiplex networks
	11:20-11:35	Christian Hennig Università di Bologna	The role of visualisation in cluster analysis
	11:40-11:55	Charles Bouveyron Université Côte d'Azur	The deep latent position block model for the clustering of nodes in multi-graphs
Friday (B. Murphy)	09:00-10:20	Luca Scrucca Università di Bologna	A model-based clustering approach for bounded data and its applications
	10:40-10:55	Pierre Mattei INRIA - Université Côte d'Azur	Establishing a new asteroid taxonomy using model-based clustering
	11:00-11:15	Naisyin Wang University of Michigan	A note about perturbed systems to preserve privacy
	11:20-11:35	Daniel Sewell University of Iowa	Camouflaged connections: Noise clusters in network community detection
	11:40-11:55	Riccardo Rastelli University College Dublin	A zero-inflated Poisson latent position cluster model

## Elena Erosheva

*Material list:*

---

## Andrea Cappozzo

### *Material list:*

Cappozzo A. (2025) Lifetime data, frailties, and random covariates: model-based clustering for COVID-19 heart failure patients. WGMBC 2025 slides.

Caldera L., Cappozzo A., Masci C., Forlani M., Leoni O., Antonelli B., Paganoni A.M., Ieva F. (2025) Cluster-weighted modeling of lifetime hierarchical data for profiling COVID-19 heart failure patients. Unpublished manuscript.

# Lifetime data, frailties, and random covariates: model-based clustering for COVID-19 heart failure patients

Working Group on Model-Based Clustering Summer Session  
Université Côte d'Azur, Nice

✉ andrea.cappozzo@unicatt.it  
👤 andreacappozzo.rbind.io  
🦋 andreacappozzo.bsky.social

July 21, 2025

1

## Coauthors



Luca Caldera



Chiara Masci



Anna Paganoni



Francesca Ieva

Members and former members of the  
**MOX laboratory of mathematical modelling and scientific computing**  
at Polimi Department of Mathematics,  
working within the **Health Analytics** research area.

2

# Problem setting

3

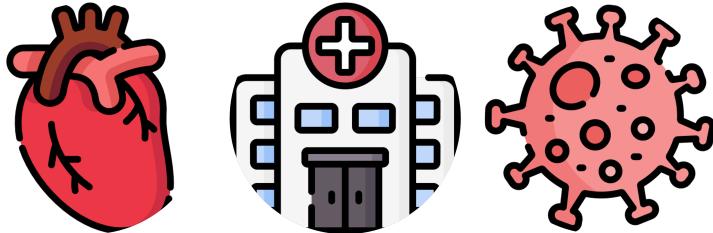
## Heart Failure (HF)



- Heart failure (HF) appears where the heart cannot supply enough blood and oxygen.
- HF often caused by **heart damage or high blood pressure**, leading to breathlessness, fatigue, and fluid retention.
- HF is **common, costly, and potentially fatal**: often causing hospital stays in older adults.

4

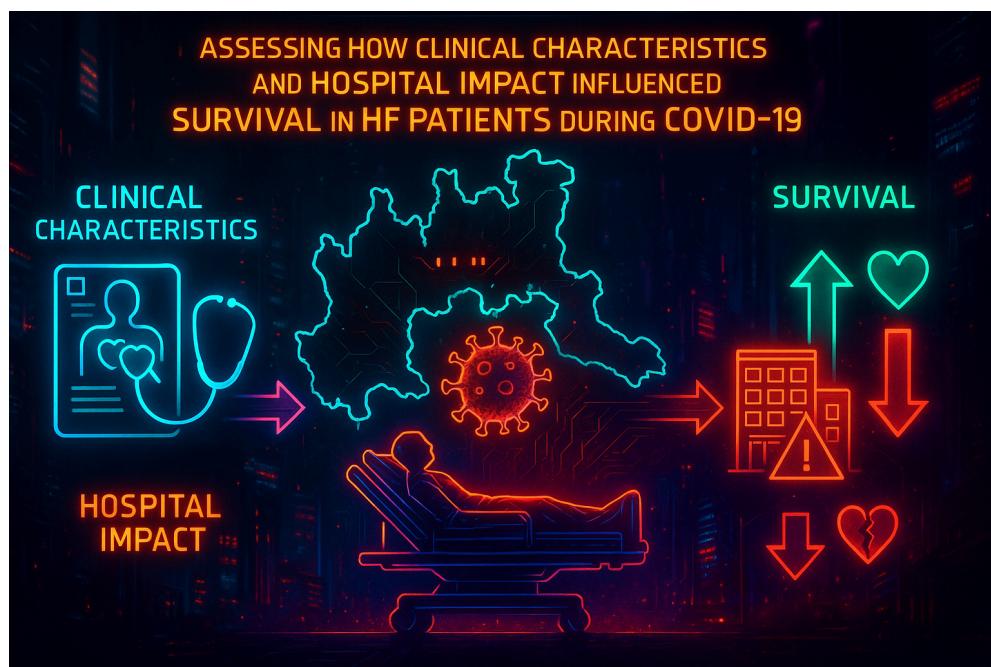
# HF, COVID-19 and Healthcare Systems



- COVID-19 pandemic worsened management of chronic conditions like HF.
- HF patients faced higher risk of severe illness and death ([Adeghate, Eid, and Singh 2021](#)).
- The Lombardy region (Italy) saw severe healthcare strain and resource shortages early in the pandemic.

5

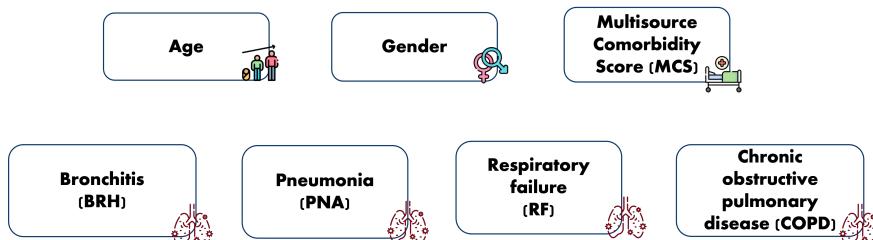
## Study aim



6

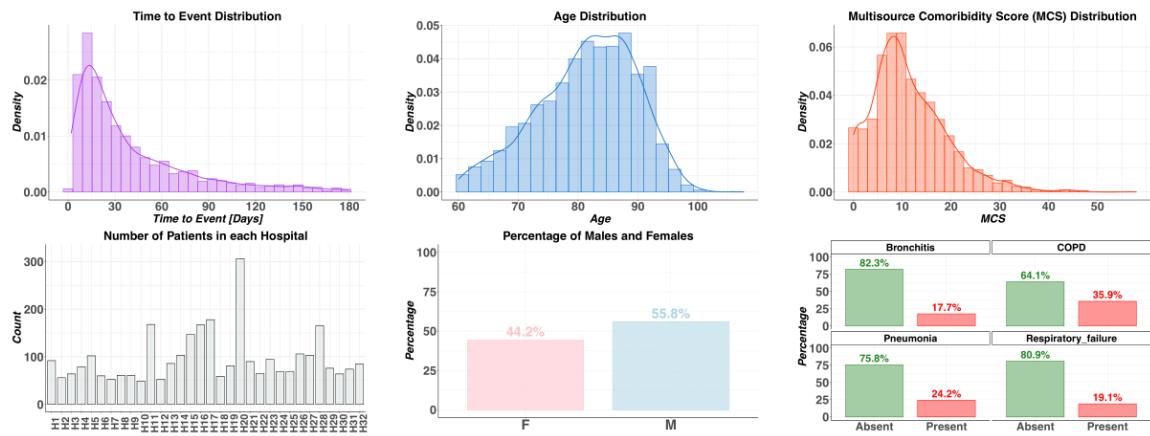
# Lombardy region dataset

- **Time span:** From January 2020 to June 2021
- $N = 3086$  **HF patients** hospitalized for COVID-19
- **Hierarchical Structure:**  $J = 32$  different hospitals
- **Response variable:** time to death within 90 days
- **Covariates:**



7

# Lombardy region dataset



8

# Details on the (modified) MCS

- Validated index summarizing **comorbidity burden** unrelated to primary diagnosis ([Corrao et al. 2017](#))
- Each disease is assigned a **score**, and the **sum of these scores** defines the MCS
- Proxy for **overall health status** (higher = worse)
- Modified to **exclude COVID-relevant respiratory diseases**, as also done in Caldera et al. ([2025](#))

9

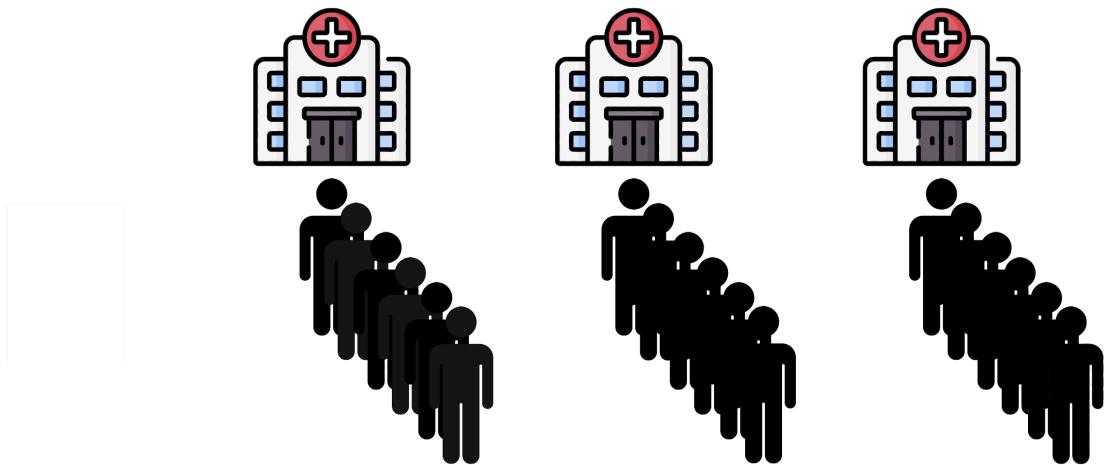
# Details on the (modified) MCS

Comorbidity	Score
Metastatic cancer	18
Alcohol abuse	11
Non-metastatic cancer	10
Tuberculosis	10
Psychosis	8
Liver diseases	8
Drugs for anxiety	6
Weight loss	6
Dementia	6
Drugs for malignancies	5
Parkinson's disease	5
Lymphoma	5
Paralysis	5
Coagulopathy	5
Fluid disorders	4
Kidney diseases	4

Comorbidity	Score
Kidney dialysis	4
Heart failure	4
Other neurological disorders	3
Rheumatic diseases	3
Brain diseases	3
Anemia	3
Diabetes	2
Gout	2
Epilepsy	2
Ulcer diseases	2
Myocardial infarction	1
Drugs for coronary	1
Valvular diseases	1
Arrhythmia	1
Obesity	1
Hypothyroidism	1

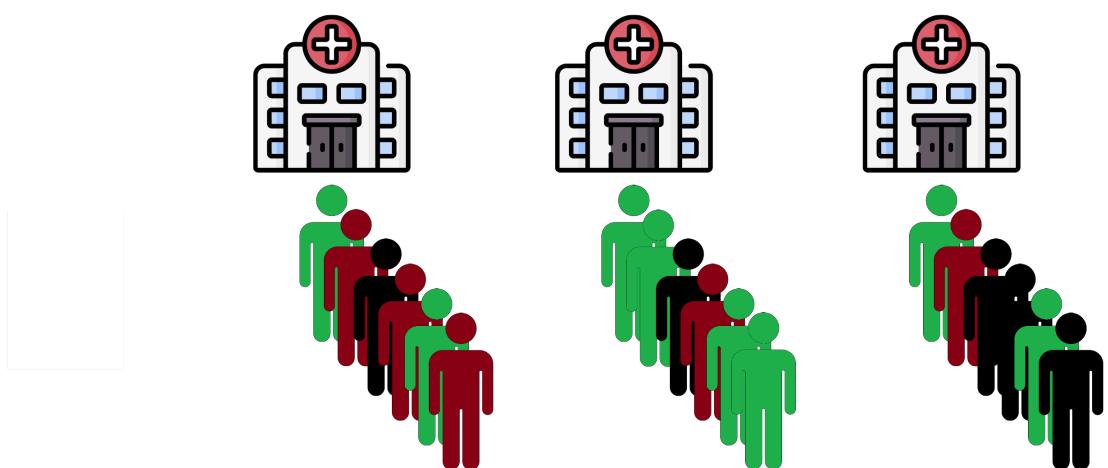
10

# Clinical goals



11

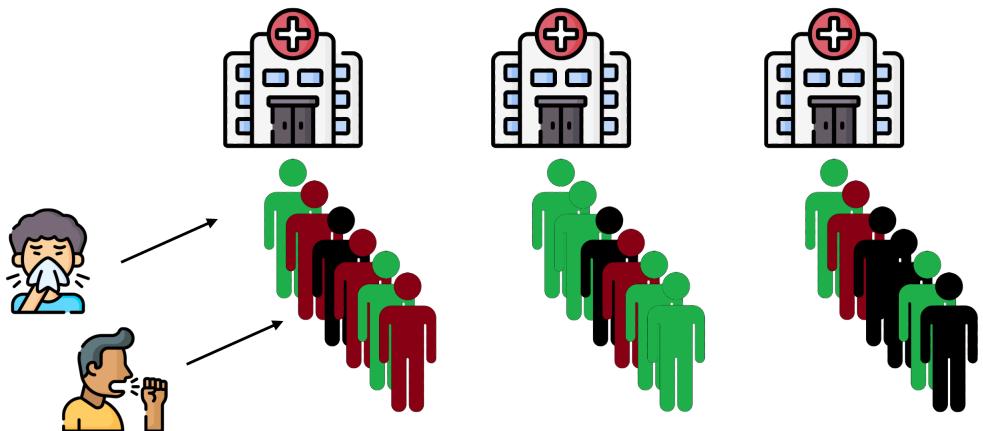
# Clinical goals



- Cluster patients based on clinical features and survival.

12

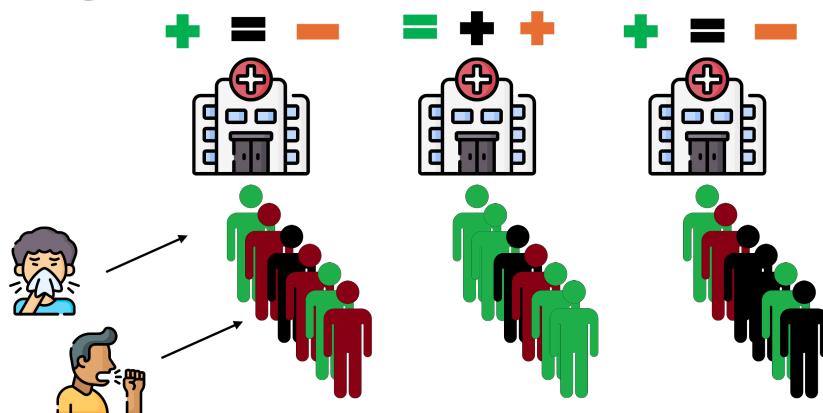
# Clinical goals



- Cluster patients based on clinical features and survival.
- Analyze cluster-specific comorbidity effects.

13

# Clinical goals



- Cluster patients based on clinical features and survival.
- Analyze cluster-specific comorbidity effects.
- Assess the hospital-effect on the different clusters.

14

# Methodology

15

## Statistical framework

### Problem

A model for time-to-event data capturing **two sources of heterogeneity**

1. Due to the **known hierarchical structure of the data**
2. Due to the **latent patient-specific characteristics**

### Idea

Integrate **Parametric Frailty Models** in the **Cluster Weighted Model** specification

16

# Prior work: Cluster Weighted Model

- Nonlinear function fitting of an input ( $X$ )-output ( $Y$ ) relation (Gershensonfeld 1997)
- CWM defines a very general **family of mixture models** (Ingrassia, Minotti, and Vittadini 2012; Ingrassia et al. 2015)

$$p(Y, X; \psi) = \sum_{g=1}^G \tau_g p(y|x; \gamma_g, \beta_g, \theta_g) \phi(u; \mu_g, \Sigma_g) \xi(v; \pi_g)$$

- $\tau_g$  mixture weights  $\tau_g > 0$  for all  $g$ ,  $\sum_{g=1}^G \tau_g = 1$
- $p(y|x; \gamma_g, \beta_g, \theta_g)$  **conditional density** of  $Y|X$
- $x = (u, v)$  where  $u$  is a  $p$ -variate vector of **continuous covariates**
- $x = (u, v)$  where  $v$  is a  $q$ -variate vector of **categorical covariates**
- $\phi(u; \mu_g, \Sigma_g)$   $p$ -variate **Gaussian density** with mean vector  $\mu_g$  and covariance  $\Sigma_g$
- $\xi(v; \pi_g)$  product of  $q$  **independent multinomials** with parameters  $\pi_g$

17

# Prior work: Parametric Frailty Models

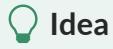
- Approach for modeling **time-to-event hierarchical data** with parametric functions for both **baseline hazard** and **frailty distribution** (Munda, Rotolo, and Legrand 2012)
- The model is defined in terms of **conditional hazard**

$$h(y_{ij}|m_j, x_{ij}; \gamma, \beta) = h_0(y_{ij}; \gamma)m_j \exp\{x_{ij}^T \beta\}$$

- $y_{ij} = \min\{t_{ij}, c_{ij}\}$ 
  - $t_{ij}$  survival time for individual  $i$  in group  $j$
  - $c_{ij}$  censoring time for individual  $i$  in group  $j$
- $h_0(\cdot; \gamma)$ : **baseline hazard function** modeling the survival parameterized by  $\gamma$
- $m_j$ : **shared frailty term** accounting for unobservable grouping-effect
- $\exp\{x_{ij}^T \beta\}$ : covariates effect on the survival governed by coefficients  $\beta$

18

# Back to the original problem



Idea

Incorporate PFM in the **conditional density**  $p(y|x; \gamma_g, \beta_g, \theta_g)$  of the CWM



Existing approaches

- MG-CWM: Multilevel Gaussian cluster-weighted model ([Berta et al. 2016](#))
- ML-CWM: Multilevel logistic cluster-weighted model ([Berta and Vinciotti 2019](#))
- Modified ML-CWM for dichotomous covariate dependence ([Caldera et al. 2025](#))



Proposed solution: MixparfmCWM

Model-Based Clustering of Lifetime Data with Frailties and Random Covariates

19

## Model formulation: notation

- Sample of hierarchical time-to-event data  $(y_{ij}, \delta_{ij}, \mathbf{x}_{ij})$
- Patient  $i, i = 1, \dots, n_j$ , within hospital  $j, j = 1, \dots, J$
- $y_{ij} = \min\{t_{ij}, c_{ij}\}$ 
  - survival time  $t_{ij}$
  - censoring time  $c_{ij}$
- $\delta_{ij} = I(t_{ij} \leq c_{ij})$  event indicator
- $\mathbf{x}_{ij} = (\mathbf{u}_{ij}, \mathbf{v}_{ij})$  vector of covariates
- $N = \sum_{j=1}^J n_j$

20

# Model formulation: $p(y|x; \gamma_g, \beta_g, \theta_g)$

- The **conditional likelihood** for patients in hospital  $j$  assigned to cluster  $g$  is (Klein and Moeschberger 2006):

$$\prod_{i \in R_{jg}} h(y_{ij} | m_{jg}, \mathbf{x}_{ij}; \gamma_g, \beta_g)^{\delta_{ij}} S(y_{ij} | m_{jg}, \mathbf{x}_{ij}; \gamma_g, \beta_g)$$

- Notice that all terms in the Parametric Frailty Model now depend on  $g$ !
- $R_{jg}$  defines the indexes of observations in hospital  $j$  assigned to cluster  $g$
- The term needed for the CWM is obtained integrating out the frailty:

$$\bar{f}_{ig}(\gamma_g, \beta_g, \theta_g) = \int_0^{+\infty} \prod_{i \in R_{jg}} h(y_{ij} | m_{jg}, \mathbf{x}_{ij}; \gamma_g, \beta_g)^{\delta_{ij}} S(y_{ij} | m_{jg}, \mathbf{x}_{ij}; \gamma_g, \beta_g) f_M(m_{jg}; \theta_g) dm_{jg}.$$

- $f_M(\cdot; \theta_g)$ : Gamma distribution with mean 1 and unknown variance  $\theta_g$

# Model formulation: objective function

- Incorporating the last term in the CWM and considering a **Classification Maximum Likelihood (CML)** approach yields the following objective function:

$$L(\boldsymbol{\psi}) = \prod_{g=1}^G \prod_{j=1}^J L_{jg}^S(\boldsymbol{\gamma}_g, \boldsymbol{\beta}_g, \theta_g) \prod_{i \in R_{jg}} \tau_g \phi(\mathbf{u}_{ij}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \xi(\mathbf{v}_{ij}; \boldsymbol{\pi}_g).$$

- $\boldsymbol{\psi} = \{\boldsymbol{\beta}_g, \theta_g, \boldsymbol{\gamma}_g, \tau_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\pi}_g\}_{g=1}^G$  set of model parameters to be estimated
- Why CML?
  - **Clustering-focused:** treats class indicators as unknown parameters.
  - **Computationally efficient:** the “all-or-nothing” assignment simplifies computation.
- **Classification Log-Likelihood:** based on derivatives of the Laplace transform of the frailty

22

# Model Estimation and Selection

- Tailored EM-algorithms are devised for maximizing the objective function
  - Classification EM ([Celeux and Govaert 1993](#))
  - Stochastic EM ([Bordes and Chauveau 2016](#))
- M-step makes use of the `parfm` subroutines ([Munda, Rotolo, and Legrand 2012](#)) to achieve **flexibility and efficiency**
- The **Bayesian Information Criterion (BIC)** is employed for selecting
  - Number of clusters  $G$
  - Baseline distribution (Exponential, Weibull, Lognormal, and more)
  - Frailty distribution (Gamma, Lognormal, and more)

23

# Application and Results

24

## Application: model setting

- Response variable  $y$ : Time from hospitalization to death
- Continuous covariates  $\mathbf{u}$  ( $p = 2$ ):
  - Age
  - Multisource Comorbidity Score (MCS)
- Categorical covariates  $\mathbf{v}$  ( $q = 3$ ):
  - Sex
  - Chronic obstructive pulmonary disease (COPD)
  - Bronchitis (BRH)
- Regression covariates  $\mathbf{x}$ :
  - Respiratory failure (RF)
  - Pneumonia (PNA)

25

# Application: model selection

Baseline	G=1	G=2	G=3	G=4	G=5
Exponential	-54198.20	-54155.42	-53297.05	-53386.16	-53313.37
Weibull	-5376.28	-53743.27	-52891.27	-52982.42	-52947.69
Lognormal	-53372.23	-53355.70	<b>-52594.26</b>	-52610.39	-52635.10
Gompertz	-54206.23	-54169.07	-53322.20	-53427.78	-53408.48

- Best model according to BIC
  - $G = 3$
  - Lognormal baseline distribution

26

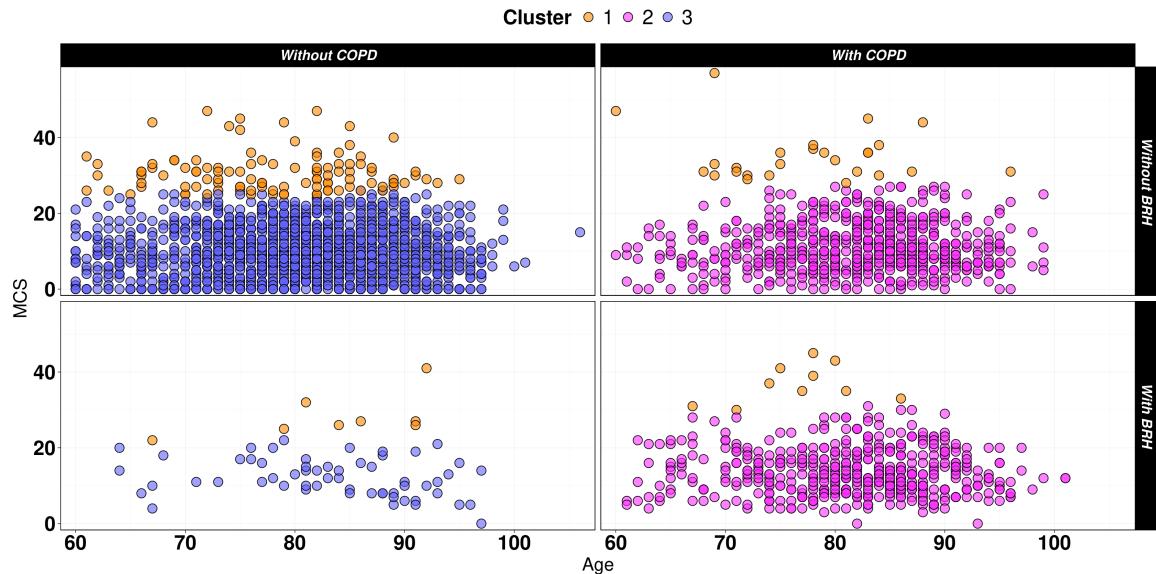
# Latent patients profiles

		Cluster 1	Cluster 2	Cluster 3
$\hat{n}_g$		151	1071	1864
$\hat{\mu}_g$	Age	78.1	81.8	81.4
	MCS	31.8	12.1	9.4
$\hat{\pi}_g$	Sex	0.39 F	0.44 F	0.45 F
	COPD	0.24 Y	0.98 Y	0.01 Y
	BRH	0.12 Y	0.44 Y	0.03 Y
$\hat{\theta}_g$		0.299	<b>0.139</b>	<b>0.188</b>
$\hat{\beta}_g$	PNA	-0.016	<b>0.235</b>	<b>0.167</b>
	RF	0.031	-0.043	0.072

- Cluster 1: Severe comorbidities
- Cluster 2: Preexisting respiratory diseases
- Cluster 3: No preexisting respiratory diseases

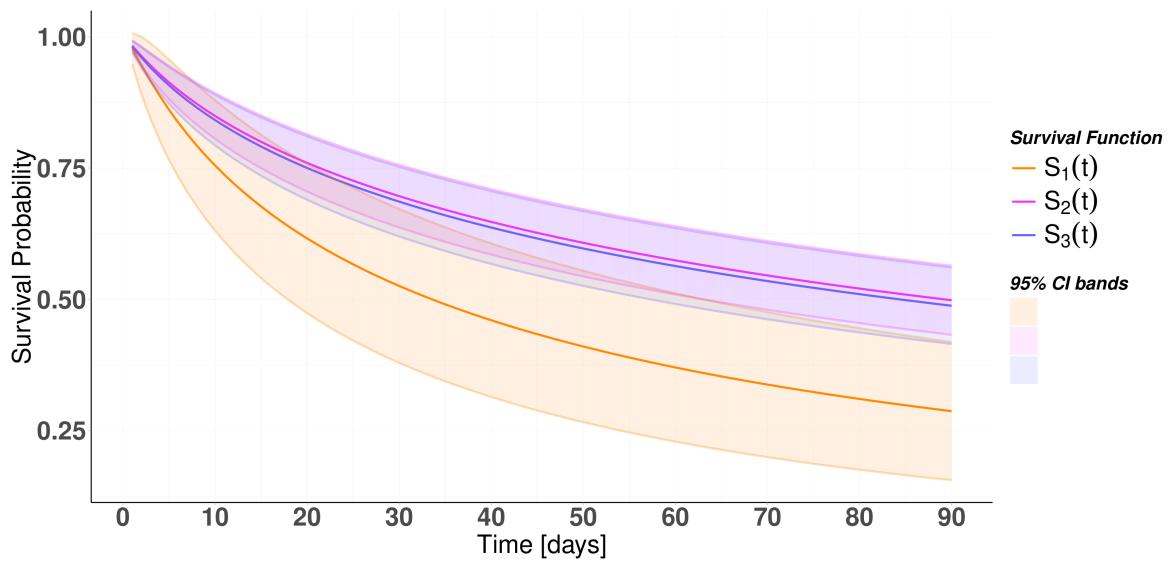
27

# Clusters in the covariates space



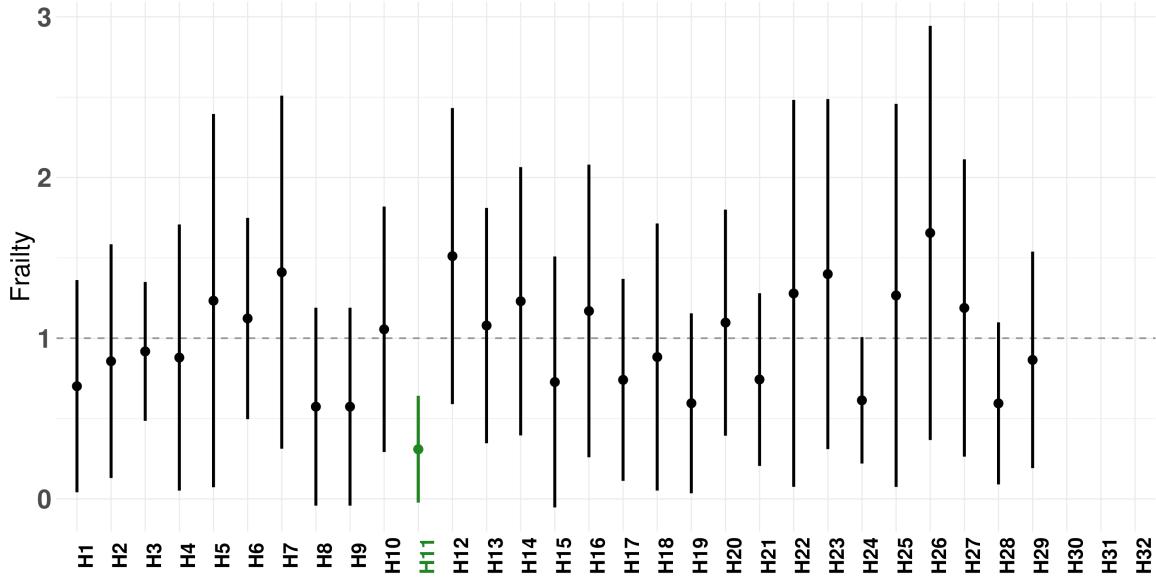
28

# Estimated survival functions



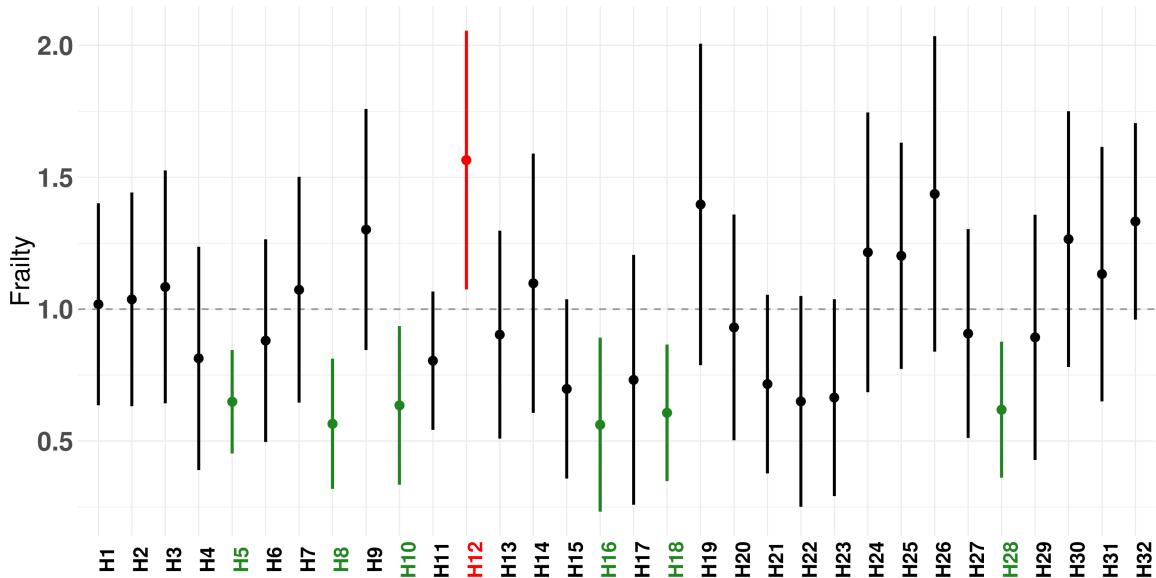
29

## Frailty Cluster 1: Severe comorbidities



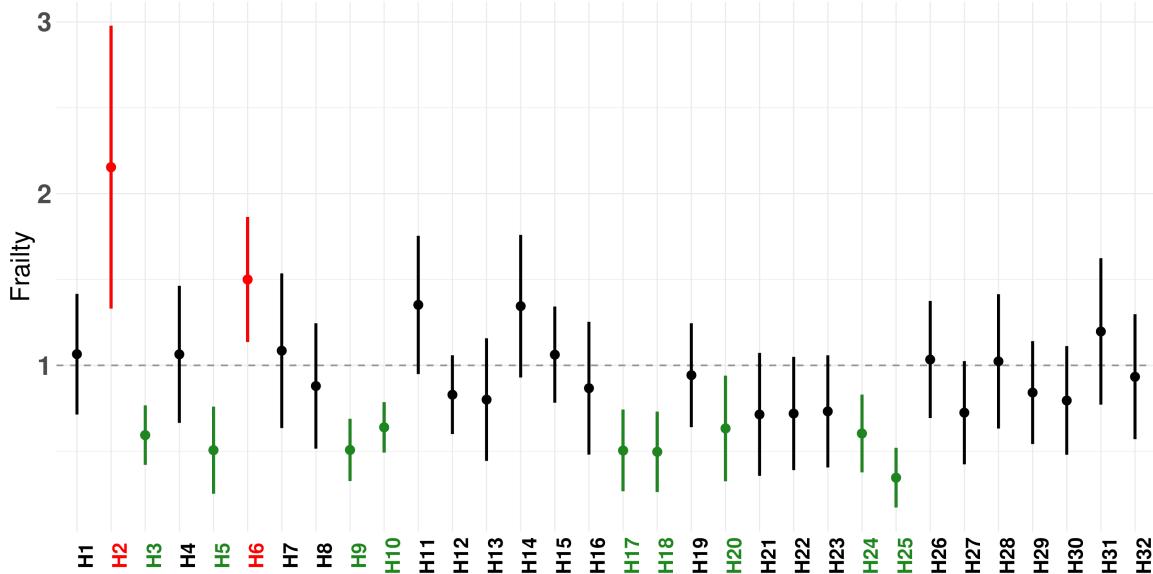
30

## Frailty Cluster 2: Preexisting respiratory diseases



31

## Frailty Cluster 3: No preexisting respiratory diseases



32

## Conclusion and discussion

- Clustering method for time-to-event data with random covariates and frailties
  - Helps identify **latent clusters** with distinct survival patterns
  - Reveals how the **hierarchy effect** impacts survival within each cluster
- Analyzed survival of **HF patients** hospitalized for **COVID-19** in Lombardy, Italy.
  - **No hospital effect** in patients with critical health conditions at admission
  - **Significant hospital effect** on generally healthy patients
- Limitations
  - **Independence** assumed for patients in same hospital but different clusters
  - Does not account for changing patient characteristics and risk **over time**

33

# Acknowledgments

This work is part of the ENHANCE-HEART project: Efficacy evaluatioN of the therapeutic-care pathWays, of the heAlthcare providers effects aNd of the risk stratifiCation in patiEnts suffering from HEART failure.



**Regione  
Lombardia**



**ARIA**  
AZIENDA REGIONALE PER  
L'INNOVAZIONE E GLI ACQUISTI

34

# References

- Adeghate, Ernest A., Nabil Eid, and Jaipaul Singh. 2021. "Mechanisms of COVID-19-induced heart failure: a short review." *Heart Failure Reviews* 26 (2): 363–69. <https://doi.org/10.1007/s10741-020-10037-x>.
- Berta, Paolo, Salvatore Ingrassia, Antonio Punzo, and Giorgio Vittadini. 2016. "Multilevel cluster-weighted models for the evaluation of hospitals." *METRON* 74 (3): 275–92. <https://doi.org/10.1007/s40300-016-0098-3>.
- Berta, Paolo, and Veronica Vinciotti. 2019. "Multilevel logistic cluster-weighted model for outcome evaluation in health care." *Statistical Analysis and Data Mining: The ASA Data Science Journal* 12 (5): 434–43. <https://doi.org/10.1002/sam.11421>.
- Bordes, Laurent, and Didier Chauveau. 2016. "Stochastic EM algorithms for parametric and semiparametric mixture models for right-censored lifetime data." *Computational Statistics* 31 (4): 1513–38. <https://doi.org/10.1007/s00180-016-0661-7>.
- Caldera, Luca, Chiara Masci, Andrea Cappozzo, Marco Forlani, Barbara Antonelli, Olivia Leoni, and Francesca Ieva. 2025. "Uncovering mortality patterns and hospital effects in COVID-19 heart failure patients: a novel multilevel logistic

- cluster-weighted modeling approach." *Biometrics* 81 (2).  
<https://doi.org/10.1093/biomtc/ujaf046>.
- Cleuz, Gilles, and Gérard Govaert. 1993. "Comparison of the mixture and the classification maximum likelihood in cluster analysis." *Journal of Statistical Computation and Simulation* 47 (3-4): 127–46.  
<https://doi.org/10.1080/00949659308811525>.
- Corrao, Giovanni, Federico Rea, Mirko Di Martino, Rossana De Palma, Salvatore Scondotto, Danilo Fusco, Adele Lallo, et al. 2017. "Developing and validating a novel multisource comorbidity score from administrative data: a large population-based cohort study from Italy." *BMJ Open* 7 (12): e019503.  
<https://doi.org/10.1136/bmjopen-2017-019503>.
- Gershenfeld, Neil. 1997. "Nonlinear Inference and Cluster-Weighted Modeling." *Annals of the New York Academy of Sciences* 808 (1 Nonlinear Sig): 18–24.  
<https://doi.org/10.1111/j.1749-6632.1997.tb51651.x>.
- Ingrassia, Salvatore, Simona C. Minotti, and Giorgio Vittadini. 2012. "Local Statistical Modeling via a Cluster-Weighted Approach with Elliptical Distributions." *Journal of Classification* 29 (3): 363–401. <https://doi.org/10.1007/s00357-012-9114-3>.
- Ingrassia, Salvatore, Antonio Punzo, Giorgio Vittadini, and Simona C. Minotti. 2015. "The Generalized Linear Mixed Cluster-Weighted Model." *Journal of Classification* 32 (1): 85–113. <https://doi.org/10.1007/s00357-015-9175-1>.

---

# CLUSTER-WEIGHTED MODELING OF LIFETIME HIERARCHICAL DATA FOR PROFILING COVID-19 HEART FAILURE PATIENTS

---

**Luca Caldera**

MOX, Department of Mathematics  
Politecnico di Milano  
Milan, IT  
luca.caldera@polimi.it

**Andrea Cappozzo**

Department of Statistical Sciences  
Università Cattolica del Sacro Cuore  
Milan, IT  
andrea.cappozzo@unicatt.it

**Chiara Masci**

Department of Economics, Management, and Quantitative Methods  
University of Milan  
Milan, IT  
chiara.masci@unimi.it

**Marco Forlani**

ARIA S.p.A. - Divisione Digital Information HUB  
Milan, IT  
marco.forlani@ext.ariaspa.it

**Barbara Antonelli**

ARIA S.p.A. - Divisione Digital Information HUB  
Milan, IT  
barbaraantonelli@ariaspa.it

**Olivia Leoni**

U.O. Osservatorio Epidemiologico, DG Welfare  
Regione Lombardia  
Milan, IT  
olivia\_leoni@regione.lombardia.it

**Anna Maria Paganoni**

MOX, Department of Mathematics  
Politecnico di Milano  
Milan, IT  
anna.paganoni@polimi.it

**Francesca Ieva**

MOX, Department of Mathematics  
Politecnico di Milano  
Milan, IT  
francesca.ieva@polimi.it

July 15, 2025

## ABSTRACT

This study investigates the heterogeneity in survival times among COVID-19 patients with Heart Failure (HF) hospitalized in the Lombardy region of Italy during the pandemic. To address this, we propose a novel mixture model for right-censored lifetime data that incorporates random effects and allows for local distributions of the explanatory variables. Our approach identifies latent clusters of patients while estimating component-specific covariate effects on survival, taking into account the hierarchical structure induced by the healthcare facility. Specifically, a shared frailty term, unique to each cluster, captures hospital-level variability enabling a twofold decoupling of survival heterogeneity across both clusters and hierarchies. Two EM-based algorithms, namely a Classification EM (CEM) and a Stochastic EM (SEM), are proposed for parameter estimation. The devised methodology effectively uncovers latent patient profiles, evaluates within-cluster hospital effects, and quantifies the impact of respiratory conditions on survival. Our findings provide new information on the

complex interplay between the impacts of HF, COVID-19, and healthcare facilities on public health, highlighting the importance of personalized and context-sensitive clinical strategies.

**Keywords** Cluster-weighted models · Frailty Survival models · Expectation–Maximization algorithm · Health care system · Hierarchical data · Multilevel models

## 1 Introduction

The COVID-19 pandemic has placed extraordinary pressure on healthcare systems worldwide, intensifying the challenges associated with managing chronic conditions such as heart failure (HF). Patients with pre-existing cardiovascular diseases, including HF, have shown increased vulnerability and increased mortality when infected with SARS-CoV-2 [Rey et al., 2020, Bader et al., 2021, Adeghate et al., 2021]. To comprehensively analyze the complex interplay between COVID-19 and HF, it is therefore crucial to consider the broader clinical and systemic context. This includes accounting for variability in patient-level clinical characteristics and hospital-related factors within a unified quantitative framework capable of capturing their intricate interactions. Such an integrated perspective is especially important for areas severely affected by the pandemic, such as the Lombardy region in Italy, which experienced one of the highest burdens during the early stages of the outbreak. Accurately assessing survival outcomes and isolating the impact of hospital-specific factors in this context requires modeling approaches capable of handling heterogeneous data sources and stratified patient populations.

To address this need, Cluster Weighted Models (CWMs) offer a flexible framework for jointly modeling covariates and outcomes in the presence of latent subpopulations [Gershenfeld, 1997, Ingrassia et al., 2012, 2014]. Although CWMs have proven effective in various domains involving mixed or structured data [Berta et al., 2016, Berta and Vinciotti, 2019, Berta et al., 2024, Caldera et al., 2025], existing formulations are not designed to handle time-to-event outcomes, a crucial limitation in the context of survival analysis.

Motivated by the problem of profiling HF patients hospitalized for COVID-19 in the Lombardy region during the pandemic, this paper proposes a novel hierarchical survival model that extends the CWM framework to handle nested time-to-event responses. The devised methodology captures cluster-specific relationships between covariates and survival outcomes through a shared frailty term, making it well suited for heterogeneous time-to-event data with an inherent hierarchical structure, such as hospital of admission in medical applications. More specifically, the frailty model introduced by Vaupel et al. [1979] accounts for the dispersion arising from the hierarchy incorporating a multiplicative factor known as frailty. The frailty can be modeled parametrically, typically with a Gamma or Lognormal distributions, or through a semiparametric approach. Detailed introductory reviews on this topic can be found in Abrahantes et al. [2007], Austin [2017] and Balan and Putter [2020].

By incorporating a parametric frailty term into the CWM specification, we propose a novel methodology that enables the joint identification of latent patient clusters with distinct survival patterns, while also uncovering the impact of the known hierarchical structure on survival within each cluster. From a clinical perspective, the main objectives of this study are the following.

- Identify relevant latent subpopulations of HF patients hospitalized for COVID-19, based on their clinical records.
- Evaluate how different medical facilities influence the hazard of death for individuals with distinct clinical profiles.
- Investigate the impact of respiratory pathologies on mortality risk between clusters of patients.
- Integrate these multiple sources of information to produce customized survival curves that stratify death risk by patient profiles, respiratory conditions, and hospital of admission.

The remainder of the paper is organized as follows. In Section 2, we present the administrative database of the Lombardy region, highlighting the clinical information of the patient that motivated our study. Section 3 details the proposed methodology, including the model setting and the estimation procedure. Section 4 presents the application of the novel method to the administrative database of the Lombardy region, along with the corresponding results. In Section 5, we perform a simulation study to assess the performance of the devised methodology under a controlled scenario. Finally, Section 6 concludes the work with key insights and suggestions for future research.

## 2 Data Description

The data set considered originates from the administrative database for health care of the Lombardy region, which is responsible for the comprehensive recording and aggregation of various health services. Specifically, we consider patients diagnosed with heart failure (HF) between January 1, 2018, and December 31, 2020. From this group, we select those who were subsequently hospitalized with COVID-19<sup>1</sup> in the Lombardy region between January 31, 2020 and June 18, 2021. The hierarchy is represented by the hospitals in which patients are admitted. Hospitals with fewer than 50 patients were excluded to ensure sufficient sample sizes for a reliable estimate of frailty-specific parameters. Patients with hospital transfers for medical reasons or repeated infections were also excluded. Our final sample includes  $N = 3086$  patients hospitalized in  $J = 32$  different hospitals in the Lombardy region. For each patient, the data set provides details about personal information, the admission facility, and clinical status. For the time-to-event analysis, we selected a 90-day observation window. Patients who survived or died after this period were treated as censored. The patient-level variables we include are:

- **Time:** it represents either the survival time or the observation period, depending on the patient's status.
- **Status:** a binary variable with a value of 1 if the event occurs within the time window and 0 otherwise.
- **Age:** The age of the patient at the time of hospital admission.
- **Gender:** The biological sex of the patient.
- **Modified Multisource-Comorbidity Score (MCS):** The MCS, introduced by Corrao et al. [2017], is a validated index that summarizes comorbidity, defined as the cumulative burden of diseases not related to the primary diagnosis of a patient. It serves as a reliable proxy for overall health status. In this study, we employ a modified version of the MCS, previously used by Caldera et al. [2025], which excludes respiratory diseases of particular medical relevance to COVID-19 infection. The resulting score is a quantitative variable that theoretically varies from 0, indicating no comorbidities, to 150, corresponding to the presence of all the conditions considered. Details on the computation of this index are provided in Supplement B.
- **Respiratory diseases:** These are represented as dichotomous variables that indicate the presence or absence of specific respiratory conditions. In detail,
  - Pneumonia (PNA): An acute inflammation of the lungs caused by infection.
  - Respiratory Failure (RF): A condition in which blood oxygen levels drop to critical low levels or carbon dioxide levels rise dangerously high.
  - Chronic obstructive pulmonary disease (COPD): A common lung disease that restricts airflow and causes breathing difficulties. It includes conditions such as emphysema and chronic bronchitis. In COPD, the lungs may be damaged or obstructed by mucus. Common symptoms include coughing (often with phlegm), shortness of breath, wheezing, and fatigue.
  - Bronchitis (BRH): Inflammation of the bronchi typically due to infection, resulting in irritation and swelling.

A brief summary of the most important descriptive statistics is provided in Table 1 for continuous variables and Table 2 for categorical variables, respectively.

Disentangling heterogeneity in survival times due to respiratory diseases and hospital effects across latent patient clusters presents a complex learning challenge. We address this by introducing a novel multilevel cluster-weighted model for lifetime data, which is detailed in the following section.

Table 1: Summary of Continuous Variables in Patients with Heart Failure and COVID-19 – Lombardy Region Dataset. Note: Descriptive statistics of variable Time refer only to observations for which the event is observed (i.e., Status = 1).

Variable	Mean	Std.Dev	1 <sup>st</sup> Quartile	Median	3 <sup>rd</sup> Quartile	Min	Max
Time [Days]	19.2	18.04	6.0	12.0	26.0	1.0	89.0
Age	81.361	8.273	76.0	82.0	88.0	60.0	106.0
MCS	11.468	7.768	6.0	10.0	16.0	0	57.0

<sup>1</sup>A patient is classified as having HF if their records show hospitalizations or ER visits under DRG code 127 (“Heart Failure and Shock”) per the Lombardy Region’s MS-DRG v40 system. This includes primary or secondary diagnoses of HF (ICD-9-CM: 428.\* or related conditions (e.g. ICD-9-CM: 402.01, 402.11, 402.91). Hospitalizations for COVID-19 are identified using a dedicated flag based on regional coding guidelines during the pandemic.

Table 2: Summary of Categorical Variables in Patients with Heart Failure and COVID-19 – Lombardy Region Dataset.

Variable	Levels	Frequency
Status	Deceased	44.75%
	Alive	55.25%
Gender	Male	55.83%
	Female	44.17%
COPD	Present	24.21%
	Not Present	75.79%
BRH	Present	17.70%
	Not Present	83.30%
PNA	Present	35.87%
	Not Present	64.13%
RF	Present	19.05%
	Not Present	80.95%

### 3 Methodology

#### 3.1 Preliminaries and related work

Consider a survival response variable  $T$  and a set of covariates  $\mathbf{X} = (\mathbf{U}, \mathbf{V})$ , where  $\mathbf{U}$  denotes a  $p$ -dimensional vector of continuous variables and  $\mathbf{V}$  represents a  $q$ -dimensional vector of categorical variables. As is customary in survival analysis, we assume that the response  $T$  may be right-censored. Therefore, the target variable is defined as  $Y = \min\{T, C\}$ , where  $C$  is a nonnegative random variable that is independent of  $T$  and represents the censoring mechanism. Furthermore, we observe a censoring indicator  $\delta$ , which is equal to 1 if  $T$  is observed and 0 otherwise. We consider  $\mathbf{X}$  and  $Y$  defined in a finite space  $\Omega$  with values in  $\mathbb{R}^{(p+q)} \times \mathbb{R}^+$ , which is assumed to be partitioned into  $G$  clusters denoted  $\Omega_1, \dots, \Omega_G$ . Given this setup, the joint probability across the clusters can be factorized as follows:

$$p(Y, \mathbf{X}; \psi) = \sum_{g=1}^G \tau_g p(y|\mathbf{x}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g, \theta_g) \phi(\mathbf{u}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \xi(\mathbf{v}; \boldsymbol{\pi}_g), \quad (1)$$

where  $\tau_g$  represents the positive mixing weights that sum to 1,  $p(y|\mathbf{x}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g, \theta_g)$  denotes the conditional density of  $Y|\mathbf{X}$  for the  $g$ -th component, whose full specification is provided in the next subsection, and the remaining terms correspond to the marginal densities of the covariates. In detail, we model continuous features  $\mathbf{U}$  using a multivariate Gaussian distribution  $\phi(\cdot; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ , with cluster-wise different mean vectors  $\boldsymbol{\mu}_g$  and covariance matrices  $\boldsymbol{\Sigma}_g$ . The density  $\xi(\cdot; \boldsymbol{\pi}_g)$  of the  $q$  categorical covariates in  $\mathbf{V}$ , each potentially possessing a different number of categories, is given by the product of  $q$  independent multinomial distributions with cluster-wise different parameter for event probabilities  $\boldsymbol{\pi}_g$ , as in Ingrassia et al. [2015] and Berta and Vinciotti [2019]. Lastly, we use  $\psi = \{\boldsymbol{\beta}_g, \theta_g, \boldsymbol{\gamma}_g, \tau_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\pi}_g\}_{g=1}^G$  to denote the complete set of model parameters to be estimated.

The joint distribution presented in Equation (1) defines a general Cluster-Weighted Model (CWM) framework, allowing the specification of various modeling approaches depending on the choice of the conditional density  $p(y|\mathbf{x}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g, \theta_g)$ . Specifically, linear Gaussian CWMs and generalized linear CWMs arise when the conditional distributions are assumed to belong to the exponential family [Ingrassia et al., 2012, 2015]. When data exhibit a hierarchical structure, multilevel linear Gaussian CWMs and multilevel generalized linear CWMs have been proposed in Berta et al. [2016] and Berta and Vinciotti [2019], respectively. A recent extension of the latter, which incorporates dependencies among dichotomous covariates through the Ising model, has been introduced by Caldera et al. [2025].

Motivated by the clinical context described in Section 2, we extend the family of multilevel Cluster-Weighted Models to accommodate time-to-event responses by incorporating a parametric frailty term into the specification of the conditional density. The detailed formulation of the model and the resulting likelihood function are presented in sections 3.2 and 3.3.

### 3.2 On the specification of the likelihood term for the survival response

To clarify the modeling structure, we distinguish between clusters and groups, which play different roles in our framework. The clusters ( $g = 1, \dots, G$ ) are latent subpopulations identified by the mixture model, each characterized by specific distributional parameters and potentially distinct survival mechanisms. In contrast, the groups ( $j = 1, \dots, J$ ) refer to observed higher-level units, such as hospitals, clinical centers, or other natural aggregations in the data, within which individual observations are nested. Building on this distinction, we now outline the proposed framework for extending the conditional density  $p(y|\mathbf{x}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g, \theta_g)$  in Equation (1) to accommodate multilevel survival responses. Start by considering hierarchical time-to-event data, for which each statistical unit  $i$ ,  $i = 1, \dots, n_j$ , within the group  $j$ ,  $j = 1, \dots, J$ , is identified by the triplet  $(y_{ij}, \delta_{ij}, \mathbf{x}_{ij})$ , where:

- $y_{ij} = \min\{t_{ij}, c_{ij}\}$ , being  $t_{ij}$  the survival time and  $c_{ij}$  the censoring time for individual  $i$  in group  $j$ ;
- $\delta_{ij} = \mathbb{1}\{t_{ij} < c_{ij}\}$  represents the event indicator for observation  $i$  in group  $j$ ;
- $\mathbf{x}_{ij} = (\mathbf{u}_{ij}, \mathbf{v}_{ij})$  denotes the vector of covariates with  $\mathbf{u}_{ij}$  and  $\mathbf{v}_{ij}$  indicating the subset of continuous and categorical predictors for the  $ij$ -th observation, respectively.

For each latent cluster  $g$ , we adopt a shared frailty model specified in terms of conditional hazard [Duchateau and Janssen, 2008, Munda et al., 2012]:

$$h(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g) = h_0(y_{ij}; \boldsymbol{\gamma}_g)m_{jg} \exp\{\mathbf{x}_{ij}^T \boldsymbol{\beta}_g\}, \quad (2)$$

where  $\boldsymbol{\beta}_g$  is a vector of covariate coefficients and  $m_{jg}$  is the shared frailty associated with group  $j$  and cluster  $g$ , for  $j = 1, \dots, J$  and  $g = 1, \dots, G$ , assumed to be independent random variables with density functions  $f_M(\cdot; \theta_g)$ . Here,  $\theta_g$  is a cluster-specific parameter that quantifies the variability of the frailty within cluster  $g$ . The frailty term operates at the group level within each latent cluster. Specifically, for each cluster  $g$ , a shared frailty term  $m_{jg}$  is assigned to each group  $j$ , capturing unobserved heterogeneity between groups within that cluster. As the frailty is both group and cluster specific, a given group  $j$  is associated with distinct frailty terms  $m_{jg}$  across the  $G$  clusters, allowing for cluster-dependent group-level effects. Lastly, the term  $h_0(\cdot; \boldsymbol{\gamma}_g)$  represents the baseline hazard function, parameterized by the vector of parameters  $\boldsymbol{\gamma}_g$ . Its specification depends on the chosen parametric distribution, with the Weibull being the most common choice for modeling the baseline. Alternative options considered in this study include the Exponential, Gompertz, and Lognormal distributions, as reported in Table 3. Further, define the (conditional) cumulative hazard function as follows:

$$H(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g) = \int_0^{y_{ij}} h(s|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g)ds = m_{jg} \exp\{\mathbf{x}_{ij}^T \boldsymbol{\beta}_g\} H_0(y_{ij}; \boldsymbol{\gamma}_g). \quad (3)$$

Starting from Equations (2) and (3) and assuming conditional independence, we write the conditional likelihood of the observations in group  $j$  assigned to cluster  $g$  as follows:

$$\prod_{i \in R_{jg}} h(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g)^{\delta_{ij}} S(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g), \quad (4)$$

where  $S(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g) = \exp\{-H(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g)\}$ , see e.g., Klein and Moeschberger [2006], and  $R_{jg}$  contains the indexes of the observations in group  $j$  assigned to cluster  $g$ . As a final step, to obtain the likelihood contribution for the survival response used in the CWM specification of Equation (1), the unobservable random effects  $m_{jg}$  must be integrated out with respect to the marginal density of the frailty, resulting in the following expression:

$$L_{jg}^S(\boldsymbol{\gamma}_g, \boldsymbol{\beta}_g, \theta_g) = \int_0^{+\infty} \prod_{i \in R_{jg}} h(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g)^{\delta_{ij}} S(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g) f_M(m_{jg}; \theta_g) dm_{jg}. \quad (5)$$

Given a sample of  $N = \sum_{j=1}^J n_j$  observation triplets  $(\mathbf{x}_{ij}, t_{ij}, \delta_{ij})$ , a Classification Maximum Likelihood criterion is formulated to estimate the model parameters, as detailed in the next subsection.

### 3.3 Objective Function of the Model

In defining the objective function of the model, we adopt a Classification Maximum Likelihood (CML) approach, treating the latent assignment of observations to the mixture components as unknown parameters [Bryant and Williamson, 1978, Celeux and Govaert, 1992]. There are two key reasons for this choice. First, the primary objective of the motivating application is to identify clusters (also referred to as profiles) of patients. The CML framework is naturally

Table 3: Parametric distributions considered in the specification of the conditional densities for the survival response. Here,  $\phi$  and  $\Phi$  respectively indicate the probability density and the cumulative distribution of a standard Gaussian, while  $\gamma$  is used as a generic notation to represent the set of parameters for the different baselines. Table adapted from Munda et al. [2012].

Distribution	$h_0(t; \gamma)$	$H_0(t; \gamma) = \int_0^t h_0(s; \gamma) ds$	Parameter Space $\gamma$
Exponential	$\lambda$	$\lambda t$	$\lambda > 0$
Weibull	$\lambda \rho t^{\rho-1}$	$\lambda t^\rho$	$\lambda, \rho > 0$
Gompertz	$\lambda \exp(\rho t)$	$\frac{\lambda}{\rho}(\exp(\rho t) - 1)$	$\lambda, \rho > 0$
Lognormal	$\frac{\phi\left(\frac{\log(t)-\eta}{\nu}\right)}{\nu t \left[1-\Phi\left(\frac{\log(t)-\eta}{\nu}\right)\right]}$	$-\log\left[1 - \Phi\left(\frac{\log(t)-\eta}{\nu}\right)\right]$	$\eta \in \mathbb{R}, \nu > 0$

aligned with this goal, as standard clustering algorithms can be viewed as specific instances of CML criteria [see, e.g., Jain and Dubes, 1988, Celeux and Govaert, 1992, García-Escudero et al., 2008, and references therein]. Second, the “all-or-nothing” assignment inherent in the CML framework simplifies computation, enabling the use of readily available routines to independently maximize the contributions of the  $G$  components (see Section 3.4.3). In detail, based on the general CWM specification outlined in Equation (1), and incorporating the contribution of the survival response derived in the previous section, the resulting likelihood takes the following form:

$$L(\psi) = \prod_{g=1}^G \prod_{j=1}^J L_{jg}^S(\gamma_g, \beta_g, \theta_g) \prod_{i \in R_{jg}} \tau_g \phi(\mathbf{u}_{ij}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \xi(\mathbf{v}_{ij}; \boldsymbol{\pi}_g), \quad (6)$$

where the conditional density of  $Y|X$  for the  $g$ -th component now explicitly corresponds to the likelihood of a parametric frailty model, with the frailties integrated out by averaging the conditional likelihood over the frailty distribution [Munda et al., 2012]. Ultimately, the overall objective function of the model can be expressed in terms of the following classification log-likelihood:

$$\ell(\psi) = \sum_{g=1}^G \left\{ \sum_{j=1}^J \left[ \sum_{i \in R_{jg}} \delta_{ij} \left( \log h_0(y_{ij}; \gamma_g) + \mathbf{x}'_{ij} \boldsymbol{\beta}_g \right) + \log \left[ (-1)^{d_{jg}} \mathcal{L}^{(d_{jg})} \left( \sum_{i \in R_{jg}} H_0(y_{ij}; \gamma_g) \exp(\mathbf{x}'_{ij} \boldsymbol{\beta}_g); \theta_g \right) \right] \right] + \sum_{j=1}^J \sum_{i \in R_{jg}} (\log \tau_g + \log \phi(\mathbf{u}_{ij}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) + \log \xi(\mathbf{v}_{ij}; \boldsymbol{\pi}_g)) \right\}, \quad (7)$$

where,  $d_{jg} = \sum_{i \in R_{jg}} \delta_{ij}$  represents the number of events for observations belonging to group  $j$  and assigned to cluster  $g$ , and  $\mathcal{L}^{(q)}(\cdot)$  denotes the  $q$ -th derivative of the Laplace transform of the frailty distribution. The complete derivation of the objective function in Equation (7) is provided in Supplement A. Direct maximization of Equation (7) poses a complex optimization problem. To address this, we propose two EM-based algorithms for parameter estimation: one incorporating a classification step (CEM algorithm), and the other relying on a stochastic step (SEM algorithm), as detailed in the next section.

### 3.4 Model Estimation

Given the objective function of our method, expressed as the classification log-likelihood of Equation (7), we maximize it using variants of the classical Expectation-Maximization algorithm [Dempster et al., 1977]. Specifically, we introduce a “hard assignment” phase between the E-step and the M-step of the standard EM procedure, implemented either through a classification step (C-step) based on the maximum a posteriori (MAP) principle, or a stochastic step (S-step) which simulates the partition according to the posterior probabilities obtained in the E-step. Depending on how the hard assignment is performed, this leads to the Classification EM (CEM) algorithm [Celeux and Govaert, 1992]

or the Stochastic EM (SEM) algorithm [Celeux, 1985] for parameter estimation. The CEM algorithm shares the theoretical guarantees of the standard EM procedure, ensuring monotonicity of the objective function at each iteration and convergence to a stationary point. However, it is highly sensitive to initialization and may become trapped in local optima. Such drawbacks are not shared by the SEM algorithm, which was specifically developed to overcome these limitations, albeit at the expense of losing the ascent property and having more complex convergence behavior [Nielsen, 2000]. The SEM algorithm has also been successfully applied in the literature to fit mixture models with censored data [Chauveau, 1995, Bordes and Chauveau, 2016].

Despite their differing advantages and limitations, both the CEM and SEM procedures share a common characteristic: they generate a hard partition at each algorithm iteration. This significantly simplifies the M-step by enabling independent maximization of each component's contribution, considering only the units assigned to that component in the current iteration. In practice, such a characteristic enables the use of standard maximum likelihood estimates for homogeneous populations, thereby leveraging existing computational routines. This proves particularly advantageous when estimating the parametric frailty term in the conditional density of our CWM specification. Detailed algorithmic steps are described in the following subsections.

### 3.4.1 E-step

As is standard in mixture models [see, for instance, Bouveyron et al., 2019], the  $(k+1)$ -th iteration of the Expectation step involves computing the posterior probability that observation  $i$  from group  $j$  belongs to cluster  $g$ , given the parameter estimates obtained in the previous step. To formalize this, we introduce assignment indicators  $z_{ijg}$ , where  $z_{ijg} = 1$  if observation  $i$  from hospital  $j$  is assigned to the  $g$ -th component, and  $z_{ijg} = 0$  otherwise, for  $i = 1, \dots, n_j$ ,  $j = 1, \dots, J$ , and  $g = 1, \dots, G$ . Then, the estimated a posteriori probabilities are routinely updated as follows:

$$\hat{z}_{ijg}^{(k+1)} = \frac{\hat{\tau}_g^{(k)} p(y_{ij} | \mathbf{x}_{ij}; \hat{\gamma}_g^{(k)}, \hat{\beta}_g^{(k)}, \hat{\theta}_g^{(k)}) \phi(\mathbf{u}_{ij}; \hat{\mu}_g^{(k)}, \hat{\Sigma}_g^{(k)}) \xi(\mathbf{v}_{ij}; \hat{\pi}_g^{(k)})}{\sum_{c=1}^G \hat{\tau}_c^{(k)} p(y_{ij} | \mathbf{x}_{ij}; \hat{\gamma}_c^{(k)}, \hat{\beta}_c^{(k)}, \hat{\theta}_c^{(k)}) \phi(\mathbf{u}_{ij}; \hat{\mu}_c^{(k)}, \hat{\Sigma}_c^{(k)}) \xi(\mathbf{v}_{ij}; \hat{\pi}_c^{(k)})}, \quad (8)$$

where the superscript  $k$  denotes the parameter estimates obtained in the previous EM iteration.

### 3.4.2 Hard assignment via C-step or S-step

Starting from the soft assignments computed in Equation (8), we employ either the MAP rule or a stochastic sampling procedure to hard-assign the units to the  $G$  clusters. Specifically, the hard assignment update in the CEM algorithm requires the following Classification step:

$$\tilde{z}_{ijg}^{(k+1)} = \begin{cases} 1 & \text{if } g = \operatorname{argmax}_{c \in \{1, \dots, G\}} \hat{z}_{ijc}^{(k+1)} \\ 0 & \text{otherwise} \end{cases}. \quad (9)$$

When the SEM algorithm is used instead to maximize Equation (7), the hard assignment update involves the following Stochastic step:

$$\tilde{z}_{ijg}^{(k+1)} \sim \operatorname{Multinomial}(1, \hat{z}_{ij}^{(k+1)}), \quad (10)$$

where the allocation is obtained by sampling from a multinomial distribution with event probabilities

$$\hat{z}_{ij}^{(k+1)} = (\hat{z}_{ij1}^{(k+1)}, \dots, \hat{z}_{ijG}^{(k+1)}).$$

Whether using the C-step in Equation (9) or the S-step in Equation (10), the units are partitioned into  $G$  clusters whose sample size is then computed as:

$$\hat{n}_g^{(k+1)} = \sum_{j=1}^J \sum_{i=1}^{n_j} \tilde{z}_{ijg}^{(k+1)}, \quad g = 1, \dots, G.$$

Note that there exists a one-to-one correspondence between the index notation  $R_{jg}$ , as introduced in Sections 3.2 and 3.3, and the assignment indicators  $z_{ijg}$ . Specifically, at each iteration of the algorithm, the sets  $R_{jg}$  are updated to include the indices of all observations in hospital  $j$  for which  $\tilde{z}_{ijg}^{(k+1)} = 1$ .

### 3.4.3 M-step

Thanks to the partition induced by the hard assignment described in the previous section, the M-step reduces to separately maximizing the likelihood contributions of  $G$  distinct subpopulations, each with a sample size equal to  $\hat{n}_g^{(k+1)}$ , for  $g = 1, \dots, G$ . Specifically, traditional maximum likelihood estimates can be easily obtained for parameters  $\{\tau_g, \mu_g, \Sigma_g, \pi_g\}_{g=1}^G$ , which govern mixing proportions and marginal distributions of both continuous and categorical covariates. Detailed derivations of these formulas can be found, for instance, in the supplementary material of Caldera et al. [2025]. Maximizing the likelihood term related to the survival response is more complex and depends heavily on the specified parametric forms. To address this challenge, we utilize the `parfm` R package [Munda et al., 2012], which provides a unified framework for fitting parametric frailty models. This approach offers two key advantages: it ensures computational efficiency and retains flexibility by allowing users to select from a wide range of parametric distributions for both the baseline hazard and frailty term. For a comprehensive discussion of the underlying maximization strategies, we refer the reader to Munda et al. [2012].

## 3.5 Additional Details on the Estimation Procedure

We hereafter discuss several practical considerations related to the estimation procedure.

- *Initialization*: the initialization of deterministic algorithms is always a critical step, and this is especially true when dealing with time-to-event data. To date, only practical data-driven heuristics have been proposed to initialize EM algorithms in mixture models for right-censored lifetime data [Bordes and Chauveau, 2016]. However, in our approach, we can leverage the postulated differences in the covariate distributions to construct an initial partition. Specifically, we employ k-prototypes, an extension of k-means designed for data sets containing both continuous and categorical variables [Huang, 1998]. Although other established alternatives such as multiple random initializations are possible, numerical experiments on both real and synthetic data suggest that the aforementioned strategy effectively guides the algorithms toward a stable convergence path.
- *Convergence*: for the CEM algorithm, convergence is assumed when the relative difference in the objective function defined in Equation (7) between two consecutive iterations falls below a threshold  $\varepsilon$ . In our analyses, we set  $\varepsilon = 10^{-5}$ . More sophisticated convergence diagnostics are required to assess the convergence of the SEM algorithm, as it does not guarantee a monotonically increasing objective function. Specifically, the final values are defined as the ergodic mean of the sequence of parameter estimates across iterations, as proposed by Bordes and Chauveau [2016] based on the asymptotic properties established in Nielsen [2000].
- *Model selection*: the Bayesian Information Criterion [BIC; Schwarz, 1978] is used to select, in a data-driven manner, the number of clusters  $G$  as well as the parametric form of both the baseline and frailty distributions. The criterion reads:

$$\text{BIC} = 2 \cdot \ell(\hat{\psi}) - d \cdot \ln(N), \quad (11)$$

where  $\ell(\hat{\psi})$  denotes the maximized log-likelihood,  $N$  is the sample size and  $d$  represents the total number of parameters:

$$d = G(1 + m) + G \left( \frac{p(p+3)}{2} \right) + G \sum_{r=1}^q (k_r - 1) + Gb + Gf + G - 1.$$

In details,  $k_r$  defines the number of categories associated with the  $r$ -th categorical variable,  $m$  is the number of covariates included in the regression component of the shared frailty survival model,  $b$  is the number of parameters of the chosen baseline distribution, and  $f$  is the number of parameters of the frailty distribution. According to the definition of Equation (11), the best model is the one with the highest BIC.

- *Implementation*: routines for implementing the proposed methodology through both CEM and SEM algorithms have been developed in R [R Core Team, 2021], with the source code freely available at <https://github.com/AndreaCappozzo/mixparfmCWM>. As discussed in Section 3.4.3, maximization of the survival term relies on the `parfm` R package [Munda et al., 2012]. However, to accommodate the specific challenges posed by our framework, the original `parfm` routines have been slightly modified and extended. An enriched version of `parfm`, available at <https://github.com/AndreaCappozzo/parfm>, is required for the devised `mixparfmCWM` R package to function correctly.

## 4 Application and Results

We apply the proposed methodology to the Lombardy region dataset, described in Section 2, with the aim of profiling patients and modeling their hazard of death based on cluster-specific respiratory conditions and hospital effect. Both

the CEM and SEM algorithms were considered for model fitting, yielding virtually identical parameter estimates. Consequently, the results presented in the following analysis are based on the CEM algorithm.

#### 4.1 Model Setting

The variables Age, Gender, MCS, COPD, and BRH represent pre-existing conditions prior to the onset of COVID-19 pathology. Consequently, they are incorporated into the marginal distribution and modeled as random covariates within the adopted CWM framework. Specifically, continuous variables (MCS and Age) are modeled jointly using a bivariate Gaussian distribution with cluster-specific parameters  $\{\mu_g, \Sigma_g\}$ . The variables Gender, COPD, and BRH are treated as independent binary categorical variables, with one parameter vector  $\pi_g$  estimated for each cluster. These pre-existing condition variables are excluded from the set of covariates in the parametric frailty model term. In contrast, the variables PNA and RF, which generally manifest as complications arising from COVID-19 infection [Du et al., 2020, Zhou et al., 2020], are included as explanatory variables in the survival regression term. This modeling strategy is designed to allow the pre-existing conditions (Age, Gender, MCS, COPD, and BRH) to primarily inform the identification of latent clusters (i.e., patient profiles), while the influence of respiratory complications is adjusted through their effect on cluster-specific survival outcomes. Accordingly, each cluster  $g$  is associated with a vector of regression coefficients,  $\beta_g$ , capturing the impact of PNA and RF on survival.

We fit the model introduced in Section 3 by varying  $G$  over the set  $\{1, 2, 3, 4, 5\}$  and considering four baseline hazard distributions: Exponential, Weibull, Lognormal, and Gompertz. For the frailty term, we assume a Gamma distribution with a fixed mean of 1 and an unknown variance, denoted by  $\theta_g$ . Following the procedure outlined in Section 3.5, the model is fitted 20 times for each combination of  $G$  and baseline hazard distribution. To initialize the algorithm, we use k-prototypes clustering based on the five variables modeled as random: MCS, Age, Gender, COPD, and BRH. The best BIC value across the 20 runs for each combination of the number of clusters  $G$  and baseline hazard distribution is reported in Table 4. The optimal model, identified by the highest BIC, corresponds to  $G = 3$  with a Lognormal baseline distribution.

Table 4: Comparison of BIC values across all combinations of number of clusters  $G$  and Baseline Hazard Distributions. The Model with the highest BIC is highlighted in bold.

Baseline Distribution	$G = 1$	$G = 2$	$G = 3$	$G = 4$	$G = 5$
Exponential	-54198.20	-54155.42	-53297.05	-53386.16	-53313.37
Weibull	-53763.28	-53745.27	-52891.27	-52982.42	-52947.69
Lognormal	-53372.23	-53355.70	<b>-52594.26</b>	-52610.39	-52635.10
Gompertz	-54206.23	-54169.07	-53322.20	-53427.78	-53408.48

#### 4.2 Results

In this section, we present the results of the best model selected using BIC, focusing on clusters description, interpretation of survival parameters, and analysis of frailty effects. Figure 1 illustrates the partition of patients into the 3 clusters within the Age-MCS space, distinguishing between patients with and without COPD and BRH. The upper portion of Table 5 provides a summary of the parameters associated with continuous and categorical variables, as well as the number of patients in each cluster. The lower portion of the table presents the estimated values and significance of the cluster-specific survival parameters, including fixed effects, the parameters that specify the Lognormal baseline hazard distribution and the estimated variance of the Gamma distribution used to model the frailty term.

In Figure 2, we represent the baseline survival function specific to each cluster, that corresponds to the estimated survival function for a patient not affected by RF and PNA and with the frailty term fixed at its expected value, namely 1. Given that a Lognormal distribution is considered for modeling the baseline hazard, the survival function  $S(t; \hat{\eta}_g, \hat{\nu}_g)$ , for each cluster  $g \in \{1, 2, 3\}$ , is given by:

$$S(t; \hat{\eta}_g, \hat{\nu}_g) = 1 - \Phi \left( \frac{\log(t) - \hat{\eta}_g}{\hat{\nu}_g} \right),$$

where  $\hat{\eta}_g$  and  $\hat{\nu}_g$  are, for each cluster, the parameter estimates reported in the lower portion of Table 5. Additionally, Figure 2 presents the corresponding 95% confidence interval bands, computed using the delta method. Further details on the computation of these intervals are provided in Supplement C. In Figure 3, we represent the hazard rate over time for each cluster together with the effect of the statistically significant covariates. The estimated fixed effects  $\hat{\beta}_{PNA_g}$  and

$\hat{\beta}_{RF_g}$  represent the log-relative hazard for an individual in cluster  $g$  with the specific disease relative to an individual in cluster  $g$  without the disease. The interpretation of the coefficient is given in terms of the hazard ratio, which reflects the relative risk of the event occurring between the two levels of a binary covariate. A positive coefficient indicates that the hazard is higher for individuals with the condition compared to those without it, meaning the event is more likely to occur in the former group. In contrast, a negative coefficient suggests a lower hazard for individuals with the condition, indicating a protective effect. Figure 4 presents cluster-specific dot plots displaying the estimated hospital frailties along with their 95% confidence intervals. These estimates capture the unobserved heterogeneity in survival outcomes across hospitals within each cluster, reflecting the extent to which individual facilities influence the baseline hazard due to unmeasured factors, such as differences in hospital practices, available resources, or other unobserved characteristics affecting patient survival. A joint examination of Figures 2, 3, and 4 enables the interpretation and characterization of the resulting clusters.

The first cluster (depicted in orange in the plots) includes 151 patients (4.9% of the total cohort). The average age of this cluster of patients is 78.09 years, with a mean MCS of 31.79. The proportion of males exceeds that of females, as in the overall dataset. Indeed, all clusters exhibit a slightly higher proportion of male patients. With respect to respiratory conditions, 24% of patients in this cluster are affected by COPD, whereas only 12% present with BRH. In general, this cluster mainly includes individuals with a high burden of comorbidity but relatively low rates of BRH and COPD. This cluster exhibits the most unfavorable survival (see the orange curve in Figure 2), reflecting the lowest life expectancy and the highest hazard rate over the entire observation period (Figure 3). The effects of PNA and RF are not statistically significant within this cluster, suggesting that these conditions do not meaningfully influence the hazard rate for patients in this profile. This may be attributed to the already high burden of comorbidities characterizing the cluster, which likely attenuates the marginal impact of additional complications. With respect to the frailty term, there is no strong evidence of substantial heterogeneity across hospitals, as indicated by the estimated  $\hat{\theta}$  reported in Table 5 and the hospital-specific frailty estimates shown in Figure 4a. Only one facility, Hospital H11, exhibits a notable protective effect for patients in this cluster. This further supports the interpretation that the severe baseline health conditions diminish the relative influence of the treating hospital, thereby reducing observable differences in survival outcomes across healthcare facilities.

The second cluster (depicted in violet) includes 1071 patients (34.7% of the total cohort). The average age of patients in this cluster is 81.79 years, with a mean MCS of 12.13. All patients in this cluster have COPD and 44% of them also have BRH. This suggests a patient profile characterized by a very high prevalence of respiratory diseases and relatively fewer other comorbidities compared to the first cluster. The hazard rate for this cluster is lower over the entire time period (Figure 3), and the survival curve shows a higher life expectancy compared to the first cluster (Figure 2). The effect of PNA is significant, with an associated estimated coefficient of 0.235, suggesting that a patient in this cluster affected by PNA has a 23.5% higher risk of death compared to a patient without PNA. The frailty effect is significant in this cluster, as shown in Table 5, suggesting heterogeneous effects of hospitals on patients survival. Specifically, hospitals H5, H8, H10, H16, H18, and H28 demonstrate a protective effect. In contrast, the confidence interval associated with hospital H12 lies entirely above 1, indicating a significantly increased hazard of death for patients in the second cluster treated at this facility (see Figure 4b).

The third cluster (depicted in blue) comprises the largest proportion of the sample, including 1864 patients, which represents 60.4% of the total cohort. The average age in this cluster is 81.38 years, with a mean MCS of 9.44. None of the patients in this cluster have COPD, and only 3% have BRH. This cluster exhibits the lowest mean MCS and the lowest percentage of patients affected by COPD and BRH among the three clusters. Consequently, the patient profile in this cluster corresponds to the healthiest subgroup within the cohort. The survival curve and hazard rate over time closely resemble those of the second cluster (see Figures 2 and 3). This similarity, despite the comparatively better health status of patients in the third cluster, particularly in terms of comorbidities and respiratory conditions, may be attributed to the prioritization of clinical monitoring and treatment for individuals with respiratory diseases, especially during the COVID-19 pandemic [Tiotiu et al., 2021, Benfante and Scichilone, 2021, Giustivi et al., 2021]. Such prioritization likely contributed to comparable survival outcomes across these two otherwise distinct patient profiles. Notably, the effect of PNA is significant in this cluster, with an associated estimated coefficient of 0.167. This indicates that a patient in this cluster affected by PNA has a 16.7% higher risk of death compared to a patient without PNA. Additionally, the frailty effect is also significant, as detailed in Table 5. Specifically, hospitals H3, H5, H9, H10, H17, H18, H20, H24, and H25 are associated to a decreased death hazard for patients treated there, whereas hospitals H2 and H6 are associated to an increased death hazard with respect to the average (see Figure 4c).

The proposed model identified three distinct patient profiles within the cohort, allowing the estimation of cluster-specific covariate effects and frailty terms that influence the risk of hazard. This enabled the evaluation of how the impact of respiratory diseases and hospitals on the hazard varies between different patient profiles.

In particular, the impact of the hospital is most evident among generally healthy but vulnerable patients, specifically those in groups 2 and 3, where timely hospital care can significantly affect survival. This key insight comes from the model's ability to uncover heterogeneity in survival outcomes related to cluster-wise different respiratory conditions and hospital-specific factors among patient profiles. The model's capability to capture such complex interactions is further demonstrated in a synthetic setting, as presented in the following section.

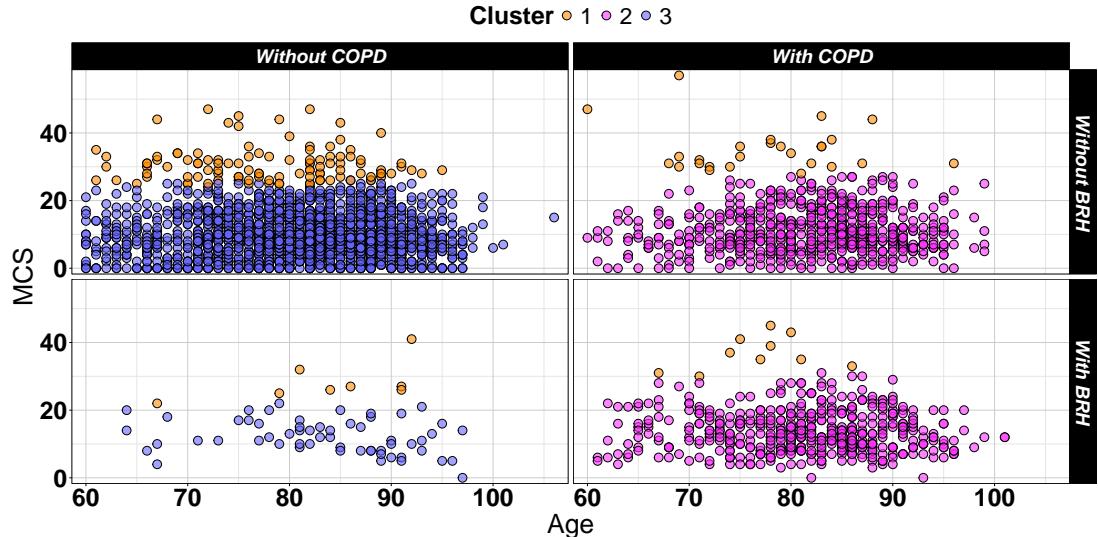


Figure 1: Scatterplots of patients in the Age–MCS feature space colored according to the three estimated clusters; stratified by Presence or Absence of COPD and BRH.

## 5 Simulation Study

We hereafter present a comprehensive simulation study to show the performance of the method in a controlled setting. We consider a Data Generating Process (DGP) that incorporates various covariates, a known hierarchy, and a latent grouping structure. Specifically, we consider the case of  $G = 3$  latent clusters and  $J = 10$  known groups. The total number of observations is set to  $N = 1500$ , equally distributed among the 10 groups. Of the 150 observations from each group, 40, 50, and 60 belong to the first, second, and third cluster, respectively. The covariates include two continuous variables ( $p = 2$ ) simulated from two independent Gaussian distributions with parameters  $\mu_g$  and  $\sigma_g$ ; one binary variable simulated using a Bernoulli distribution with parameter  $\pi_{g1}$ ; and one categorical variable with three possible outcomes simulated using a Multinomial distribution with parameter  $\pi_{g2}$ ,  $g = 1, \dots, 3$ . A Weibull distribution is considered for the baseline hazard, with cluster-wise different shape  $\rho_g$  and scale  $\lambda_g$  parameters. The frailty distribution in each cluster is modeled using a Gamma density with a mean equal to 1 and unknown variance  $\theta_g$ . To prevent frailty effects from dominating survival outcomes, we aim to keep the variance  $\theta_g$  relatively small. In the regression component, all covariates are included, resulting in a 5-dimensional coefficient vector  $\beta_g$  for each cluster  $g$ . Survival times are simulated by combining the baseline hazard with the effects of frailty and covariates. To generate the synthetic data, we utilize the `genfrail` function from the `frailtySurv` package [Monaco et al., 2018]. The complete set of true parameter values used in the simulation is reported in Table 6. In the regression component, most of the coefficients for the first cluster are specified as risk factors, meaning that the majority of entries in  $\beta_1$  are positive. In contrast, the majority of coefficients in the third cluster are specified as protective factors, with most entries in  $\beta_3$  being negative. The second cluster exhibits a mixed profile, containing both risk and protective factors. High-risk clusters are assigned larger frailty variances, whereas low-risk clusters have smaller variances. The same shape parameter  $\rho_g = 3$ ,  $\forall g = 1, 2, 3$  is applied across clusters to simulate a quadratic growth in risk, with  $\lambda_g$  varying by cluster to create distinct risk profiles. The covariate distribution parameters vary between clusters, except for the variance of the continuous covariates  $\sigma_g$ , which is fixed at  $\{1, 1\}$  for all clusters.

We simulate  $R = 100$  datasets following the DGP described above. For each dataset, we apply the proposed method by varying the number of clusters  $G \in \{1, 2, 3, 4\}$ .

Table 5: Parameters estimated by the model with  $G = 3$  clusters and a Lognormal baseline. The upper section presents the parameter estimates for the continuous covariates  $\{\hat{\mu}_g, \hat{\Sigma}_g\}_{g=1,\dots,G}$ , the categorical covariates  $\{\hat{\pi}_g\}_{g=1,\dots,G}$  and the number of patients in each cluster  $\{\hat{n}_g\}_{g=1,\dots,G}$ . The lower section provides the estimated fixed effects  $\{\hat{\beta}_g\}_{g=1,\dots,G}$ , the estimated characteristic parameters of the baseline hazard Lognormal distribution  $\{\hat{\eta}_g, \hat{\nu}_g\}_{g=1,\dots,G}$ , and the estimated parameter of the Gamma( $1, \theta$ ) distribution  $\{\hat{\theta}_g\}_{g=1,\dots,G}$  employed to model the frailty term, along with their corresponding p-values. A Wald test was performed to assess whether the frailty parameter  $\hat{\theta}_g$  differs significantly from zero in each cluster; the resulting approximate p-values are reported.

Parameter	Cluster 1	Cluster 2	Cluster 3			
$\hat{n}_g$	151	1071	1864			
$\hat{\mu}_g$	(78.09; 31.79)	(81.79; 12.13)	(81.38; 9.44)			
$\hat{\Sigma}_g$	$\begin{bmatrix} 62.69 & -2.26 \\ -2.26 & 34.13 \end{bmatrix}$	$\begin{bmatrix} 64.28 & 0.68 \\ 0.68 & 39.56 \end{bmatrix}$	$\begin{bmatrix} 70.30 & 0.14 \\ 0.14 & 36.57 \end{bmatrix}$			
$\hat{\pi}_{\text{Gender}_g}$	(0.39; 0.61)	(0.44; 0.56)	(0.45; 0.55)			
$\hat{\pi}_{\text{COPD}_g}$	(0.76; 0.24)	(0; 1)	(1; 0)			
$\hat{\pi}_{\text{BRH}_g}$	(0.88; 0.12)	(0.56; 0.44)	(0.97; 0.03)			
Parameter	Cluster 1	Cluster 2	Cluster 3			
	Estimate	$\text{Pr}(>  z )$	Estimate	$\text{Pr}(>  z )$	Estimate	$\text{Pr}(>  z )$
$\hat{\eta}_g$	-2.387		-1.409		-1.466	
$\hat{\nu}_g$	+1.754		+2.118		+2.131	
$\hat{\theta}_g$	+0.299	0.19	+0.139	0.01*	+0.188	0.004*
$\hat{\beta}_{PNA_g}$	-0.016	0.94	+0.235	0.02*	+0.167	0.08*
$\hat{\beta}_{RF_g}$	+0.031	0.91	-0.043	0.68	+0.072	0.54

Table 6: Simulation parameters related to the survival part of the model and covariates distributions in each cluster.

Cluster	J	$n_g$	$\beta_g$	$\theta_g$	$\lambda_g$	$\rho_g$		
$g = 1$	10	40	$\{0.2, -0.1, 0.3, 0.5, 0.2\}$	0.8	2	3		
$g = 2$	10	50	$\{-0.2, -0.1, 0.2, -0.3, 0.15\}$	0.6	0.7	3		
$g = 3$	10	60	$\{-0.2, 0.2, -0.3, -0.3, -0.4\}$	0.4	0.4	3		
Cluster	$\mu_g$			$\pi_{g1}$	$\pi_{g2}$			
$g = 1$	{1, -3}			{1, 1}	{0.4, 0.6}			
$g = 2$	{3, 1}			{1, 1}	{0.8, 0.2}			
$g = 3$	{5, 3}			{1, 1}	{0.2, 0.8}			
				{0.3, 0.5, 0.2}				
				{0.6, 0.1, 0.3}				
				{0.1, 0.3, 0.6}				

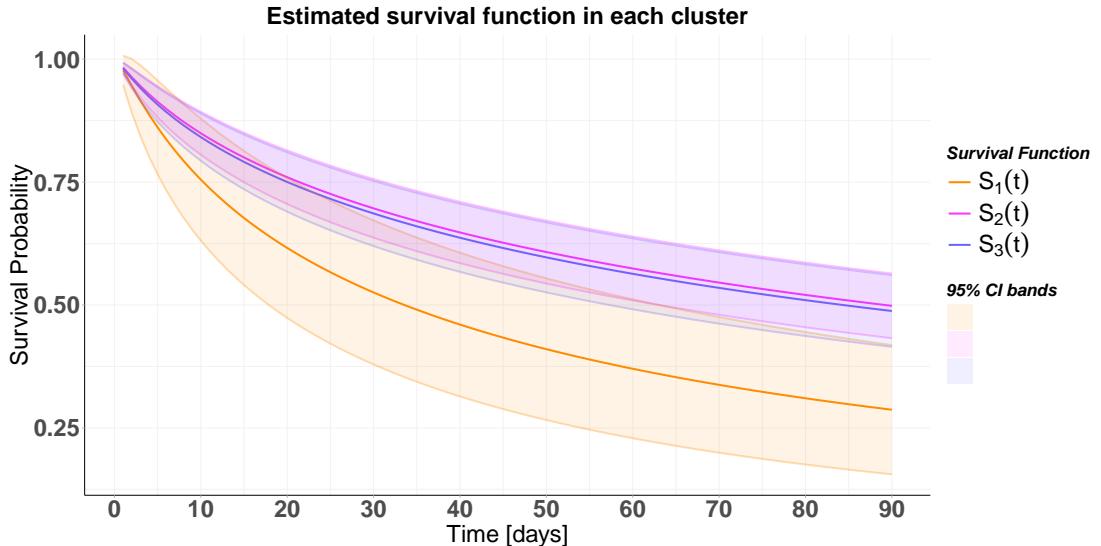


Figure 2: Estimated survival curves for the three clusters. The orange curve represents the baseline survival curve for Cluster 1, the violet curve corresponds to Cluster 2, and the blue curve represents Cluster 3. Curves refer to patients not affected by RF and PNA and with the frailty term fixed at its expected value, namely 1. Shaded bands indicate the 95% confidence intervals for the survival curves.

## 5.1 Results

For each repetition of the simulated experiment, we use the BIC to assess which model is preferred by varying the number of components  $G \in \{1, 2, 3, 4\}$ . The empirical BIC distributions shown in Figure 5 indicate that the model with  $G = 3$  clusters is preferred. Considering  $G = 3$  as the best solution, to evaluate the ability of the model to recover latent clusters, we examine the misclassification rates and the Adjusted Rand Index (ARI) [Hubert and Arabie, 1985] across the 100 runs (Figure 6). In 93 of 100 runs, the model demonstrates effective recovery of latent clusters, achieving an average misclassification rate of 0.039 ( $SD = 0.005$ ) and an ARI of 0.882 ( $SD = 0.016$ ).

In Figure 7, we compare the distributions of the estimated parameters of the simulation experiments with the true values for the case  $G = 3$ . For what concerns the survival component (Figure 7a), the parameters specific to each of the three clusters are estimated with high precision. However, the frailty variance is slightly underestimated in all three clusters. Regarding the covariates coefficients (Figure 7b), their estimates within each cluster are highly precise, effectively capturing the numerical values of risk and protective factors. Lastly, the parameters governing the random covariates are also well-estimated in most simulations (Figure 7c).

Other simulation studies, such as those that involve different numbers of latent clusters, alternative specifications of the baseline hazard function, or larger sample sizes, are certainly possible. In this work, we focus on a standard scenario that incorporates all the key components addressable by the proposed approach. The results demonstrate the strong performance of the estimation method, showing a high precision in recovering both the model parameters and the latent clusters. These findings provide a solid foundation while leaving ample room for future investigations in more complex or varied settings.

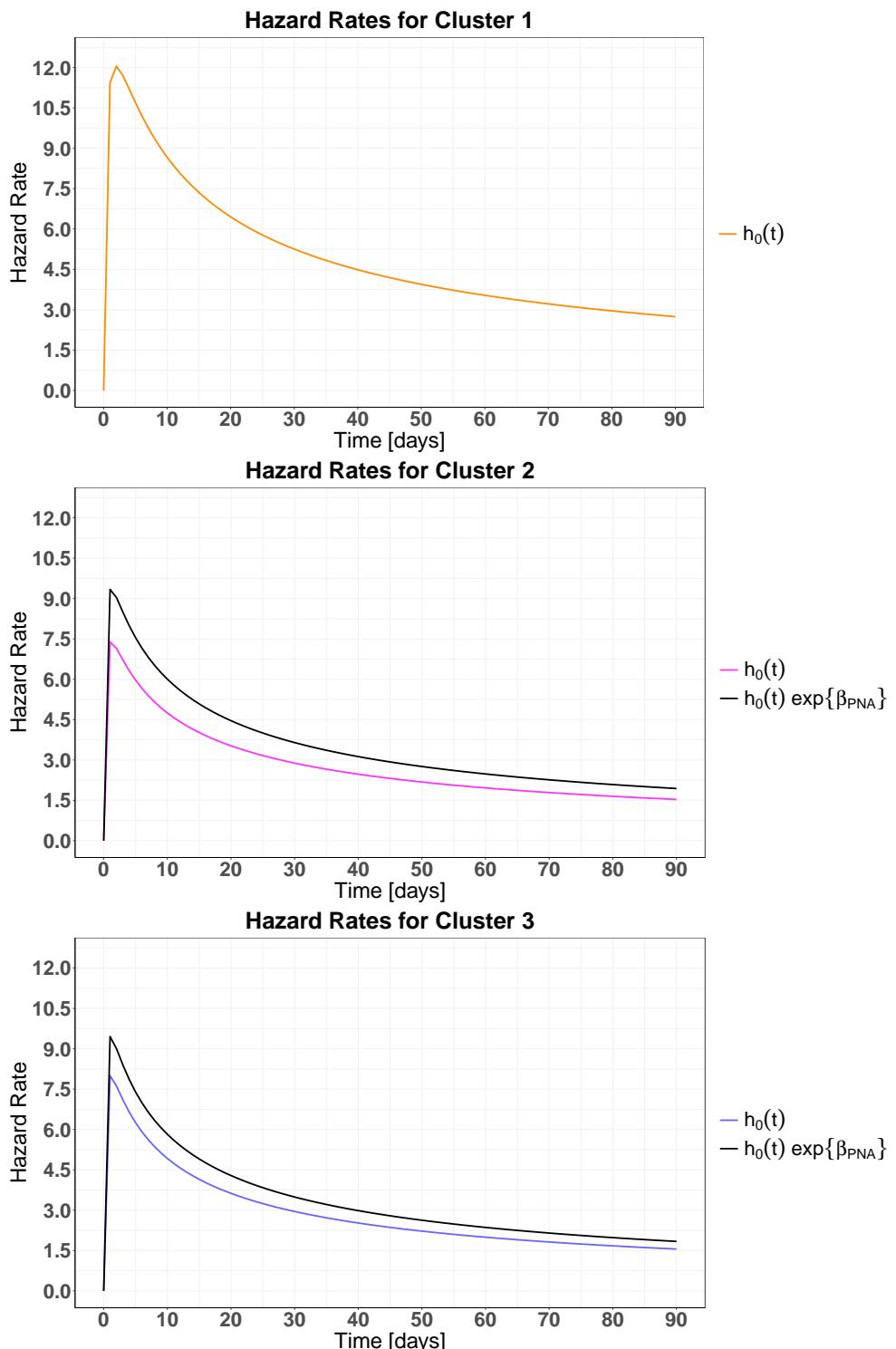


Figure 3: Estimated baseline hazard in the three clusters and effect of the significant covariates (black curve).

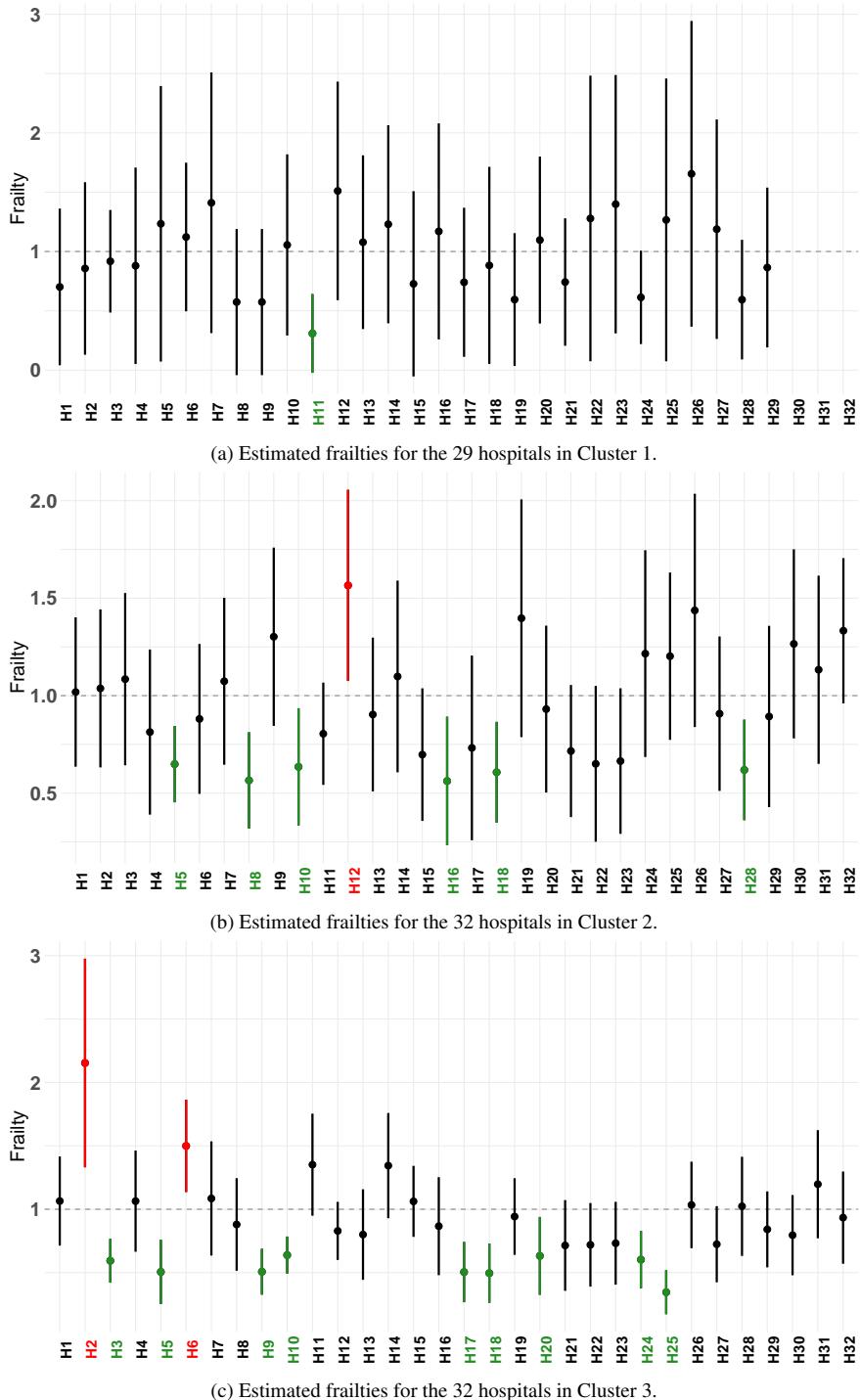


Figure 4: Estimated frailties and their 95% confidence intervals in the three clusters. Hospitals that significantly increase the hazard rate are highlighted in red, while those that decrease it are highlighted in green.

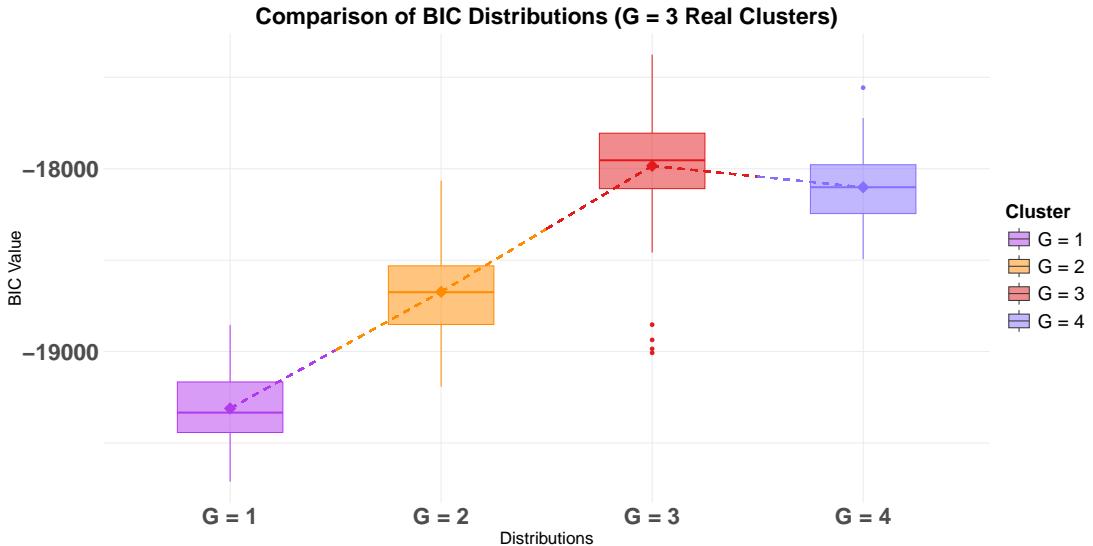


Figure 5: Comparison of BIC distributions for  $G \in \{1, 2, 3, 4\}$ . Each boxplot represents the BIC distribution for a given  $G$  across the 100 repetitions of the simulated experiment.

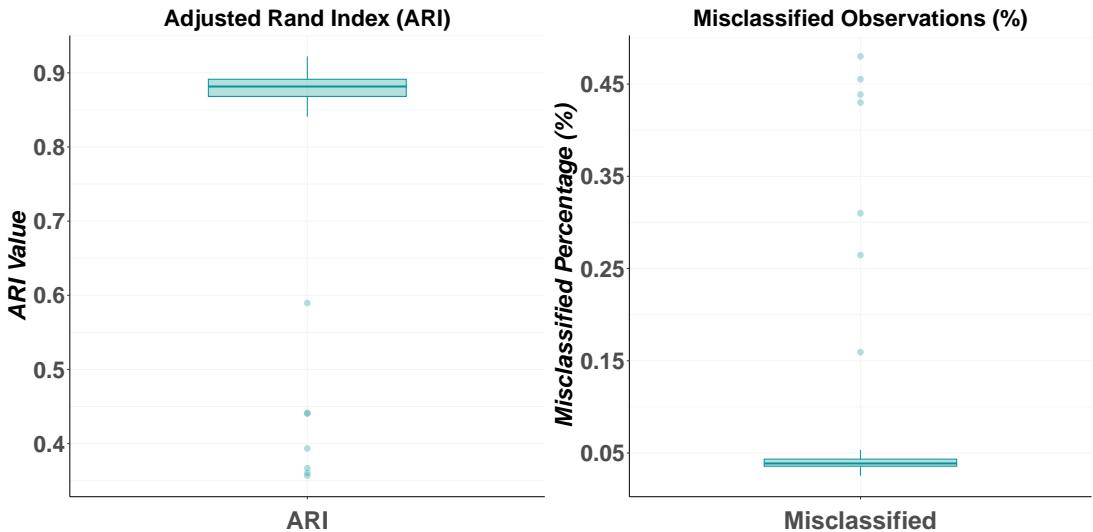


Figure 6: ARI and percentage of missclassified observations across the 100 simulations.

## 6 Discussion

This paper introduced a novel methodology that extends the CWM framework to accommodate hierarchical time-to-event outcomes. The core innovation lied in the ability of the methodology to address both the clustering of multilevel data and the inherent heterogeneity in survival analysis concurrently. The proposed methodology identified clusters with distinct survival patterns by estimating cluster-specific coefficients that influence hazard risks. This enabled the assessment of how various factors impact the hazard within each cluster. In addition, the method provided cluster-

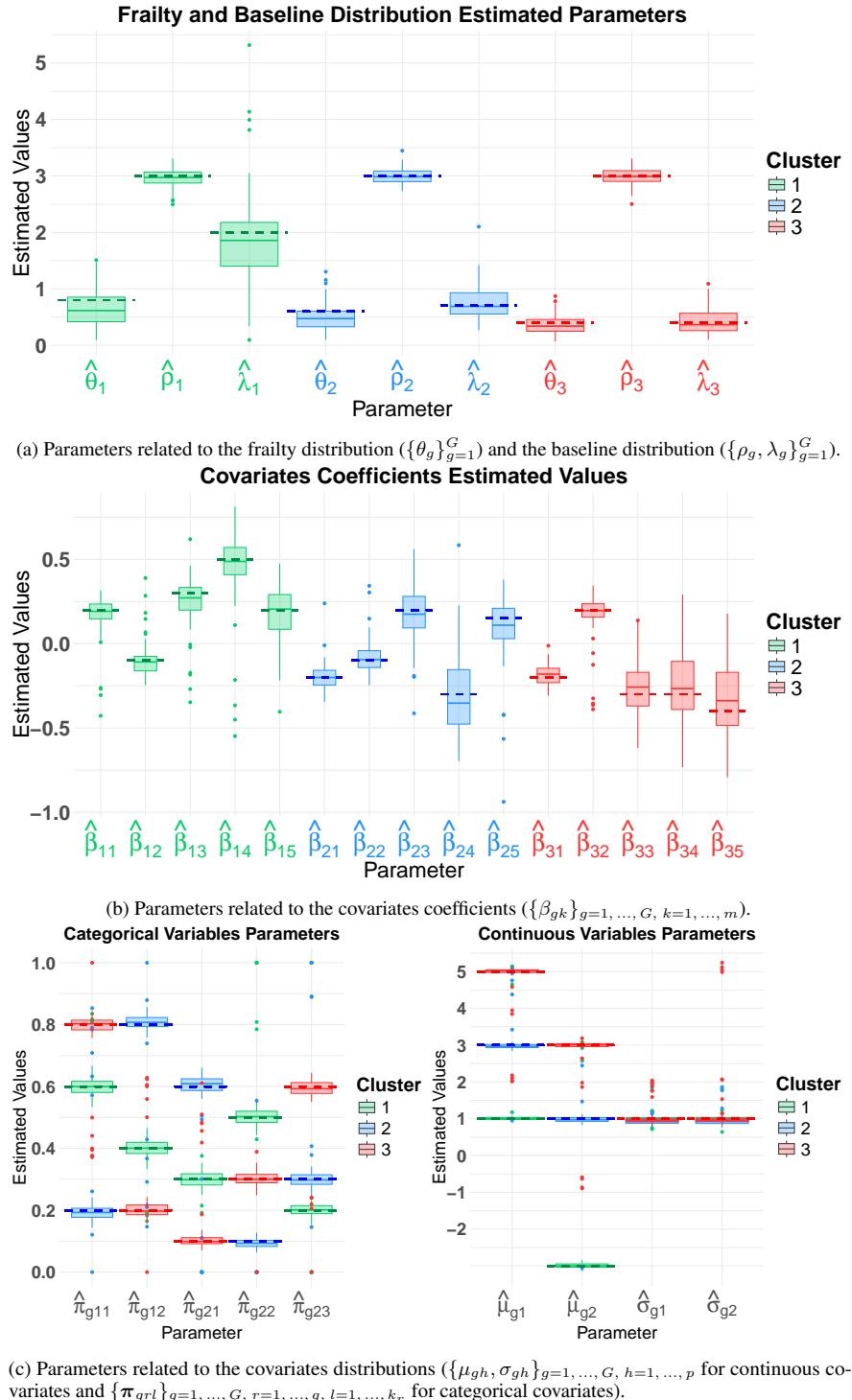


Figure 7: Comparison of the distributions of estimated parameters across the R=100 repetitions with the true values of DGP. Dashed lines represent the true values.

specific frailty estimates, capturing unobserved heterogeneity in survival outcomes and accounting for deviations from the baseline risk due to unmeasured factors. Its fully parametric structure allowed for flexible specification of both the baseline hazard and frailty distributions. Two EM-based algorithms, tailored for right-censored data, were devised for parameter estimation: one incorporating a classification step (CEM algorithm) and the other relying on a stochastic step (SEM algorithm).

This work was motivated by the analysis of a real-world administrative dataset from the Lombardy region in Italy, focusing on patients with heart failure (HF) who were hospitalized with COVID-19. Our proposed methodology enabled meaningful insights into the interaction between HF and COVID-19 during the pandemic. The analysis identified three distinct patient profiles, each exhibiting unique survival patterns. Furthermore, the approach allowed for the evaluation of respiratory conditions and hospital-level effects on individual patient profiles through cluster-specific estimates of covariate coefficients and frailty terms, respectively. These findings demonstrate the potential of the methodology to generate actionable insights for healthcare planning. By uncovering survival trends and key influencing factors, the method highlights opportunities to optimize the treatment of HF patients at the territorial level, ultimately with the aim of reducing adverse outcomes and improving the efficiency of the healthcare system.

A limitation of the proposed approach is the assumption that each patient consistently belongs to a single latent profile. While this may be reasonable in cross-sectional settings, it can be overly restrictive in longitudinal contexts, where patients' characteristics and risk profiles may change over time. Relaxing this constraint represents an important avenue for future methodological development.

Further future extensions of the proposed methodology may involve integrating the Ising model into the marginal distributions of the covariates to capture dependencies between binary variables, such as comorbidities or treatment indicators, commonly encountered in clinical and administrative datasets, along the lines of Caldera et al. [2025]. The proposed framework could also be adapted to accommodate high-dimensional data through the integration of sparsity-inducing estimation techniques. This extension would enable effective variable selection by identifying the most important covariates within each cluster, while also mitigating overfitting and improving generalizability. Developing penalized likelihood formulations specifically adapted to the cluster-weighted model of survival data would be particularly valuable for large-scale applications, such as those involving electronic health records or genomics, where the number of predictors may largely exceed the sample size. Several such approaches are currently under investigation and will be the focus of future work.

## Acknowledgments

This work is part of the ENHANCE-HEART project: Efficacy evaluatioN of the therapeutic-care patHways, of the heAlthcare providers effects aNd of the risk stratifiCation in patiEnts suffering from HEART failure. The authors thank the ‘Unità Organizzativa Osservatorio Epidemiologico Regionale’ and ARIA S.p.A for providing data and technological support. The authors gratefully acknowledge the support from the Department of Mathematics of Politecnico di Milano, which facilitated this research as part of the department’s activities of “Dipartimento di Eccellenza 2023-2027”. Chiara Masci acknowledges financial support from the Italian Ministry of University and Research (MUR) under the Department of Excellence 2023-2027 grant agreement “Centre of Excellence in Economics and Data Science” (CEEDS).

## Appendix A: Further details pertaining the model definition

### Laplace Transform and Derivatives of the Frailty Distribution

For a positive random variable  $M$  its Laplace transform is defined as:

$$\mathcal{L}(s) = \mathbb{E}[e^{-sM}] = \int_0^\infty e^{-sm} f_M(m) dm, \quad (12)$$

where  $f_M(\cdot)$  is the probability density function of  $M$ , and  $s$  is a non-negative real number. The first derivative of  $\mathcal{L}(s)$  with respect to  $s$  can be computed as follows:

$$\begin{aligned} \frac{d}{ds} \mathcal{L}(s) &= \frac{d}{ds} \int_0^\infty e^{-sm} f_M(m) dm = \text{Leibniz integral rule} \\ &= \int_0^\infty f_M(m) \frac{d}{ds} e^{-sm} dm = - \int_0^\infty m f_M(m) e^{-sm} dm. \end{aligned} \quad (13)$$

Differentiating Equation (13) again, we obtain the second derivative of  $\mathcal{L}(s)$  with respect to  $s$ :

$$\frac{d^2}{ds^2} \mathcal{L}(s) = \int_0^\infty m^2 f_M(m) e^{-sm} dm. \quad (14)$$

Therefore, the general expression of the  $q$ -th derivative of the Laplace transform  $\mathcal{L}(s)$  is given by:

$$\frac{d^q}{ds^q} \mathcal{L}(s) = \mathcal{L}^{(q)}(s) = (-1)^q \int_0^\infty m^q f_M(m) e^{-sm} dm. \quad (15)$$

In our case, the random variable  $M$  represents the frailty distribution which is parameterized by its variance  $\theta$ . Therefore, we denote the  $q$ -th derivative of the Laplace transform as dependent on  $\theta$ :

$$\frac{d^q}{ds^q} \mathcal{L}(s; \theta) = \mathcal{L}^{(q)}(s; \theta) = (-1)^q \int_0^\infty m^q f_M(m; \theta) e^{-sm} dm. \quad (16)$$

### Derivation of the classification log-likelihood

Starting from the classification likelihood defined in Equation (6) of the main paper the associated classification log-likelihood is derived as follows:

$$\begin{aligned} \ell(\psi) = \log L(\psi) &= \sum_{g=1}^G \left\{ \sum_{j=1}^J \log \left( \int_0^{+\infty} \prod_{i \in R_{jg}} h(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g)^{\delta_{ij}} S(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g) f_M(m_{jg}; \theta_g) dm_{jg} \right) + \right. \\ &\quad \left. \sum_{j=1}^J \sum_{i \in R_{jg}} (\log \tau_g + \log \phi(\mathbf{u}_{ij}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) + \log \xi(\mathbf{v}_{ij}; \boldsymbol{\pi}_g)) \right\}. \end{aligned} \quad (17)$$

where we have explicitly reported the expression of  $L_{jg}^S(\boldsymbol{\gamma}_g, \boldsymbol{\beta}_g, \theta_g)$  as per Equation (5) of the main paper. The second line of Equation (17) corresponds to the CWM contribution for the mixing proportion and the random covariates, and it directly aligns with the third row of Equation (7) in the main paper. Let us focus on the first line of Equation (17) corresponding to the contribution of the parametric frailty model for the observations in group  $j$  assigned to the  $g$ -th component:

$$\begin{aligned} &\log \left( \int_0^{+\infty} \prod_{i \in R_{jg}} h(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g)^{\delta_{ij}} S(y_{ij}|m_{jg}, \mathbf{x}_{ij}; \boldsymbol{\gamma}_g, \boldsymbol{\beta}_g) f_M(m_{jg}; \theta_g) dm_{jg} \right) = \text{Eq(2) and (3) of the main paper} \\ &= \log \left( \int_0^{+\infty} \prod_{i \in R_{jg}} (h_0(y_{ij}; \boldsymbol{\gamma}_g) m_{jg} \exp\{\mathbf{x}_{ij}^T \boldsymbol{\beta}_g\})^{\delta_{ij}} \exp\{-m_{jg} \exp\{\mathbf{x}_{ij}^T \boldsymbol{\beta}_g\} H_0(y_{ij}; \boldsymbol{\gamma}_g)\} f_M(m_{jg}; \theta_g) dm_{jg} \right) = \\ &= \log \left( \prod_{i \in R_{jg}} (h_0(y_{ij}; \boldsymbol{\gamma}_g) \exp\{\mathbf{x}_{ij}^T \boldsymbol{\beta}_g\})^{\delta_{ij}} \int_0^{+\infty} m_{jg}^{\sum_{i \in R_{jg}} \delta_{ij}} \exp \left\{ -m_{jg} \sum_{i \in R_{jg}} \exp\{\mathbf{x}_{ij}^T \boldsymbol{\beta}_g\} H_0(y_{ij}; \boldsymbol{\gamma}_g) \right\} f_M(m_{jg}; \theta_g) dm_{jg} \right). \end{aligned} \quad (18)$$

Denoting with  $s = \sum_{i \in R_{jg}} \exp\{\mathbf{x}_{ij}^T \boldsymbol{\beta}_g\} H_0(y_{ij}; \boldsymbol{\gamma}_g)$  and  $d_{jg} = \sum_{i \in R_{jg}} \delta_{ij}$  the expression in Equation (18) can be rewritten as:

$$\begin{aligned} & \log \left( \prod_{i \in R_{jg}} (h_0(y_{ij}; \boldsymbol{\gamma}_g) \exp\{\mathbf{x}_{ij}^T \boldsymbol{\beta}_g\})^{\delta_{ij}} \int_0^{+\infty} m_{jg}^{d_{jg}} \exp\{-m_{jg}s\} f_M(m_{jg}; \theta_g) dm_{jg} \right) = \text{Eq (16)} \\ &= \sum_{i \in R_{jg}} \delta_{ij} (\log h_0(y_{ij}; \boldsymbol{\gamma}_g) + \mathbf{x}_{ij}^T \boldsymbol{\beta}_g) + \log \left( (-1)^{d_{jg}} \mathcal{L}^{(d_{jg})}(s; \theta_g) \right) = \\ &= \sum_{i \in R_{jg}} \delta_{ij} (\log h_0(y_{ij}; \boldsymbol{\gamma}_g) + \mathbf{x}_{ij}^T \boldsymbol{\beta}_g) + \log \left( (-1)^{d_{jg}} \mathcal{L}^{(d_{jg})} \left( \sum_{i \in R_{jg}} \exp\{\mathbf{x}_{ij}^T \boldsymbol{\beta}_g\} H_0(y_{ij}; \boldsymbol{\gamma}_g); \theta_g \right) \right) \end{aligned} \quad (19)$$

By substituting the expression obtained from Equation (19) into Equation (17), we derive the classification log-likelihood presented in Equation (7) of the main paper.

## Appendix B: Further details pertaining the application to Lombardy region data

In this appendix, we include additional information related to the application of the proposed model for profiling COVID-19 Heart Failure patients (Section 4 of the main paper).

### Description of clinical variables involved in the analysis

In the following, we provide a brief description of the respiratory diseases considered in the analysis and of the Multi-source Comorbidity Score (MCS).

The Chronic obstructive pulmonary disease (COPD) entails a persistent constriction or blockage of the air passages resulting in an ongoing reduction in the airflow rate during exhalation. Pneumonia (PNA) refers to the abrupt inflammation of the lungs brought about by an infection. Respiratory failure (RF) manifests when the oxygen level in the blood becomes critically low or when there is a dangerous elevation in the blood's carbon dioxide level. Bronchitis (BRH) entails an inflammation of the primary air passages of the lungs, known as the bronchi, typically triggered by infection, leading to irritation and inflammation.

For what concerns the computation of the MCS, the scores associated to single diseases are presented in Web Table 7. The overall score for each patient is computed by summing the scores associated with all the diseases they have been diagnosed with. The diseases attributed to each patient are identified by analyzing their medical history pertaining the previous 5 years, a process facilitated through the utilization of ICD-9-CM codes. These codes serve as a means to extract relevant medical information and establish connections to specific diseases.

### ICD-9-CM codes for respiratory diseases

- **Chronic obstructive pulmonary disease (COPD):** 491-496, 491.2, 492.0, 492.8, 494.0-494.1
- **Pneumonia (PNA):** 480-486, 507, 011.6, 052.1, 055.1, 073.0, 130.4, 480.0-480.3, 480.8-480.9, 482.1-482.4, 482.8-482.9, 483.1, 483.8, 484.3, 484.5, 487.0, 506.0, 507.0, 507.8, 517.1, 770.0, 00322, 011.61-011.66, 115.05, 115.15, 115.95, 482.30-482.32, 482.40-482.41, 482.49, 482.81, 482.89, V12.61
- **Respiratory failure (RF):** 518.81, 518.83-518.84
- **Bronchitis (BRH):** 466, 490-491, 466.0, 491.0-491.2, 491.8-491.9, 491.20-491.22

## Appendix C: Computation of survival function confidence intervals

In order to compute confidence intervals for the survival function in Section 4.2 of the main paper, we rely on the delta method [Ver Hoef, 2012], which is a technique for approximating the variance of a non-linear function of random variables.

In our case, the survival function depends on the parameters  $\eta_g$  and  $\nu_g$  for each cluster  $g \in \{1, 2, 3\}$ . To apply the delta method, we first compute the partial derivatives of the survival function with respect to these parameters:

$$\frac{\partial S(t)}{\partial \eta_g} = -\frac{1}{\nu_g} \phi \left( \frac{\log(t) - \eta_g}{\nu_g} \right); \quad \frac{\partial S(t)}{\partial \nu_g} = -\frac{\log(t) - \eta_g}{\nu_g^2} \phi \left( \frac{\log(t) - \eta_g}{\nu_g} \right)$$

Table 7: The score associated to each disease to define the Modified Multisource-Comorbidity Score (MCS).

Comorbidity	Score	Comorbidity	Score
Metastatic cancer	18	Kidney dialysis	4
Alcohol abuse	11	Heart failure	4
Non-metastatic cancer	10	Other neurological disorders	3
Tuberculosis	10	Rheumatic diseases	3
Psychosis	8	Brain diseases	3
Liver diseases	8	Anemia	3
Drugs for anxiety	6	Diabetes	2
Weight loss	6	Gout	2
Dementia	6	Epilepsy	2
Drugs for malignancies	5	Ulcer diseases	2
Parkinson's disease	5	Myocardial infarction	1
Lymphoma	5	Drugs for coronary	1
Paralysis	5	Valvular diseases	1
Coagulopathy	5	Arrhythmia	1
Fluid disorders	4	Obesity	1
Kidney diseases	4	Hypothyroidism	1

Next, the variance of the survival function at a specific time  $t$  is calculated using the following formula:

$$\text{Var}(S(t)) = \left( \frac{\partial S(t)}{\partial \eta_g} \right)^2 \text{Var}(\eta_g) + \left( \frac{\partial S(t)}{\partial \nu_g} \right)^2 \text{Var}(\nu_g)$$

where,  $\text{Var}(\eta_g)$  and  $\text{Var}(\nu_g)$  are the squared standard errors of the fitted parameters  $\eta_g$  and  $\nu_g$ , respectively. This formula accounts for the uncertainty in the survival function resulting from the variability in the estimates of  $\eta_g$  and  $\nu_g$ . The standard error of the survival function is then the square root of the variance:

$$\text{SE}(S(t)) = \sqrt{\text{Var}(S(t))}.$$

Finally, we construct the 95% confidence intervals for the survival function at time  $t$  using the standard normal quantile  $Z = 1.96$  for a two-sided confidence level:

$$\text{CI}_{95\%} = S(t) \pm Z \cdot \text{SE}(S(t)).$$

This approach provides a method to quantify the uncertainty around our estimate of the survival function at a given time, based on the estimated parameters and their associated standard errors.

## References

- José Cortiñas Abrahantes, Catherine Legrand, Tomasz Burzykowski, Paul Janssen, Vincent Ducrocq, and Luc Duchateau. Comparison of different estimation procedures for proportional hazards model with random effects. *Computational Statistics and Data Analysis*, 51:3913–3930, 2007. ISSN 01679473. doi: 10.1016/j.csda.2006.03.009.
- Ernest A Adeghate, Nabil Eid, and Jaipaul Singh. Mechanisms of covid-19-induced heart failure: a short review. *Heart failure reviews*, 26:363–369, 2021.
- Peter C. Austin. A tutorial on multilevel survival analysis: Methods, models and applications. *International Statistical Review*, 85:185–203, 8 2017. ISSN 03067734. doi: 10.1111/insr.12214. URL <http://doi.wiley.com/10.1111/insr.12214>.
- Feras Bader, Yosef Manla, Bassam Atallah, and Randall C Starling. Heart failure and covid-19. *Heart failure reviews*, 26(1):1–10, 2021.
- Theodor A. Balan and Hein Putter. A tutorial on frailty models. *Statistical Methods in Medical Research*, 29:3424–3454, 2020. ISSN 14770334. doi: 10.1177/0962280220921889.
- Alida Benfante and Nicola Scichilone. Prioritizing care for severe asthma during sars-cov-2 pandemic. *Pulmonology*, 27(3):189–190, 2021.
- Paolo Berta and Veronica Vinciotti. Multilevel logistic cluster-weighted model for outcome evaluation in health care. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 12:434–443, 10 2019. ISSN 1932-1864. doi: 10.1002/sam.11421. URL <https://onlinelibrary.wiley.com/doi/10.1002/sam.11421>.

- Paolo Berta, Salvatore Ingrassia, Antonio Punzo, and Giorgio Vittadini. Multilevel cluster-weighted models for the evaluation of hospitals. *METRON*, 74:275–292, 12 2016. ISSN 0026-1424. doi: 10.1007/s40300-016-0098-3. URL <http://link.springer.com/10.1007/s40300-016-0098-3>.
- Paolo Berta, Salvatore Ingrassia, Giorgio Vittadini, and Daniele Spinelli. Latent heterogeneity in COVID-19 hospitalisations: a cluster-weighted approach to analyse mortality. *Australian & New Zealand Journal of Statistics*, 66(1):1–20, mar 2024. ISSN 1369-1473. doi: 10.1111/anzs.12407. URL <https://onlinelibrary.wiley.com/doi/10.1111/anzs.12407>.
- Laurent Bordes and Didier Chauveau. Stochastic em algorithms for parametric and semiparametric mixture models for right-censored lifetime data. *Computational Statistics*, 31:1513–1538, 12 2016. ISSN 0943-4062. doi: 10.1007/s00180-016-0661-7. URL <http://link.springer.com/10.1007/s00180-016-0661-7>.
- Charles Bouveyron, Gilles Celeux, T Brendan Murphy, and Adrian E Raftery. *Model-Based Clustering and Classification for Data Science*, volume 50. Cambridge University Press, 7 2019. ISBN 9781108644181. doi: 10.1017/9781108644181. URL <https://www.cambridge.org/core/product/identifier/9781108644181/type/book>.
- Peter Bryant and John A Williamson. Asymptotic Behaviour of Classification Maximum Likelihood Estimates. *Biometrika*, 65(2):273, aug 1978. ISSN 00063444. doi: 10.2307/2335205. URL <https://www.jstor.org/stable/2335205?origin=crossref>.
- Luca Caldera, Chiara Masci, Andrea Cappozzo, Marco Forlani, Barbara Antonelli, Olivia Leoni, and Francesca Ieva. Uncovering mortality patterns and hospital effects in COVID-19 heart failure patients: a novel multilevel logistic cluster-weighted modeling approach. *Biometrics*, 81(2), apr 2025. ISSN 0006-341X. doi: 10.1093/biomtc/ujaf046. URL <https://academic.oup.com/biometrics/article/doi/10.1093/biomtc/ujaf046/8121043>.
- Gilles Celeux. The sem algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational statistics quarterly*, 2:73–82, 1985.
- Gilles Celeux and Gérard Govaert. A classification em algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14:315–332, 10 1992. ISSN 01679473. doi: 10.1016/0167-9473(92)90042-E. URL <https://www.sciencedirect.com/science/article/pii/016794739290042E>.
- Didier Chauveau. A stochastic em algorithm for mixtures with censored data. *Journal of statistical planning and inference*, 46(1):1–25, 1995.
- Giovanni Corrao, Federico Rea, Mirko Di Martino, Rossana De Palma, Salvatore Scondotto, Danilo Fusco, Adele Lallo, Laura Maria Beatrice Belotti, Mauro Ferrante, Sebastiano Pollina Addario, et al. Developing and validating a novel multisource comorbidity score from administrative data: a large population-based cohort study from italy. *BMJ open*, 7(12):e019503, 2017.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39:1–22, 9 1977. ISSN 00359246. doi: 10.1111/j.2517-6161.1977.tb01600.x. URL <http://www.jstor.org/stable/2984875> <https://onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1977.tb01600.x>.
- Rong-Hui Du, Li-Rong Liang, Cheng-Qing Yang, Wen Wang, Tan-Ze Cao, Ming Li, Guang-Yun Guo, Juan Du, Chun-Lan Zheng, Qi Zhu, et al. Predictors of mortality for patients with covid-19 pneumonia caused by sars-cov-2: a prospective cohort study. *European Respiratory Journal*, 55(5), 2020.
- Luc Duchateau and Paul Janssen. *The frailty model*. Springer, 2008.
- Luis A. García-Escudero, Alfonso Gordaliza, Carlos Matrán, and Agustín Mayo-Iscar. A general trimming approach to robust cluster analysis. *The Annals of Statistics*, 36:1324–1345, 6 2008. ISSN 0090-5364. doi: 10.1214/07-AOS515. URL <http://projecteuclid.org/euclid-aos/1211819566>.
- Neil Gershenfeld. Nonlinear inference and cluster-weighted modeling. *Annals of the New York Academy of Sciences*, 808:18–24, 1 1997. ISSN 0077-8923. doi: 10.1111/j.1749-6632.1997.tb51651.x. URL <http://doi.wiley.com/10.1111/j.1749-6632.1997.tb51651.x>.
- Davide Giustivi, Francesco Bottazzini, and Mirko Belliato. Respiratory monitoring at bedside in covid-19 patients. *Journal of clinical medicine*, 10(21):4943, 2021.
- Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.
- Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 12 1985. ISSN 0176-4268. doi: 10.1007/BF01908075. URL <http://link.springer.com/10.1007/BF01908075>.

- Salvatore Ingrassia, Simona C. Minotti, and Giorgio Vittadini. Local statistical modeling via a cluster-weighted approach with elliptical distributions. *Journal of Classification*, 29:363–401, 10 2012. ISSN 0176-4268. doi: 10.1007/s00357-012-9114-3. URL <http://link.springer.com/10.1007/s00357-012-9114-3>.
- Salvatore Ingrassia, Simona C. Minotti, and Antonio Punzo. Model-based clustering via linear cluster-weighted models. *Computational Statistics and Data Analysis*, 71:159–182, 3 2014. ISSN 01679473. doi: 10.1016/j.csda.2013.02.012. URL <http://dx.doi.org/10.1016/j.csda.2013.02.012><https://linkinghub.elsevier.com/retrieve/pii/S016794731300056X>.
- Salvatore Ingrassia, Antonio Punzo, Giorgio Vittadini, and Simona C. Minotti. The generalized linear mixed cluster-weighted model. *Journal of Classification*, 32:85–113, 4 2015. ISSN 0176-4268. doi: 10.1007/s00357-015-9175-1. URL <http://link.springer.com/10.1007/s00357-015-9175-1>.
- Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
- John P Klein and Melvin L Moeschberger. *Survival analysis: techniques for censored and truncated data*. Springer Science & Business Media, 2006.
- John V. Monaco, Malka Gorfine, and Li Hsu. General semiparametric shared frailty model: Estimation and simulation with frailtysurv. *Journal of Statistical Software*, 86, 2018. ISSN 1548-7660. doi: 10.18637/jss.v086.i04. URL <http://www.jstatsoft.org/v86/i04/>.
- Marco Munda, Federico Rotolo, and Catherine Legrand. parfm : Parametric frailty models in r. *Journal of Statistical Software*, 51, 2012. ISSN 1548-7660. doi: 10.18637/jss.v051.i11. URL <http://www.jstatsoft.org/v51/i11/>.
- Søren Fedor Nielsen. The Stochastic EM Algorithm: Estimation and Asymptotic Results. *Bernoulli*, 6(3):457, jun 2000. ISSN 13507265. doi: 10.2307/3318671. URL <https://www.jstor.org/stable/3318671?origin=crossref>.
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL <https://www.R-project.org/>.
- Juan R Rey, Juan Caro-Codón, Sandra O Rosillo, Ángel M Iniesta, Sergio Castrejón-Castrejón, Irene Marco-Clement, Lorena Martín-Polo, Carlos Merino-Argos, Laura Rodríguez-Sotelo, Jose M García-Veas, et al. Heart failure in covid-19 patients: prevalence, incidence and prognostic implications. *European journal of heart failure*, 22(12): 2205–2215, 2020.
- Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 3 1978. ISSN 0090-5364. doi: 10.1214/aos/1176344136. URL <http://projecteuclid.org/euclid-aos/1176344136>.
- Angelica Tiotiu, Heriberto Chong Neto, Andras Bikov, Krzysztof Kowal, Paschalis Steiropoulos, Marina Labor, Ivan Cherrez-Ojeda, Hector Badellino, Alexander Emelyanov, Rocio Garcia, et al. Impact of the covid-19 pandemic on the management of chronic noninfectious respiratory diseases. *Expert review of respiratory medicine*, 15(8): 1035–1048, 2021.
- James W Vaupel, Kenneth G Manton, and Eric Stallard. The impact of heterogeneity in individual frailty on the dynamics of mortality. *Demography*, 16(3):439–454, 1979.
- Jay M Ver Hoef. Who invented the delta method? *The American Statistician*, 66(2):124–127, 2012.
- Fei Zhou, Ting Yu, Ronghui Du, Guohui Fan, Ying Liu, Zhibo Liu, Jie Xiang, Yeming Wang, Bin Song, Xiaoying Gu, et al. Clinical course and risk factors for mortality of adult inpatients with covid-19 in wuhan, china: a retrospective cohort study. *The lancet*, 395(10229):1054–1062, 2020.

## Florence Forbes

### *Material list:*

Geoffroy Oudoumanessah, Thomas Coudert, Carole Lartizien, Michel Dojat, Thomas Christen, Florence Forbes (2024) Scalable magnetic resonance fingerprinting: Incremental inference of high dimensional elliptical mixtures from large data volumes. arXiv 2412.10173

Geoffroy Oudoumanessah, Thomas Coudert, Luc Meyer, Aurelien Delphin, Thomas Christen, Michel Dojat, Carole Lartizien, Florence Forbes (2025) Cluster Globally, Reduce Locally: Scalable Efficient Dictionary Compression for Magnetic Resonance Fingerprinting. IEEE 22nd International Symposium on Biomedical Imaging (ISBI), Houston, TX, USA, 2025, pp. 1-5, doi: 10.1109/ISBI60581.2025.10981146.

H D Nguyen and F Forbes and G J McLachlan (2019) Mini-batch learning of exponential family finite mixture models. arXiv 1902.03335.

Dan Ma, Vikas Gulani, Nicole Seiberlich, Kecheng Liu, Jeffrey L. Sunshine, Jeffrey L. Duerk and Mark A. Griswold (2013) Magnetic resonance fingerprinting. Nature 495, 187–192.

# SCALABLE MAGNETIC RESONANCE FINGERPRINTING: INCREMENTAL INFERENCE OF HIGH DIMENSIONAL ELLIPTICAL MIXTURES FROM LARGE DATA VOLUMES

BY GEOFFROY OUDOUMANESSAH<sup>1,3,2,a</sup> , THOMAS COUDERT<sup>3,d</sup>   
CAROLE LARTIZIEN<sup>2,c</sup>  MICHEL DOJAT<sup>1,3,e</sup>   
THOMAS CHRISTEN<sup>3,f</sup>  AND FLORENCE FORBES<sup>1,b</sup> 

<sup>1</sup>Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France, <sup>a</sup>[geoffroy.oudoumanessah@inria.fr](mailto:geoffroy.oudoumanessah@inria.fr);  
<sup>b</sup>[florence.forbes@inria.fr](mailto:florence.forbes@inria.fr)

<sup>2</sup>Univ. Lyon, CNRS, Inserm, INSA Lyon, UCBL, CREATIS, UMR5220, U1294, F-69621, Villeurbanne, France,  
<sup>c</sup>[carole.lartizien@creatis.insa-lyon.fr](mailto:carole.lartizien@creatis.insa-lyon.fr)

<sup>3</sup>Univ. Grenoble Alpes, Inserm U1216, CHU Grenoble Alpes, Grenoble Institut des Neurosciences, 38000 Grenoble, France,  
<sup>d</sup>[thomas.coudert@inserm.fr](mailto:thomas.coudert@inserm.fr); <sup>e</sup>[michel.dojat@inserm.fr](mailto:michel.dojat@inserm.fr); <sup>f</sup>[thomas.christen@univ-grenoble-alpes.fr](mailto:thomas.christen@univ-grenoble-alpes.fr)

Magnetic Resonance Fingerprinting (MRF) is an emerging technology with the potential to revolutionize radiology and medical diagnostics. In comparison to traditional magnetic resonance imaging (MRI), MRF enables the rapid, simultaneous, non-invasive acquisition and reconstruction of multiple tissue parameters, paving the way for novel diagnostic techniques. In the original *matching* approach, reconstruction is based on the search for the best matches between *in vivo* acquired signals and a dictionary of high-dimensional simulated signals (fingerprints) with known tissue properties. A critical and limiting challenge is that the size of the simulated dictionary increases exponentially with the number of parameters, leading to an extremely costly subsequent matching. In this work, we propose to address this scalability issue by considering probabilistic mixtures of high-dimensional elliptical distributions, to learn more efficient dictionary representations. Mixture components are modelled as flexible ellipitic shapes in low dimensional subspaces. They are exploited to cluster similar signals and reduce their dimension locally cluster-wise to limit information loss. To estimate such a mixture model, we provide a new incremental algorithm capable of handling large numbers of signals, allowing us to go far beyond the hardware limitations encountered by standard implementations. We demonstrate, on simulated and real data, that our method effectively manages large volumes of MRF data with maintained accuracy. It offers a more efficient solution for accurate tissue characterization and significantly reduces the computational burden, making the clinical application of MRF more practical and accessible.

**1. Introduction.** Traditional Magnetic Resonance (MR) imaging relies on an analytical resolution of dynamical equations using conventional tuning of the MR hardware through sequences of pulses, each characterized by different values of parameters such as the flip angle and repetition time. Standard quantitative MRI (qMRI) methods are based on a single sequence for a single parameter measurement at a time. This leads to high scan times for multi-parametric protocols as each parameter estimate involves one MR sequence. A recent approach named Magnetic Resonance Fingerprinting (MRF, Ma et al. (2013)) has been developed to overcome these limitations. The MRF protocol involves fast undersampled acquisitions with time-varying parameters defining the MRF sequence that produces temporal signal evolutions (named *fingerprints*) in each voxel. In the original proposal, a dictionary

---

*Keywords and phrases:* Dimension reduction, Clustering, Incremental learning, High dimensional mixture models, Elliptical distributions, Expectation Maximization algorithm.

search approach is used to compare the *in vivo* fingerprints with millions of numerical simulations of MR signals for which the associated parameters are known. These millions of simulated signals compose the so-called dictionary. The values of the parameters corresponding to the closest simulated signals or *matches* are then assigned to the associated *in vivo* voxels, allowing the simultaneous reconstruction of multiple quantitative maps (images) from extremely undersampled raw images, using only one single sequence, thus saving considerable acquisition time (Poorman et al., 2020; McGivney et al., 2020). From this, standard relaxometry MRF allows reconstructing parameter maps for relaxation times  $T_1$  and  $T_2$ , over the whole human brain ( $1 \text{ mm}^3$  spatial resolution) in 3 min (Ye et al., 2017; Gu et al., 2018) compared to 30 min for a standard T1/T2 exam. Moreover, the flexibility of the numerical simulations enables correction of system imperfections as well as some patient motions by including them in the model and post-processing pipelines (Bipin Mehta et al., 2019). Thus, MRF could be a game changer for emergency patients who need to complete exams in a few minutes. The power of the MRF approach is not limited to the estimation of relaxation times, in theory, it allows the measurement of any parameter that influences nuclear magnetization (e.g., microvascular networks), and could be added to the simulation model (Wang et al., 2019a; Coudert et al., 2024). However, increasing the number of estimated parameters, even moderately, induces the design of more complex sequences and increased reconstruction times, from hours to days. This limits the clinical application of high-dimensional MRF and necessitates the development of innovative processing methods. Consequently, a significant focus is on improving MRF reconstruction methods, as reviewed by Tippareddy et al. (2021) and Monga et al. (2024). In this work, we propose to focus on reducing the reconstruction times of MRF when more than the main two parameters,  $T_1$ ,  $T_2$ , are involved, including the addition of  $\delta f$  the frequency offset, the sensitivity of the magnetic field  $B_1$ , cerebral blood volume (CBV), and microvascular geometry (e.g., vessel radius denoted as  $R$ ).

MRF reconstruction is first recast as an inverse problem that can be solved using different approaches as recalled in the next section. All approaches make use, at some stage, of a dictionary of simulated pairs (parameters, signal), which represent our knowledge of the link between tissue parameters and MR time series, through a so-called *direct* or *forward* model. The dictionary is then either used to learn an *inversion* operator, from signal to tissue parameters, or to search for the best fits between observed signals and simulated ones. The approaches scalability relies thus greatly on their ability to extract efficiently the information encoded in simulations. Efficiency has different aspects: for search-type methods, the dictionary should not be too big, while for learning-based methods, the dictionary should be informative enough. These potentially opposite requirements call for efficient representations of simulated data. We propose to explore a divide-and-conquer strategy by introducing the framework of High Dimensional Mixtures of Elliptical Distributions (HD-MED). Probabilistic mixtures of high-dimensional elliptical distributions allow us to learn more efficient dictionary representations. Mixture components are modeled as flexible elliptical shapes in low-dimensional subspaces. They are exploited to cluster similar signals and reduce their dimension locally, at the cluster level, to limit information loss. To estimate such a mixture model, we provide a new incremental algorithm capable of handling large numbers of signals, allowing us to go far beyond the hardware limitations encountered by standard implementations. We demonstrate, on simulated and real data, that our method effectively manages large volumes of MRF data with maintained accuracy. It offers a more efficient solution for accurate tissue characterization and significantly reduces the computational burden, making the clinical application of MRF more practical and accessible.

In the rest of the paper, we first recall in Section 2 the two main types of approaches that have been investigated in the literature on MRF reconstruction. We then review related work on efficient dictionary representations and specify our contributions in Section 3. The

mixtures of elliptical distributions are presented in Section 4 with their use for dimension reduction in Section 5. The incremental algorithm proposed for the mixture estimation is presented in Section 6 and illustrated on MRF in Section 7.

**2. MRF reconstruction as an inverse problem.** An inverse problem refers to a situation where one aims to determine the causes of a phenomenon from experimental observations of its effects. In MRF, the goal is to infer from an observed signal a set of tissue characteristics or parameter values that best account for the signal. Such a resolution generally starts with modeling the phenomenon under consideration, which is called the "direct" or "forward problem." It is generally assumed that at least the numerical evaluation of the forward model is available because experts have designed equations that can be solved either analytically or numerically. The most common use of the forward model is via a simulator that allows the creation of a database  $\mathcal{D}_f$ , usually referred to as a dictionary, of  $N$  signals  $\mathbf{y}_1, \dots, \mathbf{y}_N$  with  $\mathbf{y}_i \in \mathcal{Y} \subset \mathbb{R}^M$ , generated (stored or computed on the fly) by running the theoretical (physical) model  $f$  for many different tissue parameter values  $\mathbf{t}_1, \dots, \mathbf{t}_N$  with  $\mathbf{t}_i \in \mathcal{T} \subset \mathbb{R}^L$  and  $\mathbf{y}_i = f(\mathbf{t}_i)$ . The generated tissue parameter values then only partially represent the full space  $\mathcal{T}$  of possible values and correspond to a discrete grid in the full space. In this context, we can distinguish two types of methods, referred to below as optimization and learning approaches.

**2.1. Optimization vs learning (or regression) approaches.** Optimization approaches include the most used method in MRF, namely dictionary matching, often called, in other domains, grid search, look-up-table or k-nearest neighbors. They consist of minimizing over parameters  $\mathbf{t}$  a merit function  $d$  expressing the similarity between the observed signal  $\mathbf{y}_{obs}$  and simulated signal  $f(\mathbf{t})$ ,

$$(1) \quad \hat{\mathbf{t}} \in \arg \min_{\mathbf{t} \in \mathcal{T}} d(\mathbf{y}_{obs}, f, \mathbf{t}).$$

Typically,  $d(\mathbf{y}_{obs}, f, \mathbf{t}) = d(\mathbf{y}_{obs}, f(\mathbf{t})) = \|\mathbf{y}_{obs} - f(\mathbf{t})\|^2$ . Solutions are searched in the full  $\mathcal{T}$  space but solutions could also be penalized as done in grid search methods. Indeed, in grid search, the previous full search is replaced by a simpler look-up or matching operation making use of the database  $\mathcal{D}_f$ , often created beforehand offline. The search space is significantly reduced from a continuous space  $\mathcal{T}$  to a discrete and finite  $\mathcal{D}_f$ . The speed gain is significant in comparison to traditional optimization methods as retrieving a value from memory is often faster than undergoing an expensive usually iterative computation. Their disadvantage is the instability of solutions. Many questions remain on how to choose the merit function, how many  $\mathbf{y}_n$  in the look up table have to be kept to estimate parameters, how to choose the look-up table, etc. When the number of parameters is small, grid search is suitable and can provide very good predictions. However, for even moderate numbers of parameters, the required number of elements in the dictionary renders grid search either intractable or inaccurate. The technique is not amortized, for each new  $\mathbf{y}_{obs}$ , we have to compute the matching score  $d(\mathbf{y}_{obs}, \mathbf{y}_n)$  for all  $\mathbf{y}_n$  in the dictionary. When the dimension of  $\mathbf{t}$  increases, the dictionary (and  $N$ ) has to be larger too for better accuracy. The computation of  $N$  matching scores can become time consuming.

Regression or learning methods are more efficient in that sense and usually have better amortization properties. In contrast to the previous ones, this category of methods has the advantage of adapting easily to provide tractable solutions in the case of massive inversions of high-dimensional data. The main principle is to transfer the computational cost and time from individual pointwise predictions to the learning of a global inverse operator from  $\mathcal{D}_f$ . The advantage is that once the operator is learned, it can be used, at negligible cost, for very large numbers of new signals. Then, the nature of the inverse operator needs to be specified.

Traditional learning or regression methods are not specifically designed for high dimensional data but there has been a large literature covering this case, see e.g. [Giraud \(2014\)](#) for a review. In MRF, a popular approach from [Cohen, Zhu and Rosen \(2018\)](#), uses a four-layer fully connected network (DRONE) to learn the dictionary signals for reconstructing  $T_1$  and  $T_2$  parameters. Even though DRONE provides good results, this perceptron-based model loses the temporal coherence of the signal. Recently, [Cabini et al. \(2024\)](#) proposed a Recurrent Neural Network (RNN) with long-short term memory (LSTM) blocks, which yields better reconstruction results for  $T_1$  and  $T_2$  with more robustness towards noisy acquisitions. However, when more parameters need to be estimated, such as vessel oxygenation or radius ([Christen, Bolar and Zaharchuk, 2013; Christen et al., 2014](#)), the dimensionality of the signals dramatically increases. [Barrier et al. \(2024\)](#) demonstrated that a simple RNN is prone to catastrophic forgetting, where the RNN well estimates the beginning of the signals but learns the end less effectively. To mitigate this issue, they proposed using bidirectional LSTM (bi-LSTM) blocks within the RNN architecture, ensuring that both the beginning and the end of the signal are efficiently learned by the new bi-LSTM blocks. Despite the good results in reconstructing  $T_1$ ,  $T_2$ , and vascular parameters, bi-LSTM still faces challenges in learning the middle part of the signals.

Generally, the dictionary is seen as a collection of simulated signals with no particular spatial correlation. Another way to simulate signals is to acquire real parametric maps from different subjects and then simulate the MRF images. This approach was first proposed by [Soyak et al. \(2021\)](#) and later improved by [Gu et al. \(2024\)](#). Both studies utilize a UNet ([Ronneberger, Fischer and Brox, 2015](#)) to infer  $T_1$  and  $T_2$  directly using the entire MRF image as an input to the network, preserving valuable spatial information. Given the high dimensionality of the signals, the authors proposed adding attention layers ([Vaswani et al., 2017](#)) to focus on the most important dimensions of the signals. More recently, [Li and Hu \(2024\)](#) highlighted the limitations of using CNN, which have a restricted receptive field and capture spatial information only locally. To overcome these limitations, the authors proposed using a Local-Global vision Transformer to capture spatial information globally as well. However, capturing spatial information has a cost. Indeed, one needs to acquire from a large group of subjects multiple  $T_1$  and  $T_2$  maps, which takes about 30 min for a complete exam, making the data acquisition process much more costly than using more conventional dictionaries. Additionally, since  $T_1$  and  $T_2$  need to be acquired at two different times, this method introduces more errors from the registration of the acquired maps.

Finally, [Boux et al. \(2021\)](#) also proposed to use GLLiM ([Deleforge, Forbes and Horaud, 2015](#)), a model that casts MRF reconstruction into a Bayesian inverse problem and then solves it using a learning approach. This method takes into account the high-dimensional property of MRF signals by defining the low-dimensional variables as the regressors, which in our case are the tissue parameters. By doing so, they start learning the *low-to-high* regression model from which they can derive the forward model parameters and then the *high-to-low* regression model, from MRF signals to tissue parameters, as desired.

**3. Related work and positioning.** In practice, acquired MRF acquisitions come as 4D matrix, made of a time series of 3D MRI images where each voxel contains the acquired, potentially long, fingerprint signal. In this work, we focus on designing efficient representations of large highly precise grids of simulated signals counterparts.

**3.1. Parcimonious representations of dictionaries.** The curse of dimensionality goes with what is often called the *bless of dimensionality*, which refers to the fact that in high dimensional data sets, useful information actually leaves in much smaller dimensional parts of the data space. One approach to the MRF reconstruction problem is then to reduce the

dimension of the dictionary beforehand to reduce the matching or learning cost. The first to propose such an approach for efficient matching were [McGivney et al. \(2014\)](#), who applied Singular Value Decomposition (SVD) to the dictionary of signals. Once the decomposition is learned, one can project any new acquisition into the SVD low-dimensional subspace. However, when the size of the dictionary increases, computing the SVD becomes costly as it requires loading the complete dictionary into fast-access memory (*e.g.* the RAM). To address this, [Yang et al. \(2018\)](#) proposed using randomized SVD ([Halko, Martinsson and Tropp, 2011](#)). [Golbabae et al. \(2019\)](#) also suggested applying SVD to the dictionary before training a neural network. Other methods proposed a non-Euclidian analysis of the dictionary space projecting the signals into a lower-dimensional manifold ([Li and Hu, 2023](#)). Reducing the dimension of high-dimensional dictionaries assumes that most of the information in the signals can be captured and represented in a much lower-dimensional subspace. Classical techniques include principal component analysis (PCA, [Jolliffe and Cadima \(2016\)](#)), probabilistic principal component analysis (PPCA, [Tipping and Bishop \(1999a\)](#)), factor analyzers (FA), sparse models ([Zou, Hastie and Tibshirani, 2006](#); [d'Aspremont et al., 2007](#); [Archambeau and Bach, 2008](#)), and newer methods such as diffusion maps ([Coifman and Lafon, 2006](#)). More flexible approaches are based on mixtures of the previous ones, such as mixtures of factor analyzers (MFA,  [McLachlan, Peel and Bean \(2003\)](#)) introduced by [Ghahramani and Hinton \(1997\)](#) and extended by [McLachlan, Peel and Bean \(2003\)](#); [Baek and McLachlan \(2011\)](#), and mixtures of PPCA (MPPCA, [Tipping and Bishop \(1999b\)](#); [Xu, Balzano and Fessler \(2023\)](#)) with recent generalizations ([Hong et al., 2023](#); [Xu, Balzano and Fessler, 2023](#)). Another mixture approach is called HDDC in [Bouveyron, Girard and Schmid \(2007\)](#) and HD-GMM in [Bouveyron and Brunet-Saumard \(2014\)](#) for High Dimensional Gaussian Mixture Models, which encompass many forms of MFA and MPPCA and generalize them. In particular HD-GMM can be used to obtain multiple low-dimensional subspaces of different dimensions. For a review on high-dimensional clustering via mixtures, see [Bouveyron and Brunet-Saumard \(2014\)](#). Such a divide-and-conquer strategy has been used in MRF by first creating a global clustering of the signals and then reducing the dimension. This method was applied by [Cauley et al. \(2015\)](#), who performed K-way partitioning of the dictionary before using multiple cluster-associated PCA to reduce the dimension in each part. When matching a new signal, one first determines which cluster it belongs to and then applies the projection learned by the associated PCA model. More recently, [Ullah et al. \(2023\)](#) proposed a simpler approach: applying a clustering algorithm to the dictionary before utilizing GPUs for the *dictionary matching* enabling fast matching without the need for dimensionality reduction. However this method still faces issues when the number of parameters to estimate is larger than 3.

Most of these methods are designed for batch data and are thus sensitive to hardware limits such as memory, restricting the amount of data they can process. For instance, some dictionaries exceed terabyte sizes. A simple solution is to down-sample data sets before processing, potentially losing useful information. Another approach is to design incremental, also referred to as online, variants that handle data sequentially in smaller groups. A number of incremental approaches exist for dimension reduction techniques, see the recent SHASTA-PCA ([Gilman et al., 2023](#)) and references therein, or [Balzano, Chi and Lu \(2018\)](#) for a review. To our knowledge, much fewer solutions exist for mixtures. Estimation of such models is generally based on maximum likelihood estimation via the Expectation-Maximization (EM) algorithm ([McLachlan and Krishnan \(2008\)](#)). A preliminary attempt for an incremental MP-PCA can be found in [Bellas et al. \(2013\)](#) but it is based on heuristic approximations of the EM steps.

**3.2. Contribution.** Herein, we propose to explore this divide-and-conquer strategy by introducing the framework of High Dimensional Mixtures of Elliptical Distributions (HD-MED) to get over the hump of estimating more than 2 parameters, *e.g.*  $T_1, T_2, \delta f, B_1, \text{CBV}$ ,

*R.* Considering the reconstruction of 6 parameters makes the associated dictionaries both large in size (an order of 13 terabyte of signals to considerably represent the tissues heterogeneity) and dimension, making our proposal twofold. Building on HD-MED, a generalization of HD-GMM, we show how they can be used to simultaneously compress and cluster large-scale high-dimensional MRF dictionaries, and more generally any dataset. We derive a new incremental algorithm, based on a principled EM framework, to learn such a model from potentially very large data volumes. We demonstrate the effectiveness of our approach on MRF reconstruction, showing results comparable to the high-dimensional dictionary matching referred to as *full-matching* in the next sections. This approach allows us to exceed the resolution, size used in current implementations and to reconstruct a larger number of MR parameter maps with improved accuracy, thereby advancing the clinical feasibility of MRF.

**4. High Dimensional Mixtures of Elliptical Distributions.** Elliptical distributions represent a family of distributions that contains Gaussian distributions but also heavy-tailed distributions, such as the Student distributions, that are often used as a more robust alternative to the Gaussian family. In this paper, we consider a sub-class of elliptical distributions, which can be expressed as infinite mixtures of Gaussian distributions.

**4.1. Gaussian scale mixture (GSM) distributions.** Scale mixtures of multivariate Gaussian distributions are an important subclass of elliptical distributions whose definition is recalled below. Scale mixtures of Gaussians share good properties with Gaussian distributions. They are tractable, lead to tractable inference procedures and provide more robust results, in contrast to Gaussian distributions that usually suffer from sensitivity to outliers.

**DEFINITION 4.1** (Elliptical Distributions (ED)). A continuous random vector  $\mathbf{Y} \in \mathbb{R}^M$  follows a multivariate elliptical symmetric distribution if its probability density function (pdf)  $p(\mathbf{y})$  is of the following form (see [Cambanis, Huang and Simons \(1981\)](#) or [Kelker \(1970\)](#)),

$$(2) \quad p(\mathbf{y}) = C_{p,g} |\Sigma|^{-1/2} g\left((\mathbf{y} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{y} - \boldsymbol{\mu})\right),$$

where  $\Sigma \in \mathbb{R}^{M \times M}$  is the scale matrix with determinant  $|\Sigma|$ ,  $\boldsymbol{\mu} \in \mathbb{R}^M$  is the location or mean vector,  $C_{p,g}$  is a normalizing constant such that the pdf  $p(\mathbf{y})$  integrates to one. The non-negative function  $g$  is called the density generator and determines the shape of the pdf. When  $\mathbf{Y}$  has density (2), we write  $\mathbf{Y} \sim \mathcal{E}_M(\boldsymbol{\mu}, \Sigma, g)$ .

Note that the scale matrix  $\Sigma$  is not necessarily equal to the covariance matrix,  $\Sigma$  is proportional to the covariance matrix if the latter exists. The pdf of the multivariate normal distribution is a special case of ED with  $g(u) = \exp(-u^2/2)$ . Another member of the elliptical family is the multivariate Student distribution. This distribution is well-studied in the literature ([Kotz and Nadarajah, 2004](#)) and admits a useful representation as a Gaussian scale mixture. Denoting  $\mathcal{N}_M(\mathbf{y}; \boldsymbol{\mu}, \Sigma)$  a  $M$ -variate Gaussian distribution with mean  $\boldsymbol{\mu}$  and covariance matrix  $\Sigma$ , a Gaussian scale mixture distribution is a distribution of the following form.

**DEFINITION 4.2** (Gaussian scale mixture distributions (GSM)). If  $\boldsymbol{\mu}$  is a  $M$ -dimensional vector,  $\Sigma$  is a  $M \times M$  positive definite symmetric matrix and  $f$  is a pdf of a univariate positive variable  $W \in \mathbb{R}^+$ , then the  $M$ -dimensional density given by

$$(3) \quad p(\mathbf{y}) = \int_{\mathbb{R}^+} \mathcal{N}_M\left(\mathbf{y}; \boldsymbol{\mu}, \frac{\Sigma}{w}\right) f(w) dw$$

is said to be an infinite mixture of scaled Gaussians or Gaussian scale mixture (GSM) with mixing distribution function  $f$ . If vector  $\mathbf{Y}$  has density (3), we still write  $\mathbf{Y} \sim \mathcal{E}_M(\boldsymbol{\mu}, \Sigma, f)$  and refer to  $W$  as the mixing variable.

In practice, we will consider mixing distribution  $f_{\theta}$  that depends on some parameter  $\theta$  and also write  $\mathbf{Y} \sim \mathcal{E}_M(\boldsymbol{\mu}, \Sigma, \theta)$ . As already mentioned, famous GSM distributions include, the multivariate Student distribution (when  $f_{\theta}$  is the pdf of a Chi2 variable), the Pearson type VII distribution (when  $f_{\theta}$  is a gamma distribution) and the generalized Gaussian (when a power of  $W$  follows the gamma distribution). It is straightforward to see that GSM are elliptical distributions. However, not all elliptical distributions can be reduced to scale mixtures. For the previous reason, characterization and a way to represent elliptical distributions as GSM are very valuable. In [Gómez-Sánchez-Manzano, Gómez-Villegas and Marín \(2006\)](#), conditions are given under which elliptical distributions are GSMS. The issue of finding the corresponding mixing distribution is also addressed. An illustration of these results for generalized Gaussian distributions is given by [Gomez, Gomez-Villegas and Marin \(2008\)](#).

**4.2. Mixtures of High Dimensional GSM.** Consider a data set of  $N$  independent observations  $\{\mathbf{y}_1, \dots, \mathbf{y}_N\} \in \mathbb{R}^M$  assumed to be *i.i.d.* realizations of a random vector  $\mathbf{Y} \in \mathbb{R}^M$ . In addition, the data set is assumed to be made of  $K$  groups to be discovered. In this work, we consider finite mixtures of GSM, assuming that the underlying subsets are distributed according to different GSM in different proportions. In addition, to handle potentially high-dimensional observations with GSM, we propose a specific parameterization of the scale matrices.

**DEFINITION 4.3** (Finite mixture model).  $\mathbf{Y}$  follows a finite mixture model if its pdf writes as

$$(4) \quad p(\mathbf{y}) = \sum_{k=1}^K \pi_k f_k(\mathbf{y}),$$

where  $\pi_k \in [0, 1]$  are the mixing weights that sum to one, and  $f_k$  is the pdf of the  $k^{\text{th}}$  mixture component.

To simplify the notation, a finite mixture model where each pdf  $f_k$  is the pdf of a GSM distribution (3),  $\mathcal{E}(\boldsymbol{\mu}_k, \Sigma_k, \theta_k)$ , is referred to as a mixture of ED (MED). The number of parameters in MED grows quadratically with the dimension  $M$  due to the scale matrices  $\Sigma_k$ . For large  $M$ , this can be problematic for the mixture estimation from a data set. In the Gaussian mixture case, to reduce the number of parameters, [Bouveyron, Girard and Schmid \(2007\)](#) proposed a family of parsimonious Gaussian mixture models, using the eigenvalues decomposition of the covariance matrices. We extend this idea to MED by reparametrizing the scale matrices  $\Sigma_k$  as follows,

$$(5) \quad \Sigma_k = \mathbf{D}_k \mathbf{A}_k \mathbf{D}_k^T,$$

where  $\mathbf{D}_k$  is a  $M \times M$  orthogonal matrix which contains the eigenvectors of  $\Sigma_k$  and  $\mathbf{A}_k$  is a  $M \times M$  diagonal matrix that contains the associated eigenvalues in decreasing order. The key idea introduced by [Bouveyron, Girard and Schmid \(2007\)](#) is to consider that each cluster lies in a low-dimensional subspace of dimension  $d_k < M$ , which can be expressed by assuming that

$$(6) \quad \mathbf{A}_k = \text{diag}(a_{k1}, \dots, a_{kd_k}, b_k, \dots, b_k),$$

where  $a_{k1}, \dots, a_{kd_k}$  are the  $d_k$  largest eigenvalues of  $\Sigma_k$  and  $b_k$  is a small negligible value. The  $d_k$  eigenvectors associated to the first  $d_k$  eigenvalues  $\{a_{k1}, \dots, a_{kd_k}\}$  define a cluster-specific subspace  $\mathbb{E}_k$ , which captures the main cluster shape. The orthogonal subspace is denoted by  $\mathbb{E}_k^\perp$ . Let  $\tilde{\mathbf{D}}_k$  consists of the  $d_k$  first columns of  $\mathbf{D}_k$  supplemented by  $(M - d_k)$

zero columns and  $\bar{\mathbf{D}}_k = (\mathbf{D}_k - \tilde{\mathbf{D}}_k)$ . It follows that  $P_k(\mathbf{y}) = \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_k^T (\mathbf{y} - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k$  and  $P_k^\perp(\mathbf{y}) = \bar{\mathbf{D}}_k \bar{\mathbf{D}}_k^T (\mathbf{y} - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k$  are the projections of  $\mathbf{y}$  on  $\mathbb{E}_k$  and  $\mathbb{E}_k^\perp$  respectively. This parameterization allows to handle high dimensional data in a computationally efficient way. For instance the quadratic form of the Mahalanobis distance, appearing in the generator  $g$  in (2), writes

$$\begin{aligned} (\mathbf{y} - \boldsymbol{\mu}_k) \mathbf{D}_k \mathbf{A}_k^{-1} \mathbf{D}_k^T (\mathbf{y} - \boldsymbol{\mu}_k) &= (\mathbf{y} - \boldsymbol{\mu}_k)^T \tilde{\mathbf{D}}_k \mathbf{A}_k^{-1} \tilde{\mathbf{D}}_k^T (\mathbf{y} - \boldsymbol{\mu}_k) \\ &\quad + (\mathbf{y} - \boldsymbol{\mu}_k)^T \bar{\mathbf{D}}_k \mathbf{A}_k^{-1} \bar{\mathbf{D}}_k^T (\mathbf{y} - \boldsymbol{\mu}_k) \\ (7) \qquad \qquad \qquad &= \|\boldsymbol{\mu}_k - P_k(\mathbf{y})\|_{\tilde{\Sigma}_k^{-1}}^2 + \frac{1}{b_k} \|\mathbf{y} - P_k(\mathbf{y})\|^2, \end{aligned}$$

where  $\|\cdot\|_{\tilde{\Sigma}_k^{-1}}^2$  is the norm defined by  $\|\mathbf{y}\|_{\tilde{\Sigma}_k^{-1}}^2 = \mathbf{y}^T \tilde{\Sigma}_k^{-1} \mathbf{y}$  with  $\tilde{\Sigma}_k^{-1} = \tilde{\mathbf{D}}_k \mathbf{A}_k^{-1} \tilde{\mathbf{D}}_k^T$ . Equation (7) uses the definitions of  $P_k$  and  $P_k^\perp$  and  $\|\boldsymbol{\mu}_k - P_k^\perp(\mathbf{y})\|^2 = \|\mathbf{y} - P_k(\mathbf{y})\|^2$ . The gain comes from the fact that (7) does not depend on  $P_k^\perp$  and thus does not require the computation of the  $(M - d_k)$  latest columns of  $\mathbf{D}_k$ , the eigenvectors associated to the smallest eigenvalues. Similarly, determinants can be efficiently computed as  $\log(|\Sigma_k|) = (\sum_{m=1}^{d_k} \log(a_{km})) + (M - d)\log(b_k)$ . This efficient parameterization, that now only depends on matrix  $\tilde{\mathbf{D}}_k$  and not on the complete matrix  $\mathbf{D}_k$ , is indicated by denoting the corresponding ED as  $\mathcal{HE}_{Md_k}(\boldsymbol{\mu}_k, \tilde{\mathbf{D}}_k^*, \mathbf{a}_k, b_k, \boldsymbol{\theta}_k)$ , where  $\tilde{\mathbf{D}}_k^*$  the matrix  $\tilde{\mathbf{D}}_k$  with the last zeros  $M - d_k$  columns omitted. We then refer to a MED coupled with this parameterization as a high-dimensional MED (HD-MED).

**DEFINITION 4.4 (HD-MED).** A random vector  $\mathbf{Y} \in \mathbb{R}^M$  follows a HD-MED distribution if for all  $k = 1 : K$ , the pdf of the  $k^{\text{th}}$  mixture component  $f_k$  is an ED  $\mathcal{HE}_{Md_k}(\boldsymbol{\mu}_k, \tilde{\mathbf{D}}_k^*, \mathbf{a}_k, b_k, \boldsymbol{\theta}_k)$  with reparameterization given by (5) and (6). We denote

$$\mathbf{Y} \sim \mathcal{MHE}_{Md}((\pi_k, \boldsymbol{\mu}_k, \tilde{\mathbf{D}}_k^*, \mathbf{a}_k, b_k, \boldsymbol{\theta}_k)_{k=1}^K).$$

With  $\mathbf{d} = (d_1, \dots, d_k)$ ,  $\mathbf{a}_k = (a_{k1}, \dots, a_{kd_k})$ .

## 5. Dimension reduction with HD-MED.

**5.1. Latent variable dimension reduction.** Standard PCA is defined without referring to a probabilistic model. Given a set of observations in  $\mathbb{R}^M$ , their  $M \times M$  empirical covariance matrix is decomposed into eigenvalues and eigenvectors and a number  $d \ll M$  of them are retained. For any observation  $\mathbf{y} \in \mathbb{R}^M$ , a lower dimensional representation can then be obtained by considering its projection to a lower dimensional subspace  $\hat{\mathbf{y}} = \hat{\Sigma}_d^T \mathbf{y}$  where  $\hat{\Sigma}_d$  is the matrix containing the  $d$  first eigenvectors of the empirical covariance matrix. If needed, its reconstruction in  $\mathbb{R}^M$ , optimal in the sense of the squared reconstruction error, can be obtained with  $\tilde{\mathbf{y}} = \hat{\Sigma}_d \hat{\mathbf{y}}$ . Alternatively, if  $\mathbf{y}$  is assumed to be a realization of a random vector  $\mathbf{Y} \sim \mathcal{HE}_{Md}(\boldsymbol{\mu}, \tilde{\mathbf{D}}^*, \mathbf{a}, b, \boldsymbol{\theta})$ , a low dimensional representation of  $\mathbf{y}$  can be justified using the following latent variable model representation of  $\mathbf{Y}$ .

**PROPOSITION 5.1 (HD-ED latent variable model).** Let  $d \leq M - 1$ ,  $\mathbf{Y} \in \mathbb{R}^M$ ,  $\mathbf{X} \in \mathbb{R}^d$ ,  $\mathbf{E} \in \mathbb{R}^M$ ,  $W \in \mathbb{R}^+$  be random variables,  $\mathbf{V} \in \mathbb{R}^{M \times d}$  a matrix of linearly independent columns,  $\boldsymbol{\mu} \in \mathbb{R}^M$  a vector and  $f_\theta$  the pdf of a positive univariate random variable defined

by some parameter  $\boldsymbol{\theta}$ . Assume that

$$\begin{aligned}\mathbf{Y} &= \mathbf{V}\mathbf{X} + \boldsymbol{\mu} + \mathbf{E} \\ (\mathbf{X}|W=w) &\sim \mathcal{N}(\mathbf{0}_d, w^{-1}\mathbf{I}_d) \\ (\mathbf{E}|W=w) &\sim \mathcal{N}(\mathbf{0}_M, b w^{-1}\mathbf{I}_M) \\ W &\sim f_{\boldsymbol{\theta}}\end{aligned}$$

then,

$$\mathbf{Y} \sim \mathcal{HE}_{Md}(\boldsymbol{\mu}, \tilde{\mathbf{D}}^*, \mathbf{a}, b, \boldsymbol{\theta}),$$

with  $\mathbf{D}\mathbf{A}\mathbf{D}^T = b\mathbf{I}_M + \mathbf{V}\mathbf{V}^T$  the eigenvalue decomposition of  $b\mathbf{I}_M + \mathbf{V}\mathbf{V}^T$  with  $\mathbf{A} = \text{diag}(a_1, \dots, a_d, b, \dots, b)$  the ordered eigenvalues and  $\tilde{\mathbf{D}}^*$  the matrix containing the first  $d$  eigenvectors as columns.

Additionally, denoting by  $\mathbf{U} = b\mathbf{I}_d + \mathbf{V}^T\mathbf{V}$ , we have,

$$(8) \quad (\mathbf{X}|\mathbf{Y}=\mathbf{y}, W=w) \sim \mathcal{N}(\mathbf{U}^{-1}\mathbf{V}^T(\mathbf{y} - \boldsymbol{\mu}), w^{-1}b\mathbf{U}^{-1}).$$

It follows that  $\mathbb{E}[\mathbf{Y}|\mathbf{X}=\mathbf{x}] = \mathbf{V}\mathbf{x} + \boldsymbol{\mu}$  and  $\mathbb{E}[\mathbf{X}|\mathbf{Y}=\mathbf{y}] = \mathbf{U}^{-1}\mathbf{V}^T(\mathbf{y} - \boldsymbol{\mu})$ .

PROOF. It comes from the first three assumptions that

$$(\mathbf{Y}|W=w) \sim \mathcal{N}(\boldsymbol{\mu}, w^{-1}(\mathbf{V}\mathbf{V}^T + b\mathbf{I}_M))$$

and from the GSM definition 4.2 that

$$\mathbf{Y} \sim \mathcal{E}_M(\boldsymbol{\mu}, \mathbf{V}\mathbf{V}^T + b\mathbf{I}_M, f_{\boldsymbol{\theta}}).$$

Since  $\mathbf{V}\mathbf{V}^T$  is of rank  $d$ ,  $\mathbf{V}\mathbf{V}^T + b\mathbf{I}_M$  admits an eigenvalue decomposition  $\mathbf{D}\mathbf{A}\mathbf{D}^T$  as stated.

Distribution (8) follows from standard Gaussian vectors properties. Using the tower property, it comes then  $\mathbb{E}[\mathbf{Y}|\mathbf{X}=\mathbf{x}] = \mathbb{E}[\mathbb{E}[\mathbf{Y}|\mathbf{X}=\mathbf{x}, W]] = \mathbf{V}\mathbf{x} + \boldsymbol{\mu}$  and  $\mathbb{E}[\mathbf{X}|\mathbf{Y}=\mathbf{y}] = \mathbb{E}[\mathbb{E}[\mathbf{X}|\mathbf{Y}=\mathbf{y}, W]] = \mathbf{U}^{-1}\mathbf{V}^T(\mathbf{y} - \boldsymbol{\mu})$ .

□

The previous proposition states that a realization  $\mathbf{y} \in \mathbb{R}^M$  from an HD-ED, can also be seen as originating from a generative model with a lower-dimensional latent variable  $\mathbf{X} \in \mathbb{R}^d$ . Hence, a natural alternative to the standard PCA projection, is the conditional mean  $\hat{\mathbf{y}} = \mathbb{E}[\mathbf{X}|\mathbf{Y}=\mathbf{y}]$ , that is

$$(9) \quad \hat{\mathbf{y}} = Q(\mathbf{y}) = \mathbf{U}^{-1}\mathbf{V}^T(\mathbf{y} - \boldsymbol{\mu}),$$

and as a reconstruction or an approximation of the original information, the conditional mean  $\check{\mathbf{y}} = \mathbb{E}[\mathbf{Y}|\mathbf{X}=\hat{\mathbf{y}}]$ , that is  $\check{\mathbf{y}} = \mathbf{V}\hat{\mathbf{y}} + \boldsymbol{\mu}$ . Using the previous formulas, lower dimensional representations can thus be obtained using  $\mathbf{U}$  and  $\mathbf{V}$  but when estimating the parameters of the HD-ED, we get estimates for  $\mathbf{A}$  and  $\mathbf{D}$  instead. However, using that  $\mathbf{V}$  is of rank  $d$  and  $\mathbf{D}\mathbf{A}\mathbf{D}^T = b\mathbf{I}_M + \mathbf{V}\mathbf{V}^T$ , we can set

$$(10) \quad \mathbf{V} = \tilde{\mathbf{D}}^* \sqrt{\text{diag}(a_1, \dots, a_d) - b\mathbf{I}_d},$$

and deduce  $\mathbf{U}$  straightforwardly.

**5.2. Cluster globally, reduce locally.** The proposed dimension reduction and reconstruction, using the HD-ED model, generalizes to GSM, PPCA ([Tipping and Bishop, 1999c](#)) and robust PPCA ([Archambeau, Delannay and Verleysen, 2006](#)), the former using Gaussian and the latter Student distributions. Both have been extended to account for potential heterogeneity in data, considering mixtures, with the MPPCA model ([Tipping and Bishop, 1999d](#)) and its Student-based robust version ([Archambeau, Delannay and Verleysen, 2008](#)). In these models, all clusters are assumed to live in subspaces of the same dimension  $d$ . In the Gaussian case, an extension, allowing varying reduced dimensions  $d_k$  across clusters, have been proposed by [Bouveyron, Girard and Schmid \(2007\)](#) with their high dimensional Gaussian mixture model (HD-GMM). The proposed HD-GMM parameterization allows to handle high dimensional data in a computationally efficient way. However, it does not provide an actual lower dimensional representation of the data. While such a reduced-dimensional representation may often not be needed, it may be crucial to deal with hardware or software limitations. Originally, HD-GMM have not been designed for dimension reduction or compression but rather for clustering and density estimation in high-dimensional heterogeneous settings. Our HD-MED model uses the same efficient decomposition of the covariance matrix but generalizes it to the scale matrix of a GSM distribution. In addition, we describe how it can be further exploited as a dimension reduction technique. As finite mixture models, HD-MED can be used for clustering data into  $K$  clusters. For any possible observation  $\mathbf{y}$ , a HD-MED model provides a probability  $r_k(\mathbf{y})$  that  $\mathbf{y}$  is assigned to cluster  $k$  for each  $k = 1 : K$ . Using (9), a reduced-dimension representation  $\hat{\mathbf{y}}_k$  of  $\mathbf{y}$ , for each of the  $K$  different subspaces, is denoted by  $\hat{\mathbf{y}}_k = Q_k(\mathbf{y})$  and given by,

$$(11) \quad \hat{\mathbf{y}}_k = Q_k(\mathbf{y}) = \mathbf{U}_k^{-1} \mathbf{V}_k^T (\mathbf{y} - \boldsymbol{\mu}_k),$$

while its reconstruction  $\check{\mathbf{y}}_k$  in the original space is given by

$$\check{\mathbf{y}}_k = \mathbf{V}_k \hat{\mathbf{y}}_k + \boldsymbol{\mu}_k.$$

In practice, it is reasonable to use as a reduced-dimension representation of  $\mathbf{y}$  only the one corresponding to the most probable cluster  $k$ , *i.e.* with the highest  $r_k(\mathbf{y})$ . In this setting, HD-MED acts as a divide-and-conquer paradigm by initially clustering the data into  $K$  clusters and then performing cluster-specific data reduction. The divide step allows a much more effective reduction than if a single subspace was considered, while in the conquer step, little information is lost, as for any new observation  $\mathbf{y}$ , cluster assignment probabilities  $r_k(\mathbf{y})$  can be straightforwardly computed to decide on the best reduced representation to be used. However, for subsequent processing, it is important to keep track of clustering information for each observation. The reduced representations cannot be pooled back altogether, as they are likely to become impossible to distinguish across clusters. Also, as summarized in Algorithm 1 and illustrated in Section 7, each reduced cluster may have to be processed separately but this additional cost is negligible compared to the hardware and software gain of a more efficient representation.

In practice, the Expectation-Maximization (EM) algorithm ([Dempster, Laird and Rubin \(1977\)](#)), that iteratively computes and maximizes the conditional expectation of the complete-data log-likelihood, is commonly used to infer the parameters of finite mixture models. The number of clusters  $K$ , and their respective inner dimension  $d_k$  are hyper-parameters that need to be tuned prior to the EM steps. As a simple solution, we use the Bayesian Information Criterion (BIC) to tune  $K$  and the  $d_k$ 's at the same time. In contrast to MPPCA solutions which assume the same subspace dimension  $d$  for all clusters, the possibility to handle different  $d_k$ 's and to allow non Gaussian cluster shapes is important for the target applications involving datasets that are very large. Using different dimensions across clusters is likely to yield a more efficient reduced representation of the data as illustrated in Section 7.

However, the standard, or batch EM algorithm needs all the dataset to be loaded in a fast access memory (*e.g.* the RAM) which is often limited as this kind of memory is expensive. In the case of large-scale dataset, the RAM is often overloaded and supported by a slow-memory which makes the iterations of the EM very slow. Batch sizes are then limited by resource constraints, so that very large data sets need either to be downsampled or to be handled in an incremental manner. Incremental versions of EM exist and can be adapted to our setting. In section 6, we provide a way to deal with this large-scale case by using an online version of the EM algorithm.

## 6. Online Learning of High Dimensional Mixtures of Elliptical Distributions.

**6.1. Online EM, main assumptions.** When the data volume is too large the EM algorithm becomes slow because of multiple data transfer between the RAM and the store of the computer. A way to handle large volumes is to use online learning. Online learning refers to procedures able to deal with data acquired sequentially. Online variants of EM, among others, are described in Cappé and Moulines (2009); Maire, Moulines and Lefebvre (2017); Karimi et al. (2019a,b); Fort, Moulines and Wai (2020); Kuhn, Matias and Rebafka (2020); Nguyen, Forbes and McLachlan (2020). As an archetype of such algorithms, we consider the online EM of Cappé and Moulines (2009) which belongs to the family of stochastic approximation algorithms (Borkar (2009)). This algorithm has been well theoretically studied and extended. However, it is designed only for distributions that admit a data augmentation scheme, or a latent variable formulation, yielding a complete likelihood of the exponential family form, see (12) below. This case is already very broad, including *e.g.* Gaussian, gamma, Student distributions and mixtures of those. We recall the main assumptions required and the online EM iteration, based on a latent variable formulation.

Assume  $(\mathbf{Y}_i)_{i=1}^N$  is a sequence of  $N$  *i.i.d.* replicates of a random variable  $\mathbf{Y} \in \mathcal{Y} \subset \mathbb{R}^M$ , observed one at a time. Extension to successive mini-batches of observations is straightforward (Nguyen, Forbes and McLachlan (2020)). In addition,  $\mathbf{Y}$  is assumed to be the visible part of  $(\mathbf{Y}, \mathbf{Z})$ , where  $\mathbf{Z} \in \mathbb{R}^l$  is a latent variable, *e.g.* the unknown component label in a mixture model or a mixing weight in a GSM formulation, and  $l \in \mathbb{Z}_+$ . For  $i \in [1 : N]$  then, each  $\mathbf{Y}_i$  is the visible part of  $(\mathbf{Y}_i, \mathbf{Z}_i)$ . Suppose  $\mathbf{Y}$  arises from some data generating process (DGP) characterised by a probability density function  $f(\mathbf{y}; \Theta_0)$ , with unknown parameters  $\Theta_0 \in \mathbb{T} \subseteq \mathbb{R}^p$ , for  $p \in \mathbb{Z}_+$ .

Using the sequence  $(\mathbf{Y}_i)_{i=1}^N$ , the method of Cappé and Moulines (2009) allows to sequentially estimate  $\Theta_0$  provided the following assumptions are met:

(A1) The complete-data likelihood for  $(\mathbf{Y}, \mathbf{Z})$  is of the exponential family form:

$$(12) \quad f_c(\mathbf{y}, \mathbf{z}; \Theta) = h(\mathbf{y}, \mathbf{z}) \exp \left\{ [\mathbf{s}(\mathbf{y}, \mathbf{z})]^\top \phi(\Theta) - \psi(\Theta) \right\},$$

with  $h : \mathbb{R}^{M+l} \rightarrow [0, \infty)$ ,  $\psi : \mathbb{R}^p \rightarrow \mathbb{R}$ ,  $\mathbf{s} : \mathbb{R}^{M+l} \rightarrow \mathbb{R}^q$ ,  $\phi : \mathbb{R}^p \rightarrow \mathbb{R}^q$ , for  $q \in \mathbb{Z}_+$ .

(A2) The function

$$(13) \quad \bar{\mathbf{s}}(\mathbf{y}; \Theta) = \mathbb{E}[\mathbf{s}(\mathbf{Y}, \mathbf{Z}) | \mathbf{Y} = \mathbf{y}; \Theta]$$

is well-defined for all  $\mathbf{y}$  and  $\Theta \in \mathbb{T}$ , where  $\mathbb{E}[\cdot | \mathbf{Y} = \mathbf{y}; \Theta]$  is the conditional expectation when  $\mathbf{X}$  arises from the DGP characterised by  $\Theta$ .

(A3) There is a convex  $\mathbb{S} \subseteq \mathbb{R}^q$ , satisfying: (i) for all  $\gamma \in (0, 1)$ ,  $\mathbf{s} \in \mathbb{S}$ ,  $\mathbf{y} \in \mathcal{Y}$ , and  $\Theta \in \mathbb{T}$ ,  $(1 - \gamma)\mathbf{s} + \gamma\bar{\mathbf{s}}(\mathbf{y}; \Theta) \in \mathbb{S}$ ; and (ii) for any  $\mathbf{s} \in \mathbb{S}$ , the function  $Q(\mathbf{s}; \Theta) = \mathbf{s}^\top \phi(\Theta) - \psi(\Theta)$  has a unique global maximizer on  $\mathbb{T}$  denoted by

$$(14) \quad \bar{\Theta}(\mathbf{s}) = \arg \max_{\Theta \in \mathbb{T}} Q(\mathbf{s}; \Theta).$$

Let  $(\gamma_i)_{i=1}^N$  be a sequence of learning rates in  $(0, 1)$  and let  $\Theta^{(0)} \in \mathbb{T}$  be an initial estimate of  $\Theta_0$ . For each  $i \in [1 : N]$ , the online EM of Cappé and Moulines (2009) proceeds by computing

$$(15) \quad \mathbf{s}^{(i)} = \gamma_i \bar{\mathbf{s}}(\mathbf{y}_i; \Theta^{(i-1)}) + (1 - \gamma_i) \mathbf{s}^{(i-1)},$$

and

$$(16) \quad \Theta^{(i)} = \bar{\Theta}(\mathbf{s}^{(i)}),$$

where  $\mathbf{s}^{(0)} = \bar{\mathbf{s}}(\mathbf{y}_1; \Theta^{(0)})$ . It is shown in Theorem 1 of Cappé and Moulines (2009) that when  $N$  tends to infinity, the sequence  $(\Theta^{(i)})_{i=1:N}$  of estimators of  $\Theta_0$  satisfies a convergence result to stationary points of the likelihood (cf. Cappé and Moulines (2009) for a more precise statement).

**6.2. Online EM for HD-ED.** In this subsection, we derive the online EM algorithm for HD-ED. The extension to mixtures of those (HD-MED) is straightforward and is detailed in Nguyen and Forbes (2022) or in the supplementary Section 1. The weight  $W$  distribution  $f$  in the GSM formulation (3) is assumed to belong to the exponential family. This case may seem restrictive but it encompasses a number of ED as the Gaussian, Student, Normal Inverse Gamma with no skewness *etc.* distributions.

**PROPOSITION 6.1** (HD-ED exponential form). *Let  $\mathbf{Y}$  be a HD-ED distributed variable,  $\mathbf{Y} \sim \mathcal{HE}_{Md}(\boldsymbol{\mu}, \tilde{\mathbf{D}}^*, \mathbf{a}, b, \boldsymbol{\theta})$ , and  $W$  a weight variable with pdf  $f_{\boldsymbol{\theta}}$ . The set of parameters is denoted by  $\Theta = (\boldsymbol{\mu}, \tilde{\mathbf{D}}^*, \mathbf{a}, b, \boldsymbol{\theta})$ , with  $\tilde{\mathbf{D}}^*$  defined by its column vectors  $\tilde{\mathbf{D}}^* = [\mathbf{d}_1, \dots, \mathbf{d}_d]$ . If  $W$  belongs to the exponential family, i.e.  $f_{\boldsymbol{\theta}}(w) = h_w(w) \exp [\mathbf{s}_w(w)^T \boldsymbol{\phi}_w(\boldsymbol{\theta}) - \psi_w(\boldsymbol{\theta})]$ , the complete data likelihood*

$$(17) \quad f_c(\mathbf{y}, w; \Theta) = f_{\boldsymbol{\theta}}(w) \mathcal{N}_M(\mathbf{y}; \boldsymbol{\mu}, w^{-1} \mathbf{D} \mathbf{A} \mathbf{D}^T),$$

can be expressed in an exponential family form (12) with

$$(18) \quad \begin{aligned} \mathbf{s}(\mathbf{y}, w) &= \begin{bmatrix} w\mathbf{y} \\ w\text{vec}(\mathbf{y}\mathbf{y}^T) \\ w\mathbf{y}^T\mathbf{y} \\ w \\ \mathbf{s}_w(w) \end{bmatrix}, \quad \boldsymbol{\phi}(\Theta) = \begin{bmatrix} \sum_{m=1}^d \left( \frac{1}{a_m} - \frac{1}{b} \right) \mathbf{d}_m \mathbf{d}_m^T \boldsymbol{\mu} + \frac{1}{b} \boldsymbol{\mu} \\ \frac{1}{2} \sum_{m=1}^d \left( \frac{1}{b} - \frac{1}{a_m} \right) \text{vec}(\mathbf{d}_m \mathbf{d}_m^T) \\ -\frac{1}{2b} \\ \frac{1}{2} \sum_{m=1}^d \left( \frac{1}{b} - \frac{1}{a_m} \right) \boldsymbol{\mu}^T \mathbf{d}_m \mathbf{d}_m^T \boldsymbol{\mu} - \frac{1}{2b} \boldsymbol{\mu}^T \boldsymbol{\mu} \\ \phi_w(\boldsymbol{\theta}) \end{bmatrix}, \\ \psi(\Theta) &= \frac{1}{2} \sum_{m=1}^d \log a_m + \frac{M-d}{2} \log b + \psi_w(\boldsymbol{\theta}), \end{aligned}$$

where  $\text{vec}$  denotes the vectorization operator (Schott (2016)).

**PROOF.** The proof is detailed in supplementary Section 2. □

The online EM algorithm (OEM) consists, as the batch EM, of two steps, the first one is the computation of the sufficient statistics (13), and the second one is the maximization of the likelihood (14). For the first step, we need to compute  $\bar{\mathbf{s}}(\mathbf{y}; \Theta) = \mathbb{E}[\mathbf{s}(\mathbf{Y}, \mathbf{Z}) | \mathbf{Y} = \mathbf{y}; \Theta]$ . This quantity requires to compute the following expectations  $\mathbb{E}[W | \mathbf{Y} = \mathbf{y}; \Theta]$ , and  $\mathbb{E}[\mathbf{s}_w(W) | \mathbf{Y} = \mathbf{y}; \Theta]$ .

**PROPOSITION 6.2** (Expectations of sufficient statistics). *Let  $\mathbf{Y} \sim \mathcal{HE}_{Md}(\boldsymbol{\mu}, \tilde{\mathbf{D}}^*, \mathbf{a}, b, \boldsymbol{\theta})$ ,  $\boldsymbol{\Theta} = (\boldsymbol{\mu}, \tilde{\mathbf{D}}^*, \mathbf{a}, b, \boldsymbol{\theta})$ , and  $W$  the mixing variable  $W \sim f_{\boldsymbol{\theta}}$ . Then  $\mathbf{Y}$  has density (2) with generator  $g$  and Mahalanobis distance  $u = (\mathbf{y} - \boldsymbol{\mu})^T \mathbf{D} \mathbf{A}^{-1} \mathbf{D}^T (\mathbf{y} - \boldsymbol{\mu})$  defined as in (7). It follows that*

$$(19) \quad \mathbb{E}[W | \mathbf{Y} = \mathbf{y}; \boldsymbol{\Theta}] = -\frac{2}{(2\pi)^{M/2}} \frac{g'(u)}{g(u)}.$$

where  $g'$  denotes the derivative of  $g$ .

**PROOF.** According to equations (2) and (3), we have

$$\begin{aligned} \mathbb{E}[W | \mathbf{Y} = \mathbf{y}; \boldsymbol{\Theta}] &= \frac{1}{(2\pi)^{M/2} g(u)} \int_{\mathbb{R}^+} w f_{\boldsymbol{\theta}}(w) \exp(-\frac{w}{2} u) dw \\ &= -\frac{2}{(2\pi)^{M/2}} \frac{g'(u)}{g(u)}, \end{aligned}$$

which proves equation (19).  $\square$

In contrast, there is no general formula for the expectation of  $s_w(W)$ , which depends on the mixing distribution  $f_{\boldsymbol{\theta}}$ . Once the expectation of the sufficient statistics in (13) is computed, we can update it following (15).

The next OEM step is the maximization step described in (14), which gives an estimation of the parameters at each iteration. The solution for  $\boldsymbol{\theta}$  varies with  $f_{\boldsymbol{\theta}}$ , but solutions for  $\boldsymbol{\mu}$ ,  $\mathbf{A}$ , and  $\tilde{\mathbf{D}}^*$  can be derived as follows. Let  $\bar{\boldsymbol{\Theta}}(\mathbf{s})$  be defined as the unique maximizer of function  $Q(\mathbf{s}, \boldsymbol{\Theta}) = \mathbf{s}^T \phi(\boldsymbol{\Theta}) - \psi(\boldsymbol{\Theta})$  with  $\mathbf{s}$  a vector that matches the definition and dimension of  $\phi(\boldsymbol{\Theta})$  in (18), and can be conveniently written as

$$(20) \quad \mathbf{s} = \begin{bmatrix} \mathbf{s}_1 \\ \text{vec}(\mathbf{S}_2) \\ \mathbf{s}_3 \\ \mathbf{s}_4 \\ \mathbf{s}_5 \end{bmatrix},$$

with  $\mathbf{s}_1$  a  $M$ -dimensional vector,  $\mathbf{S}_2$  a  $M \times M$  matrix, and  $\mathbf{s}_3, \mathbf{s}_4, \mathbf{s}_5$  three scalar values. Parameters are updated by maximizing  $Q$  with respect to  $\boldsymbol{\Theta}$ .  $\bar{\boldsymbol{\Theta}}(\mathbf{s})$  is defined as the root of the first-order condition

$$(21) \quad \mathbf{J}_{\phi}(\boldsymbol{\Theta}) \mathbf{s} - \frac{\partial \psi}{\partial \boldsymbol{\Theta}}(\boldsymbol{\Theta}) = \mathbf{0},$$

where  $\mathbf{J}_{\phi}(\boldsymbol{\Theta}) = \frac{\partial \phi}{\partial \boldsymbol{\Theta}}$  is the Jacobian of  $\phi$ , with respect to  $\boldsymbol{\Theta}$ . Computing gradients leads to  $\bar{\boldsymbol{\Theta}}(\mathbf{s}) = (\bar{\boldsymbol{\mu}}(\mathbf{s}), \bar{\mathbf{D}}^*(\mathbf{s}), \bar{\mathbf{A}}(\mathbf{s}), \bar{\boldsymbol{\theta}}(\mathbf{s}))$ , where  $\bar{\boldsymbol{\mu}}(\mathbf{s})$ , and  $\bar{\mathbf{A}}(\mathbf{s})$  are closed form, and  $\bar{\mathbf{D}}^*(\mathbf{s})$  can be found using Riemannian optimization.

**PROPOSITION 6.3** (Maxima). *We proceed in a ECM-like procedure (Meng and Rubin (1993)) by optimizing each parameters separately and incorporating them during the optimization of each other parameters.  $(\boldsymbol{\mu}, \mathbf{A})$  can be optimized easily using (21) and computing the gradients gives*

$$(22) \quad \bar{\boldsymbol{\mu}} = \frac{\mathbf{s}_1}{\mathbf{s}_4},$$

$$(23) \quad \bar{a}_m = \mathbf{d}_m^T (\mathbf{S}_2 + s_4 \bar{\mu} \bar{\mu}^T - 2 \bar{\mu} \mathbf{s}_1^T) \mathbf{d}_m \quad \text{for } m \in [1 : d],$$

$$(24) \quad \bar{b} = \frac{1}{M-d} \left( s_4 \bar{\mu}^T \bar{\mu} + s_3 - 2 \bar{\mu}^T \mathbf{s}_1 - \sum_{m=1}^d \bar{a}_m \right).$$

The maximisation in  $\tilde{\mathbf{D}}^*$  has to take into account that  $\tilde{\mathbf{D}}^* \in St(M, d)$ , the Stiefel manifold of the  $M \times d$  matrices. Plugin-in the expressions of  $\mu$  and  $A$  above and omitting parts that depend on  $\theta$  we have,

$$(25) \quad \overline{\tilde{\mathbf{D}}}^* = \arg \max_{\tilde{\mathbf{D}}^* \in St(M, d)} \sum_{m=1}^d \left( \frac{1}{\bar{a}_m} - \frac{1}{\bar{b}} \right) \mathbf{d}_m^T (2 \bar{\mu} \mathbf{s}_1^T - \mathbf{S}_2 - s_4 \bar{\mu} \bar{\mu}^T) \mathbf{d}_m.$$

For  $\bar{\theta}$ , a general closed-form expression is not available, but if there are no particular constraints it results from solving the following equation

$$(26) \quad s_5 \frac{\partial \phi_w}{\partial \theta}(\bar{\theta}) - \frac{\partial \psi_w}{\partial \theta}(\bar{\theta}) = \mathbf{0}.$$

PROOF. We compute the gradients and use (21) for  $\mu$  and  $A$ , however for the vector  $\tilde{\mathbf{D}}^*$  we only plug-in the optimized values  $\bar{\mu}$ , and  $\bar{A}$  in (17) and solve

$$(27) \quad \arg \max_{\tilde{\mathbf{D}}^* \in St(M, d)} \mathbf{s}^T \phi \left( \bar{\mu}, \bar{A}, \tilde{\mathbf{D}}^* \right) - \psi \left( \bar{\mu}, \bar{A}, \tilde{\mathbf{D}}^* \right).$$

□

---

**Algorithm 1** Divide & Conquer high dimensional matching for MRF reconstruction

---

**Input** Dictionary of (signal, parameters) pairs  $\mathcal{D}_f = \{\mathbf{y}_i, \mathbf{t}_i\}_{i=1:N}$ ,  $N >> 1$ ,  $\mathbf{t}_i \in \mathbb{R}^L$ ,  $\mathbf{y}_i \in \mathbb{R}^M$ ,  $M >> 1$ .  
In vivo acquired signals  $\{\tilde{\mathbf{y}}_j\}_{j=1:\tilde{N}}$ ,  $\tilde{\mathbf{y}}_j \in \mathbb{R}^M$ .

1: **Reduced dimension representation of the dictionary:**  $\{\hat{\mathbf{y}}_i, \mathbf{t}_i, r_i(\mathbf{y}_i)\}_{i=1:N}$

1.1 **Online HD-MED inference from  $\{\mathbf{y}_i\}_{i=1:N}$ :**  $K$  clusters,  $d_k < M$  for  $k = 1 : K$   $\implies$  cluster assignment probabilities and cluster-wise projections  $(\mathbf{r}, \mathbf{Q}) = \{r_k(\cdot), Q_k(\cdot)\}_{k=1:K}$

1.2 **Cluster-wise fingerprint reductions:**  $\{\mathbf{y}_i\}_{i=1:N}, \mathbf{r}, \mathbf{Q} \implies \{\hat{\mathbf{y}}_i = Q_k(\mathbf{y}_i), i \in I_k\}$  with  $I_k = \{i, s.t k = \arg \max_{\ell} r_{\ell}(\mathbf{y}_i)\}$ , for  $k = 1 : K$

2: **Cluster-wise matching of acquired signals:**

2.1 **Cluster-wise invivo signal reductions:** Use learned  $(\mathbf{r}, \mathbf{Q})$  from step 1.1 to obtain  $\{Q_k(\tilde{\mathbf{y}}_j), j \in \tilde{I}_k\}$  with  $\tilde{I}_k = \{j, s.t k = \arg \max_{\ell} r_{\ell}(\tilde{\mathbf{y}}_j)\}$ , for  $k = 1 : K$

2.2 **Matching:** For  $k = 1 : K$ , for  $j \in \tilde{I}_k$ , determine  $i(\tilde{\mathbf{y}}_j) = \arg \min_{i \in I_k} d(Q_k(\tilde{\mathbf{y}}_j), \hat{\mathbf{y}}_i)$  and set  $\tilde{\mathbf{t}}_j = \mathbf{t}_{i(\tilde{\mathbf{y}}_j)}$

**Return** Matched tissue properties  $\{\tilde{\mathbf{t}}_j\}_{j=1:\tilde{N}}, \tilde{\mathbf{t}}_j \in \mathbb{R}^L$

---

**7. Application to magnetic resonance fingerprinting (MRF) reconstruction.** MRF is able to provide multiple quantitative tissue parameters images from shorter acquisition times, thanks to the simultaneous application of transient states excitation and highly undersampled  $k$ -space read-outs. These two aspects have a combined impact on acquisition times

and image reconstruction accuracy. More undersampling allows more parameter estimations in reasonable acquisition times but is also responsible for larger undersampling errors, noise and artifacts, reducing map reconstruction accuracy.

In some earlier work (Oudoumanessah et al., 2024), the Gaussian version of our procedure, referred to as HD-GMM, was evaluated, for the reconstruction, in an ideal setting, of fully sampled acquisitions, targetting  $L = 3$  parameters. In this scenario, it was reported that HD-GMM, coupled with the online EM algorithm, achieved results comparable to full dictionary matching while significantly reducing reconstruction times. However, as we report in this section, HD-GMM performance degrades when dealing with largely undersampled acquisitions and becomes inadequate for estimating six parameters. In contrast, performance can be maintained by considering an elliptical version of our procedure (HD-MED) less sensitive to outliers.

In this section, we thus show how we can accurately reconstruct six parameter maps from *in vivo* undersampled acquisitions (Section 7.1) by leveraging an extensive high-resolution dictionary (Section 7.2) and the HD-MED model. Algorithm 1 provides a schematic summary of our procedure. Figure 1 provides an illustration of our matching strategy or step 2 in Algorithm 1, once the HD-MED model has been estimated. Each acquired signal is first assigned to one of the learned clusters, and reduced accordingly to be then matched to the best dictionary signal in the corresponding cluster. It leads to improvements in both memory management complexity and reconstruction speed for parameter maps when compared to traditional dictionary matching, see Section 7.5.

We compare two instances of HD-MED, namely HD-GMM and HD-STM, where the mixture components are respectively set to Gaussian and Student distributions. The computations for the corresponding OEM are detailed in Section 3 of the Appendix. All the experiments are performed with a Python code using the JAX library (Bradbury et al., 2018) with Nvidia V100-32gb GPU except for the dictionary generation part that is done with a mix of Matlab and Python. The Python code is available at <https://github.com/geoffroyO/HD-MED>.

**7.1. Undersampled MRF acquisitions.** *In vivo* acquisitions were conducted on 6 healthy volunteers ( $28 \pm 5.5$  years old, 3 males and 3 females) using a 32-channel head receiver array on a Philips 3T Achieva dStream MRI at the IRMaGe facility (MAP-IRMaGe protocol, NCT05036629). This study was approved by the local medical ethics committee and informed consent was obtained from each volunteer prior to image acquisition. The imaging pulse sequence was based on an IR-bSSFP acquisition. 260 repetitions were acquired following the parameters proposed in Coudert et al. (2024). The acquisitions were performed using quadratic variable density spiral sampling (12 interleaves out of 13), matrix size=192x192x(4-5), voxel size=1.04x1.04x3.00 mm<sup>3</sup> for a total scan duration of 2 minutes per slice. While the acquisition time may appear high for an MRF context compared to Gu et al. (2018), where full T1 and T2 maps are generated in 2 minutes, the longer sequences in Coudert et al. (2024) account for vascular parameters without requiring contrast agents.

Spiral sampling in MRI involves acquiring data in a spiral trajectory through k-space, covering the center first and gradually moving outward, which allows for faster data acquisition and more efficient use of the scanner's gradients. However, this method leads to undersampling noise, manifesting as artifacts, because it captures less information about the image, reducing the ability to accurately reconstruct fine details (Körzdörfer et al., 2019). It results in a number  $\tilde{N}_s \approx 140,000 - 180,000$  of *in vivo* MR signals per subject  $s$ , each of dimension  $M = 260$ , and a total number  $\tilde{N}$  of approximately  $\tilde{N} = 960,000$  signals to be matched for the reconstruction of  $L = 6$  parameter maps for 6 subjects.

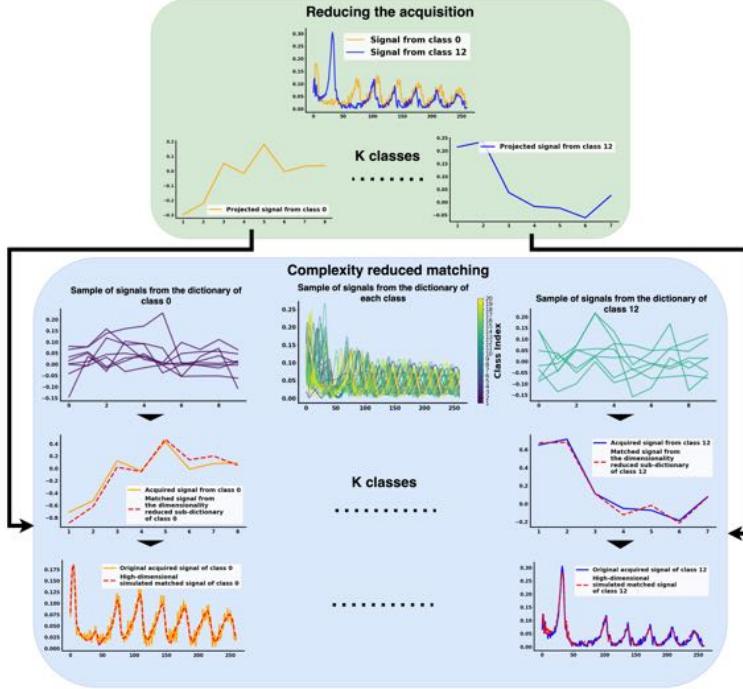


Fig 1: HD-MED-based low-dimensional signal matching step (step 2 of Algorithm 1) Green block: each acquired signal is assigned to one of the  $K$  HD-MED learned clusters and reduced accordingly using the assigned cluster projection. Blue block: reduced acquired signals are then matched with their closest reduced simulated counterpart in the cluster.

**7.2. MRF dictionary.** As introduced, the MRF sequence used here is a bSSFP-derived sequence. This type of sequence has been preferred because of its sensitivity to local frequency distributions related to microvascular structures in the imaging voxel. To account for this, the MRF dictionary is generated using the approach described in Coudert et al. (2024). First a base dictionary  $\mathcal{D}_{f_0}$  is simulated using Bloch-equations for different combinations of  $(T_1, T_2, B_1, \delta f)$  browsed through a 4-dimensional regular grid. Simulations were made at a magnetic field strength of 3.0T, on a regular parameter grid made of 20  $T_1$  values (from 200 to 3500ms), 20  $T_2$  values (from 10 to 600ms), 10  $B_1$  values (from 0.7 to 1.2) and 100 frequency offset  $\delta f$  values (from -50 to 49 Hz with an increment of 1 Hz), keeping only signals for which  $T_1 > T_2$ , resulting in an initial 390,000 entries dictionary. These Bloch simulations are obtained using an in-house mix of Python and Matlab code based on initial implementation by B. Hargreaves (Hargreaves). The relaxometric parameters vary within the dictionary to allow their estimation.  $B_1$  is varied to ensure the realism of the dictionary, given the sequence's sensitivity to this parameter, which could otherwise bias the estimation of the other parameters. Finally, varying  $\delta f$  values are necessary to compute microvascular contributions, as detailed below.

To build signals that capture additional microvascular network's blood volume (CBV) and mean vessel radius ( $R$ ) information, we follow a construction proposed by Coudert et al. (2024). Microvascular network segmentations are used to pre-compute 2500 different intra-voxel frequency distributions centred at  $\delta f$  values. A CBV and  $R$  value characterizes each distribution. Using these pre-computed distributions, new signals are obtained by summing signals from the base dictionary weighted according to each frequency distribution. It results an expanded 6-dimensional dictionary  $\mathcal{D}_f$  of almost  $N = 400,000,000$  signals of dimension  $M = 260$ , encoding for  $(T_1, T_2, B_1, \delta f, CBV, R)$ . Figure 2 gives an illustration on how

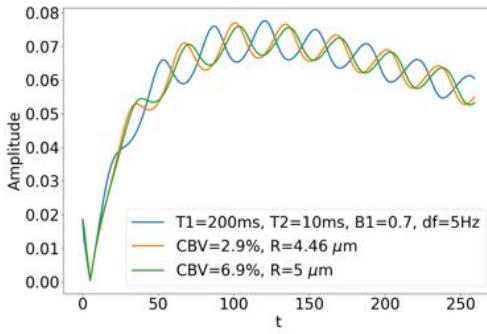


Fig 2: Illustration of the base dictionary expansion. The blue signal corresponding to a non-blood voxel with associated  $T_1$ ,  $T_2$ ,  $B_1$  and  $\delta f$  values. By convoluting this signal along the  $\delta f$  dimension with two different frequency distributions corresponding to different values of  $CBV$  and  $R$ , the resulting orange and green signals are then obtained for voxels with the same  $T_1$ ,  $T_2$ ,  $B_1$  and  $\delta f$  values as the blue signal but with different  $CBV$  and  $R$  values.

this operation changes a given signal in the base dictionary into 2 new signals depending on values of  $CBV$  and  $R$ .

**7.3. Model selection and initialization.** The only two hyperparameters that need to be set are the number of mixture components  $K$  and the desired reduced dimension vector  $\mathbf{d}$ . For mixture models,  $K$  can be selected using the Bayesian Information Criterion (BIC) (Schwarz, 1978), which requires running multiple models with varying  $K$  values and finding the elbow or the minimum of the BIC curve. The desired dimension vector  $\mathbf{d}$  is automatically set during an initialization step, which makes  $K$  the only parameter that the user needs to set prior to running the OEM algorithm.

The initialization step includes determining the reduced dimension vector  $\mathbf{d}$ . We randomly choose a subset of the dictionary that fits into memory and run a batch full covariance EM algorithm to determine a first estimation of the  $\Sigma_k$ 's. These estimations are decomposed into eigenvalues and eigenvectors to determine  $\mathbf{D}_k$ 's and  $\mathbf{A}_k$ 's. For each  $k$ ,  $d_k$  is then determined by applying a scree plot to the eigenvalues using the kneedle algorithm (Satopaa et al., 2011),  $a_k$  is then initialized to the first  $d_k$  eigenvalues, and the last eigenvalues are averaged to initialize  $b_k$ . This kind of initialization of  $\mathbf{D}_k$  and  $\mathbf{A}_k$  is called spectral initialization and is similar to the one proposed by Hong et al. (2021), proving to be relatively stable compared to other types of initializations.

In Figure 3 (left), we show the different BIC curves for HD-GMM and HD-STM models trained on the dictionary of signals with  $K$  varying from 5 to 80. The elbows of the curves are found at  $K = 30$  for HD-GMM, and at  $K = 25$  for HD-STM. Figures 3 (middle and right) show the different vectors  $\mathbf{d}$  obtained for varying  $K$ . The larger the points, the higher the number of components having the reduced dimension indicated on the  $y$ -axis. Most of these dimensions are between 5 and 40, meaning that the original  $M = 260$  signal dimension can be reduced by a factor of approximately 7 to 10 leading to a reduction from 13 To to approximately 1.5 To of signals stored in memory. To check, how much is lost in this reduction, we compute the root mean square error (RMSE) between the dictionary  $\mathbf{y}_i$  signals and their reconstructions  $\check{\mathbf{y}}_i$  from their reduced representations (11),

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{y}_i - \check{\mathbf{y}}_i)^2}.$$

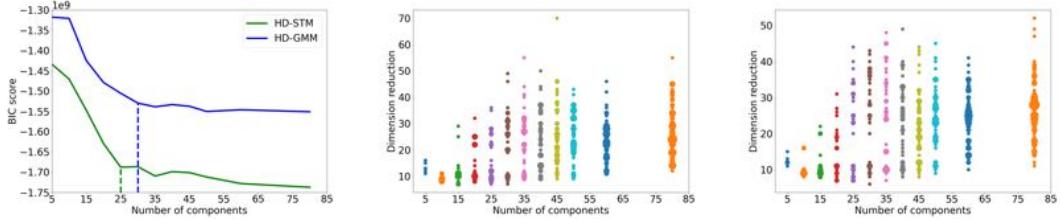


Fig 3: Model selection. BIC scores (left) and clusters dimensions, with respect to the number of components  $K$ , for HD-GMM (middle) and HD-STM (right) applied to signals of dimension  $M = 260$ . Points sizes reflect the proportion of clusters with a given dimension ( $y$ -axis).

Figure 4 shows as expected that the reconstruction RMSE decreases when  $K$  increases and is smaller for HD-STM in particular for small values of  $K$ . For the selected  $K$  values it represents about 2.5% of the signal.

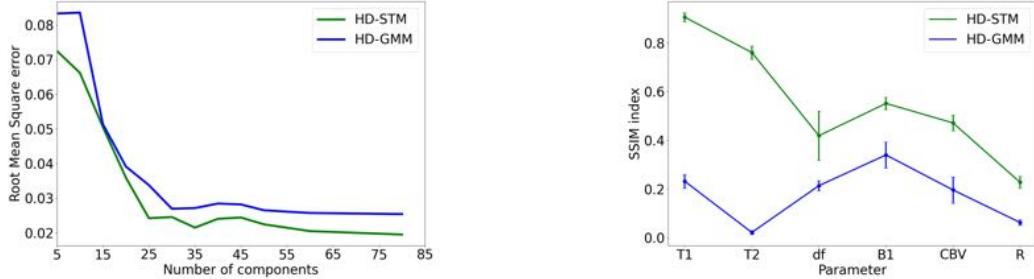


Fig 4: Dictionary reduction loss as measured by RMSE values over the dictionary signals, for HD-GMM and HD-STM, with respect to the number of components  $K$ .

Fig 5: Parameter maps comparison. Average SSIM in [-1,1], the higher the better, and standard deviation over subjects, between HD-GMM (resp. HD-STM) and full matching maps.

**7.4. Tissue and sensitivity parameters reconstruction.** For the six subjects, we compare the parameter maps obtained with our Algorithm 1 using HD-GMM and HD-STM to that obtained with traditional matching, referred to as *full matching* (*i.e.*, matching with uncompresssed signals). While *full matching* cannot be considered the ground truth, it remains the reference method due to its robustness to the slightly undersampled MRF acquisitions used in this study, as demonstrated in (Coudert et al., 2024), despite its high computational and memory demands.

Table 1 reports for each of the 6 parameters, the distances to *full matching* parameter estimations, showing the average Mean Absolute Errors (*MAE*) across voxels for all  $n_{subject} = 6$  subjects and all slices. For another comparison that takes into account the image structure, Figure 5 shows the reconstruction quality as measured by the structural similarity index measure (SSIM) (Wang et al., 2004), for both HD-STM and HD-GMM when compare to full matching, for each parameter and averaged over subjects. The SSIM is a decimal value between -1 and 1, where 1 indicates perfect similarity, 0 indicates no similarity, and -1

indicates perfect anti-correlation.

$$MAE = \frac{1}{n_{\text{subject}}} \sum_{s=1}^{n_{\text{subject}}} \frac{1}{\tilde{N}_s} \sum_{j=1}^{\tilde{N}_s} |\tilde{t}_{j,s} - t_{j,s}^{\text{match}}|,$$

$$SSIM = \frac{1}{n_{\text{subject}}} \sum_{s=1}^{n_{\text{subject}}} SSIM \left( (\tilde{t}_{j,s})_{1 \leq j \leq \tilde{N}_s}, (t_{j,s}^{\text{match}})_{1 \leq j \leq \tilde{N}_s} \right).$$

TABLE 1

*Parameter maps reconstruction. Average MAE and standard deviation, over voxels and subjects, for HD-GMM ( $K = 30$ ) and HD-STM ( $K = 25$ ) with respect to full matching. Best values in bold.*

Parameter	$T_1$ (ms)	$T_2$ (ms)	$\delta f$ (Hz)	$B_1$ sensitivity ( $10^{-3}$ )	CBV (%)	$R$ ( $\mu\text{m}$ )
HD-GMM	$462 \pm 29$	$325 \pm 17$	$10 \pm 1$	$92 \pm 18$	$5 \pm 0.8$	$2 \pm 0.03$
HD-STM	<b><math>78 \pm 14</math></b>	<b><math>5 \pm 2</math></b>	<b><math>6 \pm 1</math></b>	<b><math>40 \pm 5</math></b>	<b><math>1.5 \pm 0.5</math></b>	<b><math>1.6 \pm 0.06</math></b>

HD-STM consistently outperforms HD-GMM across all parameters, as evidenced by Figures 5 and Table 1. HD-STM achieves superior SSIM and MAE values, indicating better structural correspondence. This advantage is particularly pronounced for parameters such as  $T_1$  and  $T_2$ , where HD-STM demonstrates robust reconstruction capabilities, whereas HD-GMM fails to accommodate too much undersampling noise. However, for parameters like CBV and  $R$ , HD-STM’s performance declines, suggesting that these parameters are inherently more challenging to model accurately. In these cases, HD-GMM yields the lowest SSIM values and the highest MAE values, further underscoring its limitations. Overall, while HD-STM proves to be the more reliable method, the significant variability across subjects—particularly for  $\delta f$  and CBV—and the observed SSIM drop for  $R$  highlight the need for further refinements.

For another assessment of the maps quality, Table 2 presents the mean values and standard deviations of  $T_1$ ,  $T_2$ ,  $CBV$ , and  $R$ , computed over voxels in white and gray matter ROIs delineated on  $T_1$  maps obtained from *full matching*. Compared to the ranges for healthy subjects reported by Wansapura et al. (1999); Bjørnerud and Emblem (2010); Delphin et al. (2023), HD-STM produces values that are both more consistent with expected ranges and closer to those obtained using full matching than to those derived from HD-GMM. In particular, with HD-GMM,  $T_2$  values significantly depart from the literature reference, as also visible in Figure 6.

Indeed, the conclusions drawn from Figure 5 and Tables 1–2 are further illustrated in Figure 6. This figure highlights HD-GMM’s failure to reconstruct parameters accurately, even when some estimates fall within healthy ranges reported in the literature from Table 2. In contrast, HD-STM consistently delivers parameter estimates for  $T_1$ ,  $T_2$ ,  $\delta f$ , and  $B_1$  that are nearly artifact-free and closely aligned with expected values. Additionally, HD-STM effectively captures primary structures and generates homogeneous maps for vascular parameters such as  $CBV$  and  $R$  even though minor residual *shim* artifacts persist, these are likely due to the high degree of undersampling in the acquired images.

**7.5. Computation times.** Experiments conducted on an Nvidia V100-32gb GPU are fast. It takes only 7 minutes to reconstruct 1 slice of a subject using both HD-GMM and HD-STM, compared to 45 minutes for *full matching*. This represents a 6-fold time reduction for an HD-MED model. However, in more realistic scenarios, calculations are performed locally

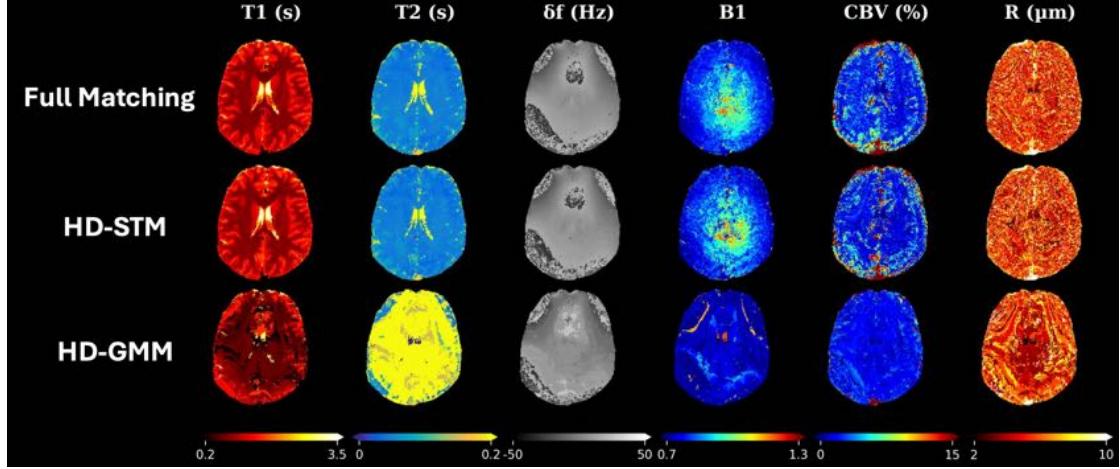


Fig 6:  $T_1$ ,  $T_2$ ,  $\delta f$ ,  $B_1$  sensitivity,  $CBV$ ,  $R$  maps (columns) with different methods (lines) - full matching, HD-STM ( $K = 25$ ) and HD-GMM ( $K = 30$ ). Results for all other slices of this subject are shown in Appendix Section 5.

TABLE 2  
Mean  $T_1$ ,  $T_2$ ,  $CBV$ ,  $R$  values and standard deviations, in white (WM) and grey (GM) matters ROIs. Values significantly departing from literature reference (last column) are in red.

Parameter	ROI	Full matching	HD-GMM	HD-STM	Literature (Wang et al., 2019b)
$T_1$ (ms)	WM	$891 \pm 6$	$650 \pm 23$	$869 \pm 19$	690-1100
	GM	$1566 \pm 12$	$1650 \pm 57$	$1631 \pm 56$	1286-1393
$T_2$ (ms)	WM	$52 \pm 0.0$	$400 \pm 15$	$48 \pm 2$	56-80
	GM	$95 \pm 4$	$370 \pm 43$	$96 \pm 12$	78-117
$CBV$ (%)	WM	$5.4 \pm 0.6$	$2.0 \pm 0.1$	$4.8 \pm 0.8$	1.7 - 3.6
	GM	$9.2 \pm 1.1$	$4.2 \pm 0.8$	$7.6 \pm 1$	3.0 - 8.0
$R$ ( $\mu s$ )	WM	$5.7 \pm 0.1$	$5.6 \pm 0.1$	$5.7 \pm 0.1$	$6.8 \pm 0.3$
	GM	$6.2 \pm 0.1$	$5.4 \pm 0.4$	$6.1 \pm 0.1$	$7.3 \pm 0.3$

by a medical practitioner on a CPU. On an Apple M2 Pro CPU, *full matching* takes days and is prone to many memory issues that needs to be solve, while the HD-GMM and HD-STM variants require only 4h30.

**8. Conclusion.** In this work, we combine robust latent variable representations, clustering and incremental learning to propose a new tractable and accurate way to represent and handle large volumes of potentially heterogeneous high-dimensional data. To our knowledge, this combination and its use as a data compression strategy is novel. The clustering structure of HD-MED allows to address data heterogeneity, for a greater dimensionality reduction without increasing information loss. Incremental learning allows to handle large volumes, resulting in significant reduction in both computational costs and information losses. In terms of implementation, the procedure is flexible and easy to interpret. It depends on two main hyperparameters, the number of clusters  $K$  and the vector of reduced dimensions  $d$ , that can be set using conventional model selection criteria or chosen by the user to meet resource constraints. Typically increasing  $K$  to reduce the size of clusters or decreasing  $d$  to increase compression.

As an illustration, we focus on a Magnetic Resonance Fingerprinting (MRF) application. The proposed method drastically reduces the computational time required on standard hardware, such as CPUs, which are commonly used in clinical environments. Beyond this crucial processing time gain, dimensionality reduction has also an interesting impact on patient data acquisition time, as it helps mitigating the effect of noise in fast-acquired *in vivo* signals. These reductions are thus not only a technical achievement but also a significant step forward in making advanced MRF techniques more accessible and usable in everyday clinical practice.

In addition, although illustrated with a simple matching procedure, which is the current reference in the target MRF application, our procedure can be coupled with other simulation-based inference (SBI) approaches (Cranmer, Brehmer and Louppe, 2020), such as approximate Bayesian computation (ABC) or other neural techniques (Barrier et al., 2024). Like MRF, simulation-based inference has to face two opposite requirements, which are the need for large and high-dimensional simulated data sets to accurately capture information on the physics under study and the issue of handling such large volumes due to computational resources constraints. ABC uses distances between observations and simulations, and inference is based on these distances. A reduced data representation can be typically coupled with ABC techniques proposing automatic summary statistics selection such as in Forbes et al. (2022); Fearnhead and Prangle (2012). It can also be used with SBI neural approaches, e.g. Häggström et al. (2024), when coupled with a regression model for high-dimensional data, see Algorithm 1 in supplementary material. Compared to matching and standard ABC, regression-based procedures are more amortized, which further reduces the reconstruction time. In terms of MRF and other medical imaging applications, future work will thus involve combining HD-MED with a regression model or a neural network, such as in Deleforge, Forbes and Horaud (2015) or Golbabaei et al. (2019).

At last, none of the previously mentioned methods, currently make use of spatial information at the voxel level, to perform map reconstruction. The idea would be to exploit voxel proximity to either improve or accelerate parameter prediction. A relatively easy extension of mixture-based methods is to account for spatial information by adding a Markov dependence on the clusters, as done for instance in Deleforge et al. (2015).

**9. Financial disclosure.** G. Oudoumanessah was financially supported by the AURA region, and was granted access to the HPC resources of IDRIS under the allocation 2022-AD011013867 made by GENCI.

The MRI facility IRMaGe is partly funded by the French program “Investissement d’avenir” run by the French National Research Agency, grant “Infrastructure d’avenir en Biologie et Santé” [ANR-11-INBS-0006]. The project is supported by the French National Research Agency [ANR-20-CE19-0030 MRFUSE].

## REFERENCES

- ARCHAMBEAU, C. and BACH, F. R. (2008). Sparse probabilistic projections. In *21st International Conference on Neural Information Processing Systems* 73–80.
- ARCHAMBEAU, C., DELANNAY, N. and VERLEYSEN, M. (2006). Robust probabilistic projections. In *Proceedings of the 23rd International conference on machine learning* 33–40.
- ARCHAMBEAU, C., DELANNAY, N. and VERLEYSEN, M. (2008). Mixtures of robust probabilistic principal component analyzers. *Neurocomputing* **71** 1274–1282.
- BAEK, J. and MCLACHLAN, G. J. (2011). Mixtures of common t-factor analyzers for clustering high-dimensional microarray data. *Bioinformatics* **27** 1269–1276.
- BALZANO, L., CHI, Y. and LU, Y. M. (2018). Streaming PCA and Subspace Tracking: The Missing Data Case. In *Proceedings of the IEEE* **106** 1293–1310.

- BARRIER, A., COUDERT, T., DELPHIN, A., LEMASSON, B. and CHRISTEN, T. (2024). MARVEL: MR Fingerprinting with Additional microVascular Estimates using bidirectional LSTMs. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* 259–269. Springer.
- BELLAS, A., BOUVEYRON, C., COTTRELL, M. and LACAILLE, J. (2013). Model-based Clustering of High-dimensional Data Streams with Online Mixture of Probabilistic PCA. *Advances in Data Analysis and Classification* **7** 281–300.
- BIPIN MEHTA, B., COPPO, S., FRANCES McGIVNEY, D., IAN HAMILTON, J., CHEN, Y., JIANG, Y., MA, D., SEIBERLICH, N., GULANI, V. and ALAN GRISWOLD, M. (2019). Magnetic resonance fingerprinting: a technical review. *Magnetic resonance in medicine* **81** 25–46.
- BJØRNERUD, A. and EMBLEM, K. E. (2010). A fully automated method for quantitative cerebral hemodynamic analysis using DSC-MRI. *Journal of Cerebral Blood Flow & Metabolism* **30** 1066–1078.
- BORKAR, V. S. (2009). *Stochastic approximation: a dynamical systems viewpoint* **48**. Springer, Gurgaon.
- BOUVEYRON, C. and BRUNET-SAUMARD, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis* **71** 52–78.
- BOUVEYRON, C., GIRARD, S. and SCHMID, C. (2007). High-dimensional data clustering. *Computational statistics & data analysis* **52** 502–519.
- BOUX, F., FORBES, F., ARBEL, J., LEMASSON, B. and BARBIER, E. L. (2021). Bayesian inverse regression for vascular magnetic resonance fingerprinting. *IEEE Transactions on Medical Imaging* **40** 1827–1837.
- BRADBURY, J., FROSTIG, R., HAWKINS, P., JOHNSON, M., LEARY, C., MACLAURIN, D., NECULA, G., PASZKE, A., VANDERPLAS, J., WANDERMAN-MILNE, S. and ZHANG, Q. (2018). JAX: composable transformations of Python+NumPy programs. <http://github.com/google/jax>.
- CABINI, R. F., BARZAGHI, L., CICOLARI, D., AROSIO, P., CARRAZZA, S., FIGINI, S., FILIBIAN, M., GAZZANO, A., KRAUSE, R., MARIANI, M. et al. (2024). Fast deep learning reconstruction techniques for preclinical magnetic resonance fingerprinting. *NMR in Biomedicine* **37** e5028.
- CAMBANIS, S., HUANG, S. and SIMONS, G. (1981). On the theory of elliptically contoured distributions. *Journal of Multivariate Analysis* **11** 368–385.
- CAPPÉ, O. and MOULINES, E. (2009). On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **71** 593–613.
- CAULEY, S. F., SETSOMPOP, K., MA, D., JIANG, Y., YE, H., ADALSTEINSSON, E., GRISWOLD, M. A. and WALD, L. L. (2015). Fast group matching for MR fingerprinting reconstruction. *Magnetic resonance in medicine* **74** 523–528.
- CHRISTEN, T., BOLAR, D. S. and ZAHARCHUK, G. (2013). Imaging brain oxygenation with MRI using blood oxygenation approaches: methods, validation, and clinical applications. *American journal of neuroradiology* **34** 1113–1123.
- CHRISTEN, T., PANNETIER, N., NI, W. W., QIU, D., MOSELEY, M. E., SCHUFF, N. and ZAHARCHUK, G. (2014). MR vascular fingerprinting: A new approach to compute cerebral blood volume, mean vessel radius, and oxygenation maps in the human brain. *Neuroimage* **89** 262–270.
- COHEN, O., ZHU, B. and ROSEN, M. S. (2018). MR fingerprinting deep reconstruction network (DRONE). *Magnetic resonance in medicine* **80** 885–894.
- COIFMAN, R. R. and LAFON, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis* **21** 5–30.
- COUDERT, T., DELPHIN, A., BARRIER, A., LEGRIS, L., WARNKING, J. M., LAMALLE, L., DONEVA, M., LEMASSON, B., BARBIER, E. L. and CHRISTEN, T. (2024). Relaxometry and contrast-free cerebral microvascular quantification using balanced Steady-State Free Precession MR Fingerprinting. <https://arxiv.org/abs/2411.03414>.
- CRANMER, K., BREHMER, J. and LOUPPE, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences* **117** 201912789.
- D'ASPREMONT, A., ELGHAOUI, L., JORDAN, M. I. and LANCKRIET, G. (2007). A direct formulation for sparse PCA using semidefinite programming. *SIAM Review* **49** 434–48.
- DELEFORGE, A., FORBES, F. and HORAUD, R. (2015). High-Dimensional Regression with Gaussian Mixtures and Partially-Latent Response Variables. *Statistics & Computing* **25** 893–911.
- DELEFORGE, A., FORBES, F., BA, S. O. and HORAUD, R. (2015). Hyper-Spectral Image Analysis With Partially Latent Regression and Spatial Markov Dependencies. *IEEE J. Sel. Top. Signal Process.* **9** 1037–1048.
- DELPHIN, A., COUDERT, T., FAN, A., MOSELEY, M. E., ZAHARCHUK, G. and CHRISTEN, T. (2023). MR Vascular Fingerprinting with 3D realistic blood vessel structures and machine learning to assess oxygenation changes in human volunteers. In *2023 ISMRM & ISMRT Annual Meeting & Exhibition*.
- DEMPSTER, A. P., LAIRD, N. M. and RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)* **39** 1–22.

- FEARNHEAD, P. and PRANGLE, D. (2012). Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **74** 419–474.
- FORBES, F., NGUYEN, H. D., NGUYEN, T. and ARBEL, J. (2022). Summary statistics and discrepancy measures for approximate Bayesian computation via surrogate posteriors. *Statistics and Computing* **32** 85.
- FORT, G., MOULINES, E. and WAI, H.-T. (2020). A stochastic path-integrated differential estimator expectation maximization algorithm. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*.
- GHAHRAMANI, Z. and HINTON, G. E. (1997). The EM algorithm for mixtures of factor analyzers Technical Report, University of Toronto.
- GILMAN, K., HONG, D., FESSLER, J. A. and BALZANO, L. (2023). Streaming Probabilistic PCA for Missing Data with Heteroscedastic Noise. *arXiv preprint arXiv:2310.06277*.
- GIRAUD, C. (2014). *Introduction to high-dimensional statistics*. Chapman and Hall/CRC.
- GOLBABAEI, M., CHEN, D., GÓMEZ, P. A., MENZEL, M. I. and DAVIES, M. E. (2019). Geometry of deep learning for magnetic resonance fingerprinting. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 7825–7829. IEEE.
- GOMEZ, E., GOMEZ-VILLEGAS, M. and MARÍN, J. (2008). Multivariate Exponential Power Distributions as Mixtures of Normal Distributions with Bayesian Applications. *Communications in Statistics-theory and Methods* **37**.
- GÓMEZ-SÁNCHEZ-MANZANO, E., GÓMEZ-VILLEGAS, M. and MARÍN, J. (2006). Sequences of elliptical distributions and mixtures of normal distributions. *Journal of multivariate analysis* **97** 295–310.
- GU, Y., WANG, C. Y., ANDERSON, C. E., LIU, Y., HU, H., JOHANSEN, M. L., MA, D., JIANG, Y., RAMOS-ESTEBANEZ, C., BRADY-KALNAY, S. et al. (2018). Fast magnetic resonance fingerprinting for dynamic contrast-enhanced studies in mice. *Magnetic resonance in medicine* **80** 2681–2690.
- GU, Y., PAN, Y., FANG, Z., MA, L., ZHU, Y., ANDROJNA, C., ZHONG, K., YU, X. and SHEN, D. (2024). Deep learning-assisted preclinical MR fingerprinting for sub-millimeter T1 and T2 mapping of entire macaque brain. *Magnetic Resonance in Medicine* **91** 1149–1164.
- HÄGGSTRÖM, H., RODRIGUES, P. L., OUDOUMANESSAH, G., FORBES, F. and PICCHINI, U. (2024). Fast, accurate and lightweight sequential simulation-based inference using Gaussian locally linear mappings. *arXiv preprint arXiv:2403.07454*.
- HALKO, N., MARTINSSON, P.-G. and TROPP, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review* **53** 217–288.
- HARGREAVES, B. Bloch Equation Simulator. <http://www-mrsrl.stanford.edu/brian/blochsim/>.
- HONG, D., GILMAN, K., BALZANO, L. and FESSLER, J. A. (2021). HePPCAT: Probabilistic PCA for data with heteroscedastic noise. *IEEE Transactions on Signal Processing* **69** 4819–4834.
- HONG, D., YANG, F., FESSLER, J. A. and BALZANO, L. (2023). Optimally Weighted PCA for High-Dimensional Heteroscedastic Data. *SIAM Journal on Mathematics of Data Science* **5** 222–250.
- JOLLIFFE, I. T. and CADIMA, J. (2016). Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences* **374** 20150202.
- KARIMI, B., MIASOJEDOW, B., MOULINES, E. and WAI, H.-T. (2019a). Non-asymptotic analysis of biased stochastic approximation scheme. *Proc. Mach. Learn. Res.* **99** 1–31.
- KARIMI, B., WAI, H.-T., MOULINES, R. and LAVIELLE, M. (2019b). On the global convergence of (fast) incremental expectation maximization methods. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*.
- KELKER, D. (1970). Distribution theory of spherical distributions and a location-scale parameter generalization. *Sankhya: The Indian Journal of Statistics, Series A* **32** 419–430.
- KÖRZDÖRFER, G., PFEUFFER, J., KLUGE, T., GEBHARDT, M., HENSEL, B., MEYER, C. H. and NITTKA, M. (2019). Effect of spiral undersampling patterns on FISP MRF parameter maps. *Magnetic resonance imaging* **62** 174–180.
- KOTZ, S. and NADARAJAH, S. (2004). *Multivariate t-distributions and their applications*. Cambridge University Press.
- KUHN, E., MATIAS, C. and REBAFKA, T. (2020). Properties of the stochastic approximation EM algorithm with mini-batch sampling. *Stat. Comp.* **30** 1725–1739.
- LI, P. and HU, Y. (2023). Learned Tensor Low-CP-Rank and Bloch Response Manifold Priors for Non-Cartesian MRF Reconstruction. *IEEE Transactions on Medical Imaging*.
- LI, P. and HU, Y. (2024). Deep magnetic resonance fingerprinting based on Local and Global Vision Transformer. *Medical Image Analysis* **95** 103198.
- MA, D., GULANI, V., SEIBERLICH, N., LIU, K., SUNSHINE, J. L., DUERK, J. L. and GRISWOLD, M. A. (2013). Magnetic resonance fingerprinting. *Nature* **495** 187–192.

- MAIRE, F., MOULINES, E. and LEFEBVRE, S. (2017). Online EM for functional data. *Computational Statistics and Data Analysis* **111** 27–47.
- MCGIVNEY, D. F., PIERRE, E., MA, D., JIANG, Y., SAYBASILI, H., GULANI, V. and GRISWOLD, M. A. (2014). SVD compression for magnetic resonance fingerprinting in the time domain. *IEEE transactions on medical imaging* **33** 2311–2322.
- MCGIVNEY, D. F., BOYACIOĞLU, R., JIANG, Y., POORMAN, M. E., SEIBERLICH, N., GULANI, V., KEENAN, K. E., GRISWOLD, M. A. and MA, D. (2020). Magnetic resonance fingerprinting review part 2: Technique and directions. *Journal of Magnetic Resonance Imaging* **51** 993–1007.
- MCLACHLAN, G. J. and KRISHNAN, T. (2008). *The EM Algorithm and Extensions*. Wiley 2nd edition.
- MCLACHLAN, G. J., PEEL, D. and BEAN, R. (2003). Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics & Data Analysis* **41** 379–388.
- MENG, X.-L. and RUBIN, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* **80** 267–278.
- MONGA, A., SINGH, D., DE MOURA, H. L., ZHANG, X., ZIBETTI, M. V. and REGATTE, R. R. (2024). Emerging Trends in Magnetic Resonance Fingerprinting for Quantitative Biomedical Imaging Applications: A Review. *Bioengineering* **11** 236.
- NGUYEN, H. D., FORBES, F. and MCLACHLAN, G. J. (2020). Mini-batch learning of exponential family finite mixture models. *Statistics and Computing* **30** 731–748.
- NGUYEN, H. and FORBES, F. (2022). Global implicit function theorems and the online expectation–maximisation algorithm. *Australian & New Zealand Journal of Statistics* **64**.
- OUDOUMANESSAH, G., COUDERT, T., MEYER, L., DELPHIN, A., DOJAT, M., LARTIZIEN, C. and FORBES, F. (2024). Cluster globally, Reduce locally: Scalable efficient dictionary compression for magnetic resonance fingerprinting. <https://arxiv.org/abs/2411.07415>.
- POORMAN, M. E., MARTIN, M. N., MA, D., MCGIVNEY, D. F., GULANI, V., GRISWOLD, M. A. and KEENAN, K. E. (2020). Magnetic resonance fingerprinting Part 1: Potential uses, current challenges, and recommendations. *Journal of Magnetic Resonance Imaging* **51** 675–692.
- RONNEBERGER, O., FISCHER, P. and BROX, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18 234–241. Springer.
- SATOPAA, V., ALBRECHT, J., IRWIN, D. and RAGHAVAN, B. (2011). Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops* 166–171. IEEE.
- SCHOTT, J. R. (2016). *Matrix analysis for statistics*. Wiley probability & statistics, Hoboken.
- SCHWARZ, G. (1978). Estimating the dimension of a model. *The Annals of Statistics* 461–464.
- SOYAK, R., NAVRUZ, E., ERSOY, E. O., CRUZ, G., PRIETO, C., KING, A. P., UNAY, D. and OKSUZ, I. (2021). Channel attention networks for robust MR fingerprint matching. *IEEE Transactions on Biomedical Engineering* **69** 1398–1405.
- TIPPAREDDY, C., ZHAO, W., SUNSHINE, J. L., GRISWOLD, M., MA, D. and BADVE, C. (2021). Magnetic resonance fingerprinting: an overview. *European Journal of Nuclear Medicine and Molecular Imaging* **48** 4189–4200.
- TIPPING, M. E. and BISHOP, C. M. (1999a). Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **61** 611–622.
- TIPPING, M. E. and BISHOP, C. M. (1999b). Mixtures of probabilistic principal component analyzers. *Neural computation* **11** 443–482.
- TIPPING, M. E. and BISHOP, C. M. (1999c). Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **61** 611–622.
- TIPPING, M. E. and BISHOP, C. M. (1999d). Mixtures of probabilistic principal component analyzers. *Neural computation* **11** 443–482.
- ULLAH, I., HASSAN, A. M., SAAD, R. M. and OMER, H. (2023). GPU accelerated grouped magnetic resonance fingerprinting using clustering techniques. *Magnetic Resonance Imaging* **97** 13–23.
- VASWANI, A., SHAZEE, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł. and POLOSKHIN, I. (2017). Attention is all you need. *Advances in neural information processing systems* **30**.
- WANG, Z., BOVIK, A. C., SHEIKH, H. R. and SIMONCELLI, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* **13** 600–612.
- WANG, Z., ZHANG, J., CUI, D., XIE, J., LYU, M., HUI, E. S. and WU, E. X. (2019a). Magnetic Resonance Fingerprinting Using a Fast Dictionary Searching Algorithm: MRF-ZOOM. *IEEE trans. bio-medical engineering* **66** 1526–1535.
- WANG, C. Y., COPPO, S., MEHTA, B. B., SEIBERLICH, N., YU, X. and GRISWOLD, M. A. (2019b). Magnetic resonance fingerprinting with quadratic RF phase for measurement of T2\* simultaneously with δf, T1, and T2. *Magnetic resonance in medicine* **81** 1849–1862.

- WANSAPURA, J. P., HOLLAND, S. K., DUNN, R. S. and BALL JR, W. S. (1999). NMR relaxation times in the human brain at 3.0 tesla. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine* **9** 531–538.
- XU, A. S., BALZANO, L. and FESSLER, J. A. (2023). HeMPPCAT: mixtures of probabilistic principal component analysers for data with heteroscedastic noise. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 1–5.
- YANG, M., MA, D., JIANG, Y., HAMILTON, J., SEIBERLICH, N., GRISWOLD, M. A. and McGIVNEY, D. (2018). Low rank approximation methods for MR fingerprinting with large scale dictionaries. *Magnetic resonance in medicine* **79** 2392–2400.
- YE, H., CAULEY, S. F., GAGOSKI, B., BILGIC, B., MA, D., JIANG, Y., DU, Y. P., GRISWOLD, M. A., WALD, L. L. and SETSOMPOP, K. (2017). Simultaneous multislice magnetic resonance fingerprinting (SMS-MRF) with direct-spiral slice-GRAPPA (ds-SG) reconstruction. *Magnetic resonance in medicine* **77** 1966–1974.
- ZOU, H., HASTIE, T. and TIBSHIRANI, R. (2006). Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics* **15** 265–286.

# CLUSTER GLOBALLY, REDUCE LOCALLY: SCALABLE EFFICIENT DICTIONARY COMPRESSION FOR MAGNETIC RESONANCE FINGERPRINTING

*Geoffroy Oudoumanessah<sup>1,2,3</sup> Thomas Coudert<sup>1</sup> Luc Meyer<sup>2</sup>  
Aurelien Delphin<sup>1</sup> Thomas Christen<sup>1</sup> Michel Dojat<sup>1,2</sup> Carole Lartizien<sup>3</sup> Florence Forbes<sup>2</sup>*

<sup>1</sup> Univ. Grenoble Alpes, Inserm U1216, CHU Grenoble Alpes, Institut des Neurosciences, France

<sup>2</sup> Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France

<sup>3</sup> Univ. Lyon, CNRS UMR 5220, Inserm U1294, INSA Lyon, UCBL, CREATIS, France

## ABSTRACT

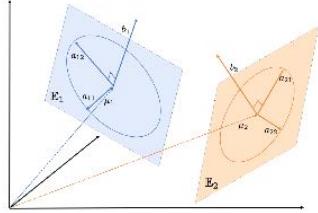
With the rapid advancements in medical data acquisition and production, increasingly richer representations exist to characterize medical information. However, such large-scale data do not usually meet computing resource constraints or algorithmic complexity, and can only be processed after compression or reduction, at the potential loss of information. In this work, we consider specific Gaussian mixture models (HD-GMM), tailored to deal with high dimensional data and to limit information loss by providing component-specific lower dimensional representations. We also design an incremental algorithm to compute such representations for large data sets, overcoming hardware limitations of standard methods. Our procedure is illustrated in a magnetic resonance fingerprinting study, where it achieves a 97% dictionary compression for faster and more accurate map reconstructions.

**Index Terms**— Massive data, Incremental Learning, Compression, Mixture of probabilistic PCA.

## 1. INTRODUCTION

The development of medical imaging has led to data acquisition and production at much larger scales, for an increasing benefit to medical decision making. The exploitation of such large-scale data poses a number of challenges. A first challenge comes from the data processing scalability. Large-scale data may not be easily stored into memory or may be collected in a distributed manner from several sources, *e.g.* hospitals. Such a limited or distributed storage may limit the use of traditional methods, which load all the data into memory before running some optimization procedure. A second challenge comes from the data dimensionality. Across a wide range of medical applications, measured observations are high dimensional, *e.g.* magnetic resonance (MR) fingerprints [1], functional MR signals, neural network latent representations of images [2], *etc.* A typical difficulty is that the number of parameters for a model of such data can then easily exceed the number of observations, leading to estimation issues. In such high-dimensional settings, it is often possible

to reduce the number of parameters by assuming that most of the information in the data can be captured and represented in a much lower dimensional subspace. Classical techniques include principal component analysis (PCA), probabilistic principal component analysis (PPCA) [3], factor analysers (FA), and newer methods such as diffusion maps [4]. More flexible approaches are also based on mixtures of the previous ones, such as mixtures of factor analysers (MFA) [5] and mixtures of PPCA (MPPCA) [6]. Another mixture approach is called HD-GMM in [7] for High Dimensional Gaussian Mixture Models. It encompasses many forms of MFA and MPPCA and generalises them, see also [7] for a review on high dimensional clustering via mixtures. However, most of these methods are designed for batch data and are thus sensitive to hardware limits such as memory, which restricts the amount of data they can process or the type of medical devices on which they can be usefully embedded. As a simple solution, most implementations downsample data sets before processing, potentially loosing useful information. Another approach is to design incremental, also referred to as online, variants handling data sequentially in smaller groups. A number of incremental approaches exist for dimension reduction techniques, see the recent SHASTA-PCA [8] and [9] for a review. To our knowledge, much fewer solutions exist for mixtures. Estimation of such models is generally based on maximum likelihood estimation via the Expectation-Maximization (EM) algorithm [10]. We can mention a preliminary attempt for an incremental MPPCA based on heuristic approximations of the EM steps [11]. In this work, considering data both large in size and dimension, our proposal is twofold. Building on HD-GMM, originally developed for high dimensional clustering and density estimation, we show how they can be used to compress high dimensional data into several reduced size data subsets. We then derive a new incremental algorithm, based on a principled EM framework, to learn such a model from very large data sets. We demonstrate the effectiveness of our approach on a MR fingerprinting (MRF) study, allowing to go far beyond the simulations resolution and size used in current implementations and to reconstruct a larger



**Fig. 1:** HD-GMM schematic illustration,  $M=3, d=2, K=2$ .

number of MR parameter maps with an improved accuracy.

## 2. DIVIDE & CONQUER REDUCTION OF LARGE DATA VOLUMES

**Identifying group-wise subspaces.** HD-GMM assume that the observations are *i.i.d.* realizations of a random variable  $\mathbf{y}$  which follows a Gaussian mixture model with  $K$  components,

$$p(\mathbf{y}; \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}_M(\mathbf{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (1)$$

where  $\mathcal{N}_M$  denotes the  $M$ -dimensional Gaussian distribution and  $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$  are respectively the  $k$ th component mean, covariance matrix and weight. An efficient parameters reduction can be obtained by using the eigendecomposition  $\boldsymbol{\Sigma}_k = \mathbf{D}_k \mathbf{A}_k \mathbf{D}_k^T$ , where  $\mathbf{D}_k$  contains the eigenvectors  $\{\mathbf{d}_1, \dots, \mathbf{d}_M\}$  of  $\boldsymbol{\Sigma}_k$ , and  $\mathbf{A}_k$  is a diagonal matrix with its eigenvalues. In HD-GMM, each  $\mathbf{A}_k$  consists of only  $d_k + 1$  different eigenvalues  $\mathbf{A}_k = \text{diag}(a_{k1}, \dots, a_{kd_k}, b_k, \dots, b_k)$ , with  $a_{kj} > b_k$ , for  $j = 1 : d_k$ , and where  $d_k \in \{1, \dots, M-1\}$  is *a priori* unknown but fixed in our work to a user decided dimension  $d$ . When  $b_k$  is negligible, this parameterization means that a group-specific subspace  $\mathbb{E}_k$ , parameterised by the  $d$  eigenvectors associated to the first  $d$  eigenvalues  $\{a_{k1}, \dots, a_{kd}\}$ , captures the main cluster shape (see Figure 1 for an illustration). The model parameters are then  $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k, k = 1 : K\}$  with  $\boldsymbol{\theta}_k = \{\pi_k, \boldsymbol{\mu}_k, \mathbf{A}_k, \mathbf{D}_k\}$ . They can be estimated using an EM algorithm. When  $d \ll M$ , a significant computation gain can be achieved. Let  $\tilde{\mathbf{D}}_k$  consist of the  $d$  first columns of  $\mathbf{D}_k$  supplemented by  $(M-d)$  zero columns and  $\bar{\mathbf{D}}_k = (\mathbf{D}_k - \tilde{\mathbf{D}}_k)$ . Then,  $P_k(\mathbf{y}) = \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_k^T (\mathbf{y} - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k$  and  $P_k^\perp(\mathbf{y}) = \bar{\mathbf{D}}_k \bar{\mathbf{D}}_k^T (\mathbf{y} - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k$  are the projections of  $\mathbf{y}$  on  $\mathbb{E}_k$  and its orthogonal space  $\mathbb{E}_k^\perp$ . The main EM computations involve quadratic quantities  $(\mathbf{y} - \boldsymbol{\mu}_k) \boldsymbol{\Sigma}_k^{-1} (\mathbf{y} - \boldsymbol{\mu}_k)$  which can be equivalently written as,

$$\|\boldsymbol{\mu}_k - P_k(\mathbf{y})\|_{\tilde{\boldsymbol{\Sigma}}_k^{-1}}^2 + \frac{1}{b_k} \|\mathbf{y} - P_k(\mathbf{y})\|^2, \quad (2)$$

where  $\|\cdot\|_{\tilde{\boldsymbol{\Sigma}}_k^{-1}}$  is the norm defined by  $\|\mathbf{y}\|_{\tilde{\boldsymbol{\Sigma}}_k^{-1}}^2 = \mathbf{y}^T \tilde{\boldsymbol{\Sigma}}_k^{-1} \mathbf{y}$  with  $\tilde{\boldsymbol{\Sigma}}_k^{-1} = \tilde{\mathbf{D}}_k \mathbf{A}_k^{-1} \tilde{\mathbf{D}}_k^T$ . Expression (2) uses the definitions of  $P_k$ ,  $P_k^\perp$  and  $\|\boldsymbol{\mu}_k - P_k^\perp(\mathbf{y})\|^2 = \|\mathbf{y} - P_k(\mathbf{y})\|^2$ . The

SNR (dB)	$d = 8$	$d = 10$	$d = 15$
SVD [12]	-	0.35	0.33
	15	0.40	0.35
	5	0.44	0.43
HD-GMM	-	0.031	0.030
	15	0.032	0.031
	5	0.16	0.15
Size (Go)	<b>0.30</b>	0.38	0.57

**Table 1:** Compressed dictionary MAEs and sizes. The lower the better, best values in bold.

gain comes from the fact that (2) does not depend on  $P_k^\perp$  and thus does not require the computation of the  $(M-d)$  latest columns of  $\mathbf{D}_k$ , the eigenvectors associated to the smallest eigenvalues. Similarly, determinants can be efficiently computed as  $\log(|\boldsymbol{\Sigma}_k|) = (\sum_{j=1}^d \log(a_{kj})) + (M-d)\log(b_k)$ . This parameterization allows to handle high dimensional data in a computationally efficient way. However, it does not provide an actual lower dimensional representation of the data. While such a reduced-dimensional representation may often not be needed, it may be crucial to deal with hardware or software limitations. Originally, HD-GMM have not been designed for this situation, but we describe next how they can be further exploited as a dimension reduction technique.

**Cluster-wise dimension reduction.** As clustering models, for any possible observation  $\mathbf{y}$ , HD-GMM provide a probability  $r_k(\mathbf{y})$  that  $\mathbf{y}$  is assigned to cluster  $k$  for each  $k = 1 : K$ . Denote  $\tilde{\mathbf{D}}_k^*$  the  $M \times d$  matrix built with the  $d$  first columns of  $\mathbf{D}_k$ . A reduced-dimensionality representation  $\hat{\mathbf{y}}_k$  of  $\mathbf{y}$  can be obtained, for each of the  $K$  different subspaces, by computing the scalar products of a centered  $\mathbf{y}$  with the columns of  $\tilde{\mathbf{D}}_k^*$ . It comes  $\hat{\mathbf{y}}_k = S_k(\mathbf{y}) = \tilde{\mathbf{D}}_k^{*T} (\mathbf{y} - \boldsymbol{\mu}_k)$ , while its reconstruction  $\tilde{\mathbf{y}}_k$  in the original space is given by  $\tilde{\mathbf{y}}_k = \tilde{\mathbf{D}}_k^* \hat{\mathbf{y}}_k + \boldsymbol{\mu}_k$ . In practice, it is reasonable to use as a reduced-dimensionality representation of  $\mathbf{y}$  only the one corresponding to the most probable group  $k$ , *i.e.* with the highest  $r_k(\mathbf{y})$ . In this setting, HD-GMM acts as a divide-and-conquer paradigm by clustering the data into  $K$  clusters and allowing cluster-specific data reduction. The divide step allows a much more effective reduction than if a single subspace is considered for the whole data set. In the conquer step, little information is lost, as for any new observation  $\mathbf{y}$ , cluster assignment probabilities  $r_k(\mathbf{y})$  can be straightforwardly computed to decide on the best reduced representation to be used. However, for subsequent processing, it is important to keep track of clustering information for each observation. The reduced representations cannot be pooled back altogether, as they would be likely to become impossible to distinguish across clusters.

**Incremental learning for large data volumes.** In practice, most approaches lie on optimization procedures requiring

all data to be loaded in a single batch. Batch sizes are then limited by resource constraints, so that very large data sets need either to be downsampled or to be handled in an incremental manner, *i.e.* with smaller data subsets processed sequentially. Incremental versions of EM exist and can be adapted to our setting. As an archetype of such algorithms, we consider the online EM of [13] which belongs to the family of stochastic approximation algorithms. We refer to [13] for details on the main assumptions required and the online EM iteration. When applied to mixture models, it can be shown [14], that it is enough to check that each mixture component has an exponential family form. For the HD-GMM parameterization, omitting the cluster index  $k$ , the exponential form of  $\mathcal{N}_M(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$  is given by  $h(\mathbf{y}) \exp(s(\mathbf{y})^T \boldsymbol{\phi}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) - \psi(\boldsymbol{\mu}, \boldsymbol{\Sigma}))$  where  $s$  and  $\boldsymbol{\phi}$  are vectors with respective elements ( $\text{vec}(\cdot)$  is the vectorization operator of a matrix)  $[\mathbf{y}, \text{vec}(\mathbf{y}\mathbf{y}^T), \mathbf{y}^T \mathbf{y}]$  and

$$\left[ \sum_{j=1}^d \left( \frac{1}{a_j} - \frac{1}{b} \right) \mathbf{d}_j \mathbf{d}_j^T \boldsymbol{\mu} + \frac{1}{b} \boldsymbol{\mu}, \frac{1}{2} \sum_{j=1}^d \left( \frac{1}{b} - \frac{1}{a_j} \right) \text{vec}(\mathbf{d}_j \mathbf{d}_j^T), -\frac{1}{2b} \right]$$

and  $\psi(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  is equal to

$$\frac{1}{2} \sum_{j=1}^d \left[ \left( \frac{1}{a_j} - \frac{1}{b} \right) \boldsymbol{\mu}^T \mathbf{d}_j \mathbf{d}_j^T \boldsymbol{\mu} + \log(a_j) \right] + \frac{1}{2b} \boldsymbol{\mu}^T \boldsymbol{\mu} + \frac{(M-d)}{2} \log(b).$$

The form of  $h(\cdot)$  is irrelevant for the computations. It follows an online EM algorithm which is closed-form except for the update of  $\tilde{\mathbf{D}}^*$ , which is estimated using a Riemannian optimization framework in the setting where  $M \gg d$  [15].

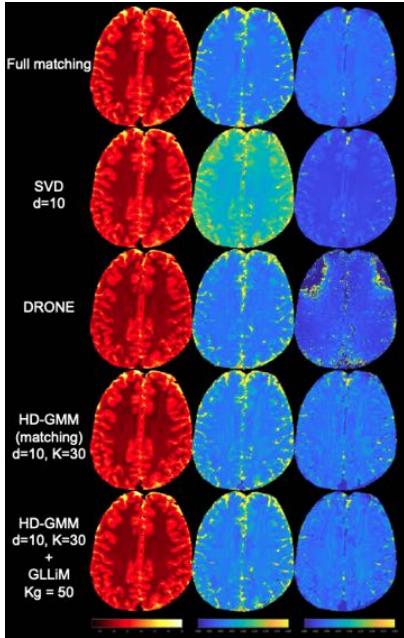
### 3. LARGE SCALE MR FINGERPRINTING (MRF)

MRF [1] allows the simultaneous acquisition and reconstruction of multiple tissue properties maps, see [16, 17] for recent reviews. In the original *matching* approach, maps reconstruction is based on the search for the best match between an observed signal and a dictionary of simulated signals (fingerprints). As an alternative, learning approaches have been studied via various neural network (NN) architectures, but improvement has been demonstrated mainly for standard  $T_1$  and  $T_2$  relaxometry parameters estimation and only up to 3 parameters in the dictionary, see *e.g.* [18] or Table 2 in [17]. To now, none of these approaches has shown real scalability properties with respect to the number of parameters to be reconstructed [18]. The main issue is that the size of the simulated dictionary increases exponentially with the number of parameters to be reconstructed. In this context, [19] showed that a dictionary-based Bayesian learning approach was more accurate and less demanding, in terms of dictionary size, than conventional dictionary matching and some NN solutions. We build on this work by using online HD-GMM estimation to handle an extensive high-resolution dictionary. This offers a gain in data storage, when traditional dictionary matching is

used, but also in accuracy and speed of map reconstructions, when Bayesian or NN methods are considered. In a matching approach, our proposal is similar in spirit to [20], which distribute the matching cost into smaller matching tasks. However [20] does not adapt straightforwardly to learning-based or NN approaches. For this illustration, we use the method introduced in [21], to simulate a dictionary for various ranges of relaxometry parameters  $T_1$ ,  $T_2$ ,  $T_2^*$ , and magnetic field parameters  $\delta f$  and  $B_1$ . In particular, transverse relaxation time  $T_2^*$  is a MR tissue property that provides insight into underlying tissue physiology and pathology, making this MR parameter a widely used biomarker of several clinical diseases. The dictionary is made of 4.750M signals in dimension 260, for a storage cost of more than 20Gb. Gaussian noise is often added to signals using different SNR values [18]. Table 1 first shows the efficiency of different dictionary compression strategies. Mean absolute error (MAE) values are computed, for different noise levels, between the original signals and their denoised reconstructions after compression. In all cases, HD-GMM show much lower compression losses than the reference SVD method [12], achieving better reconstructions even with a dimension reduced to  $d = 8$ , about half smaller than SVD ( $d = 15$ ). For HD-GMM there are two hyperparameters namely the number of components  $K$  and the reduced dimension  $d$ . Both can be chosen using a Bayesian information criterion (BIC), which in our case gives  $K = 30$  and  $d = 10$ . For initializing our EM algorithm, we use the heuristic mentioned in [22] Section 3.5, which provides good performance for our approach too. Parameters maps reconstruction is then illustrated on  $T_1$ ,  $T_2$ ,  $T_2^*$  in Figure 2. We first perform a standard matching, referred to as *full matching*, based on inner products between *in vivo* acquired MR signals and simulated signals from the original high dimensional dictionary (Figure 2 line 1). The *in vivo* acquisition was performed on one healthy volunteer with a Philips 3T Achieva dStream MRI at the IRMaGe facility. As ground truth maps are not available, full matching maps often serve as reference maps, although this has obvious limits in terms of hardware and robustness to noise. Matching results are then shown for reduced dictionary representations, using either SVD with  $d = 10$  (Figure 2 line 2) or HD-GMM with  $d = 10$  and  $K = 30$ . As learning-based alternatives, maps obtained with DRONE [23], a 4 layers fully connected NN, and with a combination of the HD-GMM reduction and the Bayesian learning model referred to as GLLiM [19], are also shown (Figure 2 line 3 and 5 respectively). The GLLiM model corresponds to a regression model for which a number of Gaussian components needs to be chosen and is set to  $K_g = 50$ . Figure 2 clearly shows that both SVD and DRONE provide unsatisfying maps, *e.g.* for  $T_2$  and  $T_2^*$ . The HD-GMM approach is visually similar to full matching. Its combination with GLLiM provides close to full matching and more spatially homogeneous maps, despite some remaining *shim* artifacts that we interpret as learning bias due to the high cross-correlation between  $\delta f$  nominal

Parameter	ROI	Full matching	SVD	DRONE	HD-GMM	HD-GMM+GLLiM	Literature [21]
$T_1$ (ms)	WM	$868 \pm 2$	$905 \pm 2$	$850 \pm 2$	$847 \pm 2$	$834 \pm 1$	690-1100
	GM	$1373 \pm 7$	<b><math>1400 \pm 7</math></b>	<b><math>1272 \pm 6</math></b>	$1360 \pm 7$	$1337 \pm 7$	1286-1393
$T_2$ (ms)	WM	<b><math>49 \pm .1</math></b>	<b><math>87 \pm .2</math></b>	<b><math>50 \pm .2</math></b>	<b><math>55 \pm .2</math></b>	<b><math>55 \pm .2</math></b>	56-80
	GM	<b><math>73 \pm 1</math></b>	$118 \pm 1$	$77 \pm 1$	$81 \pm 1$	$81 \pm .1$	78-117
$T_2^*$ (ms)	WM	$46 \pm .1$	<b><math>23 \pm .5</math></b>	<b><math>29 \pm 24</math></b>	<b><math>51 \pm .2</math></b>	<b><math>51 \pm .2</math></b>	45-48
	GM	$46 \pm .4$	<b><math>30 \pm .4</math></b>	<b><math>27 \pm 33</math></b>	<b><math>55 \pm .6</math></b>	$51 \pm .1$	42-52

**Table 2:** Mean  $T_1$ ,  $T_2$ ,  $T_2^*$  values with 99% confidence in white (WM) and grey (GM) matters ROIs. Out of range values in orange and red, red is further out.



**Fig. 2:**  $T_1$ ,  $T_2$ ,  $T_2^*$  maps (columns) for various methods (lines).

value and the resulting  $T_2^*$  value. HD-GMM superior performance is then confirmed quantitatively in Table 3 where the MAE, over all voxels, with respect to full matching is shown for all 5 parameters. In Table 3, MAEs are computed with respect to imperfect still noisy full matching maps, which explains why the GLLiM variant of HD-GMM does not provide the lowest MAE despite more satisfying maps in terms of spatial homogeneity. Another quantitative comparison is provided in Table 2, with  $T_1$ ,  $T_2$  and  $T_2^*$  mean values over voxels, respectively in white and grey matter ROIs, delineated on  $T_1$  maps. When compared to ranges in healthy subjects as provided in [21], HD-GMM variants show more in range values than most other methods. In Python with Jax library ([link to our code upon acceptance](#)), full matching takes 11s, SVD matching takes 4s and the HD-GMM variants take less than

Method	$T_1$ (s)	$T_2$ (s)	$T_2^*$ (s)	$\delta f$ (Hz)	$B_1$ sensitivity
DRONE [23]	0.14	0.022	0.096	5.0	5.5
SVD [12]	<b>0.056</b>	0.047	0.018	1.4	0.1
HD-GMM + matching	0.058	<b>0.016</b>	<b>0.010</b>	<b>1.0</b>	<b>0.04</b>
HD-GMM + GLLiM [19]	0.081	0.019	0.012	1.3	0.05

**Table 3:** MAEs over voxels with respect to full matching, for DRONE, SVD ( $d = 10$ ), and HD-GMM ( $d = 10$ ,  $K = 30$ ,  $K_g = 50$ ). Best values in bold.

0.4s on Nvidia V100 GPU. However, in more realistic situations, calculations are performed locally by a medical practitioner on a CPU. On an Apple M2 Pro CPU, full matching takes 1 to 2 hours, while HD-GMM variants take only 2min.

#### 4. CONCLUSION AND FUTURE WORK

By equipping high-dimensional Gaussian mixtures models (HD-GMM) with a dimension reduction procedure and incremental estimation of their parameters, we showed that HD-GMM could scale to both very large and high-dimensional data sets. These models can act as a divide-and-conquer paradigm by initially clustering large data volumes and then performing cluster-specific dimension reduction. The clustering structure allows to achieve larger dimension reduction for the same information loss. This ability was showcased on an MRF study using more parameters and more dictionary entries than in standard MRF settings. HD-GMM can use more informative simulations for more accurate parameters maps and thus provide a promising direction towards unleashing the full power of MRF. Future work includes testing further HD-GMM on extensive dictionaries for an increased number of parameters. However for ultra-high MRF dictionaries, the sizes of the obtained sub-dictionaries with HD-GMM may be too high to be efficiently handled via traditional matching. Coupling HD-GMM and GLLiM regression model of [19] would then provide a more tractable solution. More generally, other medical imaging applications can benefit from the scalability and flexibility on the proposed pipeline. HD-GMM allow an efficient use of higher dimensional information as can be extracted from NN latent representations, e.g. [2].

## 5. REFERENCES

- [1] D. Ma, V. Gulani, N. Seiberlich, K. Liu, et al., “Magnetic resonance fingerprinting,” *Nature*, vol. 495, no. 7440, pp. 187–192, 2013.
- [2] N. Pinon, G. Oudoumanessah, R. Trombetta, et al., “Brain subtle anomaly detection based on auto-encoders latent space analysis : application to de novo parkinson patients,” in *IEEE 20th International Symposium on Biomedical Imaging*, Cartegena de Indias, Colombia, 2023, pp. 1–4.
- [3] M. E. Tipping and C. M. Bishop, “Probabilistic principal component analysis,” *Journal of the Royal Statistical Society B: Statistical Methodology*, vol. 61, no. 3, pp. 611–622, 1999.
- [4] R. R. Coifman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, vol. 21, pp. 5–30, 2006.
- [5] G. J. McLachlan, D. Peel, and R. W. Bean, “Modelling high-dimensional data by mixtures of factor analyzers,” *Computational Statistics & Data Analysis*, vol. 41, no. 3, pp. 379–388, 2003.
- [6] M. E. Tipping and C. M. Bishop, “Mixtures of probabilistic principal component analyzers,” *Neural computation*, vol. 11, no. 2, pp. 443–482, 1999.
- [7] C. Bouveyron and C. Brunet-Sauvad, “Model-based clustering of high-dimensional data: A review,” *Computational Statistics & Data Analysis*, vol. 71, no. C, pp. 52–78, 2014.
- [8] K. Gilman, D. Hong, J. A. Fessler, et al., “Streaming Probabilistic PCA for Missing Data with Heteroscedastic Noise,” *arXiv preprint arXiv:2310.06277*, 2023.
- [9] L. Balzano, Y. Chi, and Y. M. Lu, “Streaming PCA and Subspace Tracking: The Missing Data Case,” in *Proceedings of the IEEE*, 2018, vol. 106, pp. 1293–1310.
- [10] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Wiley, 2008, 2nd edition.
- [11] A. Bellas, C. Bouveyron, M. Cottrell, et al., “Model-based Clustering of High-dimensional Data Streams with Online Mixture of Probabilistic PCA,” *Advances in Data Analysis and Classification*, vol. 7, pp. 281–300, 2013.
- [12] D. F. McGivney, E. Pierre, D. Ma, et al., “SVD compression for magnetic resonance fingerprinting in the time domain,” *IEEE Transactions on Medical imaging*, vol. 33, no. 12, pp. 2311–2322, 2014.
- [13] O. Cappé and E. Moulines, “On-line Expectation-Maximization algorithm for latent data models,” *Journal of the Royal Statistical Society B*, vol. 71, pp. 593–613, 2009.
- [14] H. D. Nguyen and F. Forbes, “Global implicit function theorems and the online expectation–maximisation algorithm,” *Australian & New Zealand Journal of Statistics*, vol. 64, no. 2, pp. 255–281, 2022.
- [15] Z. Wen and W. Yin, “A feasible method for optimization with orthogonality constraints,” *Mathematical Programming*, vol. 142, no. 1-2, pp. 397–434, 2013.
- [16] D. F. McGivney, R. Boyacioglu, Y. Jiang, et al., “Magnetic resonance fingerprinting review part 2: Technique and directions,” *Journal of Magnetic Resonance Imaging*, vol. 51, no. 4, pp. 993–1007, 2020.
- [17] A. Monga, D. Singh, H. L. de Moura, et al., “Emerging trends in magnetic resonance fingerprinting for quantitative biomedical imaging applications: A review,” *Bioengineering*, vol. 11, no. 3, pp. 236, 2024.
- [18] M. Barbieri, L. Brizi, E. Giampieri, et al., “A deep learning approach for magnetic resonance fingerprinting: Scaling capabilities and good training practices investigated by simulations,” *Physica medica*, vol. 89, pp. 80–92, 2021.
- [19] F. Boux, F. Forbes, J. Arbel, et al., “Bayesian inverse regression for vascular magnetic resonance fingerprinting,” *IEEE Transactions on Medical Imaging*, vol. 40, no. 7, pp. 1827–1837, 2021.
- [20] Z. Wang, J. Zhang, D. Cui, et al., “Magnetic Resonance Fingerprinting Using a Fast Dictionary Searching Algorithm: MRF-ZOOM,” *IEEE trans. bio-medical engineering*, vol. 66, no. 6, pp. 1526–1535, 2019.
- [21] C. Y. Wang, S. Coppo, B. B. Mehta, et al., “Magnetic resonance fingerprinting with quadratic RF phase for measurement of T2\* simultaneously with δf, T1, and T2,” *Magnetic resonance in medicine*, vol. 81, no. 3, pp. 1849–1862, 2019.
- [22] D. Hong, K. Gilman, L. Balzano, et al., “HePPCAT: Probabilistic PCA for data with heteroscedastic noise,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 4819–4834, 2021.
- [23] O. Cohen, B. Zhu, and M. S. Rosen, “MR fingerprinting deep reconstruction network (DRONE),” *Magnetic resonance in medicine*, vol. 80, no. 3, pp. 885–894, 2018.

# Mini-batch learning of exponential family finite mixture models

Hien D. Nguyen<sup>1\*</sup>, Florence Forbes<sup>2</sup>, and Geoffrey J. McLachlan<sup>3</sup>

September 9, 2019

<sup>1</sup>Department of Mathematics and Statistics, La Trobe University, Melbourne, Victoria, Australia.

<sup>2</sup>Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP<sup>†</sup>, LJK, 38000 Grenoble, France. <sup>†</sup>Institute of Engineering Univ. Grenoble Alpes.

<sup>3</sup>School of Mathematics and Physics, University of Queensland, St. Lucia, Brisbane, Australia.

\*Corresponding author: Hien Nguyen (Email: h.nguyen5@latrobe.edu.edu.au).

## Abstract

Mini-batch algorithms have become increasingly popular due to the requirement for solving optimization problems, based on large-scale data sets. Using an existing online expectation–maximization (EM) algorithm framework, we demonstrate how mini-batch (MB) algorithms may be constructed, and propose a scheme for the stochastic stabilization of the constructed mini-batch algorithms. Theoretical results regarding the convergence of the mini-batch EM algorithms are presented. We then demonstrate how the mini-batch framework may be applied

to conduct maximum likelihood (ML) estimation of mixtures of exponential family distributions, with emphasis on ML estimation for mixtures of normal distributions. Via a simulation study, we demonstrate that the mini-batch algorithm for mixtures of normal distributions can outperform the standard EM algorithm. Further evidence of the performance of the mini-batch framework is provided via an application to the famous MNIST data set.

**Key words:** expectation--maximization algorithm, exponential family distributions, finite mixture models, mini-batch algorithm, normal mixture models, online algorithm

## 1 Introduction

The exponential family of distributions is an important class of probabilistic models with numerous applications in statistics and machine learning. The exponential family contains many of the most commonly used univariate distributions, including the Bernoulli, binomial, gamma, geometric, inverse Gaussian, logarithmic normal, Poisson, and Rayleigh distributions, as well as multivariate distributions such as the Dirichlet, multinomial, multivariate normal, von Mises, and Wishart distributions. See Forbes et al. (2011, Ch. 18), DasGupta (2011, Ch. 18), and Amari (2016, Ch. 2).

Let  $\mathbf{Y}^\top = (Y_1, \dots, Y_d)$  be a random variable (with realization  $\mathbf{y}$ ) on the support  $\mathbb{Y} \subseteq \mathbb{R}^d$  ( $d \in \mathbb{N}$ ), arising from a data generating process (DGP) with probability density/mass function (PDF/PMF)  $f(\mathbf{y}; \boldsymbol{\theta})$  that is characterized by some parameter vector  $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$  ( $p \in \mathbb{N}$ ). We say that the distribution that characterizes the DGP of  $\mathbf{Y}$  is in the exponential family class, if the PDF/PMF can be written in the form

$$f(\mathbf{y}; \boldsymbol{\theta}) = h(\mathbf{y}) \exp \left\{ [\mathbf{s}(\mathbf{y})]^\top \boldsymbol{\phi}(\boldsymbol{\theta}) - \psi(\boldsymbol{\theta}) \right\}, \quad (1)$$

where  $\mathbf{s}(\cdot)$  and  $\boldsymbol{\phi}(\cdot)$  are  $p$ -dimensional vector functions, and  $h(\cdot)$  and  $\psi(\cdot)$  are 1-dimensional functions of  $\mathbf{y}$  and  $\boldsymbol{\theta}$ , respectively. If the dimensionality of  $\mathbf{s}(\cdot)$  and  $\boldsymbol{\phi}(\cdot)$  is less than  $p$ , then we say that the distribution that characterizes the DGP of  $\mathbf{Y}$  is in the curved exponential class.

Let  $Z \in [g]$  ( $[g] = \{1, \dots, g\}; g \in \mathbb{N}$ ) be a latent random variable, and write  $\mathbf{X}^\top = (\mathbf{Y}^\top, Z)$ . Suppose that the PDF/PMF of  $\{\mathbf{Y} = \mathbf{y}|Z = z\}$  can be written as  $f(\mathbf{y}; \boldsymbol{\omega}_z)$ , for each  $z \in [g]$ . If we assume that  $\mathbb{P}(Z = z) = \pi_z > 0$ , such that  $\sum_{z=1}^g \pi_z = 1$ , then we can write the marginal PDF/PMF of  $\mathbf{Y}$  in the form

$$f(\mathbf{y}; \boldsymbol{\theta}) = \sum_{z=1}^g \pi_z f(\mathbf{y}; \boldsymbol{\omega}_z), \quad (2)$$

where we put the unique elements of  $\pi_z$  and  $\boldsymbol{\omega}_z$  into  $\boldsymbol{\theta}$ . We call  $f(\mathbf{y}; \boldsymbol{\theta})$  the  $g$ -component finite mixture PDF, and we call  $f(\mathbf{y}; \boldsymbol{\omega}_z)$  the  $z$ th component PDF, characterized by the parameter vector  $\boldsymbol{\omega}_z \in \Omega$ , where  $\Omega$  is some subset of a real product space. We also say that the elements  $\pi_z$  are prior probabilities, corresponding to the respective component.

The most common finite mixtures models are mixtures of normal distributions, which were popularized by Pearson (1894), and have been prolifically used by numerous prior authors (cf. McLachlan et al., 2019). The  $g$ -component  $d$ -dimensional normal mixture model has PDF of the form

$$f(\mathbf{y}; \boldsymbol{\theta}) = \sum_{z=1}^g \pi_z \varphi(\mathbf{y}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z), \quad (3)$$

where the normal PDFs

$$\varphi(\mathbf{y}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) = |2\pi\boldsymbol{\Sigma}_z|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_z)^\top \boldsymbol{\Sigma}_z^{-1} (\mathbf{y} - \boldsymbol{\mu}_z)\right], \quad (4)$$

replace the component densities  $f(\mathbf{y}; \boldsymbol{\omega}_z)$ , in (2). Each component PDF (4) is parameterized by a mean vector  $\boldsymbol{\mu}_z \in \mathbb{R}^d$  and a positive-definite symmetric covariance matrix  $\boldsymbol{\Sigma}_z \in \mathbb{R}^{d \times d}$ . We then put each  $\pi_z$ ,  $\boldsymbol{\mu}_z$ , and  $\boldsymbol{\Sigma}_z$  into the vector  $\boldsymbol{\theta}$ .

As earlier noted, the normal distribution is a member of the exponential family, and thus (4) can be written in form (1). This can be observed by putting the unique elements of  $\boldsymbol{\mu}_z$  and  $\boldsymbol{\Sigma}_z$  into  $\boldsymbol{\omega}_z$ , and writing  $\varphi(\mathbf{y}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) = f(\mathbf{y}; \boldsymbol{\omega}_z)$  in form (1), with mappings

$$h(\mathbf{y}) = (2\pi)^{-d/2}, \quad \mathbf{s}(\mathbf{y}) = \begin{bmatrix} \mathbf{y} \\ \text{vec}(\mathbf{y}\mathbf{y}^\top) \end{bmatrix}, \quad \boldsymbol{\phi}(\boldsymbol{\omega}_z) = \begin{bmatrix} \boldsymbol{\Sigma}_z^{-1}\boldsymbol{\mu}_z \\ -\frac{1}{2}\text{vec}(\boldsymbol{\Sigma}_z^{-1}) \end{bmatrix}, \text{ and} \quad (5)$$

$$\psi(\boldsymbol{\omega}_z) = \frac{1}{2}\boldsymbol{\mu}_z^\top \boldsymbol{\Sigma}_z^{-1} \boldsymbol{\mu}_z + \frac{1}{2}\log|\boldsymbol{\Sigma}_z|. \quad (6)$$

When conducting data analysis using a normal mixture model, one generally observes an independent and identically (IID) sequence of  $n \in \mathbb{N}$  observations  $\{\mathbf{Y}_i\}_{i=1}^n$ , arising from a DGP that is hypothesized to be characterized by a PDF of the form (3), with unknown parameter vector  $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ . The inferential task is to estimate  $\boldsymbol{\theta}_0$  via some estimator that is computed from  $\{\mathbf{Y}_i\}_{i=1}^n$ . The most common computational approach to obtaining an estimator of  $\boldsymbol{\theta}_0$  is via maximum likelihood (ML) estimation, using the expectation--maximization algorithm (EM; Dempster et al., 1977). See McLachlan & Peel (2000, Ch. 3.2) for a description of the normal mixture EM algorithm. Generally, when  $g$ ,  $d$ , and  $n$  are of small to moderate size, the conventional EM approach is feasible, and is able to perform the task of ML estimation in a timely manner. Unfortunately, due to its high memory demands, costly matrix operations (Nguyen & McLachlan, 2015), and slow

convergence rates (McLachlan & Krishnan, 2008, Sec. 3.9), the conventional EM algorithm is not suited for the computational demands of analyzing increasingly large data sets, such as those that could be considered as *big data* in volumes such as Buhlmann et al. (2016), Han et al. (2017), and Hardle et al. (2018).

Over the years, numerous algorithms have been proposed as means to alleviate the computational demands of the EM algorithm for normal mixture models. Some of such approaches include the component-wise algorithm of Celeux et al. (2001), the greedy algorithm of Vlassis & Likas (2002), the sparse and incremental *kd*-tree algorithm of Ng & McLachlan (2004), the subspace projection algorithm of Bouveyron et al. (2007), and the matrix operations-free algorithm of Nguyen & McLachlan (2015).

There has been a recent resurgence in stochastic approximation algorithms, of the Robbins & Monro (1951) and Kiefer & Wolfowitz (1952) type, developed for the purpose of solving computationally challenging optimization problems, such as the ML estimation of normal mixture models. A good review of the current literature can be found in Chau & Fu (2015). Naïve and direct applications of the stochastic approximation approach to mixture model estimation can be found in Liang & Zhang (2008), Zhang & Liang (2008), and Nguyen & Jones (2018).

Following a remark from Cappé & Moulines (2009) regarding the possible extensions of the online EM algorithm, we propose mini-batch EM algorithms for the ML estimation of exponential family mixture models. These algorithms include a number of variants, among which are update truncation variants that had not been made explicit, before. Using the theorems from Cappé & Moulines (2009), we state results regarding the convergence of our algorithms. We then specialize our attention to the important case of normal mixture models, and demonstrate that the required assumptions for convergence are met in such a scenario.

A thorough numerical study is conducted in order to assess the performance of our normal mixture mini-batch algorithms. Comparisons are drawn between our algorithms and the usual batch EM algorithm for ML estimation of normal mixture models. We show that our mini-batch algorithms can be applied to very large data sets by demonstrating its applicability to the ML estimation of normal mixture models on the famous MNIST data of LeCun et al. (1998).

References regarding mixtures of exponential family distributions and EM-type stochastic approximation algorithms, and comments regarding some recent related literature are relegated to the Supplementary Materials, in the interest of brevity. Additional remarks, numerical results, and derivations are also included in these Supplementary Materials in order to provide extra context and further demonstrate the capabilities of the described framework. These demonstrations include the derivation of mini-batch EM algorithms for mixtures of exponential and Poisson distributions. The Supplementary Materials can be found at [https://github.com/hiendn/StoEMMIX/blob/master/Manuscript\\_files/SupplementaryMaterials.pdf](https://github.com/hiendn/StoEMMIX/blob/master/Manuscript_files/SupplementaryMaterials.pdf).

The remainder of the paper is organized as follows. In Section 2, we present the general results of Cappé & Moulines (2009) and demonstrate how they can be used for mini-batch ML estimation of exponential family mixture models. In Section 3, we derive the mini-batch EM algorithms for the ML estimation of normal mixtures, as well as verify the convergence of the algorithms using the results of Cappé & Moulines (2009). Via numerical simulations, we compare the performance of our mini-batch algorithms to the usual EM algorithm for ML estimation of normal mixture models, in Section 4. A set of real data study on a very large data set is presented in Section 5. Conclusions are drawn in Section 6. Additional material, such as mini-batch EM algorithms for exponential and Poisson mixture models, can be found in the Supplementary Materials.

## 2 The mini-batch EM algorithm

Suppose that we observe a single pair of random variables  $\mathbf{X}^\top = (\mathbf{Y}^\top, \mathbf{Z}^\top)$ , where  $\mathbf{Y} \in \mathbb{Y}$  is observed but  $\mathbf{Z} \in \mathbb{L}$  is latent, where  $\mathbb{Y}$  and  $\mathbb{L}$  are subsets of multivariate real-valued spaces. Furthermore, suppose that the marginal PDF/PMF of  $\mathbf{Y}$  is hypothesized to be of the form  $f(\mathbf{y}; \boldsymbol{\theta}_0)$ , for some unknown parameter vector  $\boldsymbol{\theta}_0 \in \Theta \subseteq \mathbb{R}^p$ . A good estimator for  $\boldsymbol{\theta}_0$  is the ML estimator  $\hat{\boldsymbol{\theta}}$  that can be defined as:

$$\hat{\boldsymbol{\theta}} \in \left\{ \hat{\boldsymbol{\theta}} : \log f(\mathbf{Y}; \hat{\boldsymbol{\theta}}) = \max_{\boldsymbol{\theta} \in \Theta} \log f(\mathbf{Y}; \boldsymbol{\theta}) \right\}. \quad (7)$$

When the problem (7) cannot be solved in a simple manner (e.g. when the solution does not exist in closed form), one may seek to employ an iterative scheme in order to obtain an ML estimator. If the joint PDF/PMF of  $\mathbf{X}$  is known, then one can often construct an EM algorithm in order to solve the problem in the bracket of (7).

Start with some initial guess for  $\boldsymbol{\theta}_0$  and call it the zeroth iterate of the EM algorithm  $\boldsymbol{\theta}^{(0)}$  and suppose that we can write the point PDF/PMF of  $\mathbf{X}$  as  $f(\mathbf{y}, \mathbf{z}; \boldsymbol{\theta})$ , for any  $\boldsymbol{\theta}$ . At the  $r$ th iterate of the EM algorithm, we perform an expectation (E-) step, followed by a maximization (M-) step. The  $r$ th E-step consists of obtaining the conditional expectation of the complete-data log-likelihood (i.e.  $\log f(\mathbf{y}, \mathbf{z}; \boldsymbol{\theta})$ ) given the observed data, using the current estimate of the parameter vector

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(r-1)}) = \mathbb{E}_{\boldsymbol{\theta}^{(r-1)}} [\log f(\mathbf{y}, \mathbf{Z}; \boldsymbol{\theta}) | \mathbf{Y} = \mathbf{y}],$$

which we will call the conditional expected complete-data log-likelihood.

Upon obtaining the conditional expectation of the complete-data log-likelihood, one then con-

ducts the  $r$ th M-step by solving the problem

$$\boldsymbol{\theta}^{(r)} = \arg \max_{\boldsymbol{\theta} \in \Theta} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(r-1)}).$$

The E- and M-steps are repeated until some stopping criterion is met. Upon termination, the final iterate of the algorithm is taken as a solution for problem (7). See McLachlan & Krishnan (2008) for a thorough exposition regarding the EM algorithm.

## 2.1 The online EM algorithm

Suppose that we observe a sequence of  $n$  IID replicates of the variable  $\mathbf{Y}$ ,  $\{\mathbf{Y}_i\}_{i=1}^n$ , where each  $\mathbf{Y}_i$  is the visible component of the pair  $\mathbf{X}_i = (\mathbf{Y}_i^\top, \mathbf{Z}_i^\top)$  ( $i \in [n]$ ). In the online learning context, each of the observations from  $\{\mathbf{Y}_i\}_{i=1}^n$  is observed one at a time, in sequential order.

Using the sequentially obtained sequence  $\{\mathbf{Y}_i\}_{i=1}^n$ , we wish to obtain an ML estimator for the parameter vector  $\boldsymbol{\theta}_0$ , in the same sense as in (7). In order to construct an online EM algorithm framework with provable convergence, Cappé & Moulines (2009) assume the following restrictions regarding the nature of the hypothesized DGP of  $\{\mathbf{Y}_i\}_{i=1}^n$ .

- A1      The complete-data likelihood corresponding to the pair  $\mathbf{X}$  is of exponential family form. That is,

$$f(\mathbf{x}; \boldsymbol{\theta}) = h(\mathbf{x}) \exp \left\{ [\mathbf{s}(\mathbf{x})]^\top \boldsymbol{\phi}(\boldsymbol{\theta}) - \psi(\boldsymbol{\theta}) \right\}, \quad (8)$$

where  $h(\cdot)$ ,  $\psi(\cdot)$ ,  $\mathbf{s}(\cdot)$ , and  $\boldsymbol{\phi}(\cdot)$  are as defined for (1).

- A2      The function

$$\bar{s}(\mathbf{y}; \boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}} [\mathbf{s}(\mathbf{X}) | \mathbf{Y} = \mathbf{y}] \quad (9)$$

is well defined for all  $\mathbf{y} \in \mathbb{Y}$  and  $\boldsymbol{\theta} \in \Theta$ .

A3 There exists a convex open subset  $\mathbb{S} \subseteq \mathbb{R}^p$ , which satisfies the properties that:

- (i) for all  $\mathbf{s} \in \mathbb{S}$ ,  $\mathbf{y} \in \mathbb{Y}$ ,  $\boldsymbol{\theta} \in \Theta$ ,  $(1 - \gamma)\mathbf{s} + \gamma\bar{\mathbf{s}}(\mathbf{y}; \boldsymbol{\theta}) \in \mathbb{S}$  for any  $\gamma \in (0, 1)$ , and
- (ii) for any  $\mathbf{s} \in \mathbb{S}$ , the function

$$q(\mathbf{s}; \boldsymbol{\theta}) = \mathbf{s}^\top \boldsymbol{\phi}(\boldsymbol{\theta}) - \psi(\boldsymbol{\theta})$$

has a unique global maximum over  $\Theta$ , which will be denoted by

$$\bar{\boldsymbol{\theta}}(\mathbf{s}) = \arg \max_{\boldsymbol{\theta} \in \Theta} q(\mathbf{s}; \boldsymbol{\theta}).$$

Let  $Q_n(\boldsymbol{\theta}; \boldsymbol{\theta}^{(r-1)})$  be the expected complete-data log-likelihood over data  $\{\mathbf{Y}_i\}_{i=1}^n$ , at the  $r$ th E-step of an EM algorithm for solving the problem:

$$\hat{\boldsymbol{\theta}}_n \in \left\{ \hat{\boldsymbol{\theta}} : n^{-1} \sum_{i=1}^n \log f(\mathbf{Y}_i; \hat{\boldsymbol{\theta}}) = \max_{\boldsymbol{\theta} \in \Theta} n^{-1} \sum_{i=1}^n \log f(\mathbf{Y}_i; \boldsymbol{\theta}) \right\},$$

where we say that  $\hat{\boldsymbol{\theta}}_n$  is the ML estimator, based on the data  $\{\mathbf{Y}_i\}_{i=1}^n$ . When, A1–A3 are satisfied, we can write

$$Q_n(\boldsymbol{\theta}; \boldsymbol{\theta}^{(r-1)}) = nq\left(n^{-1} \sum_{i=1}^n \bar{\mathbf{s}}(\mathbf{Y}_i; \boldsymbol{\theta}^{(r-1)}); \boldsymbol{\theta}\right) + \text{Constant},$$

which can then be maximized, with respect to  $\boldsymbol{\theta}$ , in order to yield an M-step update of the form:

$$\boldsymbol{\theta}^{(r)} = \bar{\boldsymbol{\theta}}\left(n^{-1} \sum_{i=1}^n \bar{\mathbf{s}}(\mathbf{Y}_i; \boldsymbol{\theta}^{(r-1)})\right), \quad (10)$$

where  $\boldsymbol{\theta}^{(r)}$  is a function that depends only on the average  $n^{-1} \sum_{i=1}^n \bar{s}(\mathbf{Y}_i; \boldsymbol{\theta}^{(r-1)})$ .

Now we suppose that we sample the individual observations of  $\{\mathbf{Y}_i\}_{i=1}^n$ , one at a time and sequentially. Furthermore, upon observation of  $\mathbf{Y}_i$ , we wish to compute an online estimate of  $\boldsymbol{\theta}_0$ , which we denote as  $\boldsymbol{\theta}^{(i)}$ . Based on the simplification of the EM algorithm under A1–A3, as described above, Cappé & Moulines (2009) proposed the following online EM algorithm.

Upon observation of  $\mathbf{Y}_i$ , compute the intermediate updated sufficient statistic

$$\mathbf{s}^{(i)} = \mathbf{s}^{(i-1)} + \gamma_i [\bar{s}(\mathbf{Y}_i; \boldsymbol{\theta}^{(i-1)}) - \mathbf{s}^{(i-1)}], \quad (11)$$

with  $\mathbf{s}^{(0)} = \bar{s}(\mathbf{Y}_i; \boldsymbol{\theta}^{(0)})$ . Here,  $\gamma_i$  is the  $i$ th term of the learning rate sequence that we will discuss in further details in the sequel. Observe that we can also write

$$\mathbf{s}^{(i)} = \gamma_i \bar{s}(\mathbf{Y}_i; \boldsymbol{\theta}^{(i-1)}) + (1 - \gamma_i) \mathbf{s}^{(i-1)},$$

which makes it clear that for  $\gamma_i \in (0, 1)$ ,  $\mathbf{s}^{(i)}$  is a weighted average between  $\bar{s}(\mathbf{Y}_i; \boldsymbol{\theta}^{(i-1)})$  and  $\mathbf{s}^{(i-1)}$ . Using  $\mathbf{s}^{(i)}$  and the function  $\bar{\boldsymbol{\theta}}$ , we can then express the  $i$ th iteration online EM estimate of  $\boldsymbol{\theta}_0$  as

$$\boldsymbol{\theta}^{(i)} = \bar{\boldsymbol{\theta}}(\mathbf{s}^{(i)}). \quad (12)$$

Next, we state a consistency theorem that strongly motivates the use of the online EM algorithm, defined by (11) and (12). Suppose that the true DGP that generates each  $\mathbf{Y}_i$  of  $\{\mathbf{Y}_i\}_{i=1}^n$  is characterized by the probability measure  $F_0$ . Write the expectation operator with respect to this measure as  $\mathbb{E}_{F_0}$ . In order to state the consistency result of Cappé & Moulines (2009), we require the following additional set of assumptions.

- B1      The parameter space  $\Theta$  is a convex and open subset of a real product space, and the functions  $\phi$  and  $\psi$ , in (8), are both twice continuously differentiable with respect to  $\boldsymbol{\theta} \in \Theta$ .
- B2      The function  $\bar{\boldsymbol{\theta}}$ , as defined in (10), is a continuously differentiable function with respect to  $\mathbf{s} \in \mathbb{S}$ , where  $\mathbb{S}$  is as defined in A3.
- B3      For some  $p > 2$ , and all compact  $\mathbb{K} \subset \mathbb{S}$ ,

$$\sup_{\mathbf{s} \in \mathbb{K}} \mathbb{E}_{F_0} [|\bar{\mathbf{s}}(\mathbf{Y}; \bar{\boldsymbol{\theta}}(\mathbf{s}))|^p] < \infty. \quad (13)$$

As the algorithm defined by (11) and (12) is of the Robbins-Monro type, establishment of convergence of the algorithm requires the definition of a mean field (see Chen, 2003 and Kushner & Yin, 2003 for comprehensive treatments regarding such algorithms). In the case of the online EM algorithm, we write the mean field as

$$\mathbf{h}(\mathbf{s}) = \mathbb{E}_{F_0} [\bar{\mathbf{s}}(\mathbf{Y}; \bar{\boldsymbol{\theta}}(\mathbf{s}))] - \mathbf{s}$$

and define the set of its roots as  $\Gamma = \{\mathbf{s} \in \mathbb{S} : \mathbf{h}(\mathbf{s}) = \mathbf{0}\}$ .

Define the log-likelihood of the hypothesized PDF  $f(\cdot; \boldsymbol{\theta})$  with respect to the measure  $F_0$ , as

$$\ell(f(\cdot; \boldsymbol{\theta})) = \mathbb{E}_{F_0} [\log f(\mathbf{Y}; \boldsymbol{\theta})].$$

Let  $\nabla_{\boldsymbol{\theta}}$  denote the gradient with respect to  $\boldsymbol{\theta}$ , and define the sets

$$\mathbb{W}_{\Gamma} = \{\ell(f(\cdot; \boldsymbol{\theta})) : \boldsymbol{\theta} = \bar{\boldsymbol{\theta}}(\mathbf{s}), \mathbf{s} \in \Gamma\}$$

and

$$\mathbb{M}_\Theta = \left\{ \hat{\boldsymbol{\theta}} \in \Theta : \nabla_{\boldsymbol{\theta}} \ell(f(\cdot; \boldsymbol{\theta}))|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \mathbf{0} \right\}.$$

Note that  $\mathbb{M}_\Theta$  is the set of stationary points of the log-likelihood function. Further, define the distance between a real vector  $\mathbf{a}$  and a set  $\mathbb{B}$  by

$$\text{dist}(\mathbf{a}, \mathbb{B}) = \inf_{\mathbf{b} \in \mathbb{B}} \|\mathbf{a} - \mathbf{b}\|,$$

where  $\|\cdot\|$  is the usual Euclidean metric, and denote the complement of a subset  $\mathbb{A}$  of a real product space by  $\mathbb{A}^c$ . Finally, make the following assumptions.

C1      The sequence of learning rates  $\{\gamma_i\}_{i=1}^\infty$  fulfills the conditions that  $0 < \gamma_i < 1$ , for each  $i$ ,

$$\sum_{i=1}^{\infty} \gamma_i = \infty, \text{ and } \sum_{i=1}^{\infty} \gamma_i^2 < \infty.$$

C2      At initialization  $\mathbf{s}^{(0)} \in \mathbb{S}$  and, with probability 1,

$$\limsup_{i \rightarrow \infty} |\mathbf{s}^{(i)}| < \infty, \text{ and } \liminf_{i \rightarrow \infty} \text{dist}(\mathbf{s}^{(i)}, \mathbb{S}^c) = 0.$$

C3      The set  $\mathbb{W}_\Gamma$  is nowhere dense.

**Theorem 1** (Cappe and Moulines, 2009). *Assume that A1–A3, B1–B3, and C1–C3 are satisfied, and let  $\{\mathbf{Y}_i\}_{i=1}^\infty$  be an IID sample with DGP characterized by the PDF  $f_0$ , which is hypothesized to have the form  $f(\cdot; \boldsymbol{\theta})$ , as in (8). Further, let  $\{\mathbf{s}^{(i)}\}_{i=1}^\infty$  and  $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^\infty$  be sequences generated by the*

online EM algorithm, defined by (11) and (12). Then, with probability 1,

$$\lim_{i \rightarrow \infty} \text{dist}(\mathbf{s}^{(i)}, \Gamma) = 0, \text{ and } \lim_{i \rightarrow \infty} \text{dist}(\boldsymbol{\theta}^{(i)}, \mathbb{M}_{\Theta}) = 0.$$

Notice that this result allows for a mismatch between the true probability measure  $F_0$  and the assumed pseudo-true family  $f(\cdot; \boldsymbol{\theta})$  from which  $\{\mathbf{Y}_i\}_{i=1}^{\infty}$  is hypothesized to arise. This therefore allows for misspecification, in the sense of White (1982), which is almost certain to occur in the modeling of any sufficiently complex data. In any case, the online EM algorithm will converge towards an estimate of the parameter vector  $\boldsymbol{\theta}$ , which is in the set  $\mathbb{M}_{\Theta}$ . When the DGP can be characterized by a density in the family of the form  $f(\cdot; \boldsymbol{\theta})$ , we observe that  $\mathbb{M}_{\Theta}$  contains not only the global maximizer of the log-likelihood function, but also local maximizers, minimizers, and saddle points. Thus, the online algorithm suffers from the same lack of strong convergence guarantees, as the batch EM algorithm (cf. Wu, 1983).

In the case of misspecification the set  $\mathbb{M}_{\Theta}$  will include the parameter vector  $\boldsymbol{\theta}_0$  that maximizes the log-likelihood function, with respect to the true probability measure  $F_0$ . However, as with the well-specified case, it will also include stationary points of other types, as well. We further provide characterizations of the sets  $\mathbb{W}_{\Gamma}$  and  $\mathbb{M}_{\Theta}$  in terms of the Kullback-Leibler divergence (KL; Kullback & Leibler, 1951) in the Supplementary Materials.

Assumption C1 can be fulfilled by taking sequences  $\{\gamma_i\}_{i=1}^{\infty}$  of form  $\gamma_i = \gamma_0 i^{\alpha}$ , for some  $\alpha \in (0, 1]$  and  $\gamma_0 \in (0, 1)$ . We shall discuss this point further, in the sequel. Although the majority of the assumptions can be verified or are fulfilled by construction, the two limits in C2 stand out as being particularly difficult to verify. In Cappé & Moulines (2009), the authors suggest that one method for enforcing C2 is to use the method of update truncation, but they did not provide an explicit scheme for conducting such truncation.

A truncation version of the algorithm defined by (11) and (12) can be specified via the method of Delyon et al. (1999). That is, let  $\{\mathbb{K}_m\}_{m=0}^\infty$  be a sequence of compact sets, such that

$$\mathbb{K}_m \subset \text{interior}(\mathbb{K}_{m+1}), \text{ and } \bigcup_{m=0}^\infty \mathbb{K}_m = \mathbb{S}. \quad (14)$$

We then replace (11) and (12) by the following scheme. At the  $i$ th iteration, firstly compute

$$\tilde{\mathbf{s}}^{(i)} = \mathbf{s}^{(i-1)} + \gamma_i [\bar{\mathbf{s}}(\mathbf{Y}_i; \boldsymbol{\theta}^{(i-1)}) - \mathbf{s}^{(i-1)}]. \quad (15)$$

Secondly,

$$\text{if } \tilde{\mathbf{s}}^{(i)} \in \mathbb{K}_{m_{i-1}}, \text{ then set } \mathbf{s}^{(i)} = \tilde{\mathbf{s}}^{(i)}, \boldsymbol{\theta}^{(i)} = \bar{\boldsymbol{\theta}}(\mathbf{s}^{(i)}), \text{ and } m_i = m_{i-1}, \quad (16)$$

else

$$\text{if } \tilde{\mathbf{s}}^{(i)} \notin \mathbb{K}_{m_{i-1}}, \text{ then set } \mathbf{s}^{(i)} = \mathbf{S}_i, \boldsymbol{\theta}^{(i)} = \bar{\boldsymbol{\theta}}(\mathbf{S}_i), \text{ and } m_i = m_{i-1} + 1, \quad (17)$$

where  $\{\mathbf{S}_i\}_{i=1}^\infty$  is an arbitrary random sequence, such that  $\mathbf{S}_i \in \mathbb{K}_0$ , for each  $i \in \mathbb{N}$ . We have the following result regarding the algorithm defined by (15)–(17).

**Proposition 1.** *Assume that A1–A3, B1–B3, C1 and C3 are satisfied, and let  $\{\mathbf{Y}_i\}_{i=1}^\infty$  be an IID sample with DGP characterized by the PDF  $f_0$ , which is hypothesized to have the form  $f(\cdot; \boldsymbol{\theta})$ , as in (8). Further, let  $\{\mathbf{s}^{(i)}\}_{i=1}^\infty$  and  $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^\infty$  be sequences generated by the truncated online EM algorithm, defined by (15)–(17). Then, with probability 1,*

$$\lim_{i \rightarrow \infty} \text{dist}(\mathbf{s}^{(i)}, \Gamma) = 0, \text{ and } \lim_{i \rightarrow \infty} \text{dist}(\boldsymbol{\theta}^{(i)}, \mathbb{M}_\Theta) = 0.$$

The proof of Proposition 1 requires the establishment of equivalence between A1–A3, B1–B3,

C1, and C3, and the many assumptions of Theorem 3 and 6 of Delyon et al. (1999). Thus the proof is simple and mechanical, but long and tedious. We omit it for the sake of brevity.

## 2.2 The mini-batch algorithm

At the most elementary level, a mini-batch algorithm for computation of a sequence of estimators  $\{\boldsymbol{\theta}^{(r)}\}_{r=1}^R$  for some parameter  $\boldsymbol{\theta}_0$ , from some sample  $\{\mathbf{Y}_i\}_{i=1}^n$ , where  $R \in \mathbb{N}$ , has the following property. The algorithm is iterative, and at the  $r$ th iteration of the algorithm, the estimator  $\boldsymbol{\theta}^{(r)}$  only depends on the previous iterate  $\boldsymbol{\theta}^{(r-1)}$  and some subsample, possibly with replacement, of  $\{\mathbf{Y}_i\}_{i=1}^n$ . Typical examples of mini-batch algorithms include the many variants of the stochastic gradient descent-class of algorithms; see, for example, Cotter et al. (2011), Li et al. (2014), Zhao et al. (2014), and Ghadimi et al. (2016).

Suppose that we observe a fixed size realization  $\{\mathbf{y}_i\}_{i=1}^n$  of some IID random sample  $\{\mathbf{Y}\}_{i=1}^n$ . Furthermore, fix a so-called batch size  $N \leq n$  and a learning rate sequence  $\{\gamma_r\}_{r=1}^R$ , and select some appropriate initial values  $\mathbf{s}^{(0)}$  and  $\boldsymbol{\theta}^{(0)}$  from which the sequences  $\{\mathbf{s}^{(r)}\}_{r=1}^R$  and  $\{\boldsymbol{\theta}^{(r)}\}_{r=1}^R$  can be constructed. A mini-batch version of the online EM algorithm, specified by (11) and (12) can be specified as follows. For each  $r \in [R]$ , sample  $N$  observations from  $\{\mathbf{y}_i\}_{i=1}^n$  uniformly, with replacement, and denote the subsample by  $\{\mathbf{Y}_i^r\}_{i=1}^N$ . Then, using  $\{\mathbf{Y}_i^r\}_{i=1}^N$ , compute

$$\mathbf{s}^{(r)} = \mathbf{s}^{(r-1)} + \gamma_r \left[ N^{-1} \sum_{i=1}^N \bar{s}(\mathbf{Y}_i^r; \boldsymbol{\theta}^{(r-1)}) - \mathbf{s}^{(r-1)} \right], \text{ and } \boldsymbol{\theta}^{(r)} = \bar{\boldsymbol{\theta}}(\mathbf{s}^{(r)}). \quad (18)$$

In order to justify the mini-batch algorithm, we make the following observation. The online EM algorithm, defined by (11) and (12), is designed to obtain a root in the set  $\mathbb{M}_\Theta$ , which is a vector  $\hat{\boldsymbol{\theta}} \in \Theta$  such that

$$\nabla_{\boldsymbol{\theta}} \ell(f(\cdot; \boldsymbol{\theta}))|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \mathbf{0}.$$

If  $N = 1$  (i.e., the case proposed in Cappé & Moulines, 2009, Sec. 2.5), then the DGP for generating subsamples is simply a single draw from the empirical measure:

$$F_{\text{Emp}}(\mathbf{y}) = \sum_{i=1}^n \frac{1}{n} \delta(\mathbf{y} - \mathbf{y}_i),$$

where  $\delta$  is the Dirac delta function (see, for details, Prosperetti, 2011, Ch. 2). We can write

$$\begin{aligned} \ell(f(\cdot; \boldsymbol{\theta})) &= \mathbb{E}_{F_0} [\log f(\mathbf{Y}; \boldsymbol{\theta})] \\ &= \mathbb{E}_{F_{\text{Emp}}} [\log f(\mathbf{Y}; \boldsymbol{\theta})] \\ &= \frac{1}{n} \sum_{i=1}^n \log f(\mathbf{y}_i; \boldsymbol{\theta}), \end{aligned} \tag{19}$$

which is the log-likelihood function, with respect to the realization  $\{\mathbf{y}_i\}_{i=1}^n$ , under the density function of form  $f(\cdot; \boldsymbol{\theta})$ . Thus, in the  $N = 1$  case, the algorithm defined by (18) solves for log-likelihood roots  $\hat{\boldsymbol{\theta}}$  of the form

$$\frac{1}{n} \sum_{i=1}^n \nabla_{\boldsymbol{\theta}} \log f(\mathbf{y}_i; \boldsymbol{\theta})|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \mathbf{0},$$

or equivalently, solving for an element in the set

$$\mathbb{M}_{\Theta}^{\text{Emp}} = \left\{ \hat{\boldsymbol{\theta}} \in \Theta : \sum_{i=1}^n \nabla_{\boldsymbol{\theta}} \log f(\mathbf{y}_i; \boldsymbol{\theta})|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \mathbf{0} \right\}.$$

The  $N > 1$  case follows the same argument, and is described in the Supplementary Materials (Section 2.2). Let  $F_{\text{Emp}}^N$  denote the probability measure corresponding to the DGP of  $N$  independent random samples from  $F_{\text{Emp}}$ . We have the following result, based on Theorem 1.

**Corollary 1.** For any  $N \in \mathbb{N}$ , assume that A1–A3, B1–B3, and C1–C3 are satisfied (replacing  $i$  by  $r$ , and  $F_0$  by  $F_{Emp}^N$ , where appropriate), and let  $\{\mathbf{y}_i\}_{i=1}^n$  be a realization of some IID random sequence  $\{\mathbf{Y}_i\}_{i=1}^n$ , where each  $\mathbf{Y}_i$  is hypothesized to arise from a DGP having PDF of the form  $f(\cdot; \boldsymbol{\theta})$ , as in (8). Let  $\{\mathbf{s}^{(r)}\}_{i=1}^\infty$  and  $\{\boldsymbol{\theta}^{(r)}\}_{i=1}^\infty$  be sequences generated by the mini-batch EM algorithm, defined by (11) and (12). Then, with probability 1,

$$\lim_{r \rightarrow \infty} dist(\mathbf{s}^{(r)}, \Gamma) = 0, \text{ and } \lim_{r \rightarrow \infty} dist(\boldsymbol{\theta}^{(r)}, \mathbb{M}_\Theta^{Emp}) = 0.$$

That is, as we take  $R \rightarrow \infty$ , the algorithm defined by (11) and (12) will identify elements in the sets  $\Gamma$  and  $\mathbb{M}_\Theta^{Emp}$ , with probability 1. As with the case of Theorem 1, C2 is again difficult to verify. Let  $\{\mathbb{K}_m\}_{m=0}^\infty$  be as per (14). Then, we replace the algorithm defined via (18), by the following truncated version.

Again, suppose that we observe a fixed size realization  $\{\mathbf{y}_i\}_{i=1}^n$  of some IID random sample  $\{\mathbf{Y}\}_{i=1}^n$ . Furthermore, fix a so-called batch size  $N \leq n$  and a learning rate sequence  $\{\gamma_r\}_{r=1}^R$ , and select some appropriate initial values  $\mathbf{s}^{(0)}$  and  $\boldsymbol{\theta}^{(0)}$  from which the sequences  $\{\mathbf{s}^{(r)}\}_{r=1}^R$  and  $\{\boldsymbol{\theta}^{(r)}\}_{r=1}^R$  can be constructed. For each  $r \in [R]$ , sample  $N$  observations from  $\{\mathbf{y}_i\}_{i=1}^n$  uniformly, with replacement, and denote the subsample by  $\{\mathbf{Y}_i^r\}_{i=1}^N$ . Using  $\{\mathbf{Y}_i^r\}_{i=1}^N$ , compute

$$\tilde{\mathbf{s}}^{(r)} = \mathbf{s}^{(r-1)} + \gamma_r \left[ N^{-1} \sum_{i=1}^N \bar{s}(\mathbf{Y}_i^r; \boldsymbol{\theta}^{(r-1)}) - \mathbf{s}^{(r-1)} \right]. \quad (20)$$

Then, with  $i$  being appropriately replaced by  $r$ , use (16) and (17) to compute  $\mathbf{s}^{(r)}$  and  $\boldsymbol{\theta}^{(r)}$ . We obtain the following result via an application of Proposition 1.

**Corollary 2.** For any  $N \in \mathbb{N}$ , assume that A1–A3, B1–B3, and C1–C3 are satisfied (replacing  $i$  by  $r$ , and  $F_0$  by  $F_{Emp}^N$ , where appropriate), and let  $\{\mathbf{y}_i\}_{i=1}^n$  be a realization of some IID random

sequence  $\{\mathbf{Y}_i\}_{i=1}^n$ , where each  $\mathbf{Y}_i$  is hypothesized to arise from a DGP having PDF of the form  $f(\cdot; \boldsymbol{\theta})$ , as in (8). Let  $\{\mathbf{s}^{(r)}\}_{i=1}^\infty$  and  $\{\boldsymbol{\theta}^{(r)}\}_{i=1}^\infty$  be sequences generated by the truncated mini-batch EM algorithm, defined by (20), (16), and (17). Then, with probability 1,

$$\lim_{r \rightarrow \infty} \text{dist}(\mathbf{s}^{(r)}, \Gamma) = 0, \text{ and } \lim_{r \rightarrow \infty} \text{dist}(\boldsymbol{\theta}^{(r)}, \mathbb{M}_\Theta^{\text{Emp}}) = 0.$$

### 2.3 The learning rate sequence

As previously stated, a good choice for the learning rate sequence  $\{\gamma_i\}_{i=1}^\infty$  is to take  $\gamma_i = \gamma_0 i^\alpha$ , for each  $i \in \mathbb{N}$ , such that  $\alpha \in (1/2, 1]$  and  $\gamma_0 \in (0, 1)$ . Under the assumptions of Theorem 1, Cappé & Moulines (2009, Thm. 2) showed that the learning rate choice leads to the convergence of the sequence  $\gamma_0^{1/2} i^{\alpha/2} (\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}_0)$ , in distribution, to a normal distribution with mean  $\mathbf{0}$  and covariance matrix depending on  $\boldsymbol{\theta}_0$ , for some  $\boldsymbol{\theta}_0 \in \mathbb{M}_\Theta$ . Here  $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^\infty$  is a sequence of online EM algorithm iterates, generated by (11) and (12). A similar result can be stated for the truncated online EM, mini-batch EM, and truncated mini-batch EM algorithms, by replacing the relevant indices and quantities in the previous statements by their respective counterparts.

The result above implies that the convergence rate is  $\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}_0 = o_p(i^{\alpha/2})$ , for any valid  $\alpha$ , where  $o_p$  is the usual order in probability notation (see White, 2001, Defn. 2.33). Thus, it would be tempting to take  $\alpha = 1$  in order to obtain a rate with optimal order of  $n^{1/2}$ . However, as shown in Cappé & Moulines (2009, Thm. 2), the  $\alpha = 1$  case requires constraints on  $\gamma_0$  in order to fulfill a stability assumption that is impossible to validate, in practice.

It is, however, still possible to obtain a sequence of estimators that converges to some  $\boldsymbol{\theta}_0$  at a rate with optimal order  $n^{1/2}$ . We can do this via the famous so-called Polyak averaging scheme of Polyak (1990) and Polyak & Juditsky (1992). In the current context, one takes as an input

the sequence of online EM iterates  $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{\infty}$ , and output the running average sequence  $\{\boldsymbol{\theta}_A^{(i)}\}_{i=1}^{\infty}$ , where

$$\boldsymbol{\theta}_A^{(i)} = i^{-1} \sum_{j=1}^i \boldsymbol{\theta}^{(j)}, \quad (21)$$

for each  $i \in \mathbb{N}$ . For any  $\alpha \in (1/2, 1)$ , it is provable that  $\boldsymbol{\theta}_A^{(i)} - \boldsymbol{\theta}_0 = o_p(n^{1/2})$ . As before, this result generalizes to the cases of the truncated online EM, mini-batch EM, and truncated mini-batch EM algorithms, also.

We note that the computation of the  $i$ th running average term (21) does not require the storage of the entire sequence of iterates  $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{\infty}$ , as one would anticipate by applying (21) naively. One can instead write (21) in the iterative form

$$\boldsymbol{\theta}_A^{(i)} = i^{-1} \left[ (i-1) \boldsymbol{\theta}_A^{(i-1)} + \boldsymbol{\theta}^{(i)} \right].$$

### 3 Normal mixture models

#### 3.1 Finite mixtures of exponential family distributions

We recall from Section 1, that the random variable  $\mathbf{Y}$  is said to arise from a DGP characterized by a  $g$  component finite mixture of component PDFs of form  $f(\mathbf{y}; \boldsymbol{\omega}_z)$ , if it has a PDF of the form (2). Furthermore, if the component PDFs are of the exponential family form (1), then we further write the PDF of  $\mathbf{Y}$  as

$$f(\mathbf{y}; \boldsymbol{\theta}) = \sum_{z=1}^g \pi_z h(\mathbf{y}) \exp \left\{ [\mathbf{s}(\mathbf{y})]^\top \boldsymbol{\phi}(\boldsymbol{\omega}_z) - \psi(\boldsymbol{\omega}_z) \right\}. \quad (22)$$

From the construction of the finite mixture model, we have the fact that (22) is the marginal-

ization of the joint density of the random variable  $\mathbf{X}^\top = (\mathbf{Y}^\top, Z)$ :

$$f(\mathbf{x}; \boldsymbol{\theta}) = \prod_{\zeta=1}^g \left[ \pi_\zeta h(\mathbf{y}) \exp \left\{ [\mathbf{s}(\mathbf{y})]^\top \boldsymbol{\phi}(\boldsymbol{\omega}_\zeta) - \psi(\boldsymbol{\omega}_\zeta) \right\} \right]^{\llbracket z=\zeta \rrbracket} \quad (23)$$

over the random variable  $Z \in [g]$ , recalling that  $Z$  is a categorical random variable with  $g$  categories (cf. McLachlan & Peel, 2000, Ch. 2). Here,  $\llbracket c \rrbracket$  is the Iverson bracket notation, that takes value 1 if condition  $c$  is true, and 0 otherwise (Iverson, 1967, Ch. 1). We rewrite (23) as follows:

$$\begin{aligned} f(\mathbf{x}; \boldsymbol{\theta}) &= h(\mathbf{y}) \exp \left\{ \sum_{\zeta=1}^g \llbracket z=\zeta \rrbracket \left[ \log \pi_\zeta + [\mathbf{s}(\mathbf{y})]^\top \boldsymbol{\phi}(\boldsymbol{\omega}_\zeta) - \psi(\boldsymbol{\omega}_\zeta) \right] \right\} \\ &= h(\mathbf{x}) \exp \left\{ [\mathbf{s}(\mathbf{x})]^\top \boldsymbol{\phi}(\boldsymbol{\theta}) - \psi(\boldsymbol{\theta}) \right\}, \end{aligned}$$

where  $h(\mathbf{x}) = h(\mathbf{y})$ ,  $\psi(\boldsymbol{\theta}) = 0$ ,

$$\mathbf{s}(\mathbf{x}) = \begin{bmatrix} \llbracket z=1 \rrbracket \\ \llbracket z=1 \rrbracket \mathbf{s}(\mathbf{y}) \\ \vdots \\ \llbracket z=g \rrbracket \\ \llbracket z=g \rrbracket \mathbf{s}(\mathbf{y}) \end{bmatrix}, \text{ and } \boldsymbol{\phi}(\boldsymbol{\theta}) = \begin{bmatrix} \log \pi_1 - \psi(\boldsymbol{\omega}_1) \\ \boldsymbol{\phi}(\boldsymbol{\omega}_1) \\ \vdots \\ \log \pi_g - \psi(\boldsymbol{\omega}_g) \\ \boldsymbol{\phi}(\boldsymbol{\omega}_g) \end{bmatrix},$$

and thus obtain the following general result regarding finite mixtures of exponential family distributions.

**Proposition 2.** *The complete-data likelihood of any finite mixture of exponential family distributions with PDF of the form (22) can also be written in the exponential family form (8).*

With Proposition 2, we have proved that when applying the online EM or the mini-batch EM algorithm to the problem of conducting ML estimation for any finite mixture model of exponential family distributions, A1 is automatically satisfied.

### 3.2 Finite mixtures of normal distributions

Recall from Section 1 that the random variable  $\mathbf{Y}$  is said to be distributed according to a  $g$ -component finite mixture of normal distributions, if it characterized by a PDF of the form (3). Using the exponential family decomposition from (5) and (6), we write the complete-data likelihood of  $\mathbf{X}^\top = (\mathbf{Y}^\top, Z)$  in the form (8) by setting  $h(\mathbf{x}) = (2\pi)^{-d/2}$ ,  $\psi(\boldsymbol{\theta}) = 0$ ,

$$\mathbf{s}(\mathbf{x}) = \begin{bmatrix} \llbracket z = 1 \rrbracket \\ \llbracket z = 1 \rrbracket \mathbf{y} \\ \llbracket z = 1 \rrbracket \text{vec}(\mathbf{y}\mathbf{y}^\top) \\ \vdots \\ \llbracket z = g \rrbracket \\ \llbracket z = g \rrbracket \mathbf{y} \\ \llbracket z = g \rrbracket \text{vec}(\mathbf{y}\mathbf{y}^\top) \end{bmatrix}, \text{ and } \boldsymbol{\phi}(\boldsymbol{\theta}) = \begin{bmatrix} \log \pi_1 - \frac{1}{2} \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 + \frac{1}{2} \log |\boldsymbol{\Sigma}_1| \\ \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 \\ -\frac{1}{2} \text{vec}(\boldsymbol{\Sigma}_1^{-1}) \\ \vdots \\ \log \pi_g - \frac{1}{2} \boldsymbol{\mu}_g^\top \boldsymbol{\Sigma}_g^{-1} \boldsymbol{\mu}_g + \frac{1}{2} \log |\boldsymbol{\Sigma}_g| \\ \boldsymbol{\Sigma}_g^{-1} \boldsymbol{\mu}_g \\ -\frac{1}{2} \text{vec}(\boldsymbol{\Sigma}_g^{-1}) \end{bmatrix}, \quad (24)$$

where  $\text{vec}(\cdot)$  is the matrix vectorization operator.

Using the results from McLachlan & Peel (2000, Ch. 3), we write the conditional expectation (9) in the form

$$[\bar{\mathbf{s}}(\mathbf{y}; \boldsymbol{\theta})]^\top = (\tau_1(\mathbf{y}; \boldsymbol{\theta}), \tau_1(\mathbf{y}; \boldsymbol{\theta}) \mathbf{y}, \tau_1(\mathbf{y}; \boldsymbol{\theta}) \text{vec}(\mathbf{y}\mathbf{y}^\top), \dots, \tau_g(\mathbf{y}; \boldsymbol{\theta}), \tau_g(\mathbf{y}; \boldsymbol{\theta}) \mathbf{y}, \tau_g(\mathbf{y}; \boldsymbol{\theta}) \text{vec}(\mathbf{y}\mathbf{y}^\top)),$$

where

$$\tau_z(\mathbf{y}; \boldsymbol{\theta}) = \frac{\pi_z \varphi(\mathbf{y}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)}{\sum_{\zeta=1}^g \pi_\zeta \varphi(\mathbf{y}; \boldsymbol{\mu}_\zeta, \boldsymbol{\Sigma}_\zeta)},$$

is the usual *a posteriori* probability that  $Z = z$  ( $z \in [g]$ ), given observation of  $\mathbf{Y} = \mathbf{y}$ . Again, via the results from McLachlan & Peel (2000, Ch. 3), we write the update function  $\bar{\boldsymbol{\theta}}$  in the following form. Define  $\bar{\boldsymbol{\theta}}$  to have the elements  $\bar{\pi}_z$  and  $\bar{\boldsymbol{\omega}}_z$ , for each  $z \in [g]$ , where each  $\bar{\boldsymbol{\omega}}_z$  subsequently has elements  $\bar{\boldsymbol{\mu}}_z$  and  $\bar{\boldsymbol{\Sigma}}_z$ . Furthermore, for convenience, we define for  $\mathbf{s}$  the following notation

$$\mathbf{s}^\top = (s_{11}, \mathbf{s}_{21}, \text{vec}(\mathbf{S}_{31}), \dots, s_{1g}, \mathbf{s}_{2g}, \text{vec}(\mathbf{S}_{3g})),$$

and

$$[\bar{\mathbf{s}}(\mathbf{y}; \boldsymbol{\theta})]^\top = (\bar{s}_{11}(\mathbf{y}; \boldsymbol{\theta}), \bar{s}_{21}(\mathbf{y}; \boldsymbol{\theta}), \text{vec}(\bar{\mathbf{S}}_{31}(\mathbf{y}; \boldsymbol{\theta})), \dots, \bar{s}_{1g}(\mathbf{y}; \boldsymbol{\theta}), \bar{s}_{2g}(\mathbf{y}; \boldsymbol{\theta}), \text{vec}(\bar{\mathbf{S}}_{3g}(\mathbf{y}; \boldsymbol{\theta}))),$$

with

$$\bar{s}_{1z}(\mathbf{y}; \boldsymbol{\theta}) = \tau_z(\mathbf{y}; \boldsymbol{\theta}), \bar{s}_{2z}(\mathbf{y}; \boldsymbol{\theta}) = \tau_z(\mathbf{y}; \boldsymbol{\theta}) \mathbf{y}, \text{ and } \bar{\mathbf{S}}_{3z}(\mathbf{y}; \boldsymbol{\theta}) = \tau_z(\mathbf{y}; \boldsymbol{\theta}) \mathbf{y} \mathbf{y}^\top.$$

Then the application of the M-step is equivalent to apply function  $\bar{\boldsymbol{\theta}}$  as a function of  $\mathbf{s}$ , containing the unique elements of  $\bar{\pi}_z$ ,  $\bar{\boldsymbol{\mu}}_z$ , and  $\bar{\boldsymbol{\Sigma}}_z$ , for  $z \in [g]$ , defined by

$$\bar{\pi}_z(\mathbf{s}) = \frac{s_{1z}}{\sum_{j=1}^g s_{1j}}, \quad \bar{\boldsymbol{\mu}}_z(\mathbf{s}) = \frac{\mathbf{s}_{2z}}{s_{1z}}, \quad \text{and } \bar{\boldsymbol{\Sigma}}_z(\mathbf{s}) = \frac{\mathbf{S}_{3z}}{s_{1z}} - \frac{\mathbf{s}_{2z} \mathbf{s}_{2z}^\top}{s_{1z}^2}. \quad (25)$$

This implies that the mini-batch EM and truncated mini-batch EM algorithms proceed via update rule  $\bar{\boldsymbol{\theta}}(\mathbf{s}^{(r)})$ , where  $\bar{\boldsymbol{\theta}}$  and  $\mathbf{s}^{(r)}$  are as given in (18). We start from  $\boldsymbol{\theta}^{(0)}$  and  $\mathbf{s}^{(1)} =$

$N^{-1} \sum_{i=1}^N \bar{s}(\mathbf{Y}_i; \boldsymbol{\theta}^{(0)})$ . Then,  $\boldsymbol{\theta}^{(1)\top} = [\bar{\boldsymbol{\theta}}(\mathbf{s}^{(1)})]^\top$ , which has elements

$$\bar{\pi}_z(\mathbf{s}^{(1)}) = N^{-1} \sum_{i=1}^N \tau_z(\mathbf{Y}_i; \boldsymbol{\theta}^{(0)}), \quad \bar{\mu}_z(\mathbf{s}^{(1)}) = \frac{\sum_{i=1}^N \tau_z(\mathbf{Y}_i; \boldsymbol{\theta}^{(0)}) \mathbf{Y}_i}{\sum_{i=1}^N \tau_z(\mathbf{Y}_i; \boldsymbol{\theta}^{(0)})}, \quad (26)$$

and

$$\bar{\Sigma}_z(\mathbf{s}^{(1)}) = \frac{\sum_{i=1}^N \tau_z(\mathbf{Y}_i; \boldsymbol{\theta}^{(0)}) \mathbf{Y}_i \mathbf{Y}_i^\top}{\sum_{i=1}^N \tau_z(\mathbf{Y}_i; \boldsymbol{\theta}^{(0)})} - \frac{\left[ \sum_{i=1}^N \tau_z(\mathbf{Y}_i; \boldsymbol{\theta}^{(0)}) \mathbf{Y}_i \right] \left[ \sum_{i=1}^N \tau_z(\mathbf{Y}_i; \boldsymbol{\theta}^{(0)}) \mathbf{Y}_i \right]^\top}{\left[ \sum_{i=1}^N \tau_z(\mathbf{Y}_i; \boldsymbol{\theta}^{(0)}) \right]^2}. \quad (27)$$

### 3.3 Convergence analysis of the mini-batch algorithm

In addition to Assumptions A1–A3, B1–B3, and C1–C3, make the additional assumption

- D1 The Hessian matrix of  $\sum_{i=1}^n \log f(\mathbf{y}_i; \boldsymbol{\theta})$ , evaluated at any  $\boldsymbol{\theta}_0 \in \mathbb{M}_{\Theta}^{\text{Emp}}$ , is non-singular with respect to  $\boldsymbol{\theta} \in \Theta$ .

Assumption D1 is generally satisfied for all but pathological samples  $\{\mathbf{y}_i\}_{i=1}^n$ . The following result is proved in the Supplementary Materials.

**Proposition 3.** Let  $\{\mathbf{y}_i\}_{i=1}^n$  be a realization of some IID random sequence  $\{\mathbf{Y}_i\}_{i=1}^n$ , where each  $\mathbf{Y}_i$  is hypothesized to arise from a DGP having PDF of the form (3). If  $\{\mathbf{s}^{(r)}\}_{i=1}^\infty$  and  $\{\boldsymbol{\theta}^{(r)}\}_{i=1}^\infty$  are sequences generated by the mini-batch EM algorithm, defined by (18) and (25), then for any  $N \in \mathbb{N}$ , if C1, C2, and D1 are satisfied (replacing  $i$  by  $r$ , and  $F_0$  by  $\prod_{j=1}^N F_{\text{Emp}}$ , where appropriate), then, with probability 1,

$$\lim_{r \rightarrow \infty} \text{dist}(\mathbf{s}^{(r)}, \Gamma) = 0, \text{ and } \lim_{r \rightarrow \infty} \text{dist}(\boldsymbol{\theta}^{(r)}, \mathbb{M}_{\Theta}^{\text{Emp}}) = 0.$$

Alternatively, if  $\{\mathbf{s}^{(r)}\}_{i=1}^\infty$  and  $\{\boldsymbol{\theta}^{(r)}\}_{i=1}^\infty$  are sequences generated by the truncated mini-batch EM

algorithm, defined by (20), (16), (17) and (25), then for any  $N \in \mathbb{N}$ , if C1 and D1 are satisfied (replacing  $i$  by  $r$ , and  $F_0$  by  $\prod_{j=1}^N F_{Emp}$ , where appropriate), then, with probability 1,

$$\lim_{r \rightarrow \infty} dist(\mathbf{s}^{(r)}, \Gamma) = 0, \text{ and } \lim_{r \rightarrow \infty} dist(\boldsymbol{\theta}^{(r)}, \mathbb{M}_{\Theta}^{Emp}) = 0.$$

### 3.4 A truncation sequence

In order to apply the truncated version of the mini-batch EM algorithm, we require an appropriate sequence  $\{\mathbb{K}_m\}_{m=0}^\infty$  that satisfies condition (14). This can be constructed in parts. Let us write

$$\mathbb{K}_m = \mathbb{D}_{g-1}^m \times \prod_{i=1}^g (\mathbb{B}_d^m \times \mathbb{H}_d^m), \quad (28)$$

where we shall let  $c_1, c_2, c_3 \geq 1$ ,

$$\mathbb{D}_{g-1}^m = \left\{ (\pi_1, \dots, \pi_g) \in \mathbb{R}^g : \sum_{z=1}^g \pi_z = 1, \text{ and } \pi_z \geq \frac{1}{c_1 + m}, \text{ for each } z \in [g] \right\},$$

$$\mathbb{B}_d^m = [-(c_2 + m), c_2 + m]^d,$$

and

$$\mathbb{H}_d^m = \left\{ \mathbf{H} \in \mathbb{H}_d : \lambda_1(\mathbf{H}) \geq \frac{1}{c_3 + m}, \lambda_d(\mathbf{H}) \leq c_3 + m \right\},$$

using the notation  $\lambda_1(\mathbf{H})$  and  $\lambda_d(\mathbf{H})$  to denote the smallest and largest eigenvalues of the matrix  $\mathbf{H}$ . A justification regarding this truncation scheme can be found in the Supplementary Materials.

We make a final note that the construction (28) is not a unique method for satisfying the conditions of (14). One can instead, for example, replace  $c_j + m$ , by  $c_j(1 + m)$  ( $j \in [3]$ ) in the definitions of the sets that constitute (28).

## 4 Simulation studies

We present a pair of simulation studies, based upon the famous Iris data set of Fisher (1936) and the Wreath data of Fraley et al. (2005), in the main text. A further four simulation scenarios are presented in the Supplementary Materials. In each case, we utilize the initial small data sets, obtained from the base R package (R Core Team, 2018) and the `mclust` package for R (Scrucca et al., 2016), respectively, and use them as templates to generate much larger data sets. All computations are conducted in the R programming environment, although much of the bespoke programs are programmed in C and integrated in R via the `Rcpp` and `RcppArmadillo` packages of (Eddelbuettel, 2013). Furthermore, timings of programs were conducted on a MacBook Pro with a 2.2 GHz Intel Core i7 processor, 16 GB of 1600 MHz DDR3 RAM, and a 500 GB SSD hard drive. We note that all of the code used to conduct the simulations and computations for this manuscript can be accessed from <https://github.com/hiendn/StoEMMIX>.

In the sequel, in all instances, we shall use the learning rate sequence  $\{\gamma_r\}_{r=1}^{\infty}$ , where  $\gamma_r = (1 - 10^{-10}) \times r^{6/10}$ , which follows from the choice made by Cappé & Moulines (2009) in their experiments. In all computations, a fixed number of epochs (or epoch equivalence) of 10 is allotted to each algorithm. Here, recall that the number of epochs is equal to the number of sweeps through the data set  $\{\mathbf{y}_i\}_{i=1}^n$  that an algorithm is allowed. Thus, drawing  $10n$  observations from the data  $\{\mathbf{y}_i\}_{i=1}^n$ , with replacement, is equivalent to 10 epochs. Thus, each iteration of the standard EM algorithm counts as a single epoch, whereas, for a mini-batch algorithm with batch size  $N$ , every  $n/N$  iterations counts as an epoch.

Next, in both of our studies, we consider batch sizes of  $N = n/10$  and  $N = n/5$ , we further consider Polyak averaging as well as truncation. Thus, for each study, a total of eight variants of the mini-batch EM algorithm is considered. In the truncate case, we set  $c_1, c_2, c_3 = 1000$ . Finally,

the variants of the mini-batch EM algorithm are compared to the standard (batch) EM algorithm for fitting finite mixtures of normal distributions. In the interest of fairness, each of the algorithms is initialized at the same starting value of  $\boldsymbol{\theta}^{(0)}$ , using the randomized initialization scheme suggested in McLachlan & Peel (2000, Sec. 3.9.3). That is, the same randomized starting instance is used for the EM algorithm and each of the mini-batch variants.

To the best of our knowledge, the most efficient and reliable implementation of the EM algorithm for finite mixtures of normal distributions, in R, is the `em` function from the `mclust` package. Thus, this will be used for all of our comparisons. Data generation from the template data sets is handled using the `simdataset` function from the `MixSim` package (Melnykov et al., 2012), in the Iris study, and the `simVVV` function from `mclust` in the Wreath study. Timing was conducted using the `proc.time` function.

## 4.1 Iris data

The Iris data (accessed in R via the `data(iris)` command) contain measurements of  $d = 4$  dimensions from 150 iris flowers, 50 of each are of the species *Setosa*, *Versicolor*, and *Virginica*, respectively. The 4 dimensions of each flower that are measured are *petal length*, *petal width*, *sepal length*, and *sepal width*. To each of the subpopulations of species, we fit a single multivariate normal distribution to the 50 observations (i.e., we estimate a mean vector and covariance matrix, for each species). Then, using the three mean vectors and covariance matrices, we construct a template  $g = 3$  component normal mixture model with equal mixing proportions  $\pi_z = 1/3$  ( $z \in [3]$ ), of form (3). This template distribution is then used to generate synthetic data sets of any size  $n$ .

Two experiments are performed using this simulation scheme. In the first experiment, we generate  $n = 10^6$  observations  $\{\mathbf{y}_i\}_{i=1}^n$  from the template. We then utilize  $\{\mathbf{y}_i\}_{i=1}^n$  and each of the

truncated EM algorithm variants as well as the batch EM algorithm to compute ML estimates. We use a number of measures of performance for each algorithm variant. These include the computation time, the log-likelihood, the squared error of the parameter estimates (SE; the Euclidean distance as compared to the generative parameter vector), and the adjusted-Rand index (ARI; Hubert & Arabie, 1985) between the maximum *a posteriori* clustering labels obtained from the fitted mixture model and the true generative data labels.

The ARI measures whether or not two sets of labels are in concordance or not. Here a value of 1 indicates perfect similarity, and 0 indicates discordance. Since the ARI allows for randomness in the labelling process, it is possible to have negative ARI values, which are rare and also indicates discordance in the data. Each variant is repeated  $\text{Rep} = 100$  times, and each performance measurement is recorded in order to obtain a measure of the overall performance of each algorithm. For future reference, we name this study Iris1. In the second study, which we name Iris2, we repeat the setup of Iris1 but with the number of observations increased to  $n = 10^7$ .

## 4.2 Wreath data

The Wreath data (accessed in R via the `data(wreath)` command) contain 1000 observations of  $d = 2$  dimensional vectors, each belonging to one of  $g = 14$  distinct but unlabelled subpopulations. We use the `Mclust` function from `mclust` to fit a 14 component mixture normal distributions to the data. The data, along with the means of the subpopulation normal distributions, are plotted in Figure 1. Here, each observation is colored based upon the subpopulation that maximizes its *a posteriori* probability.

As with the Iris data, using the fitted mixture model as a template, we can then simulate synthetic data sets of any size  $n$ . We perform two experiments using this scheme. In the first

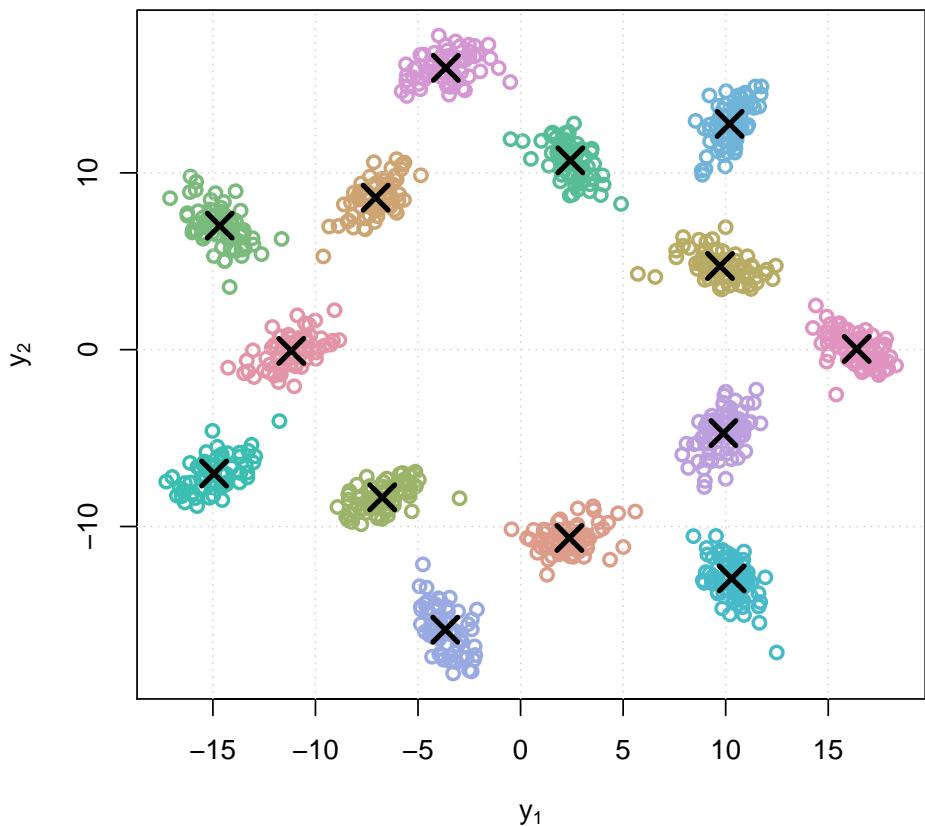


Figure 1: Plot of the 1000 observations of the Wreath data set, colored by subpopulation with subpopulation means indicated by crosses.

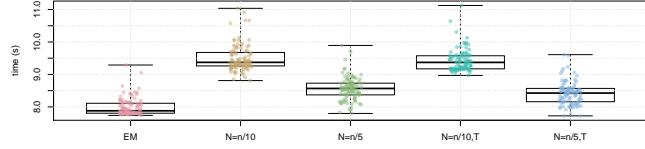
experiment, we simulate  $n = 10^6$  observations and assess the different algorithms, based on the computation time, the log-likelihood, the SE, and the ARI over Rep=100 repetitions, as per Iris1. We refer to this experiment as Wreath1. In the second experiment, we repeat the setup of Wreath1, but with  $n = 10^7$ , instead. We refer to this case as Wreath2.

### 4.3 Results

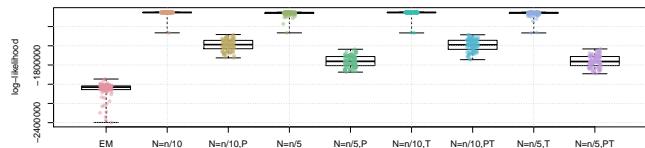
Figures 2 and 3 contain box plots that summarize the results of Iris1 and Iris2, respectively. Similarly, Figures 4 and 5 contain box plots that summarize the results of Wreath1 and Wreath2, respectively.

Firstly, we note that Polyak averaging requires no additional computational effort, for a given value of  $N$ . Thus, we do not require separate timing data for Polyak averaging variants of the mini-batch EM algorithms in each of Figures 2–5. From the timing results, we observe that the standard EM algorithm is faster than the mini-batch versions, in all scenarios, regardless of the fact that all of the algorithms were computed using 10 epochs worth of data access. This is because the mini-batch algorithms require more additional intermediate steps in each algorithm loop (e.g., random sampling from the empirical distribution), as well as a multiplicative factor of  $n/N$  more loops. From the plots, we observe that the larger value of  $N$  tends to result in smaller computing times. There appears to be no difference in timing between the use of truncation or not.

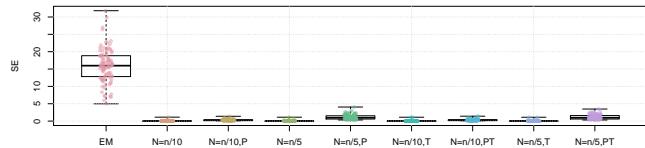
In the Iris1 and Iris2 studies, we observe that the mini-batch EM algorithms uniformly outperform the standard EM algorithm in terms of the log-likelihood, SE, and ARI. In all three measurements, we observe that  $N = n/10$  performed better than  $N = n/5$ , and also that there were no differences between truncated versions of the mini-batch algorithms and equivalent variants without truncation. Polyak averaging appears to reduce the performance of the mini-batch



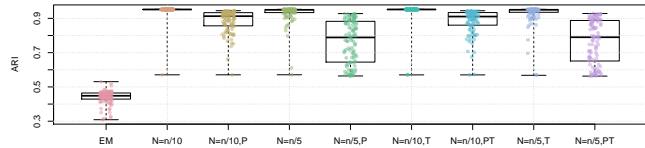
(a) Timing results, in seconds.



(b) Log-likelihood results.

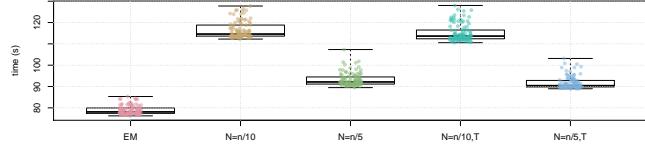


(c) Standard error results.

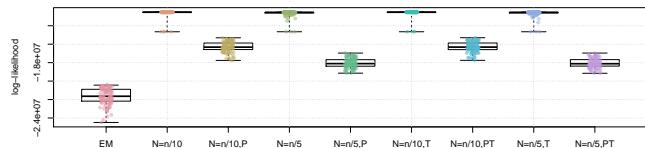


(d) Adjusted-Rand index results.

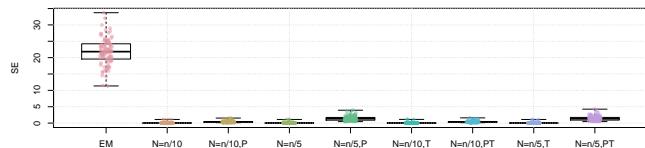
Figure 2: Results from Rep = 100 replications of the Iris1 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size  $N$  is indicated (either  $N = n/10$  or  $N = n/5$ ), and a 'P' or a 'T' designates that Polyak averaging or truncation was used, respectively.



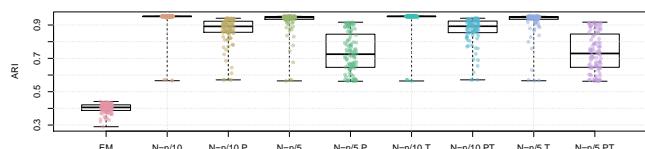
(a) Timing results, in seconds.



(b) Log-likelihood results.

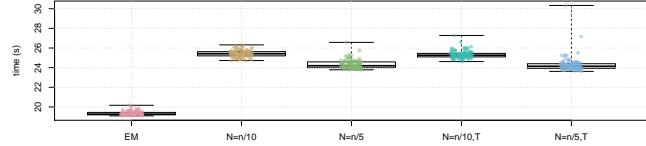


(c) Standard error results.

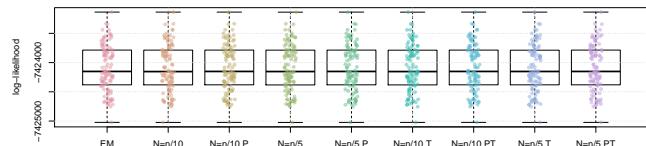


(d) Adjusted-Rand index results.

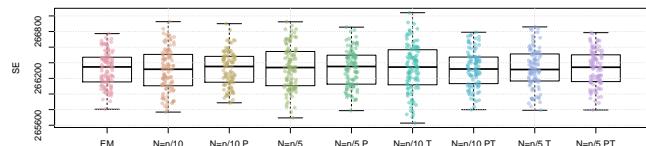
Figure 3: Results from Rep = 100 replications of the Iris2 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size  $N$  is indicated (either  $N = n/10$  or  $N = n/5$ ), and a 'P' or a 'T' designates that Polyak averaging or truncation was used, respectively.



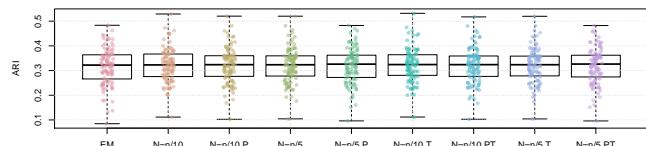
(a) Timing results, in seconds.



(b) Log-likelihood results.

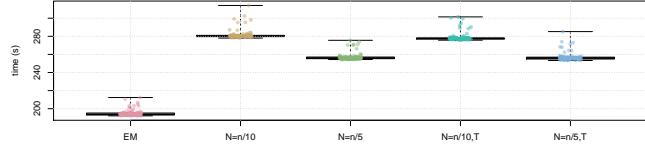


(c) Standard error results.

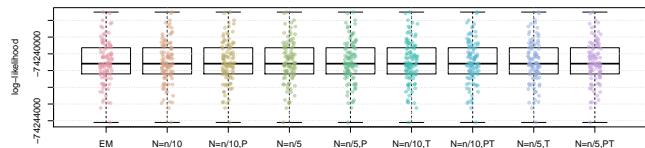


(d) Adjusted-Rand index results.

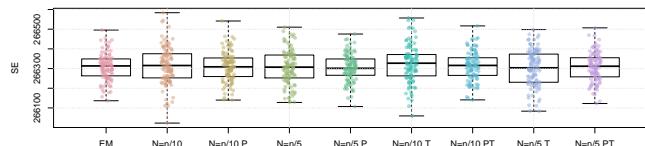
Figure 4: Results from Rep = 100 replications of the Wreath1 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size  $N$  is indicated (either  $N = n/10$  or  $N = n/5$ ), and a 'P' or a 'T' designates that Polyak averaging or truncation was used, respectively.



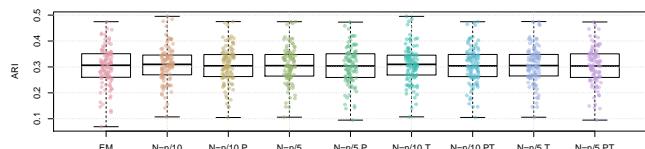
(a) Timing results, in seconds.



(b) Log-likelihood results.



(c) Standard error results.



(d) Adjusted-Rand index results.

Figure 5: Results from Rep = 100 replications of the Wreath2 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size  $N$  is indicated (either  $N = n/10$  or  $N = n/5$ ), and a 'P' or a 'T' designates that Polyak averaging or truncation was used, respectively.

algorithms, for a given level of  $N$ , with respect to each of the three measurements.

In the Wreath1 and Wreath2 studies, we observe that the standard and mini-batch EM algorithms perform virtually the same across the log-likelihood, SE, and ARI metrics. This is likely due to the high degree of separability of each of the  $g = 12$  mixture components of the Wreath data, in comparison to the overlapping components of the Iris data. Our observation is true for all of the mini-batch EM variants, with or without truncation or Polyak averaging. Upon first impression, this may appear as a weakness of the mini-batch EM algorithm, since it produces the same performance while requiring more computational time. However, we must also remember that the mini-batch EM algorithm does not require all of the data to be stored in memory at each iteration of algorithm, whereas the EM algorithm does. Thus, the mini-batch algorithm is feasible in very large data situations, since it only requires a fixed memory size, of order  $N$ , regardless of sample size  $n$ , whereas the standard EM algorithm has a memory requirement that increases with  $n$ .

To expand upon our currently presented results, we have also included a further two simulation studies regarding the fitting of finite mixtures of normal distributions using mini-batch EM algorithms. Our two studies are based on the Flea data of Wickham et al. (2011), a test scenario from the ELKI project of Schubert et al. (2015), and an original data generating process. The ELKI scenario is chosen due to its separability, in order to assess whether our conclusion regarding Wreath1 and Wreath2 are correct. The Flea data shares similarities with the Iris data but is higher dimensional. In all cases, we found that the mini-batch algorithms tended to outperform the EM algorithm in all but timing, on average. Detailed assessments of these studies can be found in the Supplementary Materials.

In addition, we have also investigated the use of the mini-batch EM algorithm for estimation

of non-normal mixture models. Namely, we present a pair of algorithms for the estimation of exponential and Poisson mixture models. We demonstrate their performance via an additional pair of simulation studies.

To conclude, we make the following recommendations. Firstly, smaller batch sizes appear to yield higher likelihood values. Secondly, averaging appears to slow convergence of the algorithm to the higher likelihood value and is thus not recommended. Thirdly, truncation appears to have no effect on the performance. This is likely due to the fact that truncation may not have been needed in any of the experiments. In any case, it is always useful to use the truncated version of the algorithm, in case there are unforeseen instabilities in the optimization process. And finally, the standard EM algorithm may be preferred to the mini-batch EM algorithm when sample sizes are small and when the data are highly separable. However, even in the face of high separability, for large  $n$ , it may not be feasible to conduct estimation by the standard EM algorithm and thus the mini-batch algorithms may be preferred due to feasibility.

It is interesting to observe that Polyak averaging tended to diminish the performance of the algorithms, in our studied scenarios. This is in contradiction to the theory that suggests that Polyak averaging should in fact increase the convergence rate to stationary solutions. We note, however, that the theory is asymptotic and the number of epochs that were used may be too short for the advantages of Polyak averaging to manifest, in practice.

## 5 Real data study

### 5.1 MNIST data

The MNIST data of LeCun et al. (1998) consists of  $n = 70,000$  observations of  $d = 28 \times 28 = 784$  pixel images of handwritten digits. These handwritten digits were sampled nearly uniformly. That is there were 6903, 7877, 6990, 7141, 6824, 6313, 6876, 7293, 6825, and 6958 observations of the digits 0–9, respectively.

Next, it is notable that not all  $d$  pixels are particularly informative. In fact, there is a great amount of redundancy in the  $d$  dimensions. Out of the  $d$  pixels, 65 are always zero, for every observation. Thus, the dimensions of the data are approximately 8.3% sparse.

We eliminate the spare pixels across all images to obtain a dense dimensionality of  $d_{\text{dense}} = 719$ . Using the  $d_{\text{dense}}$  dimensions of the data, we conduct a principal component analysis (PCA) in order to further reduce the data dimensionality; see Jolliffe, 2002 for a comprehensive treatment on PCA. Using the PCA, we extract the principal components (PCs) of each observation, and for various number of PCs  $d_{\text{PC}} \in [d_{\text{dense}}]$ . We can then use the data sets of  $n$  observations and dimension  $d_{\text{PC}}$ , to estimate mixture of normal distributions for various values of  $g$ .

### 5.2 Experimental setup

In the following study, we utilize only the truncated version of the mini-batch algorithm, having drawn the conclusions, from Section 4, that there appeared to be no penalty in performance due to truncation in practice. Again, drawing upon our experience from Section 4, we set  $N = n/10 = 7000$  as the batch size in all applications. The same learning rate sequence of  $\{\gamma_r\}_{r=1}^{\infty}$ , where  $\gamma_r = (1 - 10^{-10}) \times r^{6/10}$  is also used, and  $c_1, c_2, c_3 = 1000$ .

We apply the mini-batch algorithm to data with  $d_{\text{PC}} = 10, 20, 50, 100$ . Initialization of the parameter vector  $\boldsymbol{\theta}^{(0)}$  was conducted via the randomization scheme of McLachlan & Peel (2000, Sec. 3.9.3). The mini-batch algorithm was run 100 times for each  $d_{\text{PC}}$  and the log-likelihood values were recorded for both the fitted models using the Polyak averaging and no averaging versions of the algorithm. The standard EM algorithm, as applied via the `em` function of the `mclust` package is again used for comparison. Each of the algorithms, including the  $k$ -means algorithm, were initialized from the same initial randomization, in the interest of fairness, for each of the 100 runs. That is, a random partition of the data is generated once for each of the 100 runs, and the initial parameters for the EM, mini-batch and  $k$ -means algorithms are all computed from the same initialization. The log-likelihood values of the standard and mini-batch EM algorithms are compared along with the ARI values. Algorithms are run for 10 epochs.

We compute the ARI values obtained when comparing the maximum *a posteriori* clustering labels, obtained from each of the algorithms (cf. McLachlan & Peel, 2000, Sec. 1.15), and the true digit classes of each of the images. For a benchmark, we also compare the performance of the three EM algorithms with the  $k$ -means algorithm, as applied via the `kmeans` function in R, which implements the algorithm of Hartigan & Wong (1979). For fairness of comparison, we also allow the  $k$ -means algorithm 10 epochs in each of 100 runs. As in Section 4, we note that all codes are available at <https://github.com/hienndn/StoEMMIX>, for the sake of reproducibility and transparency.

### 5.3 Results

The results from the MNIST experiment are presented in Table 1. We observe that for  $d_{\text{PC}} \in \{10, 20, 50\}$ , all three EM variants provided better ARI than the  $k$ -means algorithm. The best

ARI values for all three EM algorithms occur when  $d_{\text{PC}} = 20$ . When  $d_{\text{PC}} = 100$ , the  $k$ -means algorithm provided a better ARI, which appeared to be somewhat uniform across the four values of  $d_{\text{PC}}$ .

Among the EM algorithms, the mini-batch algorithm provided better ARI values, with the two variants not appearing to be significantly different from one another, when considering the standard errors of the ARI values, when  $d_{\text{PC}} \in \{20, 50\}$ . When  $d_{\text{PC}} = 10$ , we observe that no averaging yielded a better ARI, whereas, when  $d_{\text{PC}} = 100$ , averaging appeared to be better, on average.

Regarding the log-likelihoods, the mini-batch EM algorithm, when applied without averaging, uniformly and significantly outperformed the standard EM algorithm. On the contrary, when applied with averaging, the EM algorithm uniformly and significantly outperformed the mini-batch algorithm. This is also in contrary with what was observed in Section 4. This is an interesting result considering that the ARI of the mini-batch algorithm, with averaging, is still better than that of the EM algorithm. As in Section 4, we can recommend the use of the mini-batch EM algorithm without averaging, as it tends to outperform the standard EM algorithm for fit and is also yields better clustering outcomes, when measured via the ARI.

## 6 Conclusions

In Section 2, we reviewed the online EM algorithm framework of Cappé & Moulines (2009), and stated the key theorems that guarantee the convergence of algorithms that are constructed under the online EM framework. We then presented a novel interpretation of the online EM algorithm that yielded our framework for constructing mini-batch EM algorithms. We then utilized the

Table 1: Tabulation of results from the 100 runs of the EM algorithms and the  $k$ -means algorithm, for each value of  $d_{PC} \in \{10, 20, 50, 100\}$ . The columns EM, Mini, and Mini Pol refer to the standard EM, the mini-batch EM, and the mini-batch EM algorithm with Polyak averaging, respectively. The SE rows contain the standard error over each of the 100 runs (i.e. the standard deviation over 10). Boldface text highlight the best results.

$d_{PC}$		ARI				log-likelihood			
		EM	Mini	Mini Pol	$k$ -means	EM	Mini	Mini Pol	
10	Mean	0.401	<b>0.443</b>	0.432	0.352	-4.98E+06	<b>-4.96E+06</b>	-5.01E+06	
	SE	0.004	0.004	0.004	0.002	1.19E+03	6.43E+02	5.90E+02	
20	Mean	0.436	0.475	<b>0.480</b>	0.367	-9.46E+06	<b>-9.44E+06</b>	-9.52E+06	
	SE	0.005	0.005	0.005	0.002	2.20E+03	1.37E+03	1.67E+03	
50	Mean	0.394	0.434	<b>0.438</b>	0.369	-2.18E+07	<b>-2.17E+07</b>	-2.20E+07	
	SE	0.005	0.005	0.006	0.002	8.47E+03	7.01E+03	4.85E+03	
100	Mean	0.326	0.356	<b>0.377</b>	0.372	-3.99E+07	<b>-3.97E+07</b>	-4.05E+07	
	SE	0.004	0.004	0.005	0.002	1.83E+04	1.68E+04	8.72E+03	

theorems of Cappé & Moulines (2009) in order to produce convergence results for this new mini-batch EM algorithm framework. Extending upon some remarks of Cappé & Moulines (2009), we also made rigorous the use of truncation in combination with both the online EM and mini-batch EM algorithm frameworks, using the construction and theory of Delyon et al. (1999).

In Section 3, we demonstrated how the mini-batch EM algorithm framework could be applied to construct algorithms for conducting ML estimation of finite mixtures of exponential family distributions. A specific analysis is made of the particularly interesting case of the normal mixture models. Here, we validate the conditions that permit the use of the Theorems from Section 2 in order to guarantee the convergence of the mini-batch EM algorithms for ML estimation of normal mixture models.

In Section 4, we conducted a set of four simulation studies in order to study the performance of the mini-batch EM algorithms, implemented in eight different variants, as compared to the standard EM algorithm for ML estimation of normal mixture models. There, we found that

regardless of implementation, in many cases, the mini-batch EM algorithms were able to obtain log-likelihood values that were better on average than the standard EM algorithm. We also found that the use of larger batch sizes and Polyak averaging tended to diminish performance of the mini-batch algorithms, but the use of truncation tended to have no effect. Although the mini-batch algorithms is generally slower than the standard EM algorithm, we note that in many cases, the fixed memory requirement of the mini-batch algorithms make them feasible where the standard EM algorithm is not.

A real data study was conducted in Section 5. There, we explored the use of the standard EM algorithm and the truncated mini-batch EM algorithm for cluster analysis of the famous MNIST data of LeCun et al. (1998). From our study, we found that the mini-batch EM algorithm was able to obtain better log-likelihood values than the standard EM algorithm, when applied without Polyak averaging. However, with averaging, the mini-batch EM algorithm was worse than the standard EM algorithm, on average. However, regardless of whether averaging was used, or not, the mini-batch EM algorithm appeared to yield better clustering outcomes, when measured via the ARI of Hubert & Arabie (1985).

This research poses numerous interesting directions for the future. First, we may extend the results to other exponential family distributions that permit the satisfaction of theorem assumptions from Section 2. We make initial steps in this direction via a pair of mini-batch algorithms for exponential and Poisson distribution mixtures. Secondly, we may use the framework to construct mini-batch algorithms for large-scale mixture of regression models (cf. Jones & McLachlan, 1992), following the arguments made by Capp   & Moulines (2009) that permitted them to construct an online EM algorithm for their mixture of regressions example analysis. Thirdly, this research theme can be extended further to the construction of mini-batch algorithms for mixture of experts models

(cf. Nguyen & Chamroukhi, 2018), which may be facilitated via the Gaussian gating construction of Xu et al. (1995).

In addition to the three previous research questions, we may also ask questions regarding the practical application of the mini-batch algorithms. For instance, we may consider the question of optimizing learning rates and batch sizes for particular application settings. Furthermore, we may consider whether the theoretical framework still applies to algorithms where we may have adaptive batch sizes and learning rate regimes. As these directions fall vastly outside the scope of the current paper, we shall leave them for future exploration.

## Acknowledgements

The authors are indebted to the Co-ordinating Editor and two Reviewers for their insightful comments that have improved the exposition of the manuscript. HDN is personally funded by Australian Research Council (ARC) grant DE170101134. GJM and HDN are also funded under ARC grant DP180101192. The work is supported by Inria project LANDER.

## References

- Amari, S. (2016). *Information Geometry and Its Applications*. Japan: Springer.
- Bouveyron, C., Girard, S., & Schmid, C. (2007). High-dimensional data clustering. *Computational Statistics and Data Analysis*, 52, 502–519.
- Buhlmann, P., Drineas, P., Kane, M., & van der Laan, M., Eds. (2016). *Handbook of Big Data*. Boca Raton: CRC Press.

- Cappé, O. & Moulines, E. (2009). On-line expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society B*, 71, 593–613.
- Celeux, G., Chretien, S., Forbes, F., & Mkhadri, A. (2001). A component-wise EM algorithm for mixtures. *Journal of Computational and Graphical Statistics*, 10, 697–712.
- Chau, M. & Fu, M. C. (2015). An overview of stochastic approximation. In M. C. Fu (Ed.), *Handbook of Simulation Optimization* (pp. 149–178). New York: Springer.
- Chen, H.-F. (2003). *Stochastic Approximation and Its Applications*. New York: Kluwer.
- Cotter, A., Shamir, O., Srebro, N., & Sridharan, K. (2011). Better mini-batch algorithms via accelerated gradient methods. In *Advances in Neural Information Processing Systems* (pp. 1647–1655).
- DasGupta, A. (2011). *Probability for Statistics and Machine Learning*. New York: Springer.
- Delyon, B., Lavielle, M., & Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, 27, 94–128.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39, 1–38.
- Eddelbuettel, D. (2013). *Seamless R and C++ Integration with Rcpp*. New York: Springer.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, (pp. 179–188).
- Forbes, C., Evans, M., Hastings, N., & Peacock, B. (2011). *Statistical Distributions*. New York: Wiley.

- Fraley, C., Raftery, A., & Wehrens, R. (2005). Incremental model-based clustering for large datasets with small clusters. *Journal of Computation and Graphical Statistics*, 14, 529–546.
- Ghadimi, S., Lan, G., & Zhang, H. (2016). Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming Series A*, 155, 267–305.
- Han, Z., Hong, M., & Wang, D. (2017). *Signal Processing and Networking for Big Data Applications*. Cambridge: Cambridge University Press.
- Hardle, W. K., Lu, H. H.-S., & Shen, X., Eds. (2018). *Handbook of Big Data Analytics*. Cham: Springer.
- Hartigan, J. A. & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society Series C*, 28, 100–108.
- Hubert, L. & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2, 193–218.
- Iverson, K. E. (1967). *A Programming Language*. New York: Wiley.
- Jolliffe, I. T. (2002). *Principal Component Analysis*. New York: Springer.
- Jones, P. N. & McLachlan, G. J. (1992). Fitting finite mixture models in a regression context. *Australian Journal of Statistics*, 34, 233–240.
- Kiefer, J. & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23, 462–466.
- Kullback, S. & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22, 79–86.

- Kushner, H. J. & Yin, G. G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications*. New York: Springer.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 2278–2324.
- Li, M., Zhang, T., Chen, Y., & Smola, A. J. (2014). Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 661–670).
- Liang, F. & Zhang, J. (2008). Estimating the false discovery rate using the stochastic approximation algorithm. *Biometrika*, 95, 961–977.
- McLachlan, G. J. & Krishnan, T. (2008). *The EM Algorithm And Extensions*. New York: Wiley.
- McLachlan, G. J., Lee, S. X., & Rathnayake, S. I. (2019). Finite mixture models. *Annual Review of Statistics and Its Application*. In press.
- McLachlan, G. J. & Peel, D. (2000). *Finite Mixture Models*. New York: Wiley.
- Melnykov, V., Chen, W.-C., & Maitra, R. (2012). MixSim: an R package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, 51, 1–25.
- Ng, S.-K. & McLachlan, G. J. (2004). Speeding up the EM algorithm for mixture model-based segmentation of magnetic resonance images. *Pattern Recognition*, 37, 1573–1589.
- Nguyen, H. D. & Chamroukhi, F. (2018). Practical and theoretical aspects of mixture-of-experts modeling: an overview. *WIREs Data Mining and Knowledge Discovery*, (pp. e1246).

- Nguyen, H. D. & Jones, A. T. (2018). Big Data-appropriate clustering via stochastic approximation and Gaussian mixture models. In M. Ahmed & A.-S. K. Pathan (Eds.), *Data Analytics: Concepts, Techniques, and Applications*. Boca Raton: CRC Press.
- Nguyen, H. D. & McLachlan, G. J. (2015). Maximum likelihood estimation of Gaussian mixture models without matrix operations. *Advances in Data Analysis and Classification*, 9, 371–394.
- Pearson, K. (1894). Contributions to the theory of mathematical evolution. *Philosophical Transactions of the Royal Society of London A*, 185, 71–110.
- Polyak, B. T. (1990). A new method of stochastic approximation type. *Automatic and Remote Control*, 51, 98–107.
- Polyak, B. T. & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal of Control and Optimization*, 30, 838–855.
- Prosperetti, A. (2011). *Advanced Mathematics for Applications*. Cambridge: Cambridge University Press.
- R Core Team (2018). *R: a language and environment for statistical computing*. R Foundation for Statistical Computing.
- Robbins, H. & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22, 400–407.
- Schubert, E., Koos, A., Emrich, T., Zufle, A., Schmid, K. A., & Zimek, A. (2015). A framework for clustering uncertain data. *Proceedings of the VLDB Endowment*, 8, 1976–1979.

- Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). mclust: clustering, classification and density estimation using Gaussian finite mixture models. *R Journal*, 8, 289–317.
- Vlassis, N. & Likas, A. (2002). A greedy EM algorithm for Gaussian mixture learning. *Neural Processing Letters*, 15, 77–87.
- White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, 50, 1–25.
- White, H. (2001). *Asymptotic Theory For Econometricians*. San Diego: Academic Press.
- Wickham, H., Cook, D., Hofmann, H., & Buja, A. (2011). tourr: an R package for exploring multivariate data with projections. *Journal of Statistical Software*, 40, 1–18.
- Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *Annals of Statistics*, 11, 95–103.
- Xu, L., Jordan, M. I., & Hinton, G. E. (1995). An alternative model for mixtures of experts. In *Advances in Neural Information Processing Systems* (pp. 633–640).
- Zhang, J. & Liang, F. (2008). Convergence of stochastic approximation algorithms under irregular conditions. *Statistica Neerlandica*, 62, 393–403.
- Zhao, T., Yu, M., Wang, Y., Arora, R., & Liu, H. (2014). Accelerated mini-batch randomized block coordinate descent method. In *Advances in Neural Information Processing Systems* (pp. 3329–3337).

# Magnetic resonance fingerprinting

Dan Ma<sup>1</sup>, Vikas Gulani<sup>1,2</sup>, Nicole Seiberlich<sup>1</sup>, Kecheng Liu<sup>3</sup>, Jeffrey L. Sunshine<sup>2</sup>, Jeffrey L. Duerk<sup>1,2</sup> & Mark A. Griswold<sup>1,2</sup>

**Magnetic resonance is an exceptionally powerful and versatile measurement technique. The basic structure of a magnetic resonance experiment has remained largely unchanged for almost 50 years, being mainly restricted to the qualitative probing of only a limited set of the properties that can in principle be accessed by this technique. Here we introduce an approach to data acquisition, post-processing and visualization—which we term ‘magnetic resonance fingerprinting’ (MRF)—that permits the simultaneous non-invasive quantification of multiple important properties of a material or tissue. MRF thus provides an alternative way to quantitatively detect and analyse complex changes that can represent physical alterations of a substance or early indicators of disease. MRF can also be used to identify the presence of a specific target material or tissue, which will increase the sensitivity, specificity and speed of a magnetic resonance study, and potentially lead to new diagnostic testing methodologies. When paired with an appropriate pattern-recognition algorithm, MRF inherently suppresses measurement errors and can thus improve measurement accuracy.**

Magnetic resonance techniques such as NMR spectroscopy and magnetic resonance imaging (MRI) are widely used throughout physics, biology and medicine because of their ability to generate detailed information about numerous important material or tissue properties, including those reflective of many common disease states<sup>1–4</sup>. However, in practice magnetic resonance acquisitions are often restricted to a qualitative or ‘weighted’ measurement of a limited set of these properties; the magnetic resonance signal intensity is almost never quantitative by itself. The same material can have different intensities in different data sets depending on many factors, including the type and set-up of the scanner, the detectors used, and so on. Because of this, the quantitative analysis of magnetic resonance results typically focuses on differences between spectral peaks, spatial locations or different points in time. Even in clinical MRI today, a tissue or material is typically referred to as being ‘hyperintense’ or ‘hypointense’ compared to another area, which may not provide a quantitative indication of the severity of the differences, and may have reduced sensitivity to global changes. Thus robust, fully quantitative multiparametric acquisition has long been the goal of research in magnetic resonance<sup>5–8</sup>. However, the quantitative methods developed to date typically provide information on a single parameter at a time, require significant scan time, and are often highly sensitive to system imperfections. Simultaneous, multiparametric measurements are almost always impractical owing to scan time limits and a high sensitivity to the measurement set-up and experimental conditions. Thus purely qualitative magnetic resonance measurements remain the standard today, particularly in clinical MRI.

Here we introduce a novel approach, namely MRF, that may overcome these constraints by taking a completely different approach to data acquisition, post-processing and visualization. Instead of using a repeated, serial acquisition of data for the characterization of individual parameters of interest, MRF uses a pseudorandomized acquisition that causes the signals from different materials or tissues to have a unique signal evolution or ‘fingerprint’ that is simultaneously a function of the multiple material properties under investigation. The processing after acquisition involves a pattern recognition algorithm to match the fingerprints to a predefined dictionary of predicted signal evolutions. These can then be translated into quantitative maps of the magnetic parameters of interest.

MRF is related to the concept of compressed sensing<sup>9–12</sup>, and shares many of its predicted benefits. For example, preliminary results show that MRF could acquire fully quantitative results in a time comparable to a traditional qualitative magnetic resonance scan, without the high sensitivity to measurement errors found in many other fast methods. Most importantly, MRF has the potential to quantitatively examine many magnetic resonance parameters simultaneously given enough scan time, whereas current magnetic resonance techniques can only examine a limited set of parameters at once. Thus MRF opens the door to computer-aided multiparametric magnetic resonance analyses, similar to genomic or proteomic analyses, that could detect important but complex changes across a large number of magnetic resonance parameters simultaneously. When an appropriate pattern recognition algorithm is used, MRF also provides a new and more robust behaviour in the presence of noise or other acquisition errors that may lead to the near complete suppression of deleterious effects stemming from these factors. Although we focus on demonstrating the feasibility for MRI in this study, it is rather straightforward to translate these results to other magnetic resonance fields, such as multiparametric NMR spectroscopy, dynamic contrast enhanced MRI and dynamic susceptibility contrast MRI<sup>13</sup>.

## Generation and recognition of MRF signals

The key assumption underlying the MRF concept is that unique signal evolutions, or fingerprints, can be generated for different materials or tissues using an appropriate acquisition scheme. Here we demonstrate that this is possible through the continuous variation of the acquisition parameters throughout the data collection. Variations in the pulse sequence parameters during acquisition have been used previously in MRI and magnetic resonance spectroscopy to reduce the signal oscillations<sup>14</sup> and to improve the spectral response<sup>15–17</sup>. However, these variations were primarily used in a preparation phase or to make the signal more constant. Randomized sampling patterns have also been used previously to aid in the separation of spatiotemporal signals in moving objects or substances with different resonance frequencies<sup>18–20</sup>. Here we demonstrate that temporal and spatial incoherence required in MRF can be achieved by varying acquisition parameters—such as the flip angle and phase of radio frequency pulses, the repetition time, echo time and sampling patterns—in a pseudorandom manner.

<sup>1</sup>Department of Biomedical Engineering, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, Ohio 44106, USA. <sup>2</sup>Department of Radiology, Case Western Reserve University and University Hospitals of Cleveland, 11100 Euclid Avenue, Cleveland, Ohio 44106, USA. <sup>3</sup>Siemens Healthcare USA, 51 Valley Stream Parkway, Malvern, Pennsylvania 19355, USA.

After the data are acquired, the separation of the signal into different material or tissue types can be achieved through pattern recognition. In its simplest form, this process is analogous to matching a person's real fingerprint to a database: once a match is made, a host of additional information about the person, such as name, address and phone number, can be obtained simultaneously once the fingerprint sample is identified. In MRF, this pattern recognition can take place through many means. In the current implementation, we construct a dictionary that contains signal evolutions from all foreseeable combinations of materials and system-related parameters—for example, the longitudinal relaxation time,  $T_1$ , the transverse relaxation time,  $T_2$  and off-resonance frequency are included in this study. Other properties could also be measured, such as diffusion and magnetization transfer using the well-established Bloch equation formalism of magnetic resonance<sup>21,22</sup>. Once this dictionary of possible signal evolutions is generated, a matching or pattern recognition algorithm<sup>23,24</sup> is then used to select a signal vector or a weighted set of signal vectors from the dictionary that best correspond to the observed signal evolution. All the parameters that were used to build this signal vector in the dictionary can then be retrieved simultaneously. At present, the calculation of a complete dictionary containing the realistic range of  $T_1$ ,  $T_2$  and off-resonance frequency requires only a few minutes on a modern desktop computer.

It should be noted that there are near-infinite possibilities for MRF-compatible pulse sequences. Other magnetic resonance parameters of interest can be investigated by identifying pulse sequence components that impart differential sensitivity to the parameters of interest. Moreover, different components can be varied simultaneously, adding the potential for a highly efficient experimental design that allows almost any material characteristic visible using magnetic resonance to be analysed in a quantitative way using MRF.

### Validation of the concept

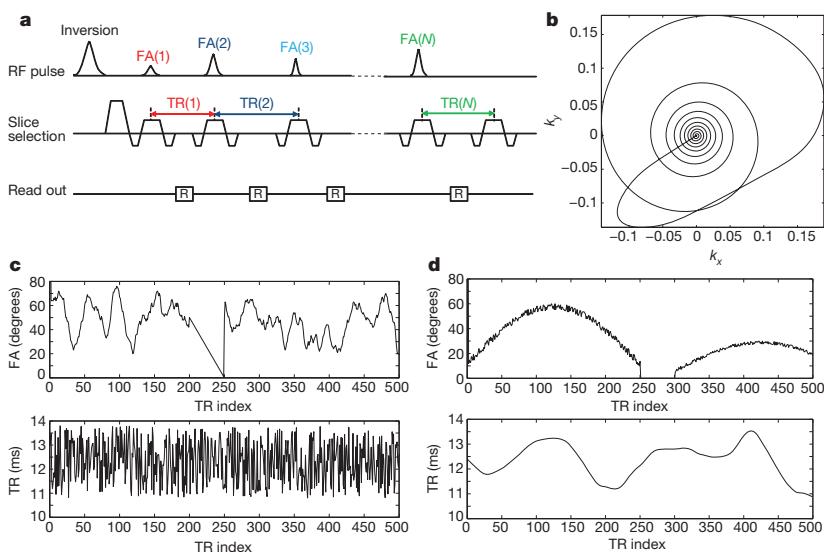
For a proof-of-principle implementation, an MRF acquisition based on an inversion-recovery balanced steady state free-precession (IR-bSSFP) sequence was used (Fig. 1a). This choice of this basic pulse sequence was based on the extensive existing knowledge about IR-bSSFP signal evolution, and its sensitivity to  $T_1$ ,  $T_2$  and off-resonance frequency<sup>25</sup>. After each radio-frequency pulse, one interleaf of a variable density spiral (VDS) readout<sup>26</sup> was acquired, as shown in Fig. 1b. Such a VDS trajectory has been used in fast imaging<sup>27</sup> and for the reduction of undersampling errors<sup>28</sup>. Two MRF acquisition patterns with randomized flip angle and repetition time were used as shown in

Fig. 1c and d in separate scans to demonstrate the flexibility of the choice of the acquisition parameters.

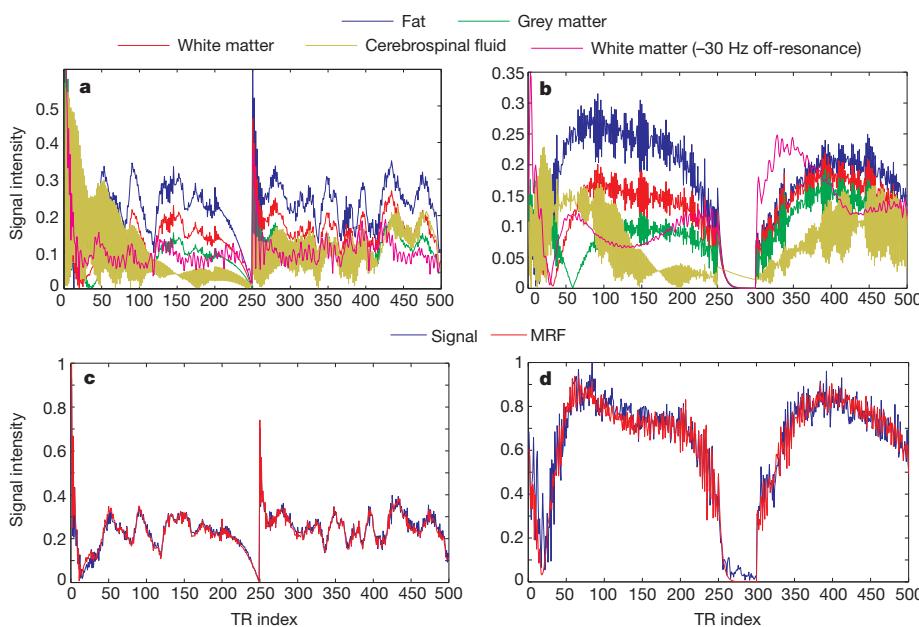
Figure 2a and b show the simulated signal evolution curves that would be expected from four commonly encountered tissues of the brain (fat, white matter, grey matter and cerebrospinal fluid) using the schematic implementation shown in Fig. 1c and d, respectively. Each tissue type has characteristic  $T_1$  and  $T_2$  values and thus each signal evolution has a different shape, which confirms that it is possible to satisfy this fundamental assumption in MRF. Note also that the signal levels in these evolutions represent a large fraction of the equilibrium magnetization (which is normalized to one in these figures.) Conventional spoiled steady-state sequences typically generate signal levels corresponding to 1–10% of the equilibrium magnetization. Figure 2c and d show an acquired signal evolution curve from fully sampled experiments on manufactured agar ‘phantoms’ and its match to the dictionary by using the acquisition pattern shown in Fig. 1c and d, along with the recovered  $T_1$ ,  $T_2$ , proton density ( $M_0$ ) and off-resonance frequency values. MRF was able to match the signal to the corresponding dictionary entry and obtain the same  $T_1$  and  $T_2$  values from both sequence patterns. A video of the signal evolution from a fully sampled *in vivo* scan is available (Supplementary Video 1), demonstrating the oscillating nature of the MRF signal observed *in vivo*.

### Accelerated MRF acquisitions

In addition to simultaneously quantifying multiple parameters, the error tolerance of MRF can be significantly better than that of conventional MRI. Because MRF is based on pattern recognition in a setting where the form of all predicted signal evolutions is known, MRF should be less sensitive to errors during the measurement. This is similar to conventional fingerprint recognition techniques, which often contend with smudges and partial fingerprint information. In particular, the interaction of the temporal and spatial incoherence possible in MRF provides new opportunities to accelerate image acquisition through rejection of spatial undersampling errors. In order to test the limits of this acceleration, the same MRF sequence as shown in Fig. 1a–c was modified to use only one spiral readout in each acquisition block. Therefore, the data collected are only 1/48th of the normally required data at each time point, resulting in a total acquisition time of 12.3 s, corresponding to 1,000 sampled time points. (See Fig. 3a and Supplementary Video 2.) The signal evolutions from all 1,000 undersampled time points were used directly to match one entry from the dictionary to quantify  $T_1$ ,  $T_2$ ,  $M_0$  and



**Figure 1 | MRF sequence pattern.** a, Acquisition sequence diagram. In each subsequent acquisition block, identified by a repetition time index (TR index; TR(1)...TR(N)), various sequence components are varied in a pseudorandom pattern. FA, flip angle. b, Here, one variable density spiral trajectory was used per repetition time. The Fourier coefficients sampled by the variable density spiral trajectory (given by the coordinates  $k_x$  and  $k_y$ ) are rotated from one repetition time to the next. c, d, Examples of the first 500 points of flip angle and repetition time patterns that were used in this study.



**Figure 2 | Signal properties and matching results from phantom study.** **a, b**, Simulated signal evolution curves corresponding to four normal brain tissues using the sequence patterns in Fig. 1c and d, respectively, as a fraction of the equilibrium magnetization. The curve from white matter with off-resonance is also plotted. **c, d**, Measured signal evolutions from one of eight phantoms using different sequence patterns and their dictionary match. The estimated  $T_1$ ,  $T_2$  and off-resonance frequencies are 340 ms, 50 ms and  $-4$  Hz (**c**) and 340 ms, 50 ms,  $-13$  Hz (**d**). The plots are normalized to their maximum value.

off-resonance simultaneously, as shown in Fig. 3b. Because these errors are incoherent with the expected MRF signals, they are largely ignored by the following processing steps. Figure 3c–f shows that high quality estimates of the magnetic resonance parameters are generated even with this significant level of undersampling. White-matter, grey-matter and cerebrospinal-fluid regions were then selected from the resultant maps. The mean  $T_1$  and  $T_2$  values obtained from each region are listed in Table 1 and are within the range of previously reported literature values<sup>29–32</sup>. The shortened  $T_2$  value in CSF is probably due to out-of-plane flow in this two-dimensional experiment. A similar effect can also be observed in conventional  $T_2$  mapping techniques<sup>33</sup>. We also note that the roughly  $-220$  Hz chemical shift of fat protons is clearly visualized in the off-resonance map.

### Motion error tolerance in MRF

Because motion is one of the most common sources of error in an MRI scan, a motion-corrupted scan was performed using the accelerated MRF acquisition described in the previous section. The subject was instructed to randomly move his head for the last 3 s of a total 15-s scan. Supplementary Video 3 shows the random motion as well as severe undersampling artefacts in the reconstructed images. Figure 4 compares the quantitative maps from the data with and without the motion-corrupted data. The maps acquired during motion show almost no sensitivity to the motion, and show nearly the same quality and anatomy as the maps from the motion-free data, thus indicating that the signal changes resulting from motion were uncorrelated with the evolutions included in the dictionary, and were largely ignored by the pattern recognition algorithm.

### Accuracy and efficiency of MRF

The accuracy and efficiency of the MRF acquisitions were compared with alternative mapping strategies, namely, standard spin-echo sequences<sup>34</sup> as well as modern rapid combined  $T_1$  and  $T_2$  mapping methods DESPOT1 and DESPOT2 (driven equilibrium single pulse observation of  $T_1$  and  $T_2$ , respectively)<sup>30</sup> using manufactured agar phantoms. Figure 5a compares the phantom  $T_1$  and  $T_2$  values from these methods. The concordance coefficient correlations for  $T_1$  and  $T_2$  between MRF and spin-echo sequence were 0.988 and 0.974, respectively. The concordance coefficient correlations for  $T_1$  and  $T_2$  between DESPOT and spin-echo sequence were 0.956 and 0.914, respectively.

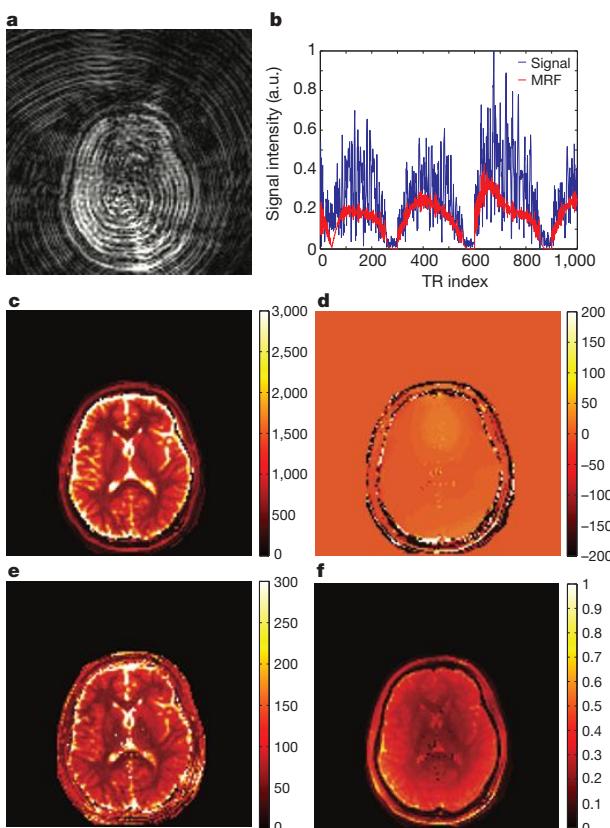
The high concordance correlation coefficients indicate that both methods are in good agreement with standard spin-echo measurements, and that MRF shows a better accuracy than DESPOT1 and DESPOT2.

The theoretical comparison of the efficiency from various mapping methods has been presented<sup>35,36</sup> and is based on a measure of precision per square root of scan time. In those publications, DESPOT1 and DESPOT2 were shown to have greater efficiency than all previously known conventional and accelerated mapping strategies<sup>36</sup>. As can be seen in Fig. 5b, MRF outperforms both DESPOT1 and DESPOT2 by an average factor of 1.87 and 1.85, respectively. For example, at a  $T_1$  of  $\sim 1,280$  ms, MRF shows an average efficiency for estimation of  $T_1$  of 24.2, whereas DESPOT1 has an average efficiency of 10.89. This means that for this  $T_1$  value, MRF achieves a precision of  $\pm 15.2$  ms (or 1.2%) in 12 s of scan time, whereas the precision in DESPOT1 would be  $\pm 33.9$  ms (or 2.6%) for the same scan time. The DESPOT methods apparently display higher efficiency from the single phantom, with  $T_1$  of 360 ms and  $T_2$  of 53 ms. However, in this one particular phantom, DESPOT overestimated the values of  $T_1$  and  $T_2$  by 23% and 42%, respectively, compared to the standard values, as can be seen in Fig. 5a, thus causing an erroneous increase in the apparent efficiency. Note that these efficiency estimates do not include the waiting times between the acquisition of the different sub-sequences in DESPOT, nor do they include the time required to reach steady state during each acquisition, and thus should be viewed as conservative estimates of the performance of MRF when compared to DESPOT1 or DESPOT2.

Because there is no steady state in the signal evolution from MRF, new information will be continuously added by longer acquisitions. Figure 5c and d illustrates the changes of mean and standard deviation as different acquisition times were used to quantify  $T_1$  and  $T_2$ , with a clear trend towards lower error at longer acquisition times. Thus one can select a trade-off between precision and scan time.

### Discussion and conclusions

The MRF concept presented here is a new approach to magnetic resonance and provides many opportunities to extend such measurements beyond their current limits. This originates from the unique pulse sequence design concept in MRF, where the goal is to generate unique signal evolutions that can be matched to theoretical signal evolutions and subsequently yield underlying quantitative information



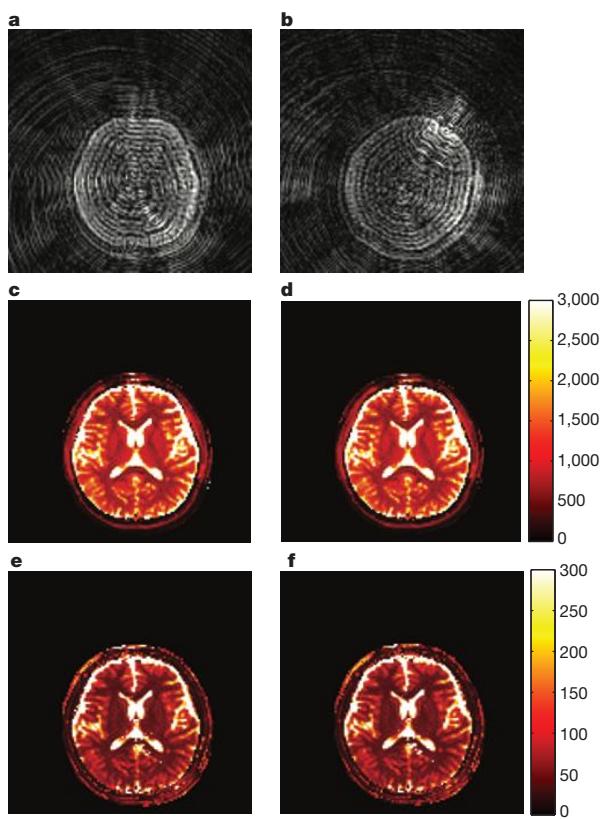
**Figure 3 | MRF results from highly undersampled data.** **a**, An image at the 5th repetition time out of 1,000 was reconstructed from only one spiral readout, demonstrating the significant errors from undersampling. **b**, One example of acquired single evolution and its match to the dictionary. Note the significant interference resulting from the undersampling. a.u., arbitrary units. **c–f**, The reconstructed parameter maps show a near complete rejection of these errors based solely on the incoherence between the underlying MRF signals and the undersampling errors; **c**,  $T_1$  (colour scale, milliseconds); **d**, off-resonance frequency (colour scale, hertz); **e**,  $T_2$  (colour scale, milliseconds); and **f**, proton density ( $M_0$ ) (normalized colour scale). These data required 12.3 s to acquire.

about the material, tissue or pathology of interest. Because there is no *a priori* requirement on the shape of the signal evolution curves, there are more degrees of freedom in designing an MRF acquisition, where parameters such as repetition time, echo time, radio frequency pulses and sampling trajectories (among others) can be varied together to produce the simultaneous sensitivity to numerous tissue properties. The ability to analyse oscillating signals in MRF also provides the opportunity to use larger fractions of the available magnetization than methods that rely on a steady-state signal, which is a significant factor contributing to the higher efficiency in MRF. In addition, the oscillatory signal in MRF allows one to sample more informative points along a longer signal evolution as compared to conventional methods which always reach a steady state level after some finite amount of

**Table 1 | In vivo data**

	$T_1$ (ms)	$T_2$ (ms)
White matter (this work)	$685 \pm 33$	$65 \pm 4$
White matter (previously reported)	$608\text{--}756$	$54\text{--}81$
Grey matter (this work)	$1,180 \pm 104$	$97 \pm 5.9$
Grey matter (previously reported)	$998\text{--}1,304$	$78\text{--}98$
Cerebrospinal fluid (this work)	$4,880 \pm 379$	$550 \pm 251$
Cerebrospinal fluid (previously reported)	$4,103\text{--}5,400$	$1,800\text{--}2,460$

Shown are comparisons of MRF results and reference values<sup>29–32</sup> in different brain regions.

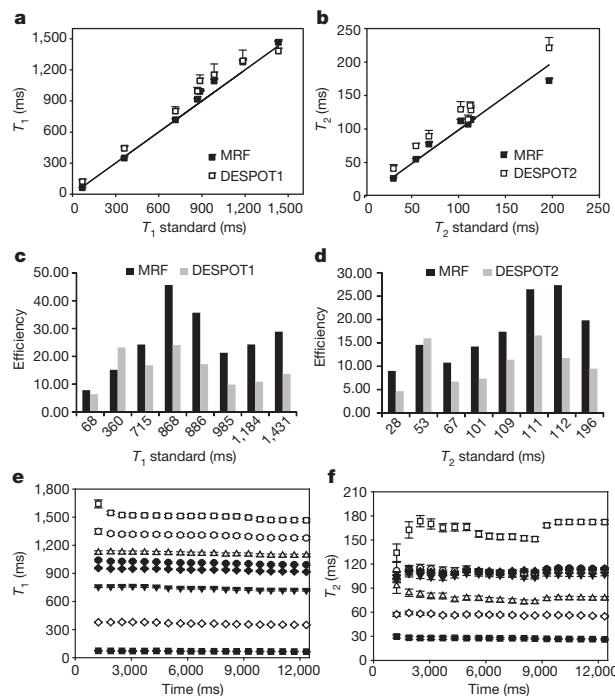


**Figure 4 | Demonstration of error tolerance in the presence of motion.** **a, b**, Reconstructed images acquired at the 12th second (**a**) and at the 15th second (**b**) demonstrate the large shift in the head position. **c–f**, The resulting MRF maps are nearly identical, demonstrating a rejection of both undersampling and motion errors that are uncorrelated with the expected signal evolution. **c, d**,  $T_1$  (colour scale, milliseconds); **e, f**,  $T_2$  (colour scale, milliseconds); **c** and **e** are from the first 12 s that has no motion, **d** and **f** are from entire 15 s that includes the motion.

time. Specifically, our initial results here demonstrate that the efficiency of MRF is approximately 1.8 times higher than the DESPOT methods, which were previously the most efficient methods for the measurement of relaxation parameters. Thus the direct prediction of the oscillating, incoherent signal evolutions through the Bloch simulation provides us the potential to obtain new quantitative information that is impractical today because of the prohibitively long scan times required, especially in biological samples and patients.

As demonstrated by the results shown here, MRF has the potential to significantly reduce the effects of errors during acquisition through its basis in pattern recognition. Acquisition errors may globally reduce the probability of a match of an observed signal to any given fingerprint, but as long as the errors do not cause another fingerprint to become the most likely match, the correct quantitative identification will still be made. Ideally, the sequence pattern will be designed so that the various fingerprints from different tissues and materials are as independent as possible, thus ensuring this robustness against motion and other practical errors.

Commercial magnetic resonance scanners include methods to minimize the effects of unavoidable system imperfections. However, these inaccuracies are becoming increasingly important as magnetic resonance technology is pushed to its limits, such as the use of very high magnetic fields or physically larger systems. MRF provides a route to model and account for system imperfections, such as inhomogeneities in both the static magnetic field ( $B_0$ ) and the



**Figure 5 | Accuracy, efficiency and error estimation for MRF and DESPOT.** **a, b**, The  $T_1$  and  $T_2$  values retrieved from MRF from eight phantoms are compared with those acquired from DESPOT1 (**a**), DESPOT2 (**b**) and a standard spin-echo sequence. **c, d**, The efficiency of MRF compared to DESPOT1 (**c**) and DESPOT2 (**d**) at different  $T_1$  and  $T_2$  values; efficiency is assessed as precision per (acquisition time) $^{1/2}$ . MRF has an average of 1.87 and 1.85 times higher efficiency than DESPOT1 and DESPOT2, respectively. **e, f**, Obtained values of  $T_1$  (**e**) and  $T_2$  (**f**) as a function of acquisition time. Data in **a–d** show mean  $\pm$  s.d. of the results over a 25-pixel region in the centre of each phantom, and are smaller than the symbols for most MRF results.

applied radio frequency field ( $B_1$ ), by adding these parameters into the dictionary simulation. Because both MRF and DESPOT2 are based on a bSSFP sequence, which is known to be sensitive to field inhomogeneities<sup>30,37</sup>, Supplementary Fig. 2 compares the  $T_2$  maps acquired from MRF and DESPOT2 from an *in vivo* scan. Because off-resonance is not taken into account in the DESPOT2 model, the  $T_2$  map from DESPOT2 shows areas of signal voids resulting from susceptibility effects at the air-tissue interfaces. MRF naturally incorporates these effects into the fingerprints, and thus the maps generated by MRF do not show these errors. Thus MRF could, for example, provide higher quality results using the current generation of magnetic resonance scanners. Alternatively, MRF could also allow the design of lower cost magnetic resonance scanners that can provide the same quality as today's high end systems through application of MRF models.

Because of its ability to provide quantitative results across many parameters simultaneously, MRF could lead to the direct identification of a material, tissue or pathology solely on the basis of its fingerprint. For example, many cancer cells show changes in multiple magnetic resonance parameters (for example,  $T_1$ ,  $T_2$  and self-diffusion tensor), a combination (though no single parameter) of which could potentially characterize them as different from all surrounding normal tissue types, and thus potentially separable. In an ideal situation, each given material, tissue or pathology would have its own signal evolution which would be orthogonal to all other signal evolutions. The MRF concept also implies that completely different acquisition schemes are possible in cases where one is only interested in the presence or absence of a particular material or disease state. For example, one could do a very rapid MRF scan of a bulk area of material or tissue

and compare the measured signal evolutions against the set of known states of interest. This measurement could either indicate the presence of the material or disease of interest, or indicate its absence within a margin of error. This feature could result in very rapid and accurate screening procedures. In particular, this feature may help to relax the required spatial resolution of an MRI examination, thus increasing the speed, and potentially reducing the cost, of such an examination. A preliminary example of this kind of visualization is shown in Supplementary Information Section 3. Using the MRF concept, the operation of the magnetic resonance unit will also be greatly simplified, because the 'all in one' scan concept of MRF has the potential to reduce the dozens of parameters currently presented to the magnetic resonance operator to a simple 'scan' button.

It is important to note that the proof-of-principle implementation of MRF shown here is but one of the many possibilities that could be used for this technique, and both the sequence design/implementation and post-processing methods will continue to be a significant open area of research, just as sequence design has advanced over the decades since the conventional methods have been introduced. Other, more advanced pattern recognition algorithms<sup>38–42</sup> will probably improve the performance of MRF. For spatially encoded MRI applications, the parameters retrieved from MRF are far fewer than the number of pixels in the images, and because the signals generated are largely incoherent, MRF has the additional potential to be highly accelerated through combination with other compressed sensing methods for accelerated spatial encoding, in addition to the now standard parallel imaging methods<sup>43–45</sup>, neither of which were included here. Any of these methods would reduce the undersampling errors seen in Fig. 3a even before the pattern recognition step, which should result in higher quality results. We have recently published data indicating that we can achieve a reduction in imaging time of about ten times for a two-dimensional slice using parallel imaging alone<sup>43,45,46</sup>. Also, it should be noted that the proof-of-principle results shown here only take advantage of two spatial dimensions for undersampling, whereas it is well known that taking full advantage of undersampling in all three spatial dimensions gives higher performance than a two-dimensional acquisition owing to the reduced power of the resulting errors at any given undersampling factor<sup>47</sup>. Thus a combination of an optimized three-dimensional MRF pulse sequence with parallel imaging and more advanced pattern matching algorithms will allow realization MRF in very short scan times.

## METHODS SUMMARY

**Sequence design.** After an initial inversion pulse, the sequence pattern shown in Fig. 1c used a pseudorandomized series (Perlin noise<sup>48</sup>) of flip angle and a random repetition time from 10.5 to 14 ms. The flip angle pattern in Fig. 1d contained repeating sinusoidal curves with a period of 250 acquisitions and alternating maximum flip angles. The repetition time was a Perlin noise pattern. The radio frequency phase for both of the patterns alternated between 0 and 180° on successive radio frequency pulses. The variable density spiral-out trajectory was designed using minimum-time gradient design<sup>49</sup>.

**Dictionary design.** A total of 563,784 signal time courses, each with 1,000 time points, with different sets of characteristic parameters ( $T_1$ ,  $T_2$  and off-resonance) were simulated for the dictionary. One dictionary entry was selected for each measured pixel location using template matching. In this case, the vector dot-product was calculated between the measured time course and all dictionary entries using complex data for both. The dictionary entry with the highest dot-product was then selected as most likely to represent the true signal evolution. The  $M_0$  value is then the multiplicative constant derived by fitting the acquired data to this dictionary entry.

**Data acquisition.** All data were acquired on a 1.5-T whole body scanner (Espree, Siemens Healthcare) with a standard 32-channel head receiver coil. Images from each acquisition block were reconstructed separately using non-uniform Fourier transform<sup>50</sup>. The resultant time series of images was used to determine the value for the parameters ( $T_1$ ,  $T_2$ ,  $M_0$  and off-resonance) as described above.

**Statistical analysis.** Quantitative estimates of the errors and efficiencies of MRF, DESPOT1 and DESPOT2 were calculated pixel-wise using a bootstrapped Monte Carlo method<sup>51</sup>. The means and standard deviations of  $T_1$  and  $T_2$  along the 50

repetitions were calculated, and averaged within a square ( $5\text{ pixel} \times 5\text{ pixel}$ ) region of interest for each phantom. The concordance correlation coefficients and efficiency were calculated as in refs 30 and 52.

**Full Methods** and any associated references are available in the online version of the paper.

Received 11 September 2012; accepted 30 January 2013.

- Bartzokis, G. *et al.* *In vivo* evaluation of brain iron in Alzheimer disease using magnetic resonance imaging. *Arch. Gen. Psychiatry* **57**, 47–53 (2000).
- Larsson, H. B. *et al.* Assessment of demyelination, edema, and gliosis by *in vivo* determination of T1 and T2 in the brain of patients with acute attack of multiple sclerosis. *Magn. Reson. Med.* **11**, 337–348 (1989).
- Pitkänen, A. *et al.* Severity of hippocampal atrophy correlates with the prolongation of MRI T2 relaxation time in temporal lobe epilepsy but not in Alzheimer's disease. *Neurology* **46**, 1724–1730 (1996).
- Williamson, P. *et al.* Frontal, temporal, and striatal proton relaxation times in schizophrenic patients and normal comparison subjects. *Am. J. Psychiatry* **149**, 549–551 (1992).
- Warrnits, J. B., Dahlqvist, O. & Lundberg, P. Novel method for rapid, simultaneous T1, T<sup>2</sup>, and proton density quantification. *Magn. Reson. Med.* **57**, 528–537 (2007).
- Warrnits, J. B., Leinhard, O. D., West, J. & Lundberg, P. Rapid magnetic resonance quantification on the brain: optimization for clinical usage. *Magn. Reson. Med.* **60**, 320–329 (2008).
- Schmitt, P. *et al.* Inversion recovery TrueFISP: quantification of T1, T2, and spin density. *Magn. Reson. Med.* **51**, 661–667 (2004).
- Ehses, P. *et al.* IR TrueFISP with a golden-ratio-based radial readout: fast quantification of T1, T2, and proton density. *Magn. Reson. Med.* **69**, 71–81 (2013).
- Donoho, D. L. Compressed sensing. *IEEE Trans. Inform. Theory* **52**, 1289–1306 (2006).
- Candes, E. J. & Tao, T. Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Trans. Inform. Theory* **52**, 5406–5425 (2006).
- Lustig, M., Donoho, D. L. & Pauly, J. M. Sparse MRI: the application of compressed sensing for rapid MR imaging. *Magn. Reson. Med.* **58**, 1182–1195 (2007).
- Bilic, B., Goyal, V. K. & Adalsteinsson, E. Multi-contrast reconstruction with Bayesian compressed sensing. *Magn. Reson. Med.* **66**, 1601–1615 (2011).
- Smith, D. S. *et al.* Robustness of quantitative compressive sensing MRI: the effect of random undersampling patterns on derived parameters for DCE- and DSC-MRI. *IEEE Trans. Med. Imaging* **31**, 504–511 (2012).
- Deshpande, V. S., Chung, Y.-C., Zhang, Q., Shea, S. M. & Li, D. Reduction of transient signal oscillations in True-FISP using a linear flip angle series magnetization preparation. *Magn. Reson. Med.* **49**, 151–157 (2003).
- Çukur, T. Multiple repetition time balanced steady-state free precession imaging. *Magn. Reson. Med.* **62**, 193–204 (2009).
- Nayak, K. & Lee, H. Wideband SSFP: alternating repetition time balanced steady state free precession with increased band spacing. *Magn. Reson. Med.* **58**, 931–938 (2007).
- Lee, K., Lee, H. & Hennig, J. Use of simulated annealing for the design of multiple repetition time balanced steady-state free precession imaging. *Magn. Reson. Med.* **68**, 220–226 (2012).
- Ernst, R. R. Magnetic resonance with stochastic excitation. *J. Magn. Reson.* **3**, 10–27 (1970).
- Scheffler, K. & Hennig, J. Frequency resolved single-shot MR imaging using stochastic k-space trajectories. *Magn. Reson. Med.* **35**, 569–576 (1996).
- Haldar, J. P., Hernando, D. & Liang, Z.-P. Compressed-sensing MRI with random encoding. *IEEE Trans. Med. Imaging* **30**, 893–903 (2011).
- Doneva, M. *et al.* Compressed sensing reconstruction for magnetic resonance parameter mapping. *Magn. Reson. Med.* **64**, 1114–1120 (2010).
- Stoecker, T., Vahedipour, K., Pracht, E., Brenner, D. & Shah, N. J. in Proc. 19th Scientific Meeting International Society for Magnetic Resonance in Medicine 381 (Int. Soc. Magn. Reson. Med., 2011).
- Davenport, M. A., Wakin, M. B. & Baraniuk, R. G. *The Compressive Matched Filter* (Tech. Rep. TREE 0610, Rice University, 2006).
- Tropp, J. A. & Gilbert, A. C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inform. Theory* **53**, 4655–4666 (2007).
- Schmitt, P. *et al.* A simple geometrical description of the TrueFISP ideal transient and steady-state signal. *Magn. Reson. Med.* **55**, 177–186 (2006).
- Lee, J. H., Hargreaves, B. A., Hu, B. S. & Nishimura, D. G. Fast 3D imaging using variable-density spiral trajectories with applications to limb perfusion. *Magn. Reson. Med.* **50**, 1276–1285 (2003).
- Marseille, G., De Beer, R., Fuderer, M., Mehlkopf, A. & Van Ormondt, D. Nonuniform phase-encode distributions for MRI scan time reduction. *J. Magn. Reson. B* **111**, 70–75 (1996).
- Tsai, C. M. & Nishimura, D. G. Reduced aliasing artifacts using variable-density K-space sampling trajectories. *Magn. Reson. Med.* **43**, 452–458 (2000).
- Vymazal, J. *et al.* T1 and T2 in the brain of healthy subjects, patients with Parkinson disease, and patients with multiple system atrophy: relation to iron content. *Radiology* **211**, 489–495 (1999).
- Deoni, S. C. L., Peters, T. M. & Rutt, B. K. High-resolution T1 and T2 mapping of the brain in a clinically acceptable time with DESPOT1 and DESPOT2. *Magn. Reson. Med.* **53**, 237–241 (2005).
- Whittall, K. P. *et al.* *In vivo* measurement of T2 distributions and water contents in normal human brain. *Magn. Reson. Med.* **37**, 34–43 (1997).
- Poon, C. S. & Henkelman, R. M. Practical T2 quantitation for clinical applications. *J. Magn. Reson. Imaging* **2**, 541–553 (1992).
- Haacke, E. M., Brown, R. W., Thompson, M. R. & Venkatesan, R. *Magnetic Resonance Imaging: Physical Principles and Sequence Design* 669–675 (Wiley & Sons, 1999).
- Hahn, E. L. Spin echoes. *Phys. Rev.* **80**, 580–594 (1950).
- Crawley, A. P. & Henkelman, R. M. A Comparison of one-shot and recovery methods in T1 imaging. *Magn. Reson. Med.* **7**, 23–34 (1988).
- Deoni, S. C. L., Rutt, B. K. & Peters, T. M. Rapid combined T1 and T2 mapping using gradient recalled acquisition in the steady state. *Magn. Reson. Med.* **49**, 515–526 (2003).
- Scheffler, K. & Lehnhardt, S. Principles and applications of balanced SSFP techniques. *Eur. Radiol.* **13**, 2409–2418 (2003).
- Needell, D. & Tropp, J. A. Cosamp: iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. Anal.* **26**, 301–321 (2009).
- Goldstein, T. & Osher, S. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.* **2**, 323–343 (2009).
- Chartrand, R. & Yin, W. in Proc. ICASSP 2008 IEEE International Conference 3869–3872 (IEEE, 2008).
- Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S. & Ma, Y. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**, 210–227 (2009).
- Turk, M. & Pentland, A. Eigenfaces for recognition. *J. Cogn. Neurosci.* **3**, 71–86 (1991).
- Griswold, M. A. *et al.* Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magn. Reson. Med.* **47**, 1202–1210 (2002).
- Pruessmann, K. P. *et al.* SENSE: sensitivity encoding for fast MRI. *Magn. Reson. Med.* **42**, 952–962 (1999).
- Seiberlich, N., Ehses, P., Duerk, J., Gilkeson, R. & Griswold, M. A. Improved radial GRAPPA calibration for real-time free-breathing cardiac imaging. *Magn. Reson. Med.* **65**, 492–505 (2011).
- Heidemann, R. M. *et al.* Direct parallel image reconstructions for spiral trajectories using GRAPPA. *Magn. Reson. Med.* **56**, 317–326 (2006).
- Barger, A. V., Block, W. F., Toropov, Y., Grist, T. M. & Mistretta, C. A. Time-resolved contrast-enhanced imaging with isotropic resolution and broad coverage using an undersampled 3D projection trajectory. *Magn. Reson. Med.* **48**, 297–305 (2002).
- Perlin, K. An image synthesizer. *Comput. Graphics* **19**, 287–296 (1985).
- Hargreaves, B. A., Nishimura, D. G. & Connolly, S. M. Time-optimal multidimensional gradient waveform design for rapid imaging. *Magn. Reson. Med.* **51**, 81–92 (2004).
- Fessler, J. A. & Sutton, B. P. Nonuniform fast Fourier transforms using min-max interpolation. *IEEE Trans. Signal Process.* **51**, 560–574 (2003).
- Riffe, M. J., Blaimer, M., Barkauskas, K. J., Duerk, J. L. & Griswold, M. A. in Proc. 15th Scientific Meeting, International Society for Magnetic Resonance in Medicine 1879 (Int. Soc. Magn. Reson. Med., 2007).
- Lin, L. I.-K. A concordance correlation coefficient to evaluate reproducibility. *Biometrics* **45**, 255–268 (1989).

**Supplementary Information** is available in the online version of the paper.

**Acknowledgements** Support for this study was provided by NIH R01HL094557 and Siemens Healthcare. We also thank H. Saybasili and G. Lee for technical assistance during the implementation of these concepts; M. Lustig and W. Grissom for discussions regarding this work; and A. Exner, S. Brady-Kalnay, E. Karathanasis, E. Lavik and H. Salz for their assistance in preparing the manuscript.

**Author Contributions** D.M., concept development, technical implementation, data collection and analysis, manuscript development and editing; V.G., concept development, manuscript development and editing; N.S., concept development, manuscript development and editing; K.L., concept development, technical implementation, manuscript development and editing; J.L.S., concept development, manuscript development and editing; J.L.D., concept development, manuscript development and editing; M.A.G., concept development, data collection and analysis, manuscript development and editing.

**Author Information** Reprints and permissions information is available at [www.nature.com/reprints](http://www.nature.com/reprints). The authors declare competing financial interests: details are available in the online version of the paper. Readers are welcome to comment on the online version of the paper. Correspondence and requests for materials should be addressed to M.A.G. (mark.griswold@case.edu).

## METHODS

**Sequence design.** After an initial inversion pulse, the first sequence pattern shown in Fig. 1c used a pseudorandomized series (Perlin noise<sup>48</sup>) of flip angle and a random repetition time between 10.5 ms and 14 ms based on a uniform random number generator. A linear ramp was added to the flip angle train since we have seen that this can increase differential sensitivity to both  $T_1$  and  $T_2$ .

The second flip angle pattern in Fig. 1d used a series of repeating sinusoidal curves with a period of 250 repetition times and alternating maximum flip angles. In the odd periods, the flip angle (FA) is calculated as  $FA_t = 10 + \sin(\frac{2\pi}{500}t) \times 50 + \text{random}(5)$ , where  $t$  is from 1 to 250, and  $\text{random}(5)$  is a function to generate uniformly distributed values with a standard deviation of 5. In the even periods, we divide the previous period's flip angle by 2. A 600 ms delay was added between each of the periods to allow for both differential magnetization recovery according to  $T_1$  and differential signal decay according to  $T_2$ . In this case, the repetition time was a Perlin noise pattern. The radio frequency phase for both of the patterns in Fig. 1 alternated between 0 and 180° on successive pulses.

The variable density spiral-out trajectory was designed to have 5.8 ms readout time in each repetition time and to have zero and first moment gradient compensation using minimum-time gradient design<sup>19</sup>. (The code used for this design is available at <http://www-mrsrl.stanford.edu/~brian/mintgrad/>). This trajectory required one interleaf to sample the inner 10 × 10 region, while 48 interleaves were required to fully sample the outer portions of  $k$ -space. During acquisition, the spiral trajectory rotated 7.5° from one time point to the next, so that each time point had a slightly different spatial encoding.

**Dictionary design.** The dictionary used in the matching algorithm was simulated using MATLAB (The MathWorks). Signal time courses with different sets of characteristic parameters ( $T_1$ ,  $T_2$  and off-resonance) were simulated. The ranges of  $T_1$  and  $T_2$  for the *in vivo* study were chosen according to the typical physiological limits of tissues in the brain:  $T_1$  values were taken to be between 100 and 5,000 ms (in increments of 20 ms below a  $T_1$  of 2,000 ms and in an increment of 300 ms above). The  $T_2$  values included the range between 20 and 3,000 ms (with an increment of 5 ms below a  $T_2$  of 100 ms, an increment of 10 ms between 100 ms and 200 ms, and an increment of 200 ms above a  $T_2$  of 200 ms). Since magnetic resonance is sensitive to parts per million (p.p.m.) level deviations in the  $B_0$  field, different off-resonance frequencies (1 Hz increment between ±40 Hz, 2 Hz between ±40 to ±80 Hz, 10 Hz between ±90 to ±250 Hz, and 20 Hz between ±270 to ±400 Hz) were simulated for each combination of  $T_1$  and  $T_2$  parameters to incorporate the effects of signal evolutions in different  $B_0$  fields. A total of 563,784 dictionary entries, each with 1,000 time points, were generated in 399 s on a standard desktop computer. One dictionary entry was selected for each measured pixel location using template matching. In this case, the vector dot-product was calculated between the measured time course and all dictionary entries (appropriately normalized to each having the same sum squared magnitude) using the complex data for both. The dictionary entry with the highest dot-product was then selected as most likely to represent the true signal evolution. The proton density ( $M_0$ ) of each pixel was calculated as the scaling factor between the measured signal and the simulated time course from the dictionary. For this experiment, four parameters were retrieved simultaneously from each of the 128 × 128 pixels using MRF. This calculation required about 3 min on a standard desktop computer.

**Data acquisition.** All MRI and MRF data were acquired on a 1.5 T whole body scanner (Siemens Espree, Siemens Healthcare) with a 32 channel head receiver coil (Siemens Healthcare). A square field of view of 300 mm × 300 mm was covered with a matrix of 128 × 128 pixels. The slice thickness was 5 mm. Images from each acquisition block were reconstructed separately using non-uniform Fourier transform (NUFFT)<sup>50</sup>. The resultant time series of images was

used to determine the value for the parameters ( $T_1$ ,  $T_2$ ,  $M_0$  and off-resonance) as described above.

*In vivo* experiments were performed with IRB guidelines, including written informed consent. For the fully sampled spiral acquisition shown in Supplementary Video 1, 48 repetitions were acquired, each with a different interleaf of the total acquisition. A recovery time of 5 s was used in between various acquisitions and this was taken into account in the simulated dictionary.

For the phantom study shown in Figs 2 and 4, eight cylindrical phantoms were constructed with varying concentrations of GdCl<sub>3</sub> (Aldrich) and agarose (Sigma) to yield different  $T_1$  and  $T_2$  values ranging from 67 to 1,700 ms and 30 to 200 ms, respectively. Standard spin echo sequences were used to quantify  $T_1$  and  $T_2$  separately ( $T_1$  quantification: 13 repetition times (TRs) ranging from 50 to 5,000 ms, echo time TE = 8.5 ms, total acquisition time = 33.4 min;  $T_2$  quantification: spin echo sequences with TRs = [15, 30, 45, 60, 90, 150, 200, 300, 400] ms, TR = 10,000 ms, total acquisition time = 3.2 h).  $T_1$  values were calculated pixel-wise using a standard three-parameter nonlinear least squares fitting routine to solve the equation:  $S(\text{TR}) = a + be^{-\text{TE}/T_1}$ .  $T_2$  values were determined in a pixel-wise fashion using a two-parameter nonlinear least squares fitting routine to solve the equation  $S(\text{TE}) = ae^{-\text{TE}/T_2}$ . DESPOT1 and DESPOT2 sequences using a fully sampled spiral readout were implemented based on the acquisition values from ref. 30: DESPOT1: FA: 4° and 55°, TR: 13.6 ms, DESPOT2: FA: 15° and 55°, TR = 10.8 ms. The  $T_1$  and  $T_2$  values were calculated from the equations provided in ref. 30. A 20 s waiting period was used in between the different acquisitions. The initial 10 s of data acquisition was not used in order to ensure that the signal was in steady-state for each of the DESPOT acquisitions. In the following analysis of efficiency, only the pure time of data acquisition for the steady-state DESPOT images is used. For DESPOT1 this was 1.27 s and for DESPOT2 it was 2.29 s (which includes the time for the required DESPOT1 acquisition.)

**Statistical analysis.** Quantitative estimates of the errors and efficiencies of MRF, DESPOT1 and DESPOT2 were calculated pixel-wise using a bootstrapped Monte Carlo method<sup>51</sup>. Two sets of raw data were acquired for each sequence: the encoded signal and a separate acquisition that only contained noise. Fifty reconstructions were then calculated by randomly resampling the acquired noise and adding it to the raw data before reconstruction and quantification. The means and standard deviations of  $T_1$  and  $T_2$  along the 50 repetitions were calculated, and both were averaged within a 5 pixel × 5 pixel square region of interest for each phantom. The concordance correlation coefficients ( $\rho_c$ ) were calculated using the equation<sup>52</sup>:

$$\rho_c = \frac{2S_{12}}{S_1^2 + S_2^2 + (\bar{Y}_2 - \bar{Y}_1)^2}$$

where  $Y_1$  and  $Y_2$  denotes the  $T_1$  or  $T_2$  values from two different methods,  $n$  is the number of phantoms,  $\bar{Y}_j = \frac{1}{n} \sum_{i=1}^n Y_{ij}$ ,  $S_j^2 = \frac{1}{n} \sum_{i=1}^n (Y_{ij} - \bar{Y}_j)^2$ ,  $j = 1, 2$  and  $S_{12} = \frac{1}{n} \sum_{i=1}^n (Y_{i1} - \bar{Y}_1)(Y_{i2} - \bar{Y}_2)$ .

The efficiency of the methods was calculated using:

$$\text{Efficiency} = \frac{T_n \text{NR}}{\sqrt{T_{\text{seq}}}}, n = 1, 2$$

where  $T_n \text{NR}$  is the  $T_1$  or  $T_2$  to noise ratio (defined as the  $T_1$  or  $T_2$  value divided by the estimated error).  $T_{\text{seq}}$  is the total acquisition time for MRF, and the relevant acquisition times for DESPOT1 and DESPOT2 (where the waiting times required for the approach to steady state and the time between each of the DESPOT1 and DESPOT2 scans to allow for complete recovery of magnetization were ignored).

## Cinzia Viroli

### *Material list:*

Viroli C. (2025) Some Contributions to Microclustering: A Frequentist Perspective. WGMBC 2025 slides.

## Some Contributions to Microclustering: A Frequentist Perspective

Edoardo Redivo, Cinzia Viroli  
(University of Bologna, Italy)

July 23, 2025

WGMBC 2025 - Nice

### The Microclustering Setting

We're interested in data structures where each unit belongs to a small groups — think of families, firms, biological samples.

## The Microclustering Setting

We're interested in data structures where each unit belongs to a small groups — think of families, firms, biological samples.

These tasks naturally require **microclustering**, where the number of clusters  $k$  grows with the sample size  $n$ , but each cluster remains small.

## The Microclustering Setting

We're interested in data structures where each unit belongs to a small groups — think of families, firms, biological samples.

These tasks naturally require **microclustering**, where the number of clusters  $k$  grows with the sample size  $n$ , but each cluster remains small.

Traditional clustering models, such as Gaussian mixtures, are ill-suited for this setting because they produce well-separate clusters and cluster sizes grow linearly with  $n$ .

## Motivating Examples

### Motivating Example 1: Noisy Longitudinal Records (SIPP)

The Survey of Income and Program Participation (SIPP) is a longitudinal study that records multiple waves of data for the same individuals. However, inconsistencies may arise across waves due to reporting errors, changes in self-identification, or data entry mistakes.

pid	wave	sex	birth_year	birth_month	state	race
7001	4	Female	1977	5	Alabama	White alone
7001	5	Female	1977	5	Georgia	White alone
4669001	5	Female	1956	11	California	Asian alone
4669001	6	Female	1956	1	California	Asian alone
4669001	7	Female	1956	1	California	White alone
39895002	6	Male	1973	12	New York	White alone
39895002	8	Male	1973	12	New York	Black alone

**Notice:** the same pid (person ID) appears across waves with small but meaningful changes — in state, race, or birth month — that complicate exact matching.

## Why Microclustering?

- Each unique individual should be assigned to a single group.
- Most clusters contain only one record (no duplicates).
- A few clusters contain 2 or more near-duplicate records (same person).

## Why Microclustering?

- Each unique individual should be assigned to a single group.
- Most clusters contain only one record (no duplicates).
- A few clusters contain 2 or more near-duplicate records (same person).

### Key idea:

- Standard clustering assumes clusters grow as  $n$  increases.
- But in record linkage, cluster sizes remain tiny even for large datasets.
- We need a model that favours many small clusters.

## Why Microclustering?

- Each unique individual should be assigned to a single group.
- Most clusters contain only one record (no duplicates).
- A few clusters contain 2 or more near-duplicate records (same person).

### Key idea:

- Standard clustering assumes clusters grow as  $n$  increases.
- But in record linkage, cluster sizes remain tiny even for large datasets.
- We need a model that favours many small clusters.

This is the essence of **microclustering**.

## Motivating Example 2: DNA Data Storage

In DNA data storage, digital data is encoded into reference DNA sequences. Sequencing produces many noisy copies ("reads") of each reference.

## Motivating Example 2: DNA Data Storage

In DNA data storage, digital data is encoded into reference DNA sequences.  
Sequencing produces many noisy copies ("reads") of each reference.  
Each group of reads should be clustered to recover the original references.

## Motivating Example 2: DNA Data Storage

In DNA data storage, digital data is encoded into reference DNA sequences.  
Sequencing produces many noisy copies ("reads") of each reference.  
Each group of reads should be clustered to recover the original references.

Reference ID	Read Sequence	Error Type	Group
R1	ACTGATGCACTGA	Exact copy	Cluster 1
R1	ACTGTTGCACTGA	Substitution	Cluster 1
R1	ACTGATGCAGTG	Deletion	Cluster 1
R2	TGACGTAGGCCCTA	Exact copy	Cluster 2
R2	TGACGTAGGCCCTA	Insertion	Cluster 2
R3	GCTAGCTAGCTA	Exact copy	Cluster 3

## Motivating Example 2: DNA Data Storage

In DNA data storage, digital data is encoded into reference DNA sequences. Sequencing produces many noisy copies ("reads") of each reference. Each group of reads should be clustered to recover the original references.

Reference ID	Read Sequence	Error Type	Group
R1	ACTGATGCACTGA	Exact copy	Cluster 1
R1	ACTGTTGCACTGA	Substitution	Cluster 1
R1	ACTGATGCACTG	Deletion	Cluster 1
R2	TGACGTAGGCCTA	Exact copy	Cluster 2
R2	TGACGTAGGCCCTA	Insertion	Cluster 2
R3	GCTAGCTAGCTA	Exact copy	Cluster 3

Reads from the same reference form a small cluster (size 2–10), with small differences due to sequencing noise.

Definition and first results

## Microclustering: A Mathematical Definition

### Classical Clustering Assumption:

- Cluster sizes grow **linearly** with the number of observations  $n$ .
- For example: in a  $k$ -component mixture with  $\pi_h = 1/k$ , the expected size is  $E[M_h] = n/k$ .

## Microclustering: A Mathematical Definition

### Classical Clustering Assumption:

- Cluster sizes grow **linearly** with the number of observations  $n$ .
- For example: in a  $k$ -component mixture with  $\pi_h = 1/k$ , the expected size is  $E[M_h] = n/k$ .

### Microclustering Assumption:

- The largest cluster grows **sublinearly** with  $n$ .

## Microclustering: A Mathematical Definition

### Classical Clustering Assumption:

- Cluster sizes grow **linearly** with the number of observations  $n$ .
- For example: in a  $k$ -component mixture with  $\pi_h = 1/k$ , the expected size is  $E[M_h] = n/k$ .

### Microclustering Assumption:

- The largest cluster grows **sublinearly** with  $n$ .

### Formal Definition:

Let  $\mathcal{P}_n$  be a random partition of  $n$  units, and  $M_{(k)}$  the size of the largest cluster. We say that  $\mathcal{P}_n$  satisfies the **microclustering property** if:

$$\frac{M_{(k)}}{n} \xrightarrow{P} 0 \quad \text{as} \quad n \rightarrow \infty.$$

## Bayesian Approaches to Microclustering

Microclustering has recently emerged as a distinct paradigm in the Bayesian framework:

- **Zanella, Betancourt, Wallach, Miller, Zaidi, Steorts (2016), NeurIPS**  
*"Flexible Models for Microclustering with Application to Entity Resolution"*  
⇒ Introduces a class of Bayesian models designed to favor microclustering behavior via flexible prior specifications. Focuses on entity resolution and allows for overlapping clustering structures. Uses a novel generalization of the Chinese Restaurant Process (CRP) to capture microclustering.
- **Johndrow, Lum, Dunson (2018), Biometrika**  
*"Theoretical limits of microclustering for record linkage"*  
⇒ Shows that accurate entity resolution via microclustering is often fundamentally limited, even under ideal conditions. Suggests a shift toward coarser inferential goals (e.g., population size estimation).
- **Di Benedetto, Caron, Teh (2021), Annals of Statistics**  
*"Nonexchangeable random partition models for microclustering"*  
⇒ Proposes a class of nonexchangeable partition models that yield cluster sizes growing sublinearly with  $n$ . Based on completely random measures and Poisson embeddings.
- **Betancourt, Zanella, Steorts (2022), JASA**  
*"Random Partition Models for Microclustering Tasks"*  
⇒ Develops a general framework for Bayesian microclustering via exchangeable sequences of clusters (not of points). Ensures identifiability and tractable inference.

## ESC-D model (BZS 2022)

**Hyper-prior on the base size pmf**

$$r \sim \text{Gamma}(a_r, b_r), \quad p \sim \text{Beta}(a_p, b_p),$$

$$\mu_s^{(0)} = \Pr[\text{NegBin}(r, p) = s], \quad s = 1, \dots, H.$$

$\mu^{(0)} = (\mu_1^{(0)}, \mu_2^{(0)}, \dots)$  is a base vector.

## ESC-D model (BZS 2022)

**Hyper-prior on the base size pmf**

$$r \sim \text{Gamma}(a_r, b_r), \quad p \sim \text{Beta}(a_p, b_p),$$

$$\mu_s^{(0)} = \Pr[\text{NegBin}(r, p) = s], \quad s = 1, \dots, H.$$

$\mu^{(0)} = (\mu_1^{(0)}, \mu_2^{(0)}, \dots)$  is a base vector.

**Dirichlet prior on the cluster-size distribution**

$$\mu = (\mu_1, \mu_2, \dots, \mu_H) \sim \text{Dirichlet}(\alpha \mu_1^{(0)}, \dots, \alpha \mu_H^{(0)}), \quad \sum_{s=1}^H \mu_s = 1.$$

$\alpha > 0$  controls the concentration around  $\mu^{(0)}$ .

## ESC-D model (BZS 2022)

### Hyper-prior on the base size pmf

$$r \sim \text{Gamma}(a_r, b_r), \quad p \sim \text{Beta}(a_p, b_p),$$

$$\mu_s^{(0)} = \Pr[\text{NegBin}(r, p) = s], \quad s = 1, \dots, H.$$

$\mu^{(0)} = (\mu_1^{(0)}, \mu_2^{(0)}, \dots)$  is a base vector.

### Dirichlet prior on the cluster-size distribution

$$\mu = (\mu_1, \mu_2, \dots, \mu_H) \sim \text{Dirichlet}(\alpha \mu_1^{(0)}, \dots, \alpha \mu_H^{(0)}), \quad \sum_{s=1}^H \mu_s = 1.$$

$\alpha > 0$  controls the concentration around  $\mu^{(0)}$ .

### Prior on a partition $\pi$

Given  $\mu$  and  $\alpha$ ,

$$P(\pi | \mu, \alpha) = \frac{\alpha^k \Gamma(\alpha)}{\Gamma(\alpha + n)} \prod_{\ell=1}^k \mu_{n_\ell} \quad (\text{Exchangeable Partition Probability Function}).$$

## ESC-D model (BZS 2022)

### Hyper-prior on the base size pmf

$$r \sim \text{Gamma}(a_r, b_r), \quad p \sim \text{Beta}(a_p, b_p),$$

$$\mu_s^{(0)} = \Pr[\text{NegBin}(r, p) = s], \quad s = 1, \dots, H.$$

$\mu^{(0)} = (\mu_1^{(0)}, \mu_2^{(0)}, \dots)$  is a base vector.

### Dirichlet prior on the cluster-size distribution

$$\mu = (\mu_1, \mu_2, \dots, \mu_H) \sim \text{Dirichlet}(\alpha \mu_1^{(0)}, \dots, \alpha \mu_H^{(0)}), \quad \sum_{s=1}^H \mu_s = 1.$$

$\alpha > 0$  controls the concentration around  $\mu^{(0)}$ .

### Prior on a partition $\pi$

Given  $\mu$  and  $\alpha$ ,

$$P(\pi | \mu, \alpha) = \frac{\alpha^k \Gamma(\alpha)}{\Gamma(\alpha + n)} \prod_{\ell=1}^k \mu_{n_\ell} \quad (\text{Exchangeable Partition Probability Function}).$$

**Micro-clustering property** With the above hyper-prior,

$$\frac{M_{(k)}}{n} \xrightarrow{P} 0 \quad (n \rightarrow \infty),$$

so every cluster remains asymptotically negligible.

## Our Contribution: A Frequentist Approach to Microclustering

- Existing literature on microclustering is mostly Bayesian, relying on random partition models or nonexchangeable priors.

## Our Contribution: A Frequentist Approach to Microclustering

- Existing literature on microclustering is mostly Bayesian, relying on random partition models or nonexchangeable priors.
- We propose a **frequentist framework** for microclustering, with formal sufficient conditions for sublinear cluster growth.

## Our Contribution: A Frequentist Approach to Microclustering

- Existing literature on microclustering is mostly Bayesian, relying on random partition models or nonexchangeable priors.
- We propose a **frequentist framework** for microclustering, with formal sufficient conditions for sublinear cluster growth.
- The core of our approach is a novel mixture model that distinguishes between:
  - ▶ *Unmatched units* (singletons),
  - ▶ *Matched units* (true duplicates).

## Our Contribution: A Frequentist Approach to Microclustering

- Existing literature on microclustering is mostly Bayesian, relying on random partition models or nonexchangeable priors.
- We propose a **frequentist framework** for microclustering, with formal sufficient conditions for sublinear cluster growth.
- The core of our approach is a novel mixture model that distinguishes between:
  - ▶ *Unmatched units* (singletons),
  - ▶ *Matched units* (true duplicates).
- We prove:
  - ▶ Non-monotonicity of the log-likelihood in  $k$ ,
  - ▶ Existence of a finite optimal number of clusters.

## Our Contribution: A Frequentist Approach to Microclustering

- Existing literature on microclustering is mostly Bayesian, relying on random partition models or nonexchangeable priors.
- We propose a **frequentist framework** for microclustering, with formal sufficient conditions for sublinear cluster growth.
- The core of our approach is a novel mixture model that distinguishes between:
  - ▶ *Unmatched units* (singletons),
  - ▶ *Matched units* (true duplicates).
- We prove:
  - ▶ Non-monotonicity of the log-likelihood in  $k$ ,
  - ▶ Existence of a finite optimal number of clusters.
- Simulations confirm that our model effectively identifies microclusters in realistic settings (e.g., record linkage).

## Sufficient Conditions for Microclustering

We let the number of clusters depend on the sample size  $n$ ; write  $K_n$  for this (possibly random) quantity and  $k$  for a fixed realisation.

### Theorem (sufficient conditions)

If, conditionally on  $K_n = k$ ,

## Sufficient Conditions for Microclustering

We let the number of clusters depend on the sample size  $n$ ; write  $K_n$  for this (possibly random) quantity and  $k$  for a fixed realisation.

### Theorem (sufficient conditions)

If, conditionally on  $K_n = k$ ,

- (i) **Bounded weights:**  $\pi_h \leq c/k$  ( $\forall h$ ) for some constant  $c > 0$  (*no cluster can dominate*);

## Sufficient Conditions for Microclustering

We let the number of clusters depend on the sample size  $n$ ; write  $K_n$  for this (possibly random) quantity and  $k$  for a fixed realisation.

### Theorem (sufficient conditions)

If, conditionally on  $K_n = k$ ,

- (i) **Bounded weights:**  $\pi_h \leq c/k$  ( $\forall h$ ) for some constant  $c > 0$  (*no cluster can dominate*);
- (ii) **Linear growth in  $n$ :**  $k = \Theta(n)$  (*total mass spread over  $\asymp n$  clusters*),

## Sufficient Conditions for Microclustering

We let the number of clusters depend on the sample size  $n$ ; write  $K_n$  for this (possibly random) quantity and  $k$  for a fixed realisation.

### Theorem (sufficient conditions)

If, conditionally on  $K_n = k$ ,

- (i) **Bounded weights:**  $\pi_h \leq c/k$  ( $\forall h$ ) for some constant  $c > 0$  (*no cluster can dominate*);
- (ii) **Linear growth in  $n$ :**  $k = \Theta(n)$  (*total mass spread over  $\asymp n$  clusters*),

then

$$\frac{M_{(k)}}{n} \xrightarrow{P} 0 \quad (n \rightarrow \infty),$$

so every empirical cluster becomes vanishingly small.

## Sufficient Conditions for Microclustering

### What if a condition fails?

- (i) violated  $\Rightarrow$  a few clusters may still capture  $O(n)$  points.
- (ii) violated ( $k = O(1)$ )  $\Rightarrow$  some cluster weight  $\geq 1/k$ , so microclustering impossible.

## Sufficient Conditions for Microclustering

### What if a condition fails?

- (i) violated  $\Rightarrow$  a few clusters may still capture  $O(n)$  points.
- (ii) violated ( $k = O(1)$ )  $\Rightarrow$  some cluster weight  $\geq 1/k$ , so microclustering impossible.

Two issues:

- 1 Theorem 1 provides sufficient conditions but they imply microclustering only if  $n \rightarrow \infty$ .

## Sufficient Conditions for Microclustering

### What if a condition fails?

- (i) violated  $\Rightarrow$  a few clusters may still capture  $O(n)$  points.
- (ii) violated ( $k = O(1)$ )  $\Rightarrow$  some cluster weight  $\geq 1/k$ , so microclustering impossible.

Two issues:

- 1 Theorem 1 provides sufficient conditions but they imply microclustering only if  $n \rightarrow \infty$ . What about finite  $n$ ?

## Sufficient Conditions for Microclustering

### What if a condition fails?

- (i) violated  $\Rightarrow$  a few clusters may still capture  $O(n)$  points.
- (ii) violated ( $k = O(1)$ )  $\Rightarrow$  some cluster weight  $\geq 1/k$ , so microclustering impossible.

Two issues:

- 1 Theorem 1 provides sufficient conditions but they imply microclustering only if  $n \rightarrow \infty$ . What about finite  $n$ ?
- 2 The theorem is purely conditional on fixed  $k$ . Do we need a Bayesian prior on  $K_n$ ?

## A finite-sample upper bound on the largest cluster size

- Asymptotic guarantees may not suffice when  $n$  is moderate.
- In practical applications (e.g., record linkage),  $n$  is fixed and we want explicit control on the size of the largest cluster.
- Can we bound  $M_{(k)}$  for finite  $n$  under the same assumptions?

## A finite-sample upper bound on the largest cluster size

- Asymptotic guarantees may not suffice when  $n$  is moderate.
- In practical applications (e.g., record linkage),  $n$  is fixed and we want explicit control on the size of the largest cluster.
- Can we bound  $M_{(k)}$  for finite  $n$  under the same assumptions?

Assume the cluster sizes follow a multinomial distribution:

$$(M_1, \dots, M_k) \sim \text{Multinomial}(n; \pi_1, \dots, \pi_k), \quad \pi_h \leq \frac{c}{k}.$$

## A finite-sample upper bound on the largest cluster size

- Asymptotic guarantees may not suffice when  $n$  is moderate.
- In practical applications (e.g., record linkage),  $n$  is fixed and we want explicit control on the size of the largest cluster.
- Can we bound  $M_{(k)}$  for finite  $n$  under the same assumptions?

Assume the cluster sizes follow a multinomial distribution:

$$(M_1, \dots, M_k) \sim \text{Multinomial}(n; \pi_1, \dots, \pi_k), \quad \pi_h \leq \frac{c}{k}.$$

Then, for any  $\alpha \in (0, 1)$ ,

$$\Pr\left(\frac{M_{(k)}}{n} > \delta_{n,k}(\alpha, c)\right) \leq \alpha, \quad \text{with} \quad \delta_{n,k}(\alpha, c) = \frac{c}{k} + \sqrt{\frac{\log(k/\alpha)}{2n}}.$$

## A finite-sample upper bound on the largest cluster size

- Asymptotic guarantees may not suffice when  $n$  is moderate.
- In practical applications (e.g., record linkage),  $n$  is fixed and we want explicit control on the size of the largest cluster.
- Can we bound  $M_{(k)}$  for finite  $n$  under the same assumptions?

Assume the cluster sizes follow a multinomial distribution:

$$(M_1, \dots, M_k) \sim \text{Multinomial}(n; \pi_1, \dots, \pi_k), \quad \pi_h \leq \frac{c}{k}.$$

Then, for any  $\alpha \in (0, 1)$ ,

$$\Pr\left(\frac{M_{(k)}}{n} > \delta_{n,k}(\alpha, c)\right) \leq \alpha, \quad \text{with} \quad \delta_{n,k}(\alpha, c) = \frac{c}{k} + \sqrt{\frac{\log(k/\alpha)}{2n}}.$$

**Interpretation:** With probability at least  $1 - \alpha$ , the largest cluster occupies at most a proportion  $\delta_{n,k}(\alpha, c)$  of the sample.

## Decay of the Upper Bound $\delta_{n,k}(\alpha, c)$

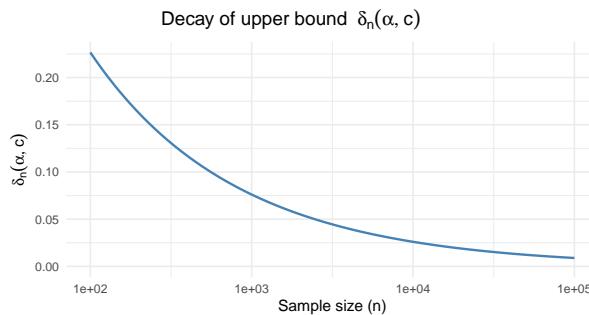
**Setup:**

- Let  $k = n/3$ ,  $\alpha = 0.05$ ,  $c = 1.5$ .
- Evaluate the upper bound  $\delta_{n,k}(\alpha, c)$  for  $n$  from 100 to 100,000.

## Decay of the Upper Bound $\delta_{n,k}(\alpha, c)$

### Setup:

- Let  $k = n/3$ ,  $\alpha = 0.05$ ,  $c = 1.5$ .
- Evaluate the upper bound  $\delta_{n,k}(\alpha, c)$  for  $n$  from 100 to 100,000.

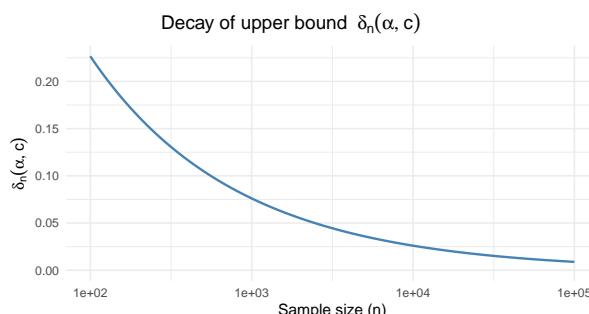


Decay of the upper bound  $\delta_{n,k}(\alpha, c)$  as  $n$  increases (log-scale on x-axis). The bound rapidly decreases, confirming that even moderate values of  $n$  yield small maximum cluster sizes with high probability.

## Decay of the Upper Bound $\delta_{n,k}(\alpha, c)$

### Setup:

- Let  $k = n/3$ ,  $\alpha = 0.05$ ,  $c = 1.5$ .
- Evaluate the upper bound  $\delta_{n,k}(\alpha, c)$  for  $n$  from 100 to 100,000.



Decay of the upper bound  $\delta_{n,k}(\alpha, c)$  as  $n$  increases (log-scale on x-axis). The bound rapidly decreases, confirming that even moderate values of  $n$  yield small maximum cluster sizes with high probability.

**Take-home:** For moderate sample sizes ( $n < 1,000$ ) the largest cluster can still absorb 5-10% of the data. In real applications we will need an additional mechanism to keep clusters genuinely “micro”. *More on this when we discuss estimation.*

## Do we need a Bayesian prior on $K_n$ ?

No: the theorem is purely conditional on  $k$ . Two workflows are possible:

- 1 *Frequentist*: Choose  $k$  deterministically (e.g. via a penalised likelihood) and verify it scales as  $\Theta(n)$ .
- 2 *Bayesian*: Place a prior on  $K_n$  (see in the following)

Either path satisfies the theorem once (i)–(ii) hold.

## Moment Conditions for a “Good” Prior on $K_n$

### Theorem (moment-based sufficiency)

Let  $K_n$  be the random number of clusters. If

$$\mathbb{E}[K_n] = \Theta(n), \quad \text{Var}(K_n) = o(n^2),$$

then

$$\frac{K_n}{n} \xrightarrow{p} c > 0, \quad \text{and} \quad \frac{M_{(K_n)}}{n} \xrightarrow{p} 0.$$

Hence the microclustering property holds *in probability*.

## Moment Conditions for a “Good” Prior on $K_n$

### Theorem (moment-based sufficiency)

Let  $K_n$  be the random number of clusters. If

$$\mathbb{E}[K_n] = \Theta(n), \quad \text{Var}(K_n) = o(n^2),$$

then

$$\frac{K_n}{n} \xrightarrow{P} c > 0, \quad \text{and} \quad \frac{M(K_n)}{n} \xrightarrow{P} 0.$$

Hence the microclustering property holds *in probability*.

### Interpretation

- Linear expectation guarantees *enough* clusters.
- Sub-quadratic variance prevents explosive dispersion.
- No explicit Bayesian machinery is *required*—the result only needs the two moment bounds.

## Concrete Priors Satisfying the Moment Conditions

### Good priors

- Discrete Uniform on  $[n - \sqrt{n}, n]$
- Binomial  $B(n, p)$  with fixed  $p \in (0, 1)$
- Hypergeometric, Beta–Binomial with fixed hyper-parameters

### Fails for microclustering

- Uniform on  $[1, n] \Rightarrow$  too much weight near small  $k$ .
- Any law with  $\mathbb{E}[K_n] = O(1) \Rightarrow$  some cluster mass  $\geq 1/k$ .

## Concrete Priors Satisfying the Moment Conditions

### Good priors

- Discrete Uniform on  $[n - \sqrt{n}, n]$
- Binomial  $B(n, p)$  with fixed  $p \in (0, 1)$
- Hypergeometric, Beta–Binomial with fixed hyper-parameters

### Fails for microclustering

- Uniform on  $[1, n] \Rightarrow$  too much weight near small  $k$ .
- Any law with  $\mathbb{E}[K_n] = O(1) \Rightarrow$  some cluster mass  $\geq 1/k$ .

A discrete uniform prior on  $[n - a\sqrt{n}, n]$ , for some fixed constant  $a > 0$ , satisfies the moment conditions and guarantees microclustering in probability.

## Modeling Microclustering

## Matched vs. Unmatched Observations

Define a binary latent variable  $U$ .

- **Matched points** ( $u = 0$ ): belong to a cluster with  $\geq 2$  near-duplicates.
- **Unmatched points** ( $u = 1$ ): genuine singletons. Under microclustering they are *the majority*.

For every unit  $x_i$ ,

$$\Pr(u_i = 1) = \pi_u, \quad \Pr(u_i = 0) = 1 - \pi_u.$$

## Matched vs. Unmatched Observations

Define a binary latent variable  $U$ .

- **Matched points** ( $u = 0$ ): belong to a cluster with  $\geq 2$  near-duplicates.
- **Unmatched points** ( $u = 1$ ): genuine singletons. Under microclustering they are *the majority*.

For every unit  $x_i$ ,

$$\Pr(u_i = 1) = \pi_u, \quad \Pr(u_i = 0) = 1 - \pi_u.$$

**Key modelling move:**

- Matched points  $\rightarrow$  usual component densities  $f_h(\cdot | \theta_h)$ .
- Unmatched points  $\rightarrow$  *constant* density  $\xi$  (uniform on the sample space).

## Mixture Specification with a Uniform “Singleton” Component

Conditionally on  $K_n = k$  clusters, reorder them so that  $h = 1, \dots, k_m$  are matched clusters and  $h = k_m + 1, \dots, k$  are singletons.

$$\underbrace{f(\mathbf{x} | \Psi)}_{\text{complete model}} = \sum_{h=1}^{k_m} \pi_h f_h(\mathbf{x} | \theta_h) + \underbrace{(\pi_{k_m+1} + \dots + \pi_k)}_{\pi_u} \xi, \quad (\star)$$

$$\sum_{h=1}^{k_m} \pi_h + \pi_u = 1.$$

- $\pi_u$  collects all singleton mass.

## Mixture Specification with a Uniform “Singleton” Component

Conditionally on  $K_n = k$  clusters, reorder them so that  $h = 1, \dots, k_m$  are matched clusters and  $h = k_m + 1, \dots, k$  are singletons.

$$\underbrace{f(\mathbf{x} | \Psi)}_{\text{complete model}} = \sum_{h=1}^{k_m} \pi_h f_h(\mathbf{x} | \theta_h) + \underbrace{(\pi_{k_m+1} + \dots + \pi_k)}_{\pi_u} \xi, \quad (\star)$$

$$\sum_{h=1}^{k_m} \pi_h + \pi_u = 1.$$

- $\pi_u$  collects all singleton mass.
- $\xi$  is *not* a tuning constant (cf. next slide) but the density for unmatched units.

## Mixture Specification with a Uniform “Singleton” Component

Conditionally on  $K_n = k$  clusters, reorder them so that  $h = 1, \dots, k_m$  are matched clusters and  $h = k_m + 1, \dots, k$  are singletons.

$$\underbrace{f(\mathbf{x} | \Psi)}_{\text{complete model}} = \sum_{h=1}^{k_m} \pi_h f_h(\mathbf{x} | \theta_h) + \underbrace{(\pi_{k_m+1} + \dots + \pi_k)}_{\pi_u} \xi, \quad (\star)$$
$$\sum_{h=1}^{k_m} \pi_h + \pi_u = 1.$$

- $\pi_u$  collects all singleton mass.
- $\xi$  is *not* a tuning constant (cf. next slide) but the density for unmatched units.
- $f_h(\mathbf{x} | \theta_h)$  is a multivariate density that depends on the data type.  
For instance, in record linkage with categorical variables, a latent class model with conditionally independent categorical marginals can be used.

## Relation to the “Improper” Gaussian Mixture

Coretto & Hennig (2016, 2017)

Add a *uniform* component to a Gaussian mixture to down-weight outliers and improve robustness. The constant density is a fixed tuning parameter, selected by OTRIMLE.

## Relation to the “Improper” Gaussian Mixture

### Coretto & Hennig (2016, 2017)

Add a *uniform* component to a Gaussian mixture to down-weight outliers and improve robustness. The constant density is a fixed tuning parameter, selected by OTRIMLE.

**Our setting differs:**

- Focus is microclustering  $\Rightarrow$  singletons are abundant, not rare outliers.
- $\xi$  must therefore adapt to the distribution of these unmatched points, not merely “lie below” the Gaussian tails.
- We derive  $\xi$  through an information-theoretic argument.

## Choosing $\xi$ via KL Divergence and Entropy

- Maximum-likelihood estimation of  $(*) \iff$  minimising  $D_{KL}(P \parallel f_\psi)$ .
- At the optimum  $D_{KL} = 0$  we have  $\mathbb{E}_P[\log f_\psi(X)] = -H(P)$ , where  $H(P)$  is the entropy of the true distribution  $P$ .

$$\boxed{\xi = \exp\{-H(P_{\text{unmatched}})\}}$$

## Choosing $\xi$ via KL Divergence and Entropy

- Maximum-likelihood estimation of  $(\star)$   $\iff$  minimising  $D_{KL}(P \parallel f_\psi)$ .
- At the optimum  $D_{KL} = 0$  we have  $\mathbb{E}_P[\log f_\psi(X)] = -H(P)$ , where  $H(P)$  is the entropy of the true distribution  $P$ .

$$\boxed{\xi = \exp\{-H(P_{\text{unmatched}})\}}$$

### Implications

- $\xi$  automatically scales with the “spread” of singleton data.
- No extra tuning: entropy can be consistently estimated (plug-in or kernel estimator) from the empirical singletons.

## Choosing the Optimal $k$

## Choice of $k$ in the Microclustering Model

The total log-likelihood function  $\mathcal{L}_k$  is not monotonic in  $k \in [1, n]$ , unlike in standard mixture models.

**Why?** Because we add a penalty from unmatched units (entropy-related).

## Choice of $k$ in the Microclustering Model

The total log-likelihood function  $\mathcal{L}_k$  is not monotonic in  $k \in [1, n]$ , unlike in standard mixture models.

**Why?** Because we add a penalty from unmatched units (entropy-related).

There exists an optimal number of clusters  $k^*$  such that:

$$k^* = \arg \max_k \log \mathcal{L}_k.$$

## Choice of $k$ in the Microclustering Model

The total log-likelihood function  $\mathcal{L}_k$  is not monotonic in  $k \in [1, n]$ , unlike in standard mixture models.

**Why?** Because we add a penalty from unmatched units (entropy-related).

There exists an optimal number of clusters  $k^*$  such that:

$$k^* = \arg \max_k \log \mathcal{L}_k.$$

**Decomposition of the log-likelihood:**

$$\log \mathcal{L}_k = \underbrace{\sum_{i=1}^n u_i \xi}_{\mathcal{L}_k^{(u)}} + \underbrace{\sum_{i=1}^n (1 - u_i) \log \sum_{h=1}^{k_m} \pi_h f_h(x_i | \theta_h)}_{\mathcal{L}_k^{(m)}}$$

## Log-likelihood Decomposition

### Matched vs. Unmatched contribution

$$\log \mathcal{L}_k(\mathbf{x}) = \underbrace{\sum_{i=1}^n u_i \xi}_{\mathcal{L}_k^{(u)}(\mathbf{x})} + \underbrace{\sum_{i=1}^n (1 - u_i) \log \left[ \sum_{h=1}^{k_m} \pi_h f_h(x_i | \theta_h) \right]}_{\mathcal{L}_k^{(m)}(\mathbf{x})} \quad \text{where } u_i = \mathbb{1}\{\text{unmatched}\}.$$

- $\xi = e^{-H(P)}$  (entropy of the unmatched-data distribution).
- $k_m = \text{number of matched clusters}$ ,  $\pi_u = \sum_{h=k_m+1}^k \pi_h$ .
- Key intuition: more clusters help the matched term but harm the unmatched term.

## Two Lemmas on the Components

### Lemma 1 (Unmatched part)

For  $n \geq k$ ,

$$\mathcal{L}_k^{(u)}(\mathbf{x}) \text{ is strictly decreasing in } k \quad \text{and} \quad \lim_{k \rightarrow \infty} \mathcal{L}_k^{(u)} = -\infty.$$

### Lemma 2 (Matched part)

With mixing weights bounded as  $\pi_h \leq c/k$ ,

$$\mathcal{L}_k^{(m)}(\mathbf{x}) \leq n_m \log(cM) \quad \forall k,$$

where  $n_m = \sum_i (1 - u_i)$  and  $M = \sup_{x,\theta} f^{(m)}(x | \theta) < \infty$ .

⇓ Unmatched term pulls the curve down, matched term is uniformly bounded above.

## Existence of a Finite Optimum $k^*$

### Theorem

Assume:

- $\pi_h \leq c/k$  (bounded weights), with  $c \geq 1$ ,
- $k = \Theta(n)$  (microclustering regime).

Then:

$$k^* = \arg \max_{k \in \{1, \dots, n\}} \log \mathcal{L}_k(\mathbf{x}) \quad \text{exists and } k^* < \infty.$$

## Existence of a Finite Optimum $k^*$

### Theorem

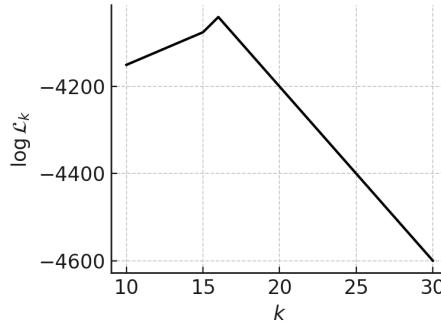
Assume:

- $\pi_h \leq c/k$  (bounded weights), with  $c \geq 1$ ,
- $k = \Theta(n)$  (microclustering regime).

Then:

$$k^* = \arg \max_{k \in \{1, \dots, n\}} \log \mathcal{L}_k(\mathbf{x}) \quad \text{exists and } k^* < \infty.$$

**Behavior of  $\log \mathcal{L}_k$ :**



- It increases until matched likelihood dominates; beyond  $k^*$ , the entropy penalty prevails.

## Practical Notes

**Finding  $k^*$ :** Evaluate the log-likelihood  $\log \mathcal{L}_k$  for  $k = 1, \dots, n$ , or use a faster search / penalized criterion, then select the maximizer.

## Practical Notes

**Finding  $k^*$ :** Evaluate the log-likelihood  $\log \mathcal{L}_k$  for  $k = 1, \dots, n$ , or use a faster search / penalized criterion, then select the maximizer.

**Influence of unmatched count  $n_u$ :** The penalty term  $-n_u H$  increases with the number of unmatched units, and shifts the optimum:

$$n_u \uparrow \Rightarrow k^* \downarrow.$$

## Practical Notes

**Finding  $k^*$ :** Evaluate the log-likelihood  $\log \mathcal{L}_k$  for  $k = 1, \dots, n$ , or use a faster search / penalized criterion, then select the maximizer.

**Influence of unmatched count  $n_u$ :** The penalty term  $-n_u H$  increases with the number of unmatched units, and shifts the optimum:

$$n_u \uparrow \Rightarrow k^* \downarrow.$$

**Relation to ICL:** The entropy adjustment is reminiscent of the *Integrated Completed Likelihood* (ICL; Biernacki et al., 2000, 2010):

$$\text{ICL} = \log p(\mathbf{x} | \hat{\theta}) - \text{separation penalty}.$$

## Practical Notes

**Finding  $k^*$ :** Evaluate the log-likelihood  $\log \mathcal{L}_k$  for  $k = 1, \dots, n$ , or use a faster search / penalized criterion, then select the maximizer.

**Influence of unmatched count  $n_u$ :** The penalty term  $-n_u H$  increases with the number of unmatched units, and shifts the optimum:

$$n_u \uparrow \Rightarrow k^* \downarrow.$$

**Relation to ICL:** The entropy adjustment is reminiscent of the *Integrated Completed Likelihood* (ICL; Biernacki et al., 2000, 2010):

$$\text{ICL} = \log p(\mathbf{x} | \hat{\theta}) - \text{separation penalty}.$$

However, there is a key difference:

- ICL penalizes uncertain assignment for *all* units; the likelihood accounts for the entire dataset.
- Our penalty targets *unmatched units only*, encouraging a clearer separation between genuine matches and noise (small clusters). The likelihood component refers only to matched units.

## Empirical Analysis

## Reparametrization via the “Unmatched Group”

In principle, one could define a latent binary variable

$$U_i = \begin{cases} 1 & \text{if unit } i \text{ is matched (assigned to a real cluster)} \\ 0 & \text{if unit } i \text{ is unmatched (noise, singleton)} \end{cases}$$

## Reparametrization via the “Unmatched Group”

In principle, one could define a latent binary variable

$$U_i = \begin{cases} 1 & \text{if unit } i \text{ is matched (assigned to a real cluster)} \\ 0 & \text{if unit } i \text{ is unmatched (noise, singleton)} \end{cases}$$

**But:** we do not need to introduce  $U_i$  explicitly.

- Each unit has a latent label  $Z_i \in \{1, \dots, k\}$
- Clusters are reordered:  $h = 1, \dots, k_m$  are *matched*,  $h = k_m + 1, \dots, k$  are *singletons*
- Then:

$$U_i = \mathbb{1}_{\{Z_i \leq k_m\}}$$

## Reparametrization via the “Unmatched Group”

In principle, one could define a latent binary variable

$$U_i = \begin{cases} 1 & \text{if unit } i \text{ is matched (assigned to a real cluster)} \\ 0 & \text{if unit } i \text{ is unmatched (noise, singleton)} \end{cases}$$

**But:** we do not need to introduce  $U_i$  explicitly.

- Each unit has a latent label  $Z_i \in \{1, \dots, k\}$
- Clusters are reordered:  $h = 1, \dots, k_m$  are *matched*,  $h = k_m + 1, \dots, k$  are *singletons*
- Then:

$$U_i = \mathbb{1}_{\{Z_i \leq k_m\}}$$

**Advantage:** No need to model  $U$  directly. The “unmatched group” is handled through the joint prior on  $Z$ , and its total weight is:

$$\pi_u = \sum_{h=k_m+1}^k \pi_h$$

## Choosing a Practical Weight-Bound

**Finite-sample guarantee (Thm. 2).** For  $(M_1, \dots, M_k) \sim \text{Mult}(n; \pi_1, \dots, \pi_k)$ :

$$\Pr(M_{(k)} > S) \leq k \exp\left\{-2n\left(\frac{S}{n} - \frac{\pi_h}{k}\right)^2\right\}.$$

- Setting  $c \leq c_{\max}(S, \alpha)$  controls  $\Pr(M_{(k)} > S) \leq \alpha$ , but for  $n=1,000$ ,  $k=200$ ,  $S=5$ ,  $\alpha=10^{-4}$   $c_{\max} \approx 16 \Rightarrow \pi_h \leq 0.08$  — far too loose.

## Choosing a Practical Weight-Bound

**Finite-sample guarantee (Thm. 2).** For  $(M_1, \dots, M_k) \sim \text{Mult}(n; \pi_1, \dots, \pi_k)$ :

$$\Pr(M_{(k)} > S) \leq k \exp\left\{-2n\left(\frac{S}{n} - \frac{c}{k}\right)^2\right\}.$$

- Setting  $c \leq c_{\max}(S, \alpha)$  controls  $\Pr(M_{(k)} > S) \leq \alpha$ , but for  $n=1,000$ ,  $k=200$ ,  $S=5$ ,  $\alpha=10^{-4}$   $c_{\max} \approx 16 \Rightarrow \pi_h \leq 0.08$  — far too loose.
- In the microclustering regime ( $n\pi_h = O(1)$ ) each  $M_h$  is approximately  $\text{Pois}(\lambda)$  with  $\lambda = n\pi_h$ . Require

$$k [1 - F_{\text{Pois}}(\lambda)(S)] \leq \alpha \implies \pi_h \leq \frac{\lambda_{\max}(S, \alpha)}{n}.$$

$\alpha$	$\lambda_{\max}$	bound $\pi_h^{\max} = \lambda_{\max}/n$
$10^{-2}$	0.63	$6.3 \times 10^{-4}$
$10^{-3}$	0.42	$4.2 \times 10^{-4}$
$10^{-4}$	0.28	$2.8 \times 10^{-4}$

With  $\pi_h \leq 2.8 \times 10^{-4}$  the expected cluster size is  $\lambda_{\max} \approx 0.28$  and  $\Pr(M_{(k)} > 5) \leq 10^{-4}$ .

## Estimation via the EM Algorithm

**Component model.** Matched components follow a latent-class structure

$$f_h(\mathbf{x}_i \mid \theta_h) = \prod_{j=1}^p \prod_{l=1}^{L_j} \theta_{hjl}^{\mathbb{1}[x_{ij}=l]}.$$

### Initialisation.

- Compute pairwise Jaccard distances.
- Records with no close neighbour (dist.  $> 0.25$ )  $\Rightarrow$  initialise as *unmatched*.
- Remaining records clustered into  $k_m$  matched groups (agglomerative).
- Starting values  $\pi_h, \theta_h$ : smoothed empirical frequencies;  $\xi = \exp\left(\sum_{j,l} \frac{n_{jl}}{n} \log \frac{n_{jl}}{n}\right)$ .

### E-step. Posterior responsibilities

$$\tau_{ih} = \frac{\pi_h f(\mathbf{x}_i \mid z_i = h)}{\sum_{r=1}^{k_m+1} \pi_r f(\mathbf{x}_i \mid z_i = r)}.$$

## EM Algorithm: M-step with constrained weight updates

**Goal.** Enforce the Poisson-based bound on the mixing weights:

$$\pi_h \leq b \quad \text{with} \quad b = \frac{\lambda_{\max}(S, \alpha)}{n}.$$

**Constrained M-step.** We maximize the expected complete-data log-likelihood:

$$\max_{\pi} \sum_{h=1}^k n_h \log \pi_h \quad \text{subject to} \quad \sum_h \pi_h = 1, \quad 0 \leq \pi_h \leq b.$$

This is a concave optimization problem with linear constraints, solved via the method of Lagrange multipliers.

## EM Algorithm: M-step with constrained weight updates

**Goal.** Enforce the Poisson-based bound on the mixing weights:

$$\pi_h \leq b \quad \text{with} \quad b = \frac{\lambda_{\max}(S, \alpha)}{n}.$$

**Constrained M-step.** We maximize the expected complete-data log-likelihood:

$$\max_{\pi} \sum_{h=1}^k n_h \log \pi_h \quad \text{subject to} \quad \sum_h \pi_h = 1, \quad 0 \leq \pi_h \leq b.$$

This is a concave optimization problem with linear constraints, solved via the method of Lagrange multipliers.

**Parameter update.** With updated  $\pi$ , the component parameters for the matched clusters are updated by:

$$\theta_{hjl} = \frac{\sum_i \tau_{ih} \mathbb{1}[x_{ij} = l]}{\sum_i \tau_{ih}}, \quad h \leq k_m.$$

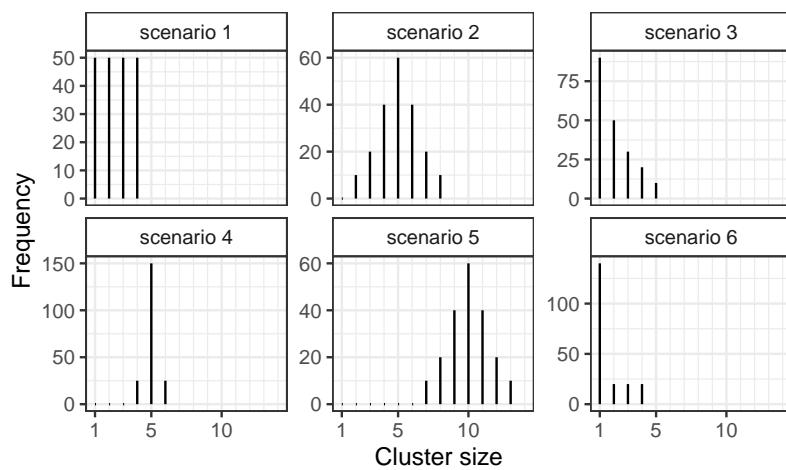
## Simulation Study: Microclustering Partitions

We evaluate our model for microclustering on synthetic data, replicating the setup in Betancourt et al. (2016). Six different ground-truth partitions are considered, all with  $k = 200$  groups but varying cluster-size distributions.

- Data simulated using the graphical RL model from Steorts et al. (2016), via the `microclustr` R package.
- Each unit is described by  $p = 5$  categorical variables with  $L_j = 10$  categories.
- Distortion levels:  $\beta \in \{0.01, 0.05, 0.1\}$  – higher values imply more noise.

## Cluster Size Distributions in Simulated Data

- All six scenarios have  $k = 200$  clusters.
- Scenarios 1–5 replicate Betancourt et al. (2016); Scenario 6 adds a classical record linkage case with many singletons.



## Competing Methods

**Baseline:** Exchangeable Sequence of Clusters (ESC) from Betancourt et al. (2016)

- Fitted with 20,000 MCMC samples after 5,000 burn-in.
- Two priors considered: ESC with Dirichlet (ESCD) and with Negative Binomial (ESCNB).
- Results shown only for ESCD (best performing).

**Our method:**

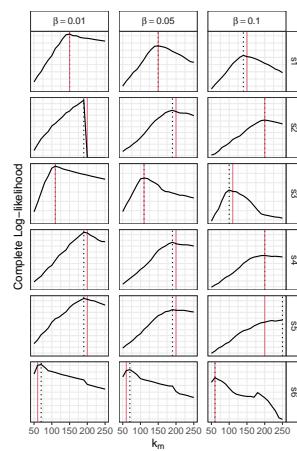
- Penalized likelihood EM with  $\alpha_0 = 1.5$ ,  $\alpha_1 = 100$ .
- Grid search over  $k_m \in [50, 250]$ .
- Model with highest complete log-likelihood selected.

## Model Selection via Penalized Log-Likelihood

- Selection over  $k_m$  based on complete log-likelihood.
- This acts like a discrete uniform prior over a broad grid:

$$k_m \in [n - a\sqrt{n}, n], \quad 0 < a < \sqrt{n}$$

- Empirical log-likelihood curves show non-monotonic behavior, as expected.



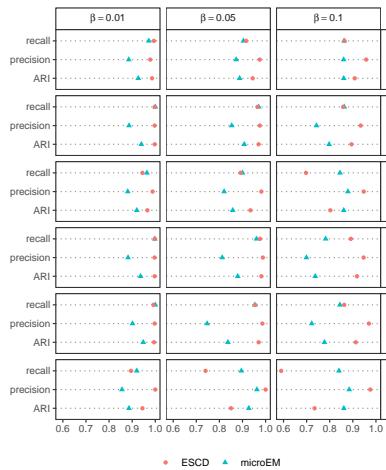
## Clustering Accuracy: ARI, Recall, Precision

### Evaluation Metrics:

- **Adjusted Rand Index (ARI):** standard clustering metric.
- **Recall & Precision:** computed on pairwise co-assignment indicators.

### Post-processing:

- Unmatched units in our model are labeled with unique singleton IDs.
- ESCD metrics are averaged over MCMC samples.



## Conclusions (1/2)

### ■ Frequentist micro-clustering (“microEM”) in a nutshell

- ▶ deterministic EM; unmatched units absorbed by an *entropy–driven uniform* component;
- ▶ finite-sample bounds on the largest cluster ( $M_{(k)}/n$ )  $\Rightarrow$  *provable* micro-behaviour.

### ■ Real data (SIPP1000) performance

- ▶ increasing cluster-size mix (0.09 — 0.12 — 0.12 — 0.16 — **0.50**);
- ▶ ARI = **0.93** > supervised Fellegi-Sunter;
- ▶ below ESC-D ( $\approx 0.98$ ) that has a truncated Negative-Binomial base making sizes  $\geq 6$  rapidly unlikely.
- ▶ our method performs better when these proportions *flip* (sim scenarios 4-6).

## Discussion & Outlook (2/2)

### ■ Key advantages vs. ESC-D

- ▶ **Runtime:** minutes vs. hours of MCMC;
- ▶ no hyper-priors, no convergence diagnostics;
- ▶ deterministic, fully reproducible, easy to parallelise.
- ▶ **Automatic  $k$  selection:** the log-likelihood is *non-monotonic* in  $k$ , so we pick the optimal number of clusters at its peak — no external criteria like BIC or AIC needed.

### ■ Limitations & next steps

- ▶ add a *data-driven size penalty* for highly skewed size profiles;
- ▶ hybrid workflow: EM warm-start + short MCMC for posterior uncertainty;
- ▶ extend likelihood to mixed-type features (categorical, discrete-triangular, continuous).

## Claire Gormley

### *Material list:*

Majumdar K., Jaffrézic F., Rau A., Gormley I.C., Murphy T.B. (2024) Integrated differential analysis of multi-omics data using a joint mixture model: idiffomix. arXiv:2412.17511, <https://arxiv.org/abs/2412.17511>

# Integrated differential analysis of multi-omics data using a joint mixture model: idiffomix

Koyel Majumdar<sup>1</sup>, Florence Jaffrézic<sup>2</sup>, Andrea Rau<sup>2</sup>, Isobel Claire Gormley<sup>\*1</sup>, and Thomas Brendan Murphy<sup>1</sup>

<sup>1</sup>School of Mathematics and Statistics, University College Dublin, Ireland.

<sup>2</sup>INRAE, UMR1313 AgroParisTech, GABI, Universit Paris-Saclay, France.

## Abstract

Gene expression and DNA methylation are two interconnected biological processes and understanding their relationship is important in advancing understanding in diverse areas, including disease pathogenesis, environmental adaptation, developmental biology, and therapeutic responses. Differential analysis, including the identification of differentially methylated cytosine-guanine dinucleotide (CpG) sites (DMCs) and differentially expressed genes (DEGs) between two conditions, such as healthy and affected samples, can aid understanding of biological processes and disease progression. Typically, gene expression and DNA methylation data are analysed independently to identify DMCs and DEGs which are further analysed to explore relationships between them, or methylation data are aggregated to gene level for analysis. Such approaches ignore the inherent dependencies and biological structure within these related data.

A joint mixture model is proposed that integrates information from the two data types at the modelling stage to capture their inherent dependency structure, enabling simultaneous identification of DMCs and DEGs. The model leverages a joint likelihood function that accounts for the nested structure in the data, with parameter estimation performed using an expectation-maximisation algorithm.

---

<sup>\*</sup>claire.gormley@ucd.ie

Performance of the proposed method, ***idiffomix***, is assessed through a thorough simulation study and the approach is used to analyse RNA-Seq and DNA methylation array data from matched healthy and breast invasive carcinoma tumour samples from The Cancer Genome Atlas (TCGA) project. Several genes, identified as non-differentially expressed when the data types were modelled independently, had high likelihood of being differentially expressed when associated methylation data were integrated into the analysis. Subsequent gene ontology analysis indicated that many of the differentially methylated CpG sites and differentially expressed genes identified by the joint mixture model are involved in important, in some cases cancer related, biological processes and pathways.

The ***idiffomix*** approach highlights the advantage of an integrated analysis via a joint mixture model over independent analyses of the two data types; genome-wide and cross-omics information is simultaneously utilised providing a more comprehensive view. An open source R package is available to facilitate widespread use of ***idiffomix***.

**Keywords:** DNA methylation, gene expression, integrated analysis, joint mixture model, EM algorithm.

## 1 Introduction

The epigenetic process of methylation/demethylation of a cytosine-guanine dinucleotide (CpG) site in DNA is an important biomarker in cancer studies [Berger et al., 2009]. This addition or removal of a methyl group from the C5 of a cytosine ring is a heritable change that does not result in DNA sequence alteration [Moore et al., 2013]. Gene promoter regions are the DNA sequences located in the upstream direction, i.e. 5' direction of the transcription start site (TSS). Methylation in these promoter regions can block the binding of necessary proteins and transcription factors to the DNA, preventing initiation of transcription of the gene to messenger RNA (mRNA). Methylated DNA is also known to modify histones to produce compact chromatin structure making the DNA less accessible for transcription [Moore et al., 2013]. As DNA methylation makes DNA less accessible for transcription, it thus plays a significant role in controlling gene expression and cellular growth [Bird, 2002, Suzuki and Bird, 2008]. Silencing of tumour suppressor genes due to hypermethylation of promoter genes has been studied as an important biomarker in cancer settings [Jones and Takai, 2001, Jones and Baylin, 2002].

Genes responsible for cell growth and division, including some oncogenes have been observed to become hyperactive due to hypomethylation [Van Tongelen et al., 2017]. The relationship between gene expression regulation and DNA methylation has been explored to identify key genes and assess how methylation impacts their function in various neurodegenerative diseases, such as Alzheimer’s disease [Semick et al., 2019]. Differential analysis of gene expression and DNA methylation is therefore important for understanding various biological processes and for early detection and treatment of various diseases. Comprehensive measurement of gene expression and DNA methylation has been made possible on a genome wide scale by high-throughput technologies.

A number of statistical models have been proposed to model gene expression and DNA methylation datasets separately in order to respectively identify differentially expressed genes (DEGs) and differentially methylated CpG sites (DMCs). For gene expression data, models like `limma-voom`, `edgeR` and `DESeq2` employ data transformation for empirical Bayesian modelling or negative binomial distribution models to identify DEGs [Robinson and Oshlack, 2010, Law et al., 2014, Love et al., 2014]. For DNA methylation analysis, statistical models like `limma`, `PanDM`, `FastDMA` and `betaclust` have been developed to identify DMCs by employing techniques like empirical Bayesian modelling, nonparametric tests and mixture modelling [Wu et al., 2013, Ritchie et al., 2015, Shi et al., 2020, Majumdar et al., 2024]. Xu et al. [2019] identify DEGs and DMCs independently and subsequently employ a correlation analysis to study the associations between them while Xie et al. [2011] identified DEGs and employed *t*-tests and regression analysis to infer if changes in methylation were correlated to expression change. Alivand et al. [2021] used network analysis to combine DEGs and DMCs to study their association. To find tumour specific candidate genes, Wang et al. [2018] examined the association between differentially methylated regions (DMRs) and DEGs. In these “two-step” approaches, subtle yet biologically relevant associations between the two data types can be missed when they are modelled independently; integrating the data through a single model would allow simultaneous identification of DEGs and DMCs, providing a more comprehensive analysis of gene regulation being affected by aberrant methylation.

An integrated approach to modelling gene expression and DNA methylation data was proposed by Spainhour et al. [2019] where Pearson’s correlation was employed to study the associ-

ations between the two data types . The Bioconductor package `iNETgrate` [Sajedi et al., 2023] incorporates the two data types at the modelling stage by combining them to form a single gene network. The `INTEND` model [Itai et al., 2023] uses Lasso regression to predict the gene expression level based on associated methylation values. Kormaksson et al. [2012] proposed an integrated mixture model to cluster omics data independently as well as jointly to identify biologically meaningful clusters. The `EBADIMEX` method [Madsen et al., 2019] proposes moderated *t*-tests and *F*-tests with empirical Bayes priors to integrate the two data types for differential analysis and sample classification while the `BioMethyl` package [Wang et al., 2019] uses linear regression models to analyse the impact of each CpG site's methylation on gene expression. Jeong et al. [2010] proposed an integrated empirical Bayes model using marginal linear models to integrate the two data types and applied several user-defined thresholds to classify the genes. While these models jointly consider gene expression and methylation array data, they typically operate at the gene level, neglecting the nested dependency structure between CpG sites and their corresponding genes. There is therefore a need to simultaneously analyse both data types, while accounting for their inherent dependency structure, when identifying DEGs and DMCs. Specifically, the nested dependency structure, created by mapping CpGs to genes based on their positions within promoter regions, needs to be accounted for – while some genes may experience differential methylation in associated CpG sites without corresponding differential expression, others could exhibit differential expression due to epigenetic factors other than methylation.

Here, a joint mixture model approach, termed `idiffomix`, is proposed for the integrated differential analysis of gene expression data and methylation array data that accounts for their nested dependency structure. A key dependency matrix parameter is used in the joint mixture model to allow the methylation state of a CpG site to depend on the expression level status of the gene to which it is associated. The model parameters are estimated via an EM algorithm [Dempster et al., 1977]. In the E-step, as the expected values of the latent variables are intractable, approximate but tractable estimates are employed. As chromosomes can be assumed to be independent, the joint mixture model is parallelized across chromosomes to enhance computational efficiency. The performance of `idiffomix` is assessed through a thorough simulation study and application to a publicly available breast cancer dataset where the data are derived from bulk tissue. Several genes implicated in breast cancer, identified to be non-DEGs

when the two data types are modelled independently, are inferred to be differentially expressed when their corresponding CpG sites' methylation is taken into consideration. Further, gene enrichment analysis revealed the identified DEGs and DMCs to be associated with significant biological processes and several cancer related pathways, e.g., *MAPK cascade*, *cAMP signaling pathway*, and *Hippo signaling pathway*. To facilitate widespread implementation of ***idiffomix***, an open source R package is available at [Github](#).

## 2 Gene expression and methylation data

### 2.1 Quantifying gene expression and DNA methylation

Gene expression levels can be quantified as count values using RNA sequencing (RNA-Seq) technology [Wang et al., 2009] where the count values represent the number of sequencing reads that align to each gene. An increase in gene expression count between biological conditions suggests upregulation of a gene, while a decrease in count suggests downregulation. Differences in gene expression count levels between biological conditions are analysed to identify DEGs. For accurate comparison of gene expression between conditions, RNA-Seq data needs to be normalised to account for differences in library size, which refers to the total number of reads sequenced for each sample. Library size scaling factors are calculated using the Trimmed Mean of M-values approach [Robinson and Oshlack, 2010], and raw counts for each gene are divided by the normalized library sizes and multiplied by one million to get counts per million (CPM) values. Highly expressed genes can increase the variance in the data therefore the CPM values are typically transformed to log counts per million (log-CPM) to stabilise the variance and facilitate analysis using Gaussian models [Law et al., 2014].

Methylation arrays quantify the level of methylation at a CpG site as a *beta* value [Du et al., 2010]. These values are continuous and bounded in nature ranging from 0 to 1, where a value close to 0 indicates hypomethylation, a value close to 0.5 indicates hemimethylation and a value close to 1 indicates hypermethylation at a CpG site. Differences in methylation levels are analysed across different biological conditions to identify DMCs. For analysis, as bounded *beta* values violate Gaussian assumptions, typically *beta* values are transformed using a *logit* transformation to produce *M*-values. Such a transformation results in real valued data which

is more appropriate for the application of Gaussian models.

From each RNA sample, gene expression levels of  $G$  genes are quantified while from each DNA sample, methylation levels at  $C$  CpG sites are quantified, where typically  $C \gg G$ . The RNA-Seq and methylation array data, each collected from the same set of individuals, exhibit a many-to-one mapping, in that each CpG site can be assigned to a single gene, although some genes may have no associated CpG sites. This nested structure results in a complex relationship between gene expression and methylation patterns, as variations in methylation levels at multiple CpG sites within a genomic locus may collectively be associated with variations in gene expression.

## 2.2 Motivating breast cancer study

Gene expression and methylation data in matched healthy and breast invasive carcinoma tumour samples from The Cancer Genome Atlas ([TCGA](#)) are publicly available in the Genomic Data Commons ([GDC](#)) data portal. Here, a sub-selection of data from  $N = 5$  patients with the *duct and lobular neoplasm* disease type and for whom both gene expression and methylation array data were available, were analysed. The donor ID (case ID in parenthesis) of patients selected are: DO1253 (TCGA-E9-A1NG), DO1254 (TCGA-BH-A0AU), DO1257 (TCGA-BH-A0DG), DO1283 (TCGA-BH-A1EU) and DO1299 (TCGA-BH-A0BM). For each patient, RNA-Seq data along with 450K methylation array data were obtained from normal (herein benign) and primary tumour breast tissues using Illumina HiSeq Sequencing and Illumina HumanMethylation450 BeadChip array [[Steemers and Gunderson, 2005](#)], respectively. The data are derived from bulk tissue rather than single-cell samples, capturing the aggregate molecular signals across all cell types within the tissue. The RNA-Seq data comprised of gene expression levels for 20,499 genes while the methylation data comprised of methylation levels for 394,356 CpG sites.

The RNA-Seq data consisted of raw counts depicting the gene expression levels. To ensure data quality, only genes whose sum of expression counts across both biological conditions  $> 5$  were retained, resulting in  $G = 15,722$  genes. The data were normalized to account for differences in library sizes. The normalized count data were used to obtain CPM values which were further log-transformed to obtain log-CPM values. Given the paired design of the motivating

setting, the log-fold changes between the tumour and benign samples were calculated for each gene in every patient and used in the subsequent analyses. For the methylation array data, CpG sites located within promoter regions were selected, specifically those within 200 nucleotides of the transcription start site (TSS200), the first exon, or the 5' untranslated region (UTR) region, resulting in  $C = 94,873$  CpG sites. The *beta* values at these CpG sites were then *logit* transformed to  $M$ -values. Similar to the RNA-Seq data, given the paired design, the difference in  $M$ -values between tumour and benign samples were calculated for each CpG site in every patient and used in the subsequent analyses.

### 3 The `idiffomix` method

#### 3.1 A joint mixture model for gene expression and methylation data

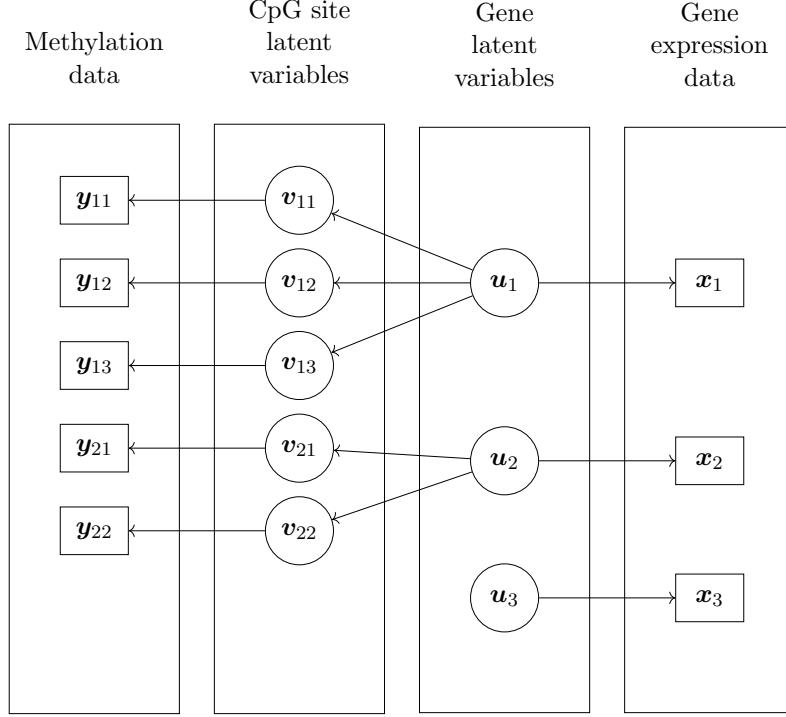
Given the paired experimental design of the motivating setting, the log-fold change of the normalized RNA-Seq data between the two biological conditions for the  $g^{th}$  gene is denoted  $\mathbf{x}_g$ . Specifically,  $\mathbf{x}_g = (x_{g1}, \dots, x_{gn}, \dots, x_{gN})$ , where  $x_{gn}$  signifies the log-fold change in gene expression levels for the  $g^{th}$  gene from the  $n^{th}$  patient's RNA sample. The collection of log-fold change values for all  $G$  genes and  $N$  patients is denoted  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_g, \dots, \mathbf{x}_G)$ , a  $G \times N$  matrix. Similarly, the difference in  $M$ -values for CpG site  $c$  located on gene  $g$  between the two biological conditions is denoted  $\mathbf{y}_{gc} = (y_{gc1}, \dots, y_{gcn}, \dots, y_{gCN})$ , where  $y_{gcn}$  represents the difference in  $M$ -values at CpG site  $c$ , located on gene  $g$  from patient  $n$ 's DNA sample. Each gene  $g$  has  $C_g$  associated CpG sites such that  $C = \sum_{g=1}^G C_g$ . It is possible for a gene to have no associated CpG sites, in which case  $C_g = 0$  for that gene. The collection of  $M$ -value differences between the two biological conditions for all  $C$  CpG sites and  $N$  patients is denoted  $\mathbf{Y} = (\mathbf{y}_{11}, \dots, \mathbf{y}_{1C_1}, \dots, \mathbf{y}_{g1}, \dots, \mathbf{y}_{gC_g}, \dots, \mathbf{y}_{G1}, \dots, \mathbf{y}_{GC_G})$ , a  $C \times N$  matrix.

To simultaneously identify DEGs and DMCs, while accounting for the nested structure between genes and CpG sites, a joint mixture modelling approach, termed `idiffomix`, is proposed. Expression levels at each gene are assumed to undergo one of three possible state changes between the benign and tumour conditions: if expression levels decrease between the tumour and matched normal samples, the gene is considered downregulated (E-) and will have a large negative log-fold change. Conversely, if expression levels increase in the tumour sample, the gene is

deemed upregulated (E+) with large positive log-fold change value. If no change is observed, a gene is categorized as non-differentially expressed (E0), with log-fold change values close to 0. Thus, under `idiffomix`, the log-fold changes in the RNA-Seq data are assumed to have been generated from a heterogeneous population modelled by a  $K = 3$  component mixture model. Similarly, methylation levels at a CpG site are also assumed to undergo one of three possible state changes between the two biological conditions: a CpG site is considered hypomethylated (M-) if the methylation level decreases between the tumour and benign samples, resulting in large negative differences in  $M$ -values. A CpG site is deemed to be hypermethylated (M+) if methylation increases in the tumour sample compared to the benign sample, represented by large positive differences in  $M$ -values. A CpG site will be non-differentially methylated (M0) if the difference in  $M$ -values between the two conditions is close to 0. Thus, the differences in  $M$ -values are assumed to be generated from an  $L = 3$  component mixture model.

As is typical in mixture models, an incomplete data approach is employed to facilitate inference. For gene  $g$ , an indicator variable  $u_{gk}$  is introduced, where  $u_{gk} = 1$  if the gene  $g$  belongs to cluster  $k$ , for  $g = 1, 2, \dots, G$  and  $k = 1, 2, \dots, K$ . The collection of indicator variables for  $G$  genes is denoted  $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_g, \dots, \mathbf{u}_G)$ , a  $G \times K$  matrix. Further, within each component, the log-fold change data are assumed to be independent and identically Gaussian distributed, i.e.,  $x_{gn}|(u_{gk} = 1) \sim N(\mu_k, \sigma_k^2)$ , for  $g = 1, 2, \dots, G$ ,  $n = 1, 2, \dots, N$  and  $k = 1, 2, \dots, K$ . Similarly, for CpG site  $C$ , an indicator variable  $v_{gcl}$  is introduced, where  $v_{gcl} = 1$  if CpG site  $c$ , located on gene  $g$ , belongs to cluster  $l$ . The collection of indicator variables for  $C$  CpG sites is denoted  $\mathbf{V} = (\mathbf{v}_{11}, \dots, \mathbf{v}_{1C_1}, \dots, \mathbf{v}_{g1}, \dots, \mathbf{v}_{gC_g}, \dots, \mathbf{v}_{G1}, \dots, \mathbf{v}_{GC_G})$ , a  $C \times L$  matrix. The differences in  $M$ -values are also assumed to be independent and identically Gaussian distributed within a component, i.e.,  $y_{gcn}|(v_{gcl} = 1) \sim N(\lambda_l, \rho_l^2)$  for  $g = 1, 2, \dots, G$ ,  $c = 1, 2, \dots, C_g$ ,  $n = 1, 2, \dots, N$  and  $l = 1, 2, \dots, L$ .

To account for the nested dependency structure between genes and CpG sites, the two mixture models are integrated. Firstly, the proportion of genes belonging to each cluster is denoted as  $\boldsymbol{\tau} = (\tau_1, \dots, \tau_k, \dots, \tau_K)$ . The dependencies between the genes and CpG sites are then accounted for through a key  $L \times K$  matrix parameter  $\boldsymbol{\pi}$ . The value  $\pi_{l|k}$  is the probability of a CpG site belonging to cluster  $l$ , given its associated gene belongs to cluster  $k$ , for  $l = 1, 2, \dots, L$  and  $k = 1, 2, \dots, K$ . Figure 1 provides a graphical model of the `idiffomix`



**Figure 1: Graphical model of `idiffomix`.**

The joint mixture model for integrated differential analysis of gene expression and methylation data. Here, for example, gene expression data arises from 3 genes: gene 1 has 3 associated CpG sites while gene 2 has 2 associated CpG sites. Gene 3 has no associated CpG sites.

model structure.

Denoting the parameters in the gene expression mixture model collectively as  $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K)$ , where  $\boldsymbol{\theta}_k = (\mu_k, \sigma_k^2)$ , and the parameters in the methylation data clusters collectively as  $\boldsymbol{\phi} = (\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_L)$ , where  $\boldsymbol{\phi}_l = (\lambda_l, \rho_l^2)$ , the joint probability density function for the log-fold change RNA-Seq data  $\mathbf{X}$ , differences in  $M$ -value methylation data  $\mathbf{Y}$  and latent variables  $\mathbf{U}$  and  $\mathbf{V}$  is,

$$P(\mathbf{X}, \mathbf{Y}, \mathbf{U}, \mathbf{V} | \boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \prod_{g=1}^G \left\{ \prod_{k=1}^K P(\mathbf{x}_g | \boldsymbol{\theta}_k)^{u_{gk}} \prod_{c=1}^{C_g} \prod_{l=1}^L P(\mathbf{y}_{gc} | \boldsymbol{\phi}_l)^{v_{gcl}} \right\} \\ \times \prod_{g=1}^G \prod_{k=1}^K \left\{ \tau_k \prod_{c=1}^{C_g} \prod_{l=1}^L \pi_{l|k}^{v_{gcl}} \right\}^{u_{gk}}. \quad (1)$$

If a gene has no associated CpG sites then the products over  $c$  and  $l$  in (1) are equal to one.

From (1), it is clear that if  $\pi_{l|k} = \pi_{l|k'}$  for all  $k, k'$ , then the status of CpG sites and genes are independent. In such a case, the model is equivalent to two independent mixture models.

Assuming that the data from the  $N$  patients are conditionally independent given their component membership then the probability of the RNA-Seq data is  $P(\mathbf{x}_g|\boldsymbol{\theta}_k) = \prod_{n=1}^N p(x_{gn}|\boldsymbol{\theta}_k)$ , and similarly, for the methylation data,  $P(\mathbf{y}_{gc}|\boldsymbol{\phi}_l) = \prod_{n=1}^N p(y_{gcn}|\boldsymbol{\phi}_l)$ . Thus the complete data log-likelihood function for the joint mixture model is,

$$\begin{aligned}\ell_C(\boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\phi} | \mathbf{X}, \mathbf{Y}, \mathbf{U}, \mathbf{V}) &= \sum_{g=1}^G \sum_{k=1}^K \sum_{n=1}^N u_{gk} \log p(x_{gn}|\boldsymbol{\theta}_k) \\ &\quad + \sum_{g=1}^G \sum_{c=1}^{C_g} \sum_{l=1}^L \sum_{n=1}^N v_{gcl} \log p(y_{gcn}|\boldsymbol{\phi}_l) \\ &\quad + \sum_{g=1}^G \sum_{k=1}^K u_{gk} \log \tau_k + \sum_{g=1}^G \sum_{k=1}^K \sum_{c=1}^{C_g} \sum_{l=1}^L u_{gk} v_{gcl} \log \pi_{l|k}.\end{aligned}\tag{2}$$

### 3.2 Inference via an EM algorithm

The expectation-maximisation (EM) algorithm [Dempster et al., 1977] is used to compute, from (2), the maximum likelihood estimates of the model parameters, i.e.,  $\hat{\boldsymbol{\tau}}$ ,  $\hat{\boldsymbol{\pi}}$ ,  $\hat{\boldsymbol{\theta}}$ , and  $\hat{\boldsymbol{\phi}}$  and the expected values of the latent variables, conditional on the current parameter estimates and the observed data.

At the E-step of the EM algorithm, the expected value of the complete data log-likelihood function (2) with respect to the conditional distributions of the latent variables is calculated, given the observed data and the current estimates of the model parameters. The required expected values are therefore,

$$\begin{aligned}\hat{u}_{gk} &= \mathbb{E}(u_{gk} | \mathbf{x}_g, \boldsymbol{\tau}_k, \boldsymbol{\pi}_{l|k}, \boldsymbol{\theta}_k), \\ \hat{v}_{gcl} &= \mathbb{E}(v_{gcl} | \mathbf{y}_{gc}, \boldsymbol{\pi}_{l|k}, \boldsymbol{\phi}_l) \\ \widehat{u_{gk} v_{gcl}} &= \mathbb{E}(u_{gk} v_{gcl} | \mathbf{x}_g, \mathbf{y}_{gc}, \boldsymbol{\tau}_k, \boldsymbol{\pi}_{l|k}, \boldsymbol{\theta}_k, \boldsymbol{\phi}_l),\end{aligned}$$

for  $g = 1, 2, \dots, G$ ,  $k = 1, 2, \dots, K$ ,  $c = 1, 2, \dots, C_g$  and  $l = 1, 2, \dots, L$ . While these are intractable, a tractable approximation is available by considering the conditional expected values of each latent variable given the other and employing an algorithm similar to that proposed in Salter-Townshend and Murphy [2013] and Chamroukhi and Huynh [2018] at the E-step (see Appendix A.1 for full details). For the M-step, the expected complete data log-likelihood function is maximised with respect to the model parameters  $\boldsymbol{\tau}$ ,  $\boldsymbol{\pi}$ ,  $\boldsymbol{\theta}$  and  $\boldsymbol{\phi}$ ; closed

form solutions are available and full details are given in Appendix A.1.

To initialise the EM algorithm, here the latent variables  $\mathbf{U}$  and  $\mathbf{V}$  are initialised. Clustering algorithms such as  $k$ -means or `mclust` [Scrucca et al., 2016] could be used to independently cluster the two data types into  $K = 3$  and  $L = 3$  clusters. Here we instead chose to employ a quantile initialization approach to ensure that the initial clusters were well separated. The quantile initialization approach was applied to the two data types independently such that the E- and M- clusters consisted of genes and CpG sites, respectively, with the lowest 10% differential expression and methylation levels, the E+ and M+ clusters contained genes and CpG sites, respectively, with the highest 10% of differential expression and methylation levels and the remaining genes and CpG sites were allocated to clusters E0 and M0 respectively. Given the resulting initialised values of  $\mathbf{U}$  and  $\mathbf{V}$ , an M-step was used to calculate starting values for the model parameters  $\boldsymbol{\tau}$ ,  $\boldsymbol{\pi}$ ,  $\boldsymbol{\theta}$ , and  $\boldsymbol{\phi}$ . The E-step and M-step were then iterated until convergence which here was deemed to be achieved when the absolute change in all parameter estimates between successive iterations was less than the threshold of  $1 \times 10^{-5}$ , to yield estimates  $\hat{\boldsymbol{\tau}}$ ,  $\hat{\boldsymbol{\pi}}$ ,  $\hat{\boldsymbol{\theta}}$  and  $\hat{\boldsymbol{\phi}}$ .

On convergence of the EM-algorithm, the estimates of the latent variables  $\mathbf{U}$  and  $\mathbf{V}$  provide the posterior probabilities of cluster membership for genes and CpG sites respectively. Cluster assignment is then performed using the maximum a posteriori (MAP) rule, where each gene or CpG site is assigned to the cluster for which it has highest posterior probability of membership. Thus, DEGs are deemed to be those assigned to clusters E- and E+ while DMCs are deemed to be those assigned to clusters M- and M+, based on the joint analysis of both datasets. The uncertainty of gene  $g$ 's cluster assignment is available as  $1 - \max_{k=1,\dots,K}(\hat{u}_{gk})$  while the uncertainty of CpG site  $c$ 's cluster assignment is available as  $1 - \max_{l=1,\dots,L}(\hat{v}_{gc_l})$ , providing deeper insight than available via a hard clustering approach.

Due to independence of chromosomes and to ease the computational burden, the model is fitted to each chromosome independently in parallel. Assuming a genome has  $J$  chromosomes the joint mixture model in (1) is fitted separately  $J$  times, once for each chromosome  $j = 1, 2, \dots, J$ . In each instance the model is fitted to the  $G_j$  genes and  $C_j$  CpG sites associated with gene  $j$ .

### 3.3 A generalised joint mixture model

While the joint mixture model in (1) was formulated based on the paired experimental design of the motivating study outlined in Section 2.2, the `idiffomix` method can be generalised to other experimental designs and omic datasets, for example where two omics data types are collected from the same individuals in  $R \geq 2$  different biological conditions or time points. In such a generalized scenario, while the observed omics data may be transformed to e.g., achieve Gaussianity or stabilise variances, the (transformed) observed data, rather than fold changes or differences between the  $R$  replicates, could be modelled directly. In this generalised `idiffomix` model, the RNA-Seq data, say, at gene  $g$  are denoted  $\mathbf{x}_g = (x_{g11}, \dots, x_{gnr}, \dots, x_{gNR})$ , where  $x_{gnr}$  represents the (transformed) gene expression level at the  $g^{th}$  gene, collected from the  $n^{th}$  patient's  $r^{th}$  biological condition or time point. Similarly, at the  $c^{th}$  CpG site the methylation data  $\mathbf{y}_{gc} = (y_{gc11}, \dots, y_{gcnr}, \dots, y_{gcNR})$  where  $y_{gcnr}$  denotes the (transformed) *beta* value at the  $c^{th}$  CpG site located on the  $g^{th}$  gene, collected from the  $n^{th}$  patient's  $r^{th}$  biological condition or time point. The values of  $K$  and  $L$  may vary depending on the study objective, similar to the approach taken in [Majumdar et al. \[2024\]](#). Finally, while here the `idiffomix` model assumes Gaussian distributions for  $P(\mathbf{x}_g)$  and  $P(\mathbf{y}_{gc})$ , this need not be the case and other distributions could be specified as appropriate for the data and research question at hand, with or without data transformations as needed.

## 4 Results

### 4.1 Simulation study

#### 4.1.1 Data generation

To assess the performance of `idiffomix`, a simulation study was conducted where the simulation settings were chosen to mirror those in the motivating breast cancer study data. One hundred RNA-Seq and methylation array datasets were simulated based on (1) from two biological conditions (conditions A and B), from  $N = 4$  patients and  $K = L = 3$ . For the simulated RNA-Seq data,  $G = 500$  genes. Based on the TCGA data, the number of CpG sites linked to each gene ranged from 1 to 100, with 95% of genes having fewer than 30 CpG sites. Therefore, for the simulated methylation data, the number of CpGs linked to each gene was drawn from

a uniform distribution  $U(3, 30)$  resulting in the simulated methylation datasets having 5,000 CpG sites, on average.

The RNA-Seq raw counts under condition A were generated from a negative binomial distribution with mean of 10,000 and dispersion parameter of 5. Ten percent of the  $G = 500$  genes were simulated to be downregulated (i.e., in cluster E-) and raw counts for these genes in condition B were simulated from a negative binomial distribution with mean and dispersion parameters of 4,000 and 5 respectively. Similarly, 10% of the genes were simulated to be upregulated (i.e., in cluster E+) with counts generated for condition B from a negative binomial with mean and dispersion parameters of 60,000 and 5 respectively. For non-differential genes (i.e., those in cluster E0), raw counts for condition B were generated as for condition A, i.e., from a negative binomial distribution with mean and dispersion parameters of 10,000 and 5 respectively. To emulate the characteristics of the real data, random Poisson noise was added to each generated count, where the Poisson mean was the generated count value.

Given the nested structure of CpG sites on genes, simulation of the methylation data depends on the cluster membership of CpG sites' associated genes, as quantified by  $\pi$ . To assess the robustness of the model across different dependency scenarios, methylation data were generated under three different settings of  $\pi$ , as detailed in Table 1. In case 1, probabilities are similar to those resulting from fitting the joint mixture model to the breast cancer dataset. Case 2 considers a high level of dependency, represented by a diagonal-heavy  $\pi$  matrix. In case 3, no dependence between the two data types is assumed.

**Table 1:** Three settings of  $\pi$  considered in the simulation study.

(a) Case 1			(b) Case 2			(c) Case 3					
	E-	E0	E+	M+	E-	E0	E+	M+	E-	E0	E+
M+	0.4	0.05	0.1	M+	0.8	0.1	0.1	M+	0.2	0.6	0.2
M0	0.5	0.9	0.5	M0	0.1	0.8	0.1	M0	0.2	0.6	0.2
M-	0.1	0.05	0.4	M-	0.1	0.1	0.8	M-	0.2	0.6	0.2

Values represent the probabilities of a CpG site belonging to cluster M+, M0 or M-, conditional of their associated gene belonging to cluster E-, E0 or E+.

Given the cluster membership of a CpG site's associated gene, if a CpG site is assumed to be hypermethylated (i.e., belongs to cluster M+) in condition B compared to A, *beta* values were generated from a Beta(3, 20) distribution for condition A and from a Beta(20, 3) for condition B. Similarly, if the CpG site is hypomethylated (i.e. belongs to cluster M-) in condition B

compared to A, *beta* values were generated from Beta(20, 3) for condition A and Beta(3, 20) for condition B. Non-differentially methylated CpG sites arise when they are either hypermethylated in both conditions, hypomethylated in both conditions or hemimethylated in both conditions. Therefore, if a CpG site is assumed to be non-differential (i.e., belongs to cluster M0), the *beta* values for both conditions A and B were generated from either Beta(3, 20) (both hypomethylated), Beta(20, 3) (both hypermethylated) or Beta(4, 3) (both hemimethylated) distributions. Zero centred Gaussian noise with standard deviation 0.05 was added to each of the generated *beta* values to emulate the variation in the breast cancer data.

For analysis, the simulated RNA-Seq data were normalized, log-transformed to log-CPM values and log-fold changes from condition A to B calculated. Similarly, differences in logit-transformed *beta* values between conditions A and B were calculated for all CpG sites. The **idiffomix** joint mixture model was then fitted to these simulated log-fold change RNA-Seq values and to the differences in the simulated methylation *M*-values.

#### 4.1.2 Simulation study results

Results under the **idiffomix** approach were compared to those obtained when the two simulated data types were analysed independently using the **mclust** and **limma** approaches [Ritchie et al., 2015, Scrucca et al., 2016]. For **mclust**, three clusters and the EEI, EVI and EII models were considered and cluster assignment was via the MAP rule. For **limma**, a threshold of 0.05 for Benjamini-Hochberg adjusted *p*-values was used to identify significant differential expression and methylation. Performance of the three methods was evaluated based on false discovery rate (FDR), sensitivity, specificity and adjusted Rand index (ARI) [Hubert and Arabie, 1985].

Table 2 details the performance metrics across the 100 simulated datasets generated given  $\pi$  from case 1 in Table 1. In terms of identifying DEGs, **idiffomix** outperforms **mclust** and **limma** with lower mean FDR and higher mean sensitivity, specificity and ARI values. Similarly when identifying DMCs, **idiffomix** performs well overall, with only **limma** having a slightly higher mean sensitivity. Interestingly, **idiffomix** has much stronger performance than **mclust** and **limma** when identifying DEGs, while all 3 methods show similar performance when identifying DMCs; when information from CpG sites associated with the genes is appropriately accounted for through the joint mixture model's dependency structure, DEG identification

improves. Table 3 shows similar performance patterns for data generated using  $\pi$  under case 2 from Table 1 where the level of dependency is high. Results under case 3 show similar clustering solutions between the three methods suggesting the joint mixture model performs as two independent mixture models; results are provided in Appendix A.2.

**Table 2:** Mean performance metrics for 100 simulated datasets given  $\pi$  under case 1 from Table 1.

(a) DEG identification performance				
	FDR	Sensitivity	Specificity	ARI
<b>idiffomix</b>	<b>0.014</b> (0.011)	<b>0.976</b> (0.015)	<b>0.997</b> (0.003)	<b>0.966</b> (0.017)
<b>mclust</b>	0.102 (0.049)	0.873 (0.046)	0.975 (0.015)	0.800 (0.041)
<b>limma</b>	0.038 (0.021)	0.764 (0.064)	0.993 (0.005)	0.760 (0.059)
(b) DMC identification performance				
	FDR	Sensitivity	Specificity	ARI
<b>idiffomix</b>	<b>0.016</b> (0.005)	0.999 (0.001)	<b>0.997</b> (0.001)	<b>0.986</b> (0.004)
<b>mclust</b>	0.019 (0.006)	0.999 (0.001)	0.996 (0.001)	0.983 (0.005)
<b>limma</b>	0.058 (0.006)	<b>1.000</b> (<0.001)	0.987 (0.002)	0.948 (0.006)

\*Standard deviations in parentheses and the top performing method for each metric is highlighted in boldface.

**Table 3:** Mean performance metrics for 100 simulated datasets given  $\pi$  under case 2 from Table 1.

(a) DEG identification performance				
	FDR	Sensitivity	Specificity	ARI
<b>idiffomix</b>	<b>0.003</b> (0.006)	<b>0.997</b> (0.005)	<b>0.999</b> (0.001)	<b>0.995</b> (0.007)
<b>mclust</b>	0.102 (0.049)	0.873 (0.046)	0.975 (0.015)	0.800 (0.041)
<b>limma</b>	0.038 (0.021)	0.764 (0.064)	0.993 (0.005)	0.760 (0.059)
(b) DMC identification performance				
	FDR	Sensitivity	Specificity	ARI
<b>idiffomix</b>	<b>0.009</b> (0.003)	1.000 (< 0.001)	<b>0.995</b> (0.002)	<b>0.987</b> (0.004)
<b>mclust</b>	0.011 (0.004)	1.000 (< 0.001)	0.994 (0.002)	0.984 (0.005)
<b>limma</b>	0.050 (0.005)	1.000 (< 0.001)	0.973 (0.003)	0.930 (0.007)

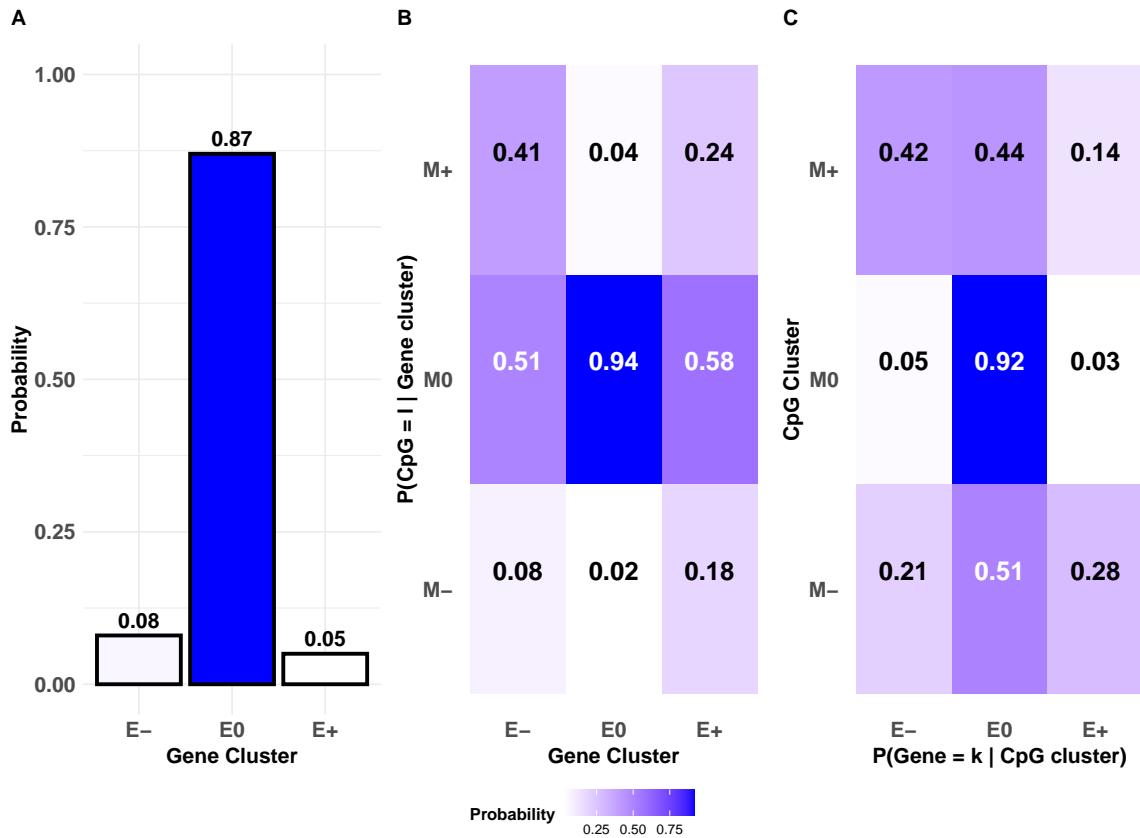
\*Standard deviations in parentheses and the top performing method for each metric is highlighted in boldface.

All computations were performed using R (version 4.3.3) [R Core Team, 2024] on a Windows 11 operating system with an Intel Core i7 CPU (2.70GHz) and 16GB of RAM. In terms of computational cost, for example, fitting **idiffomix** to one simulated dataset of RNA-Seq and methylation data took 5.34 minutes. To explore the impact of a larger value of  $N$  on computational cost, further simulation studies were performed where  $N$  varied from  $N = 4$  to  $N = 200$ . The computational cost increased linearly with  $N$ , consistent with the form of the model's

likelihood function, suggesting a time complexity of  $O(N)$ . Further details on computational cost are provided in Appendix A.3.

## 4.2 Application to breast cancer gene expression and methylation data

The `idiffomix` model is fit to the publicly available breast cancer RNA-Seq and methylation array data outlined in Section 2.2. The joint mixture model was applied to data from each chromosome individually; for clarity, results for a single chromosome (chromosome 7) are provided here with results for all other chromosomes provided in Appendix A.4. Analysing the RNA-Seq and methylation data independently indicated large numbers of DMCs and DEGs on chromosome 7 and hence, the results for chromosome 7 are given here.



**Figure 2:** `Idiffomix` applied to chromosome 7 of TCGA breast cancer data.  
 (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .

Figure 2 panel A displays the probability of a gene in chromosome 7 belonging to each of

the E-, E0 or E+ clusters, with a large majority deemed to be non-differentially expressed and allocated to the E0 cluster. Panel B in Figure 2 details the estimated matrix  $\hat{\pi}$  of conditional probabilities of a CpG site's cluster membership, given its gene's cluster. Interestingly, a CpG site has highest probability of being in cluster M0 (i.e., non-differentially methylated) regardless of whether its gene is in either E-, E0 or E+. However, if a gene is differentially expressed i.e., in clusters E- or E+, an associated CpG site, if also differentially methylated, has highest probability of being hypermethylated, i.e. in M+ (with probabilities 0.41 and 0.24, respectively).

To illustrate the dependence between gene and CpG cluster memberships, panel C in Figure 2 details the conditional probabilities of a gene belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ , computed using Bayes' theorem, given  $\hat{\tau}$  and  $\hat{\pi}$ . That is,

$$\mathbb{P}\{\text{Gene in cluster } k | \text{CpG in cluster } l\} = \frac{\hat{\tau}_k \hat{\pi}_{l|k}}{\sum_{k'=1}^K \hat{\tau}_{k'} \hat{\pi}_{l|k'}}.$$

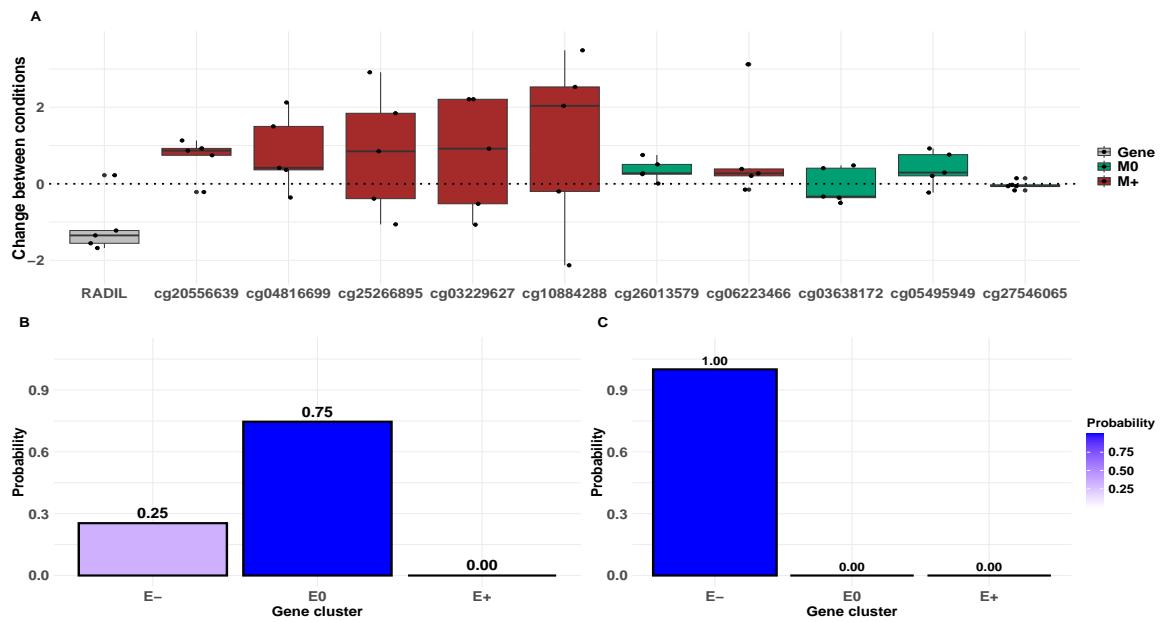
Here, a gene has highest probability of being in cluster E0 (i.e., non-differentially expressed) regardless of whether its CpG site is in M-, M0 or M+. Thereafter, if a CpG site is in M-, its associated gene, if also differentially expressed, has highest probability of being in E+ (with probability 0.28), while if a CpG site is in M+, its associated gene, if also differentially expressed, has highest probability of being E- (with probability 0.43).

For comparison purposes, as the log-fold changes and differences in  $M$ -values are approximately Gaussian distributed, `mclust` is fitted to each data type on chromosome 7 independently. Out of the 772 genes located on chromosome 7, `idiffomix` identified 30 genes as DEGs and 22 genes as non-DEGs that were not detected as such by `mclust`. For the 4,790 CpG sites associated with promoter regions of genes on chromosome 7, `idiffomix` identified 36 CpG sites as DMCs and 41 CpG sites as non-DMCs that were not detected as such by `mclust`.

#### 4.2.1 Genes of interest

Genes for which the inferred differential expression status differed between the independent and integrated analyses are of particular interest. When the two data types are modelled jointly, 6 of the 10 CpG sites linked to the *RADIL* gene were deemed to be DMCs in the M+ cluster and 4 were inferred to belong to the M0 cluster (Figure 3, panel A). Interestingly, when the

RNA-Seq data were modelled independently the *RADIL* gene, with a median log-fold change below  $-1$ , was identified to be non-differentially expressed, as illustrated in panel B of Figure 3. However, modelling the RNA-Seq data and the methylation data together under `idiffomix` suggested *RADIL* was a DEG, belonging to the E- cluster, as illustrated in panel C of Figure 3.

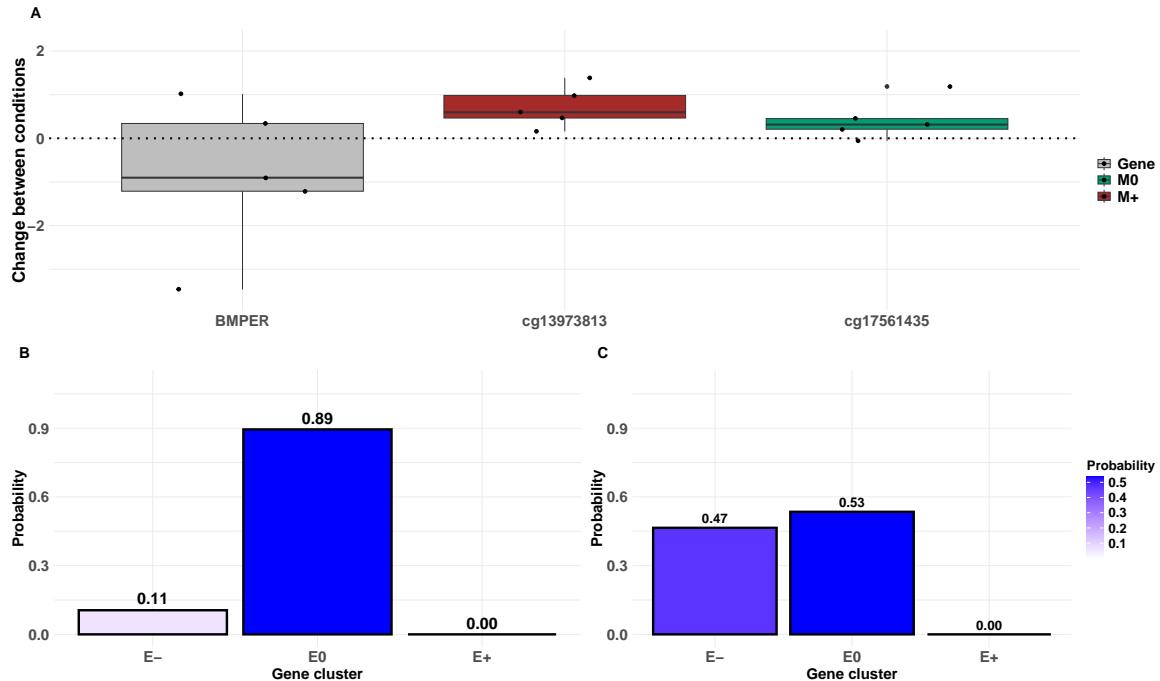


**Figure 3: Comparison of results for independent and integrated analyses for *RADIL* on chromosome 7**

(A) Log-fold change in gene expression levels (grey) and differences in *M*-values between tumour and normal samples, coloured by inferred idiffomix cluster (hypermethylated CpG sites, M+ in brown; non-differentially methylated CpG sites, M0 in green); (B) posterior probability of *RADIL* belonging to the E-, E0 and E+ clusters under `mclust`, (C) posterior probability of *RADIL* belonging to the E-, E0 and E+ clusters when jointly modelled with methylation data under `idiffomix`. Larger posterior probabilities are represented by increasingly dark shades of blue.

Another aspect of interest are the gene's clustering uncertainties. Figure 4 shows, in panel A, the log-fold change and difference in *M*-values for *BMPER* and its associated CpG sites while panels B and C show the posterior probabilities of cluster membership for *BMPER* under `mclust` and `idiffomix` respectively. The resulting clustering uncertainty for *BMPER* is higher under `idiffomix` than under `mclust` as, while the gene expression levels (panel A) suggest the gene is likely to be non-differential, of the two CpG sites linked to the gene, one is inferred to be hypermethylated and the other as non-differential. Consequently, the posterior probability of *BMPER* belonging to the non-differential E0 cluster under `idiffomix` is smaller than that under `mclust`, while the probability of *BMPER* being a DEG in the E- cluster is

larger. Thus, while the MAP cluster for *BMPER* is E0 under both independent and integrated approaches, the clustering uncertainty is intuitively higher under `idiffomix`, given the observed data. Clustering uncertainties for other genes of interest are provided in Appendix A.5.

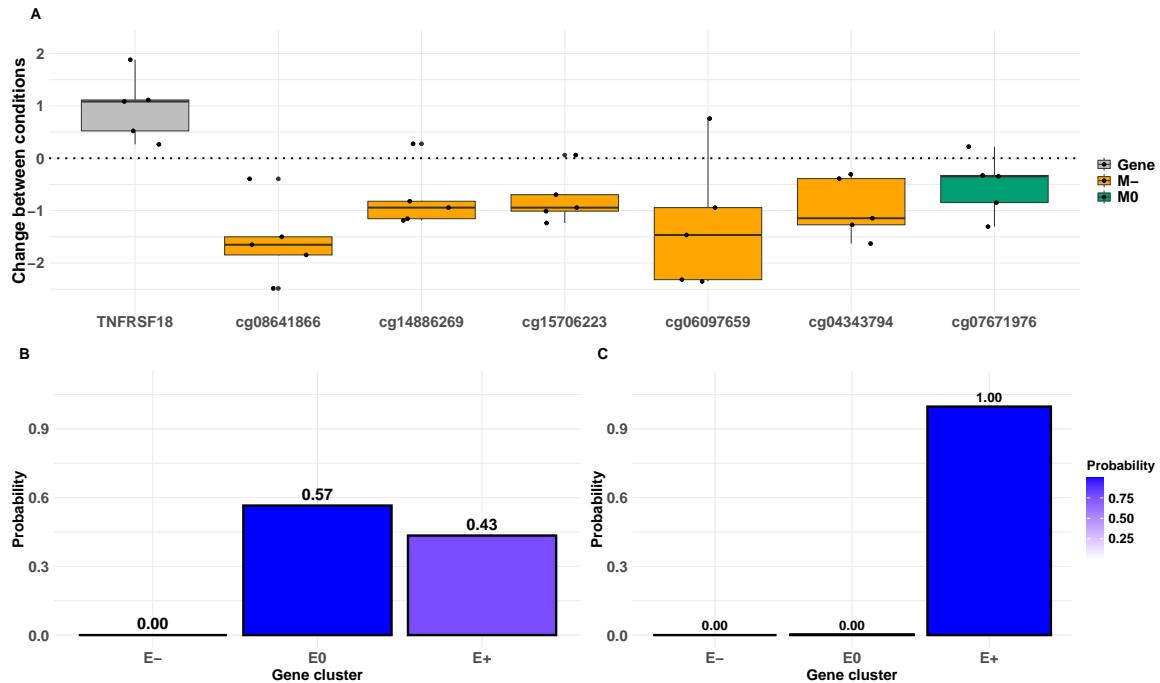


**Figure 4: Comparison of results for independent and integrated analyses for *BMPER* on chromosome 7.**

(A) Log-fold change in gene expression levels (grey) and differences in *M*-values between tumour and normal samples, coloured by inferred `idiffomix` cluster (hypermethylated CpG sites, M+ in brown; non-differentially methylated CpG sites, M0 in green); (B) posterior probability of *BMPER* belonging to the E-, E0 and E+ clusters under `mclust` (C) posterior probability of *BMPER* belonging to the E-, E0 and E+ clusters when jointly modelled with methylation data under `idiffomix`. Larger posterior probabilities are represented by increasingly dark shades of blue.

*TNFRSF18*, *GPX7* and *RAD51* are of particular interest as they are known to play key roles in the development and progression of breast cancer [Wiegmans et al., 2014, Rusolo et al., 2017, Xiong et al., 2019]. Figure 5 (panel A) shows that the median log-fold change for *TNFRSF18* is  $>1$ , and that of the 6 CpG sites linked to the gene, 5 are DMCs in M- and 1 is non-differential in M0 when the two data types are modelled jointly. When the expression data of *TNFRSF18* are modelled independently under `mclust` (Figure 5 panel B), *TNFRSF18* has posterior probabilities of 0.57 and 0.43 of being in clusters E0 and E+ respectively. However, when the expression and methylation data are modelled jointly using `idiffomix` (Figure 5 panel C), the posterior probability of *TNFRSF18* being upregulated in cluster E+ is 1. These results show that when the RNA-Seq and methylation data are jointly modelled, different insights

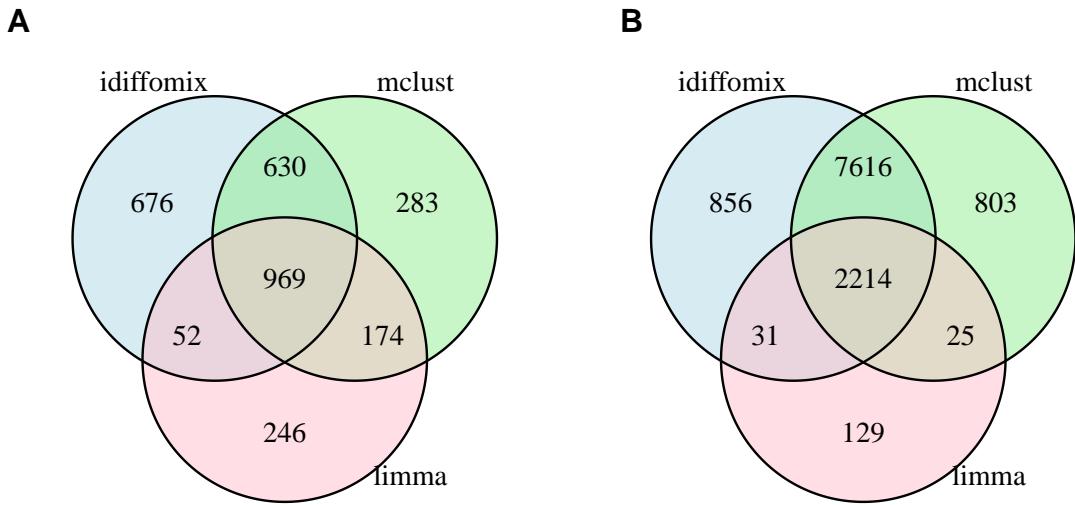
are revealed than when the two data types are modelled independently. Posterior probability cluster membership distributions, and their corresponding insights, for *GPX7* and *RAD51* are available in Appendix A.6.



**Figure 5: Comparison of results for independent and integrated analyses *TNFRSF18* on chromosome 1**

(A) Log-fold change in gene expression levels (grey) and differences in *M*-values between tumour and normal samples, coloured by inferred idiffomix cluster (hypomethylated CpG sites, M- in yellow; non-differentially methylated CpG sites, M0 are green); (B) posterior probability of *TNFRSF18* belonging to the E-, E0 and E+ clusters under **mclust** (C) posterior probability of *TNFRSF18* belonging to the E-, E0 and E+ clusters when jointly modelled with methylation data under **idiffomix**. Larger posterior probabilities are represented by increasingly dark shades of blue.

The two data types were also modelled independently using **limma** for comparison purposes. In contrast to **idiffomix** which identified 2,327 DEGs and 10,717 DMCs, **mclust** identified 2,056 DEGs and 10,658 DMCs while **limma** identified 1,441 DEGs and 2,399 DMCs. Figure 6 shows Venn diagrams illustrating the intersection of DEGs and DMCs, in panels A and B respectively, identified under **idiffomix**, **mclust**, and **limma**, highlighting their common and method-specific findings. Out of the 5 genes of interest discussed here, **idiffomix** identified 3 to be DEGs, while **mclust** identified *RAD51* as a DEG, and under **limma** only *TNFRSF18* was identified to be a DEG.



**Figure 6: Comparison of results for DEGs and DMCs between independent and integrated analyses.**

Venn diagrams showing the intersection of (A) DEGs and (B) DMCs identified under `idiffomix`, `mclust`, and `limma`.

#### 4.2.2 Gene enrichment analysis

A gene enrichment analysis was performed for the DEGs and DMCs identified under the joint `idiffomix` model, and under the independent `mclust` and `limma` approaches using Gene Ontology (GO) and Kyoto Encyclopedia of Genes and Genomes (KEGG) terms [Kanehisa et al., 2017, Aleksander et al., 2023]. The GO enrichment analysis of `idiffomix` DEGs indicated they were associated with 1,299 significant biological pathways, while DEGs under `mclust` and `limma` applied to the RNA-Seq data identified 1,442 and 647 significant biological pathways respectively. Similarly, the GO analysis of the `idiffomix` DMCs unveiled 457 significant biological pathways whereas `mclust` and `limma` identified 414 and 83 respectively. Several biological processes identified under `idiffomix`, but not under the other two methods, such as *MAPK cascade*, *ERK1 and ERK2 cascade*, *extracellular matrix organization* and *BMP signalling pathway* play an essential role in breast cancer development and prognosis [Whyte et al., 2009, Zabkiewicz et al., 2017, Lepucki et al., 2022]. The KEGG analysis of `idiffomix` DEGs revealed associations with 27 significant pathways (adjusted  $p$ -value  $< 0.05$ ), while `mclust` and `limma` DEGs identified 32 and 13 significant pathways, respectively. Similarly, the KEGG analysis mapped `idiffomix` DMCs to genes associated with 17 significant metabolic and signalling

pathways, while DMCs under `mclust` and `limma` were mapped to genes associated with 18 and 4 significant pathways respectively. Several pathways associated with `idiffomix` DEGs, but not identified under the other two methods, such as the *cAMP signaling pathway*, *ECM-receptor interaction*, *Hippo signaling pathway* and *Cell adhesion molecules* have also been shown to play key roles in breast cancer development [Kyriazoglou et al., 2021, Ruan et al., 2022, Zhang et al., 2024]. The top 10 biological processes and pathways associated with the `idiffomix` DEGs and DMCs are presented in Appendix A.7.

## 5 Discussion

While there are inherent, biological dependencies between gene sequencing and methylation array data, analyses that aim to identify differential gene expression and methylation typically do so via independent analyses of both data types. Here, a joint mixture model approach `idiffomix` is proposed that integrates both data types at the modelling stage by directly modelling the nested structure of CpG sites in gene promoter regions. The method does not assume a fixed pattern of associations between expression and promoter methylation (e.g., global anti-correlation for all genes/CpGs); rather, it learns global patterns from the data. This allows for a genome-wide, cross-omics analysis that simultaneously identifies DMCs and DEGs. Simulation studies and application to data from a breast cancer study demonstrated the benefit of integrating both data types at the modelling stage, providing a joint analysis.

While log-fold changes of transformed RNA-Seq and differences in *beta* valued methylation data were modelled here, allowing for a joint Gaussian mixture model, such transformations can make results less biologically interpretable. Relaxing the Gaussian assumption and employing distributions that model the inherent data distributions directly could improve model performance and interpretability.

Information on the complete set of proteins that is expressed by a cell or tissue under specific conditions is captured in proteomics data, which can be obtained using high-throughput technologies like mass spectroscopy. Integrating proteomics with other omics data, such as gene expression data, has been shown to capture the complexity of biological systems and provide deeper insights into gene regulation and cellular functions [Kumar et al., 2016]. The joint mixture model approach could be enhanced to jointly model proteomics data along with gene

expression and DNA methylation data.

It is well established that methylation patterns and gene expression regulation are not only dependent on one another but also on several other factors including environmental stress, food habits, etc [Muralidharan et al., 2022]. The **idiffomix** method could be further developed to facilitate modelling of the influence on the two data types of such environmental factors. For example, employing a mixtures of experts approach [Gormley and Frühwirth-Schnatter, 2019], where cluster membership probabilities are modelled as dependent on concomitant factors, would allow for such covariates to be incorporated within **idiffomix**'s joint mixture model framework.

## 6 Conclusions

The development and application of the **idiffomix** joint mixture model for integrated differential analysis of multi-omics data illustrated the advantages of integrating RNA-Seq and methylation data when identifying DEGs and DMCs, over analysing such data separately. The **idiffomix** model provides insight to the complex associations between gene expression and methylation changes, leading to a deeper understanding of the transcriptional and epigenetic landscape. Future work on this model will aim to refine and expand its capabilities, building on our current findings.

## 7 Acknowledgements

This publication has emanated from research conducted with the financial support of Research Ireland under grant number 18/CRT/6049. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

## References

- Hany Alashwal, Remi Dosunmu, and Nasser H Zawia. Integration of genome-wide expression and methylation data: relevance to aging and Alzheimer’s disease. *Neurotoxicology*, 33(6):1450–1453, 2012.
- Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris, et al. The gene ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031, 2023.
- Mohammad Reza Alivand, Sajad Najafi, Sajjad Esmaeili, Dara Rahmanpour, Hossein Zhaleh, and Yazdan Rahmati. Integrative analysis of DNA methylation and gene expression profiles to identify biomarkers of glioblastoma. *Cancer Genetics*, 258:135–150, 2021.
- Shelley L Berger, Tony Kouzarides, Ramin Shiekhattar, and Ali Shilatifard. An operational definition of epigenetics. *Genes & Development*, 23:781–783, 2009.
- Marina Bibikova and Jian-Bing Fan. Genome-wide dna methylation profiling. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 2(2):210–223, 2010.
- Adrian Bird. DNA methylation patterns and epigenetic memory. *Genes & Development*, 16:6–21, 2002.
- Faicel Chamroukhi and Bao Tuyen Huynh. Regularized maximum-likelihood estimation of mixture-of-experts for regression and clustering. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.
- Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.
- P Du, X Zhang, C-C Huang, N Jafari, W A Kibbe, L Hou, and S M Lin. Comparison of beta-value and M-value methods for quantifying methylation levels by microarray analysis. *BMC Bioinformatics*, 11:587, 2010.
- IC Gormley and S Frühwirth-Schnatter. Mixtures of experts models. In *Handbook of mixture analysis*, pages 271–307. CRC Press, 2019.

- L Hubert and P Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.
- Yonatan Itai, Nimrod Rappoport, and Ron Shamir. Integration of gene expression and DNA methylation data across different experiments. *Nucleic Acids Research*, 51(15):7762–7776, 2023.
- Jaesik Jeong, Lang Li, Yunlong Liu, Kenneth P Nephew, Tim Hui-Ming Huang, and Changyu Shen. An empirical Bayes model for gene expression and methylation profiles in antiestrogen resistant breast cancer. *BMC Medical Genomics*, 3:1–9, 2010.
- Peter A Jones and Stephen B Baylin. The fundamental role of epigenetic events in cancer. *Nature Reviews Genetics*, 3(6):415–428, 2002.
- Peter A Jones and Daiya Takai. The role of DNA methylation in mammalian epigenetics. *Science*, 293(5532):1068–1070, 2001.
- Minoru Kanehisa, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima. Kegg: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research*, 45(D1):D353–D361, 2017.
- D C Koestler, B C Christensen, C J Marsit, K T Kelsey, and E A Houseman. Recursively partitioned mixture model clustering of dna methylation data using biologically informed correlation structures. *Statistical Applications in Genetics and Molecular Biology*, 12(2):225–240, 2013. doi:10.1515/sagmb-2012-0068.
- Matthias Kormaksson, James G Booth, Maria E Figueroa, and Ari Melnick. Integrative model-based clustering of microarray methylation and expression data. *The Annals of Applied Statistics*, 6(3):1327–1347, 2012.
- Dhirendra Kumar, Gourja Bansal, Ankita Narang, Trayambak Basak, Tahseen Abbas, and Debasis Dash. Integrating transcriptome and proteome profiling: strategies and applications. *Proteomics*, 16(19):2533–2544, 2016.
- Anastasios Kyriazoglou, Michalis Liontos, Roubini Zakopoulou, Maria Kaparelou, Anna Tsiora, Alkistis Maria Papatheodoridi, Rebecca Georgakopoulou, and Flora Zagouri. The role of

the Hippo pathway in breast cancer carcinogenesis, prognosis, and treatment: a systematic review. *Breast Care*, 16(1):6–15, 2021.

Charity W Law, Yunshun Chen, Wei Shi, and Gordon K Smyth. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology*, 15:1–17, 2014.

Arkadiusz Lepucki, Kinga Orlińska, Aleksandra Mielczarek-Palacz, Jacek Kabut, Paweł Olczyk, and Katarzyna Komosińska-Vassev. The role of extracellular matrix proteins in breast cancer. *Journal of Clinical Medicine*, 11(5):1250, 2022.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with *deseq2*. *Genome Biology*, 15:1–21, 2014.

Zhanyu Ma and Andrew E Teschendorff. A variational bayes beta mixture model for feature selection in dna methylation studies. *Journal of Bioinformatics and Computational Biology*, 11(4):1350005, 2013.

Tobias Madsen, Michał Świtnicki, Malene Juul, and Jakob Skou Pedersen. EBADIMEX: an empirical Bayes approach to detect joint differential expression and methylation and to classify samples. *Statistical Applications in Genetics and Molecular Biology*, 18(6):20180050, 2019.

Koyel Majumdar, Romina Silva, Antoinette S. Perry, Ronald W. Watson, Andrea Rau, Florence Jaffrézic, et al. A novel family of beta mixture models for the differential analysis of dna methylation data: An application to prostate cancer. *PLoS ONE*, 19(12):e0314014, 2024.

Lisa D Moore, Thuc Le, and Guoping Fan. Dna methylation and its basic function. *Neuropharmacology*, 38:23–38, 2013.

Sachin Muralidharan, Sarah Ali, Lilin Yang, Joshua Badshah, Syeda Farah Zahir, Rubbiya A Ali, Janin Chandra, Ian H Frazer, Ranjeny Thomas, and Ahmed M Mehdi. Environmental pathways affecting gene expression (E. PAGE) as an R package to predict gene–environment associations. *Scientific Reports*, 12(1):18710, 2022.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2024. URL <https://www.R-project.org/>.

Matthew E Ritchie, Belinda Phipson, DI Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic Acids Research*, 43(7):e47, 2015.

Mark D. Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of rna-seq data. *Genome Biology*, 11:R25, 2010. URL <https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-3-r25>.

Yongsheng Ruan, Libai Chen, Danfeng Xie, Tingting Luo, Yiqi Xu, Tao Ye, Xiaona Chen, Xiaoqin Feng, and Xuedong Wu. Mechanisms of cell adhesion molecules in endocrine-related cancers: a concise outlook. *Frontiers in Endocrinology*, 13:865436, 2022.

Fabiola Rusolo, Francesca Capone, Raffaella Pasquale, Antonella Angiolillo, Giovanni Colonna, Giuseppe Castello, Maria Costantini, and Susan Costantini. Comparison of the seleno-transcriptome expression between human non-cancerous mammary epithelial cells and two human breast cancer cell lines. *Oncology Letters*, 13(4):2411–2417, 2017.

Sogand Sajedi, Ghazal Ebrahimi, Raheleh Roudi, Isha Mehta, Amirreza Heshmat, Hanie Samimi, Shiva Kazempour, Aamir Zainulabdeen, Thomas Roderick Docking, Sukeshi Patel Arora, et al. Integrating DNA methylation and gene expression data in a single gene network using the iNETgrate package. *Scientific Reports*, 13(1):21721, 2023.

Michael Salter-Townshend and Thomas Brendan Murphy. Variational Bayesian inference for the latent position cluster model for network data. *Computational Statistics & Data Analysis*, 57(1):661–671, 2013.

L Scrucca, M Fop, T B Murphy, and A E Raftery. mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):289–317, 2016. PMCID: PMC5096736.

Stephen A Semick, Rahul A Bharadwaj, Leonardo Collado-Torres, Ran Tao, Joo Heon Shin, Amy Deep-Soboslay, James R Weiss, Daniel R Weinberger, Thomas M Hyde, Joel E Kleinman, et al. Integrated dna methylation and gene expression profiling across multiple brain regions implicate novel genes in alzheimer’s disease. *Acta Neuropathologica*, 137:557–569, 2019.

Mai Shi, Stephen Kwok-Wing Tsui, Hao Wu, and Yingying Wei. Pan-cancer analysis of differential dna methylation patterns. *BMC Medical Genomics*, 13 (Suppl 10), 154, 2020.

K D Siegmund, P W Laird, and I A Laird-Offringa. A comparison of cluster analysis methods using dna methylation data. *Bioinformatics*, 20(12):1896–1904, 2004. doi:10.1093/bioinformatics/bth176.

John CG Spainhour, Hong Seo Lim, Soojin V Yi, and Peng Qiu. Correlation patterns between DNA methylation and gene expression in the Cancer Genome Atlas. *Cancer Informatics*, 18: 1176935119828776, 2019.

Frank J Steemers and Kevin L Gunderson. Illumina, Inc. *Pharmacogenomics*, 6(7):777–782, 2005.

Miho M Suzuki and Adrian Bird. DNA methylation landscapes: provocative insights from epigenomics. *Nature Reviews Genetics*, 9(6):465–476, 2008.

Aurélie Van Tongelen, Axelle Loriot, and Charles De Smet. Oncogenic roles of DNA hypomethylation through the activation of cancer-germline genes. *Cancer Letters*, 396:130–137, 2017.

Juan Wang, Yong Duan, Qing-He Meng, Rong Gong, Chong Guo, Ying Zhao, and Yanliang Zhang. Integrated analysis of DNA methylation profiling and gene expression profiling identifies novel markers in lung cancer in Xuanwei, China. *PLOS One*, 13(10):e0203155, 2018.

Yue Wang, Jennifer M Franks, Michael L Whitfield, and Chao Cheng. Biomethyl: an R package for biological interpretation of DNA methylation data. *Bioinformatics*, 35(19):3635–3641, 2019.

Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009.

Jacqueline Whyte, Orla Bergin, Alessandro Bianchi, Sara McNally, and Finian Martin. Key signalling nodes in mammary gland development and cancer. mitogen-activated protein kinase signalling in experimental models of breast cancer progression and in mammary gland development. *Breast Cancer Research*, 11:1–14, 2009.

Adrian P Wiegmans, Fares Al-Ejeh, Nicole Chee, Pei-Yi Yap, Julia J Gorski, Leonard Da Silva, Emma Bolderson, Georgia Chenevix-Trench, Robin Anderson, Peter T Simpson, et al. Rad51 supports triple negative breast cancer metastasis. *Oncotarget*, 5(10):3261, 2014.

Dingming Wu, Jin Gu, and Michael Q Zhang. Fastdma: An infinium humanmethylation450 beadchip analyzer. *PLOS One*, 8(9):e74275, 2013.

Linglin Xie, Brent Weichel, Joyce Ellen Ohm, and Ke Zhang. An integrative analysis of DNA methylation and RNA-seq data for human heart, kidney and liver. *BMC Systems Biology*, 5:1–11, 2011.

Donghai Xiong, Yian Wang, and Ming You. Tumor intrinsic immunity related proteins may be novel tumor suppressors in some types of cancer. *Scientific Reports*, 9(1):10918, 2019.

Wanxue Xu, Mengyao Xu, Longlong Wang, Wei Zhou, Rong Xiang, Yi Shi, Yunshan Zhang, and Yongjun Piao. Integrative analysis of DNA methylation and gene expression identified cervical cancer-specific diagnostic biomarkers. *Signal Transduction and Targeted Therapy*, 4(1):55, 2019.

Catherine Zabkiewicz, Jeyna Resaul, Rachel Hargest, Wen Guo Jiang, and Lin Ye. Bone morphogenetic proteins, breast cancer, and bone metastases: striking the right balance. *Endocrine-related Cancer*, 24(10):R349–R366, 2017.

Hongying Zhang, Yongliang Liu, Jieya Liu, Jinzhu Chen, Jiao Wang, Hui Hua, and Yangfu Jiang. cAMP-PKA/EPAC signaling and cancer: the interplay in tumor microenvironment. *Journal of Hematology & Oncology*, 17(1):5, 2024.

Lin Zhang, Jia Meng, Hui Liu, and Yufei Huang. A nonparametric Bayesian approach for clustering bisulfate-based DNA methylation profiles. *BMC Genomics*, 13(6):S20, 2012. doi:10.1186/1471-2164-13-S6-S20.

# A Appendix to ‘Integrated differential analysis of multi-omics data using a joint mixture model: `idiffomix`’ by Majumdar et al.

## A.1 EM algorithm for joint mixture model

The complete data log-likelihood function for the joint mixture model is,

$$\begin{aligned} \ell_C(\boldsymbol{\tau}, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\phi} | \mathbf{U}, \mathbf{V}, \mathbf{X}, \mathbf{Y}) = & \sum_{g=1}^G \sum_{k=1}^K \sum_{n=1}^N u_{gk} \log p(x_{gn} | \boldsymbol{\theta}_k) \\ & + \sum_{g=1}^G \sum_{c=1}^{C_g} \sum_{l=1}^L \sum_{n=1}^N v_{gcl} \log p(y_{gcn} | \boldsymbol{\phi}_l) \\ & + \sum_{g=1}^G \sum_{k=1}^K u_{gk} \log \tau_k + \sum_{g=1}^G \sum_{k=1}^K \sum_{c=1}^{C_g} \sum_{l=1}^L u_{gk} v_{gcl} \log \pi_{l|k} \end{aligned} \quad (3)$$

**M-step of EM algorithm for the joint mixture model** The complete data log-likelihood function in (3) is maximised w.r.t. the parameters  $\tau_k$ ,  $\pi_{l|k}$ ,  $\boldsymbol{\theta}_k = (\mu_k, \sigma^2)$ , and  $\boldsymbol{\phi}_l = (\lambda_l, \rho^2)$  to calculate  $\hat{\tau}_k$ ,  $\hat{\pi}_{l|k}$ ,  $\hat{\sigma}^2$  and  $\hat{\rho}^2$ . For parsimony, the standard deviations are constrained to be equal across clusters and are calculated as the weighted average of the individual standard deviations, such that  $\hat{\sigma}^2 = \sum_{k=1}^K \tau_k \sigma_k^2$  and  $\hat{\rho}^2 = \sum_{l=1}^L (\sum_{k=1}^K \pi_{l|k} \tau_k) \sigma_k^2$ . The other maximised parameter estimates are then:

$$\begin{aligned} \hat{\tau}_k &= \frac{\sum_{g=1}^G u_{gk}}{G}, & \hat{\pi}_{l|k} &= \frac{\sum_{g=1}^G \sum_{c=1}^{C_g} u_{gk} v_{gcl}}{\sum_{g=1}^G u_{gk} C_g}, \\ \hat{\mu}_k &= \frac{\sum_{g=1}^G \sum_{n=1}^N u_{gk} x_{gn}}{N \sum_{g=1}^G u_{gk}}, & \hat{\sigma}_k^2 &= \frac{\sum_{g=1}^G \sum_{n=1}^N u_{gk} (x_{gn} - \hat{\mu}_k)^2}{N \sum_{g=1}^G u_{gk}}, \\ \hat{\lambda}_l &= \frac{\sum_{g=1}^G \sum_{c=1}^{C_g} \sum_{n=1}^N v_{gcl} y_{gcn}}{N \sum_{g=1}^G \sum_{c=1}^{C_g} v_{gcl}}, & \hat{\rho}_l^2 &= \frac{\sum_{g=1}^G \sum_{c=1}^{C_g} \sum_{n=1}^N v_{gcl} (y_{gcn} - \hat{\lambda}_l)^2}{N \sum_{g=1}^G \sum_{c=1}^{C_g} v_{gcl}}. \end{aligned}$$

**E-step of EM algorithm for the joint mixture model** In the E-step, the conditional expected values of  $u_{gk}$ ,  $v_{gcl}$  and  $u_{gk} v_{gcl}$  are calculated given the observed data  $\mathbf{X}$ ,  $\mathbf{Y}$  and the

estimated model parameters  $\tau_k^{(t)}$ ,  $\pi_{l|k}^{(t)}$ ,  $\boldsymbol{\theta}_k^{(t)}$ , and  $\boldsymbol{\phi}_l^{(t)}$ , at iteration  $t$  i.e., we have to compute  $\mathbb{E}(u_{gk}|\mathbf{x}_g, \tau_k, \boldsymbol{\pi}_{l|k}, \boldsymbol{\theta}_k)$ ,  $\mathbb{E}(v_{gcl}|\mathbf{y}_{gc}, \boldsymbol{\pi}_{l|k}, \boldsymbol{\phi}_l)$ , and  $\mathbb{E}(u_{gk}v_{gcl}|\mathbf{x}_g, \mathbf{y}_{gc}, \tau_k, \boldsymbol{\pi}_{l|k}, \boldsymbol{\theta}_k, \boldsymbol{\phi}_l)$ . However, these expected values are intractable.

Here, instead, the following expected values are considered and can be derived from the complete data log-likelihood function (3):

$$\mathbb{E}(u_{gk}|\mathbf{x}_g, v_{g11}, v_{g12}, \dots, v_{gC_gL}, \tau_k, \boldsymbol{\theta}_k, \pi_{1|k}, \dots, \pi_{L|k}) = \frac{\tau_k \prod_{n=1}^N p(x_{gn}|\boldsymbol{\theta}_k) \prod_{c=1}^{C_g} \prod_{l=1}^L (\pi_{l|k})^{v_{gcl}}}{\sum_{k=1}^K \left\{ \tau_k \prod_{n=1}^N p(x_{gn}|\boldsymbol{\theta}_k) \prod_{c=1}^{C_g} \prod_{l=1}^L (\pi_{l|k})^{v_{gcl}} \right\}}, \quad (4)$$

for  $g = 1, 2, \dots, G$ , and  $k = 1, 2, \dots, K$ , and

$$\mathbb{E}(v_{gcl}|\mathbf{y}_{gc}, u_{g1}, u_{g2}, \dots, u_{gK}, \boldsymbol{\phi}_l, \pi_{l|1}, \dots, \pi_{l|K}) = \frac{\prod_{n=1}^N p(y_{gcn}|\boldsymbol{\phi}_l) \prod_{k=1}^K (\pi_{l|k})^{u_{gk}}}{\sum_{l=1}^L \left\{ \prod_{n=1}^N p(y_{gcn}|\boldsymbol{\phi}_l) \prod_{k=1}^K (\pi_{l|k})^{u_{gk}} \right\}}, \quad (5)$$

for  $g = 1, 2, \dots, G$ ,  $c = 1, 2, \dots, C_g$ , and  $l = 1, 2, \dots, L$ . Given the conditional expected values in (4) and (5) then:

$$\begin{aligned} \mathbb{P}(u_{gk} = 1|\mathbf{x}_g, v_{g11}, v_{g12}, \dots, v_{gC_gL}, \tau_k, \boldsymbol{\theta}_k, \pi_{1|k}, \dots, \pi_{L|k}) = \\ \frac{\tau_k \prod_{n=1}^N p(x_{gn}|\boldsymbol{\theta}_k) \prod_{c=1}^{C_g} \prod_{l=1}^L (\pi_{l|k})^{v_{gcl}}}{\sum_{k=1}^K \left\{ \tau_k \prod_{n=1}^N p(x_{gn}|\boldsymbol{\theta}_k) \prod_{c=1}^{C_g} \prod_{l=1}^L (\pi_{l|k})^{v_{gcl}} \right\}}, \end{aligned} \quad (6)$$

$$\begin{aligned} \mathbb{P}(v_{gcl} = 1|\mathbf{y}_{gc}, u_{g1}, u_{g2}, \dots, u_{gK}, \boldsymbol{\phi}_l, \pi_{l|1}, \dots, \pi_{l|K}) = \\ \frac{\prod_{n=1}^N p(y_{gcn}|\boldsymbol{\phi}_l) \prod_{k=1}^K (\pi_{l|k})^{u_{gk}}}{\sum_{l=1}^L \left\{ \prod_{n=1}^N p(y_{gcn}|\boldsymbol{\phi}_l) \prod_{k=1}^K (\pi_{l|k})^{u_{gk}} \right\}}, \end{aligned} \quad (7)$$

for  $g = 1, 2, \dots, G$ ,  $k = 1, 2, \dots, K$ ,  $c = 1, 2, \dots, C_g$ , and  $l = 1, 2, \dots, L$ .

The probabilities given in (6) and (7) could be used to sample values of  $\mathbf{U}$  and  $\mathbf{V}$  using an MCMC algorithm. However, applying an MCMC algorithm to the large dataset considered here where  $G = 15,722$  and  $C = 94,873$  would be computationally expensive. Therefore, for computational efficiency, an algorithm similar to that employed in Salter-Townshend and Murphy

[2013] and Chamroukhi and Huynh [2018] is used to compute the required conditional expected values. Under this approach, the expected values given in (4) and (5) are iteratively computed until convergence (after  $S$  iterations). The computed values are then used at convergence to calculate the required expected values:

$$\begin{aligned}\mathbb{E}(u_{gk} | \dots) &\approx u_{gk}^{(S)} = \hat{u}_{gk}, & \mathbb{E}(v_{gcl} | \dots) &\approx v_{gcl}^{(S)} = \hat{v}_{gcl}, \text{ and} \\ \mathbb{E}(u_{gk}v_{gcl} | \dots) &\approx u_{gk}^{(S)}v_{gcl}^{(S)} = \widehat{u_{gk}v_{gcl}}.\end{aligned}$$

In practice, here  $S \approx 10$  iterations were required to achieve convergence per iteration of the EM algorithm.

## A.2 Performance metrics for simulated datasets

**Table 4:** Mean performance metrics (standard deviations in parentheses) for 100 simulated datasets given  $\pi$  under case 3 from Table 1.

(a) DEG identification performance				
	FDR	Sensitivity	Specificity	ARI
<b>idiffomix</b>	<b>0.036</b> (0.025)	0.771 (0.072)	<b>0.993</b> (0.006)	0.766 (0.061)
<b>mclust</b>	0.102 (0.049)	<b>0.873</b> (0.046)	0.975 (0.015)	<b>0.800</b> (0.041)
<b>limma</b>	0.038 (0.021)	0.764 (0.064)	<b>0.993</b> (0.005)	0.760 (0.059)

(b) DMC identification performance				
	FDR	Sensitivity	Specificity	ARI
<b>idiffomix</b>	<b>0.009</b> (0.003)	<b>1.000</b> (< 0.001)	<b>0.994</b> (0.002)	<b>0.985</b> (0.004)
<b>mclust</b>	<b>0.009</b> (0.003)	<b>1.000</b> (< 0.001)	<b>0.994</b> (0.002)	<b>0.985</b> (0.004)
<b>limma</b>	0.046 (0.004)	1.000 (< 0.001)	0.968 (0.003)	0.924 (0.006)

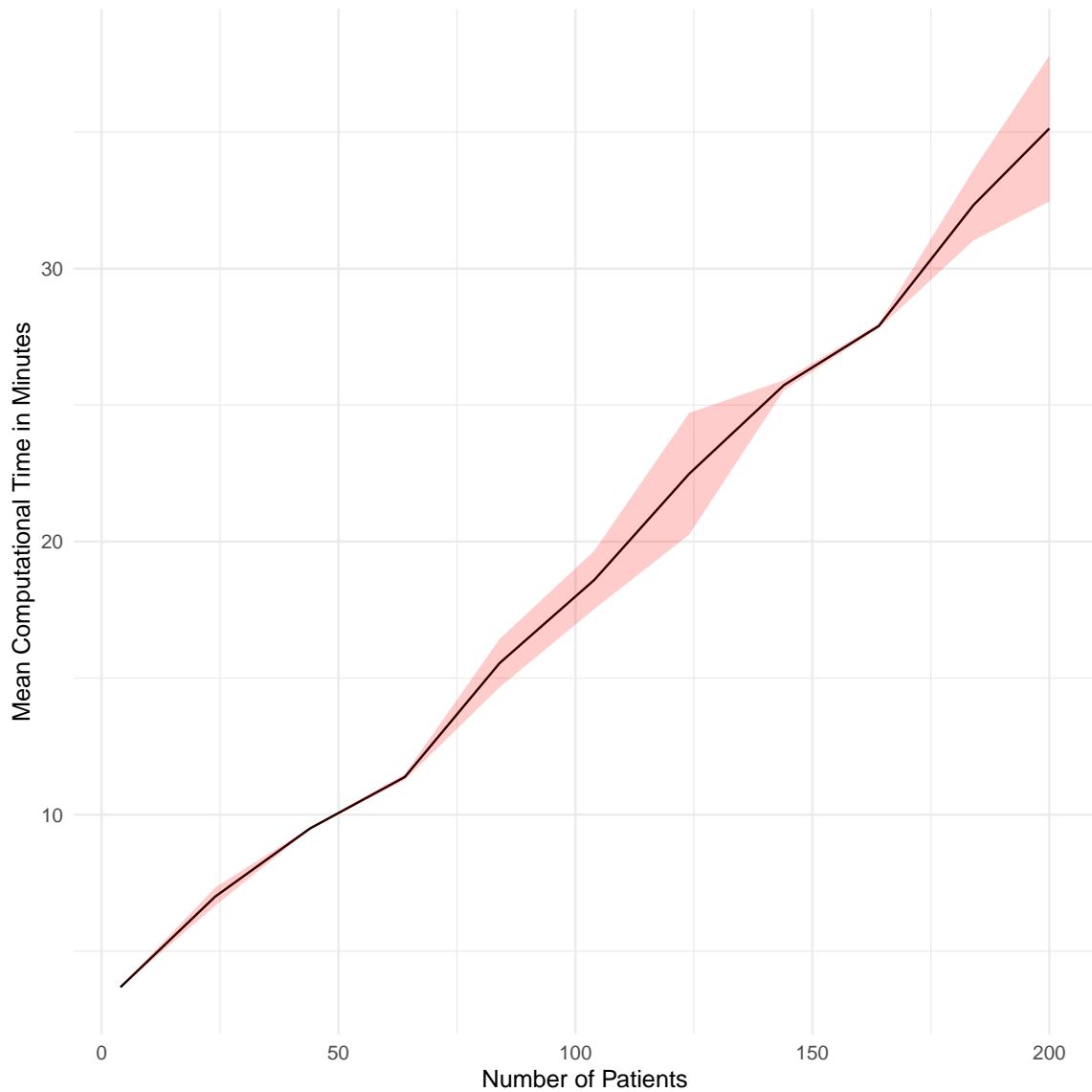
**Table 5:** Mean ARI values (with standard deviations in parentheses) for 100 simulated datasets given  $\pi$  under case 3 from Table 1 for comparing clustering solutions between methods.

(a) ARI for DEG identification	
Method	Mean ARI
<b>idiffomix vs mclust</b>	0.837 (0.100)
<b>idiffomix vs limma</b>	0.936 (0.041)

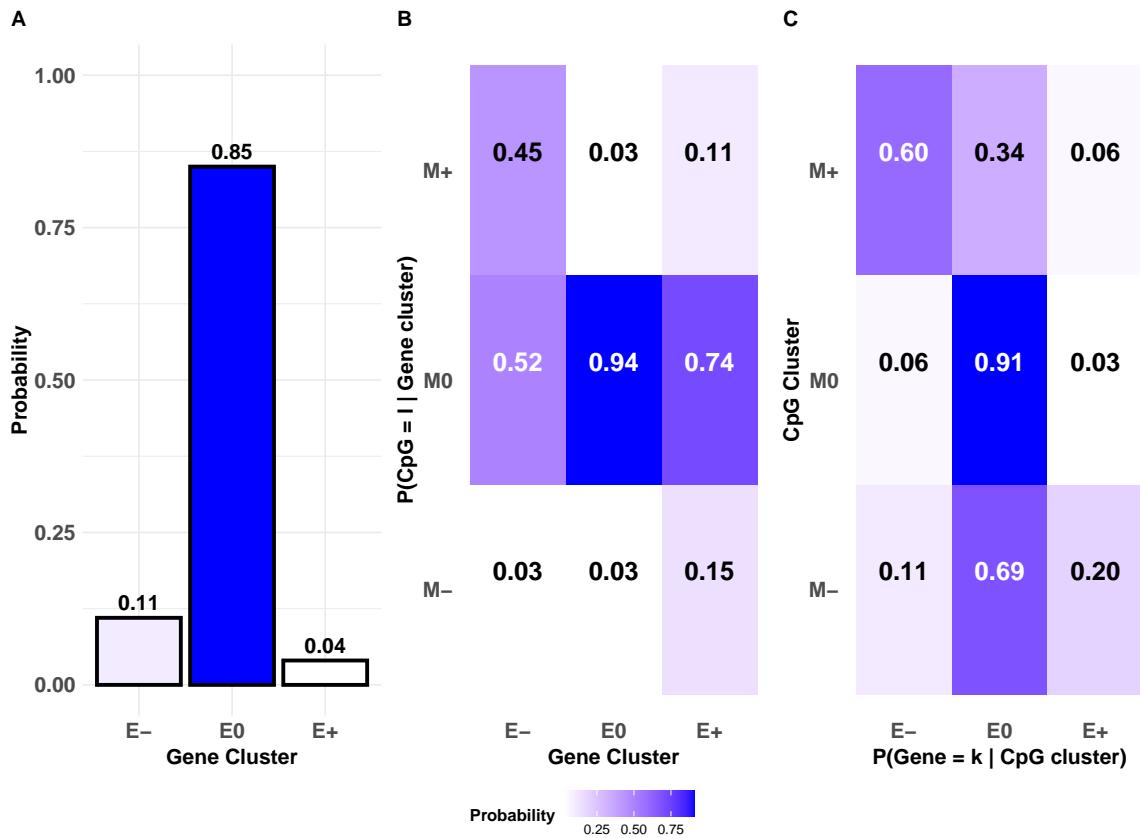
(b) ARI for DMC identification	
Method	Mean ARI
<b>idiffomix vs mclust</b>	0.999 (0.001)
<b>idiffomix vs limma</b>	0.915 (0.007)

### A.3 Time complexity

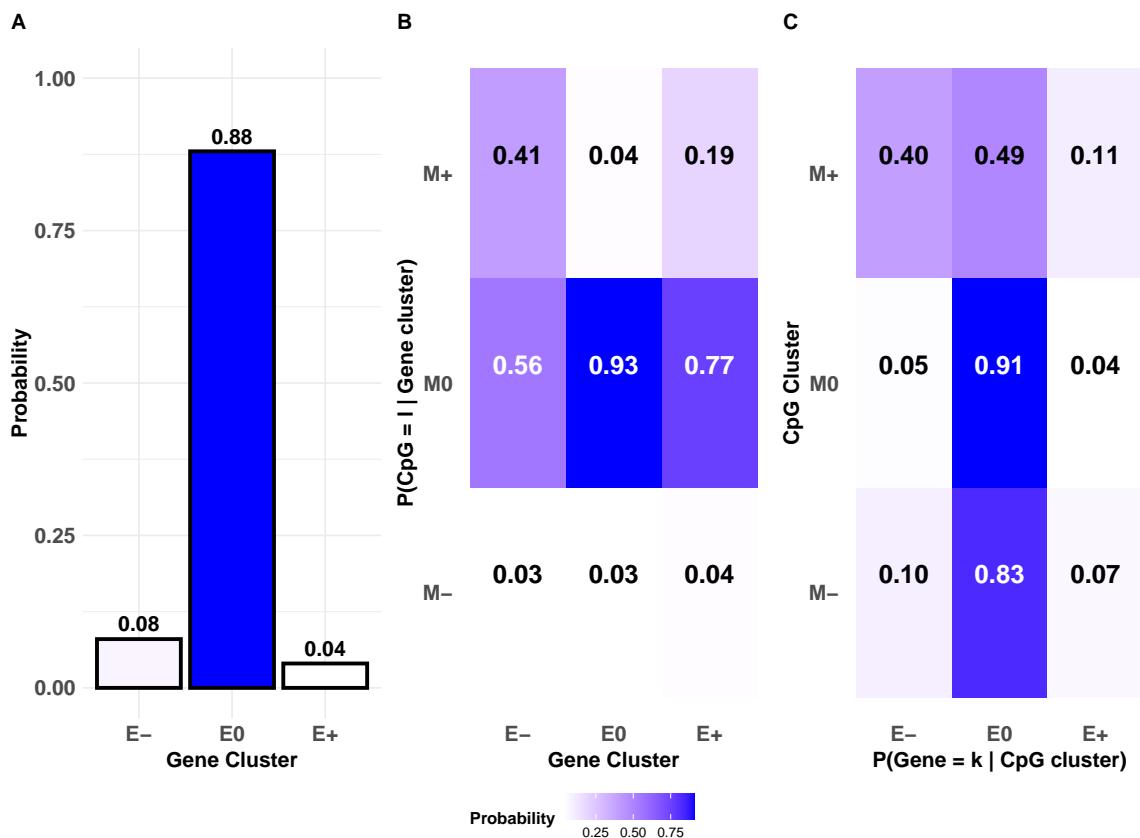


**Figure 7:** Mean computational time for fitting the `idiffomix` model, with 95% confidence intervals, as the number of patients,  $N$ , is increased from 4 to 200. As the complexity of the algorithm with respect to  $N$  is proportional to  $N$ , as the number of patients increases the computational cost scales linearly.

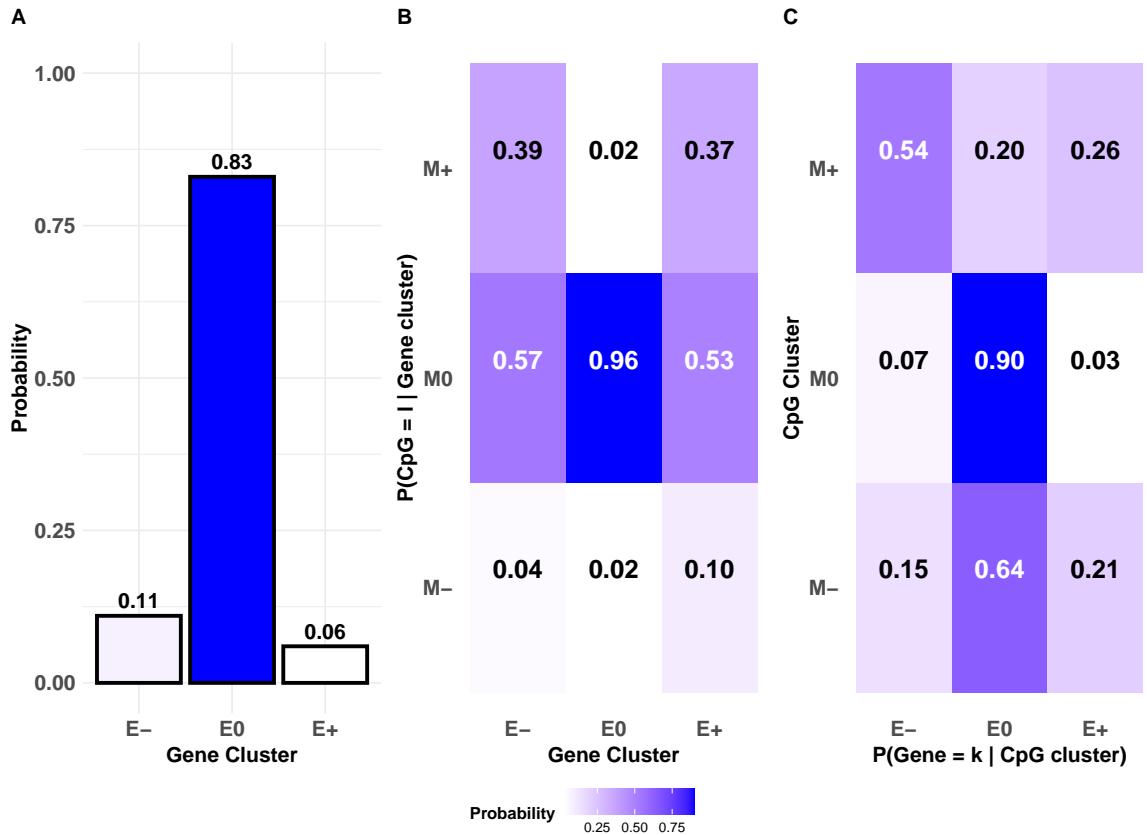
#### A.4 Idiffomix results for all chromosomes from TCGA breast cancer data



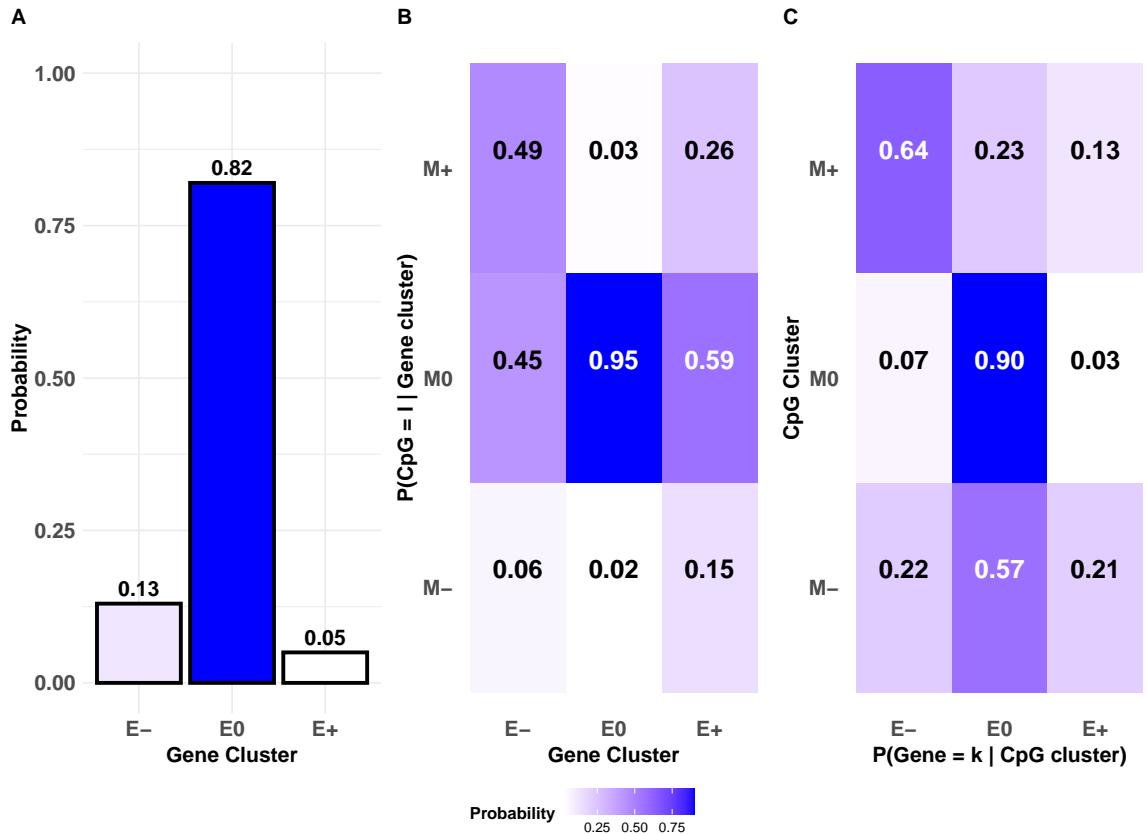
**Figure 8:** Idiffomix applied to chromosome 1 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



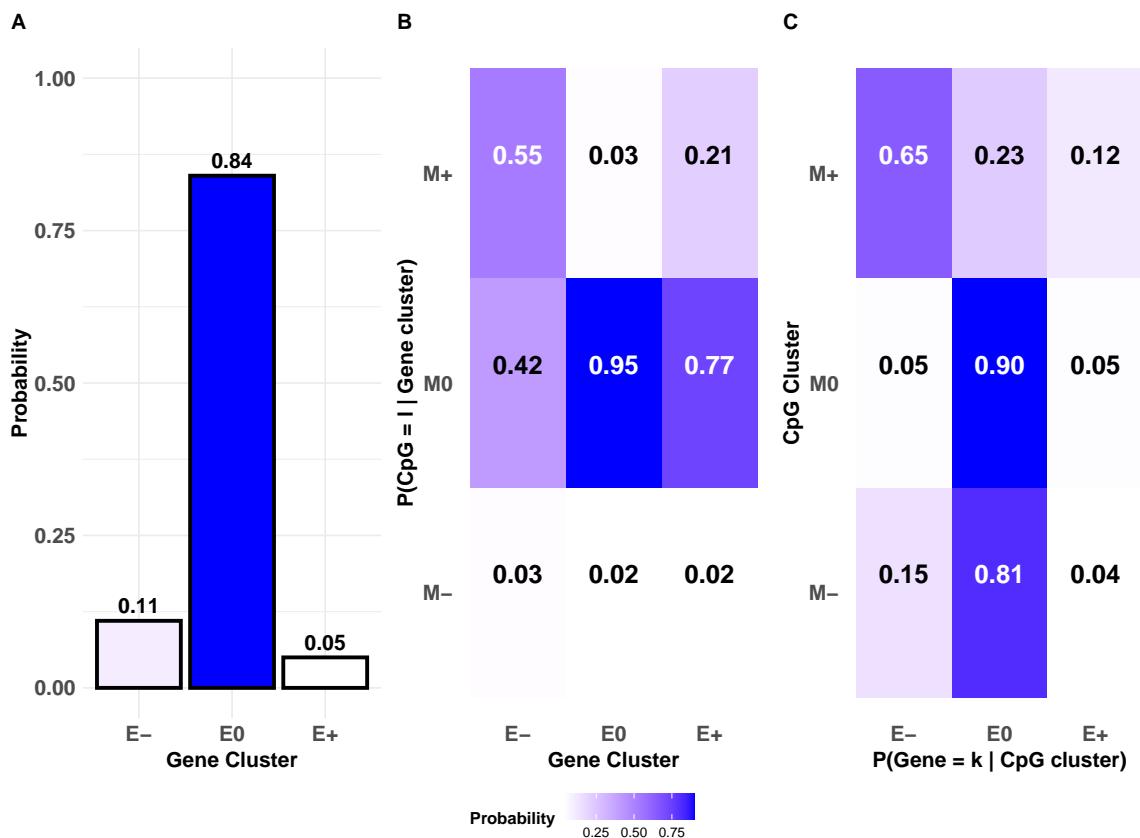
**Figure 9:** Idiffomix applied to chromosome 2 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



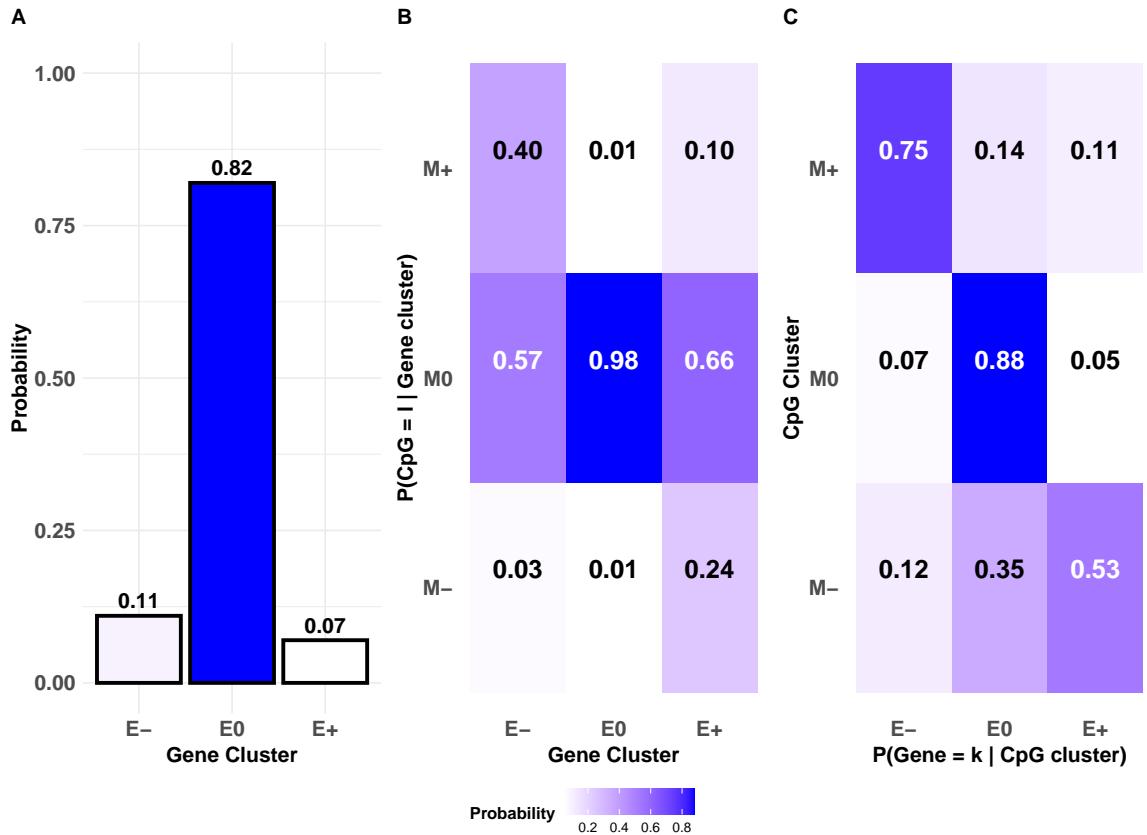
**Figure 10:** Idiffomix applied to chromosome 3 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



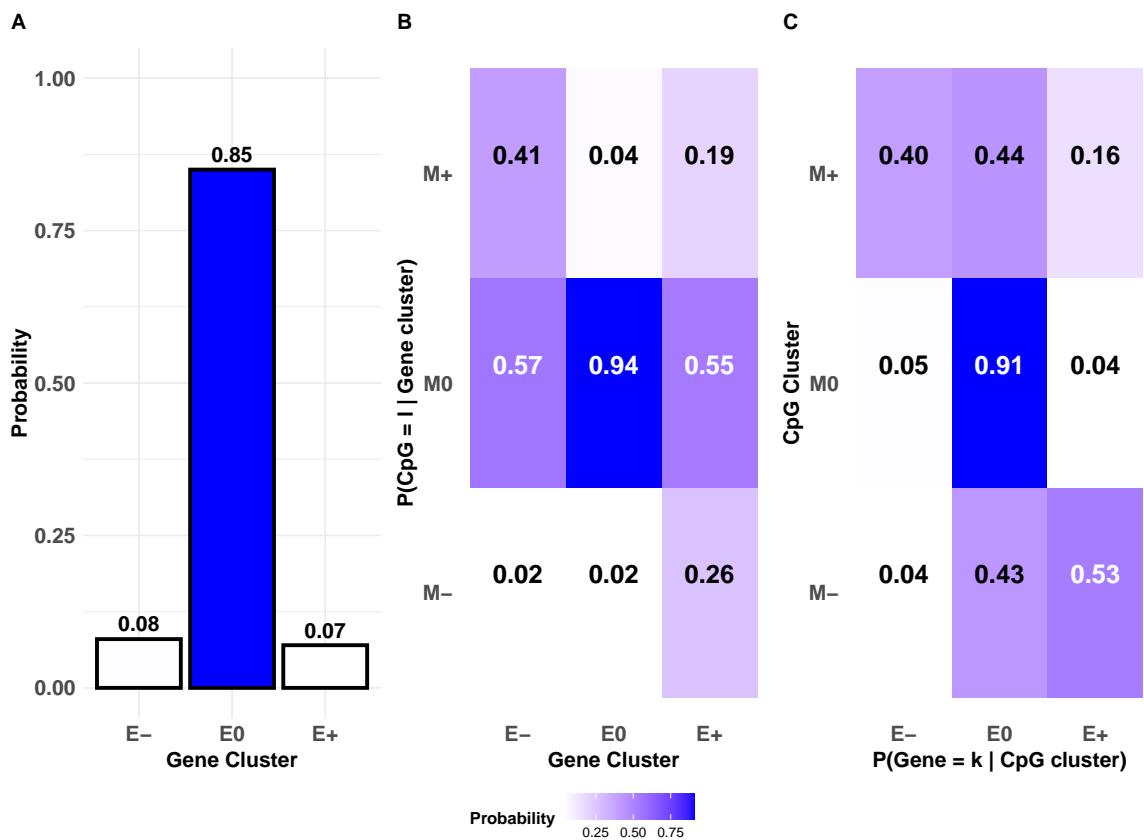
**Figure 11:** Idiffomix applied to chromosome 4 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



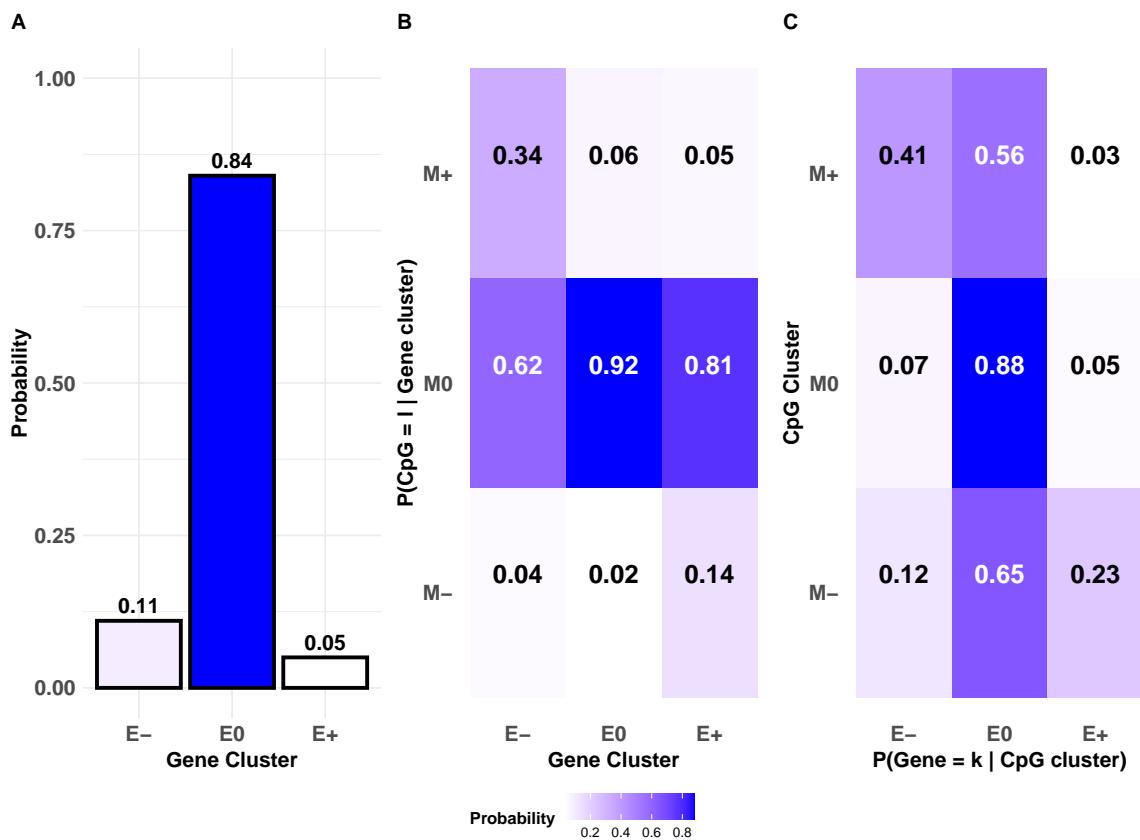
**Figure 12:** Idiffomix applied to chromosome 5 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



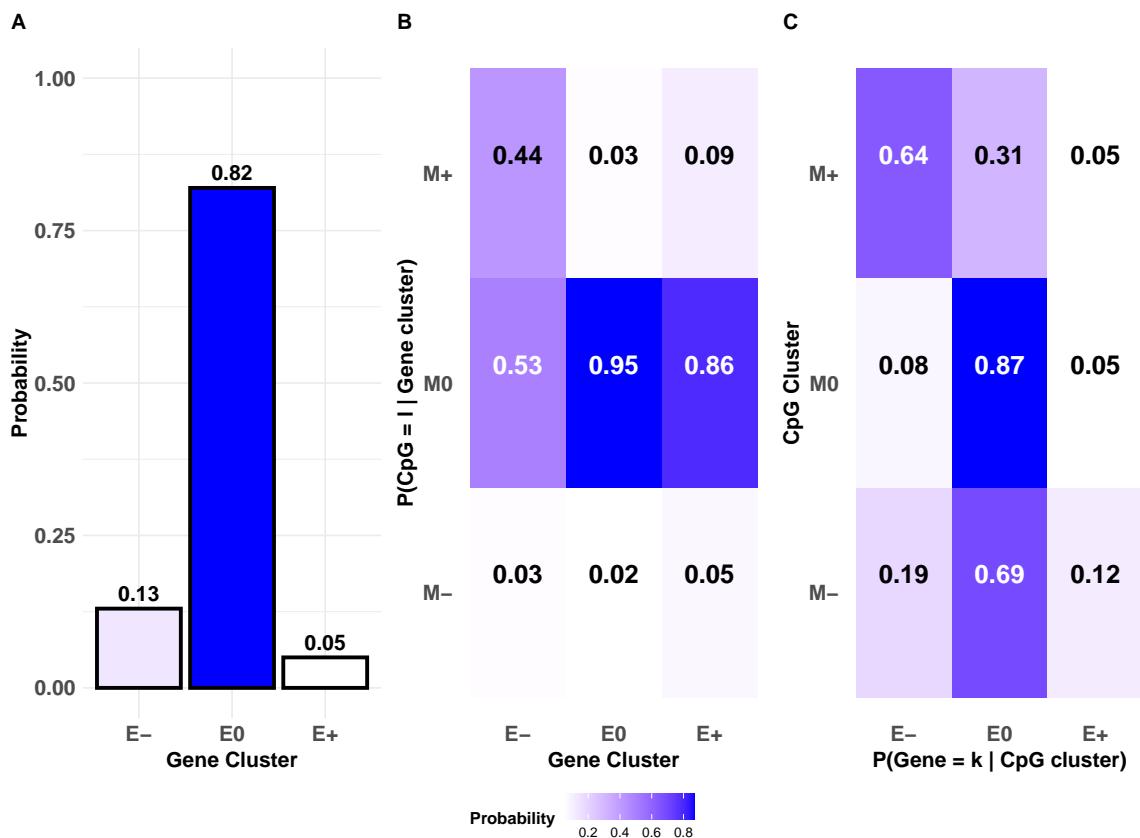
**Figure 13:** Idiffomix applied to chromosome 6 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



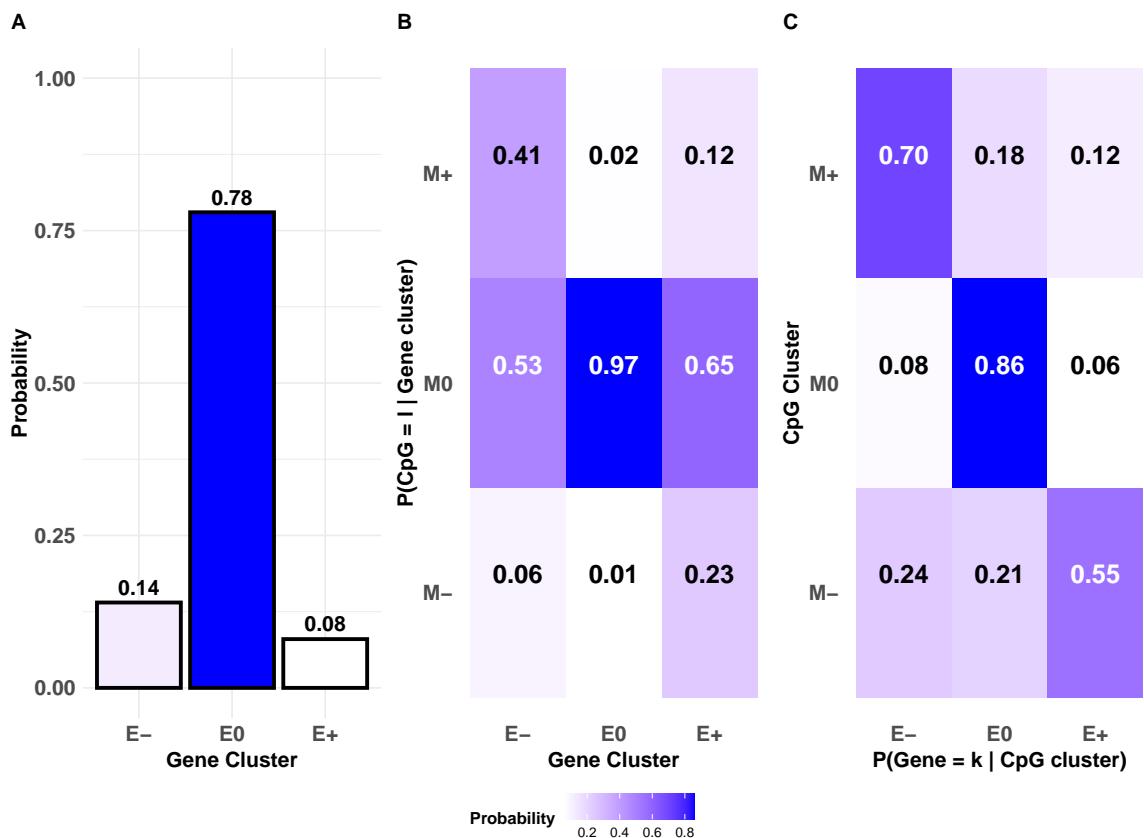
**Figure 14:** Idiffomix applied to chromosome 8 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



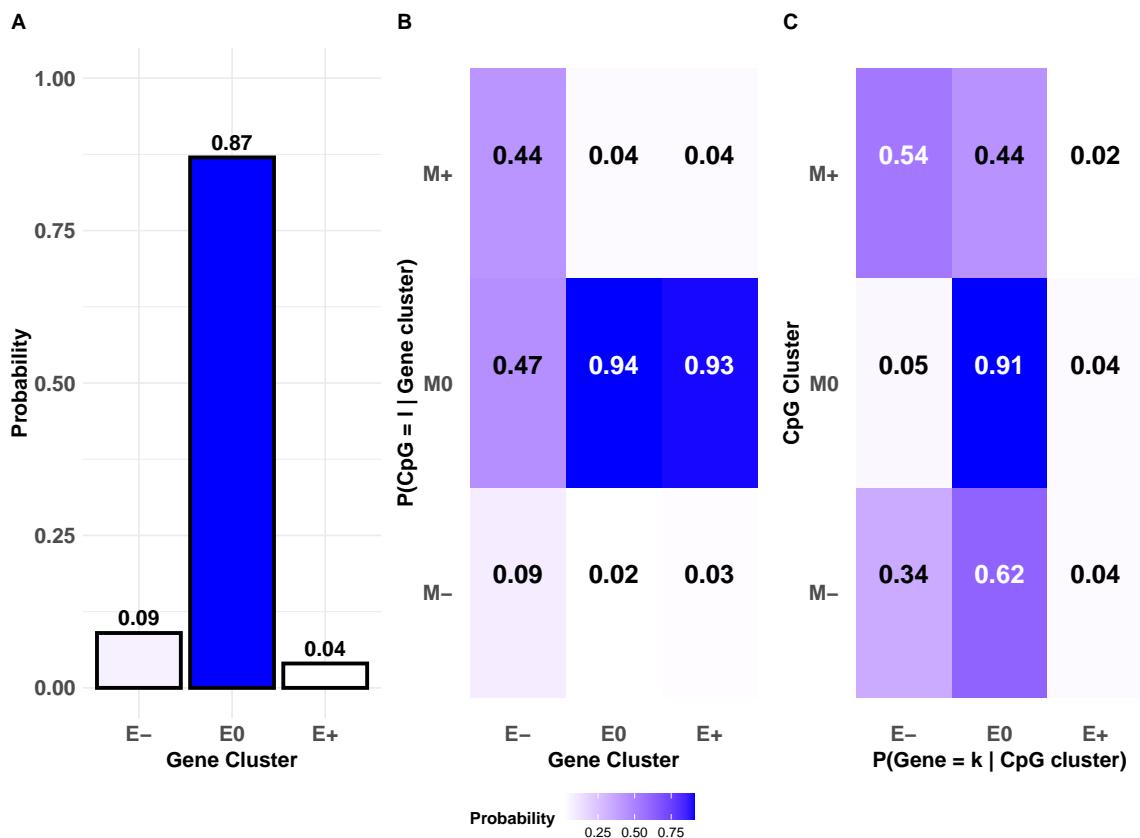
**Figure 15:** Idiffomix applied to chromosome 9 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



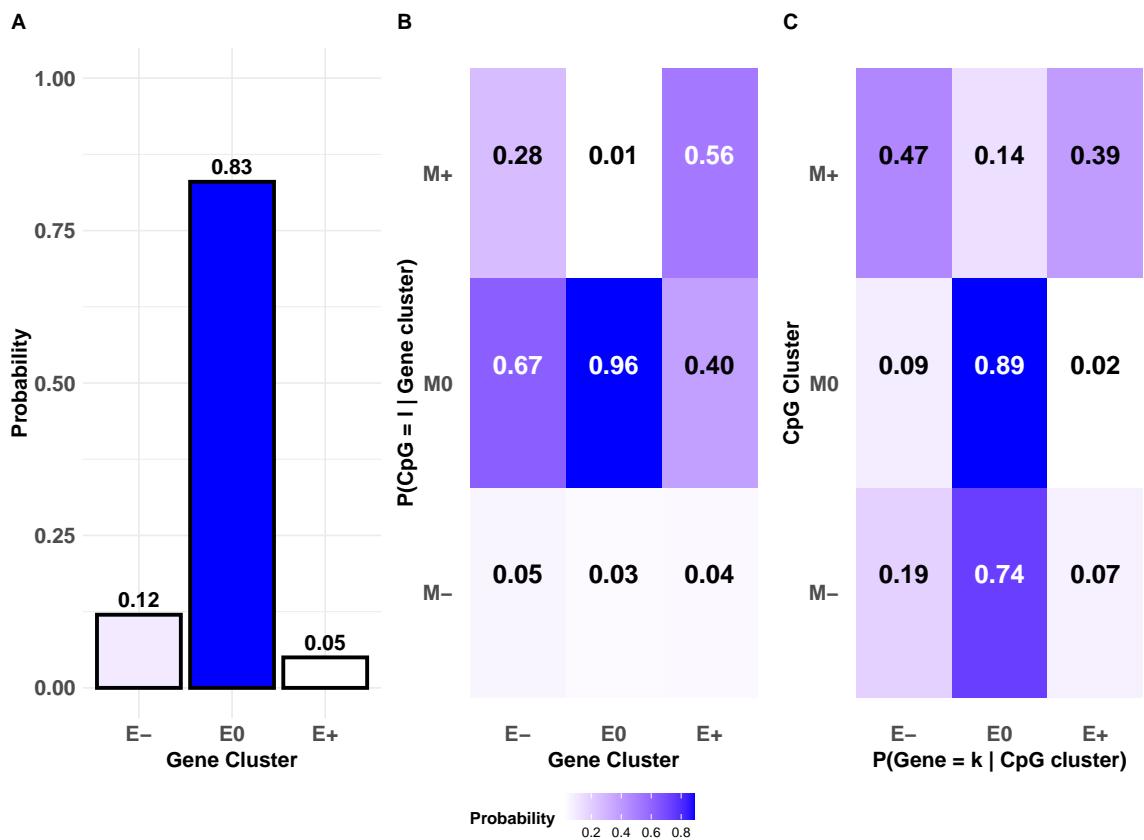
**Figure 16:** Idiffomix applied to chromosome 10 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



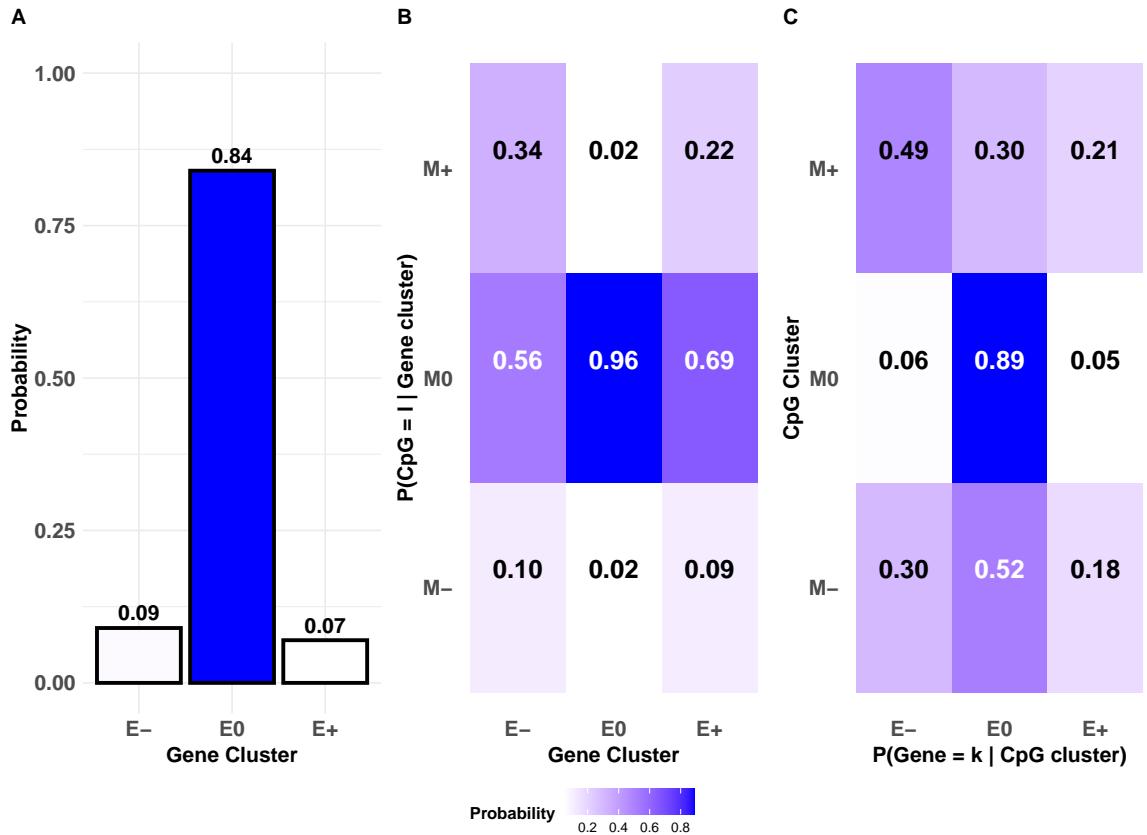
**Figure 17:** Idiffomix applied to chromosome 11 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



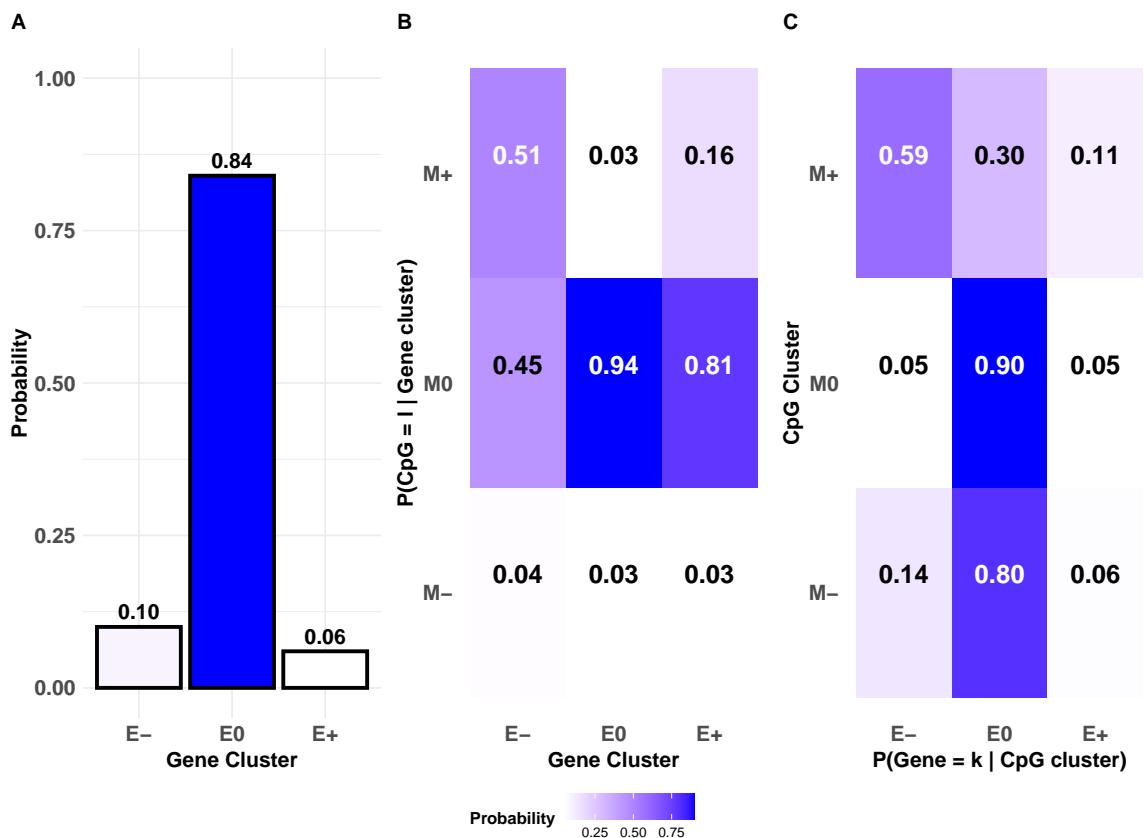
**Figure 18:** Idiffomix applied to chromosome 12 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



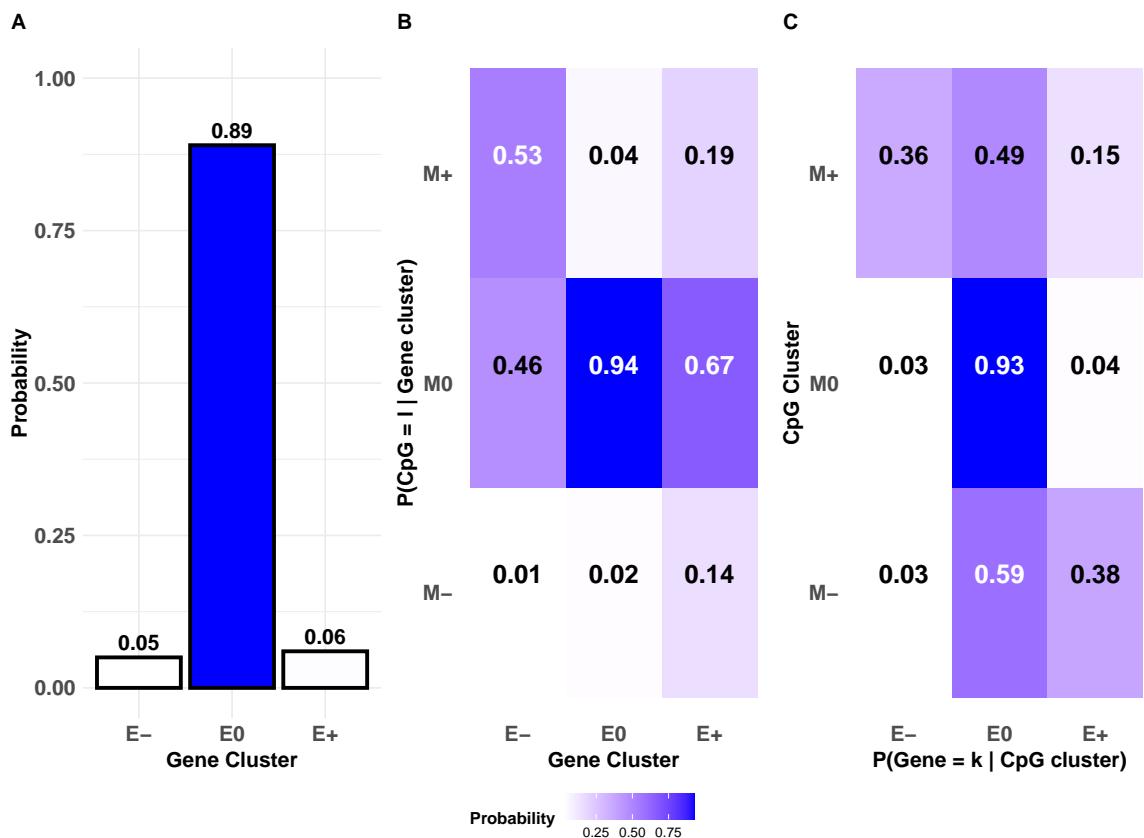
**Figure 19:** Idiffomix applied to chromosome 13 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



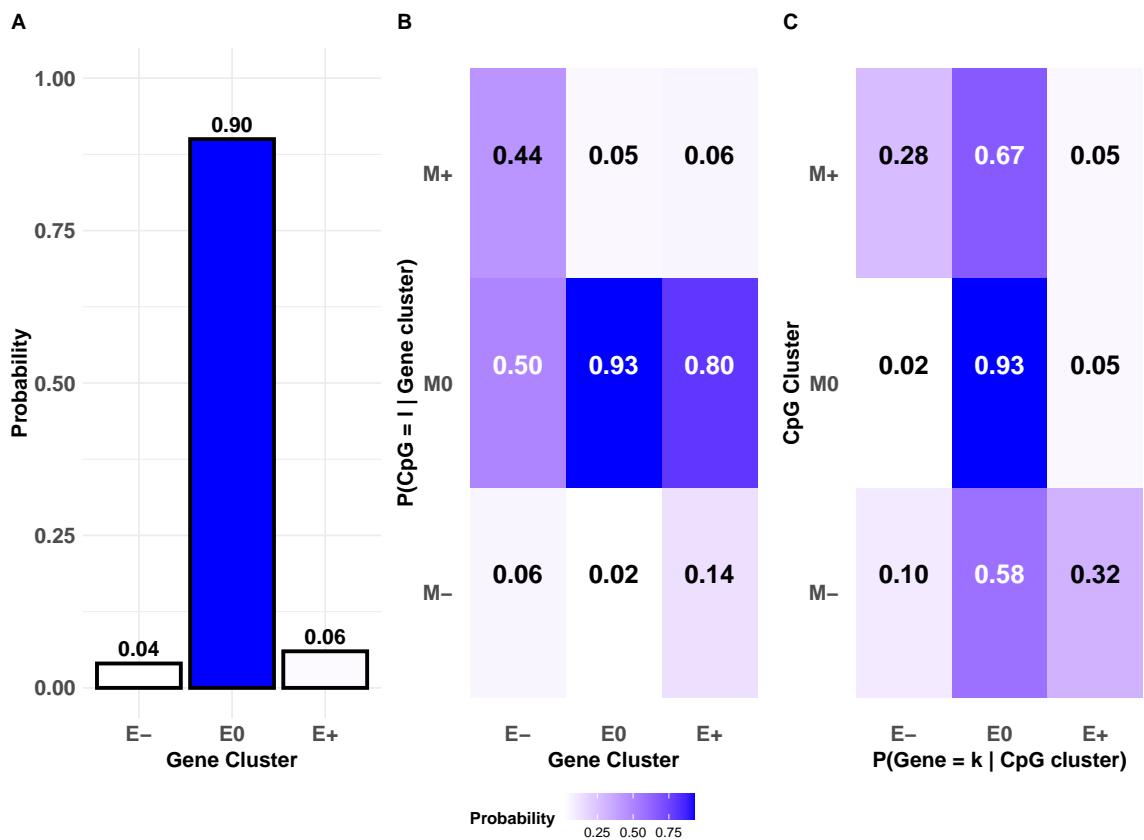
**Figure 20:** Idiffomix applied to chromosome 14 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



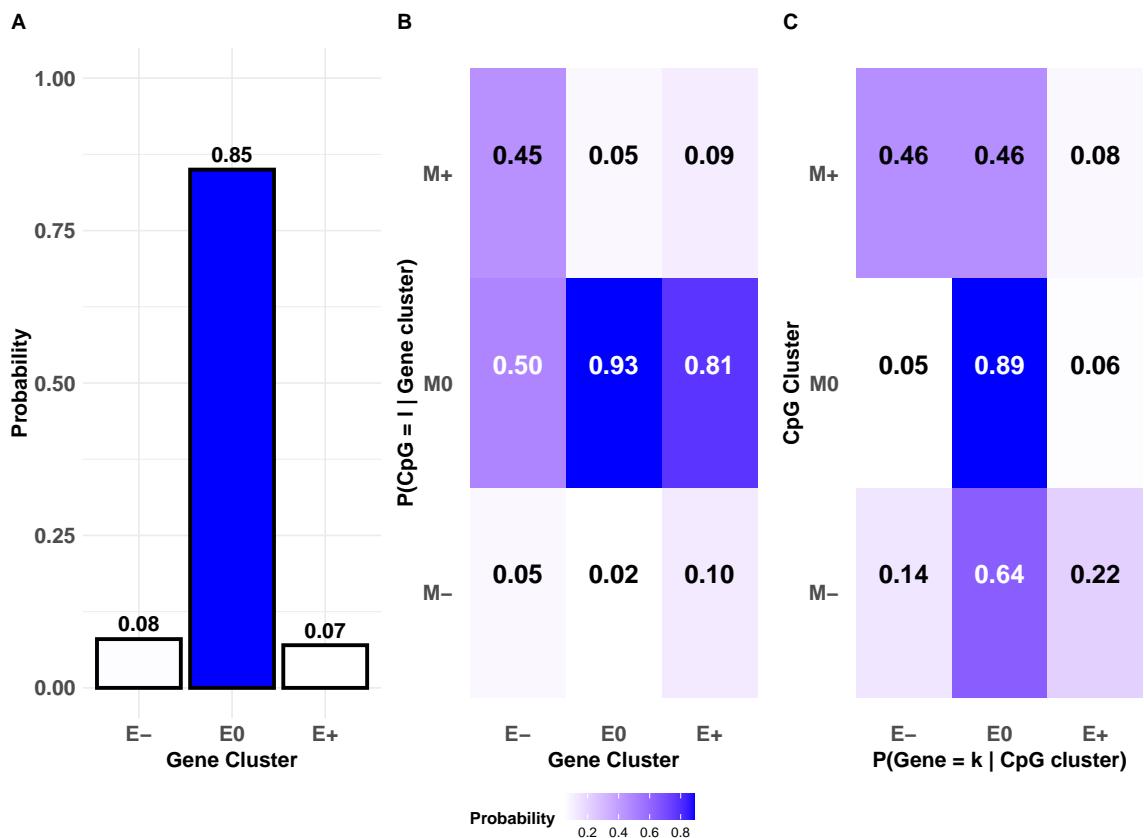
**Figure 21:** Idiffomix applied to chromosome 15 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



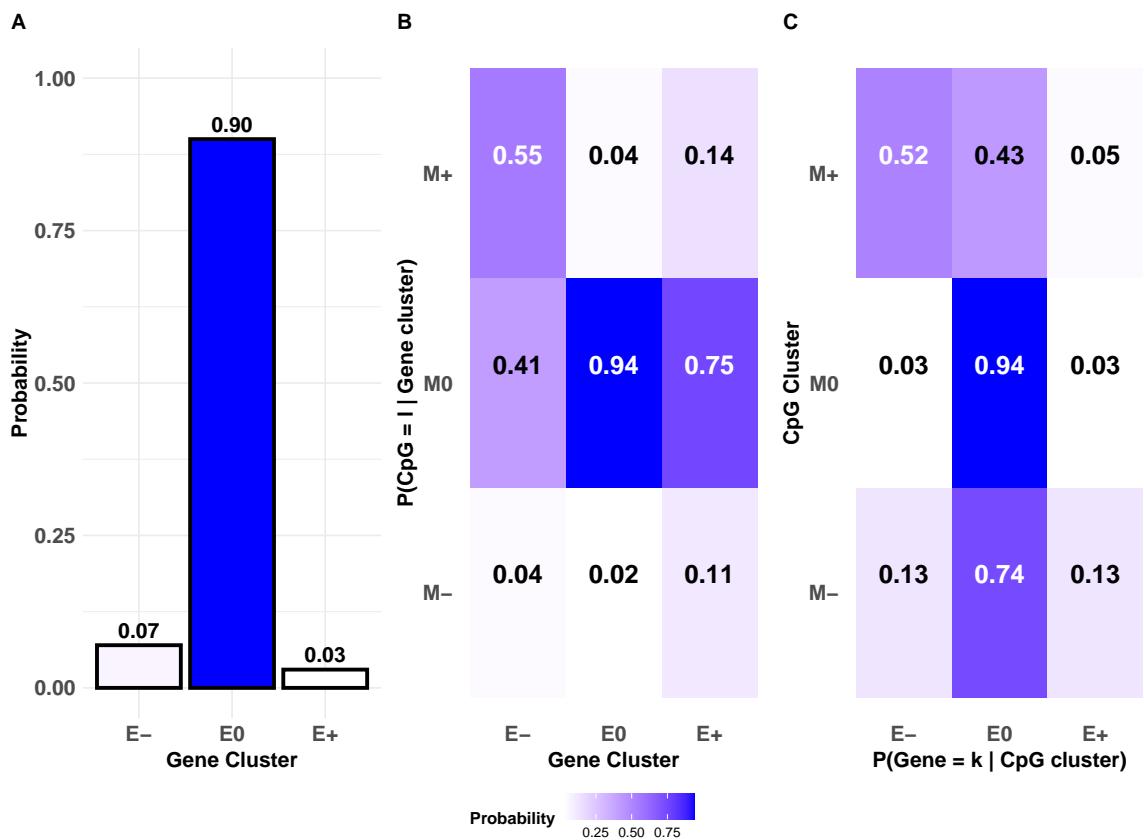
**Figure 22:** Idiffomix applied to chromosome 16 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



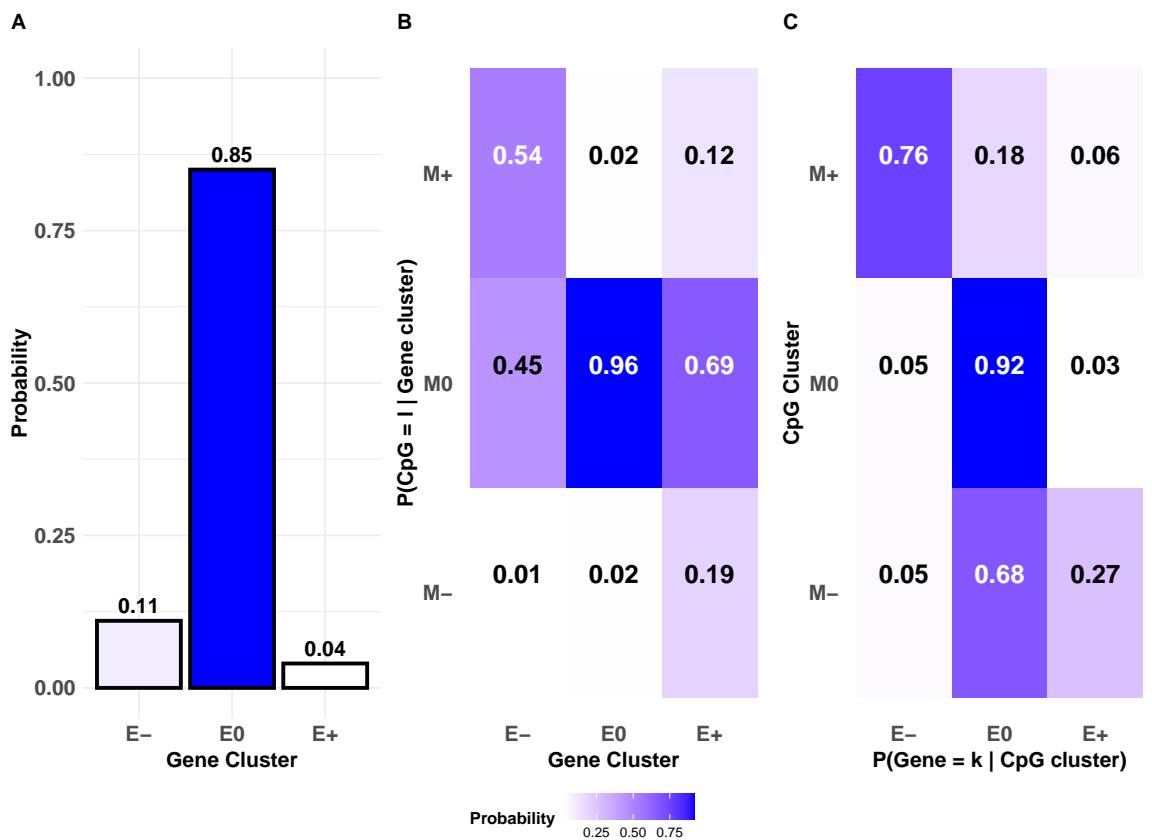
**Figure 23:** Idiffomix applied to chromosome 17 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



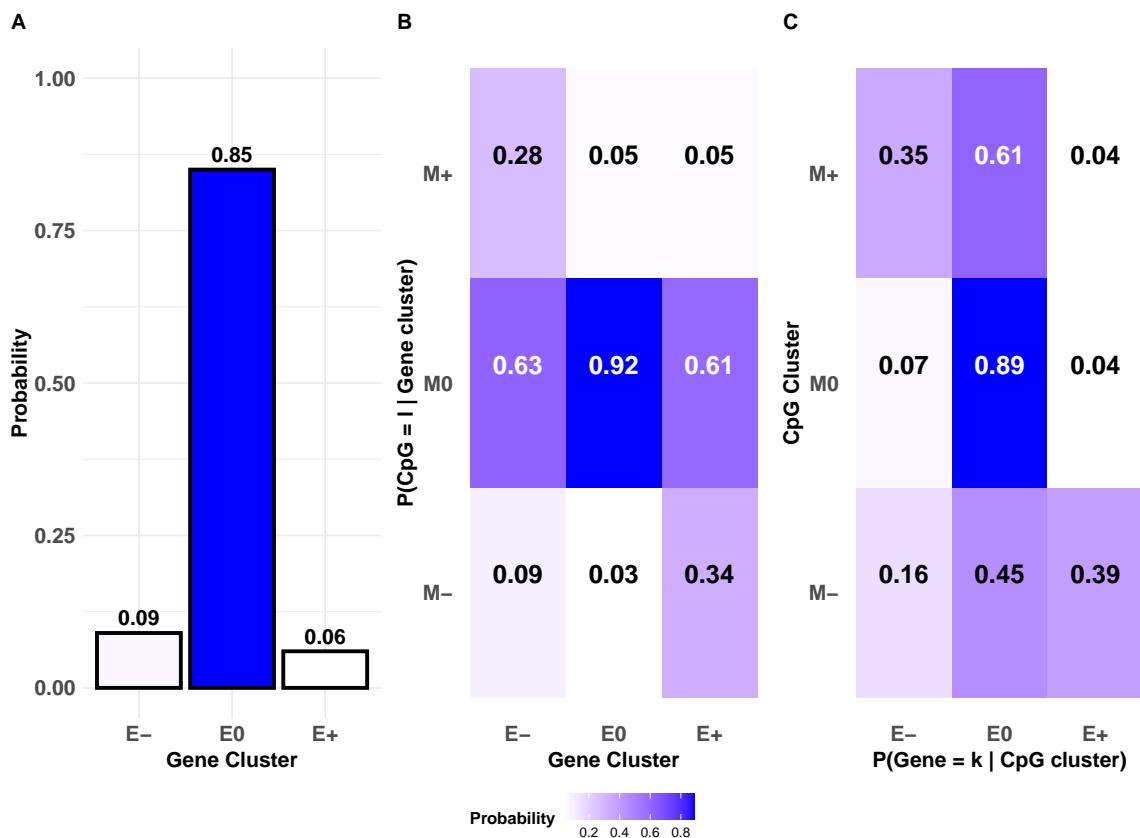
**Figure 24:** Idiffomix applied to chromosome 18 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



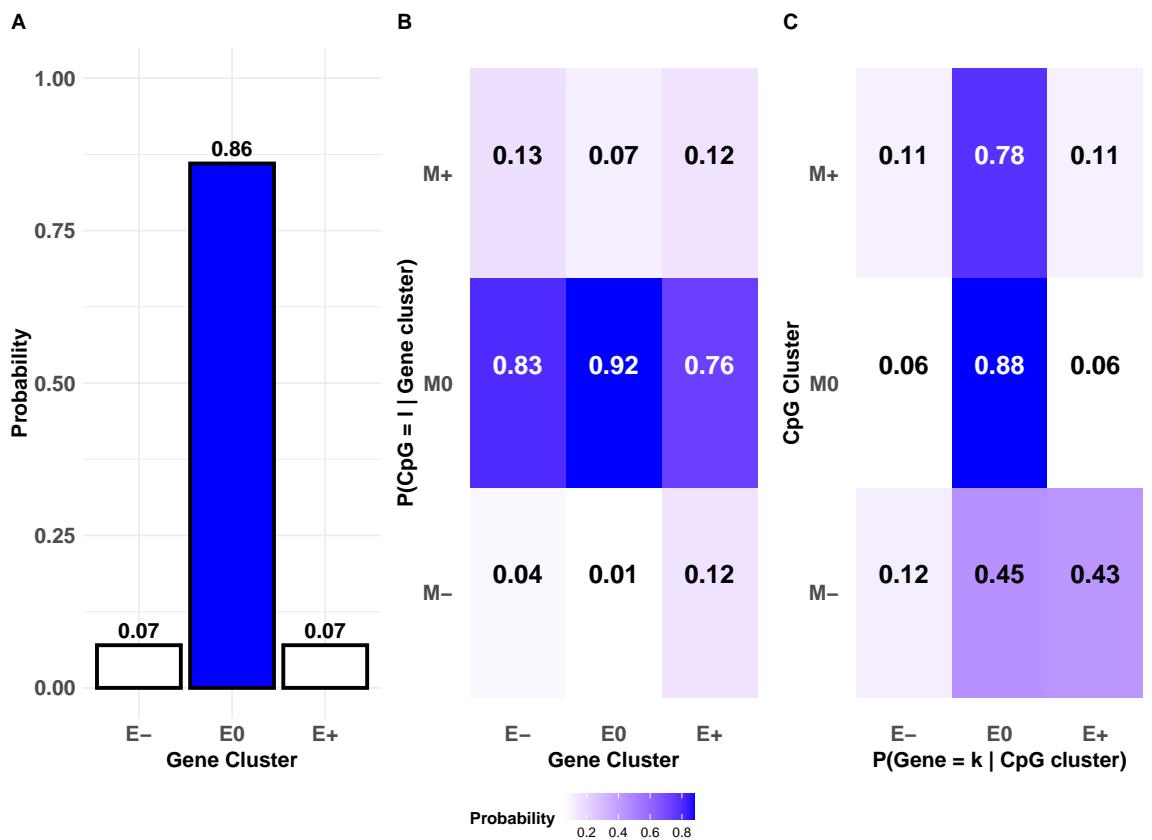
**Figure 25:** Idiffomix applied to chromosome 19 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



**Figure 26:** Idiffomix applied to chromosome 20 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .

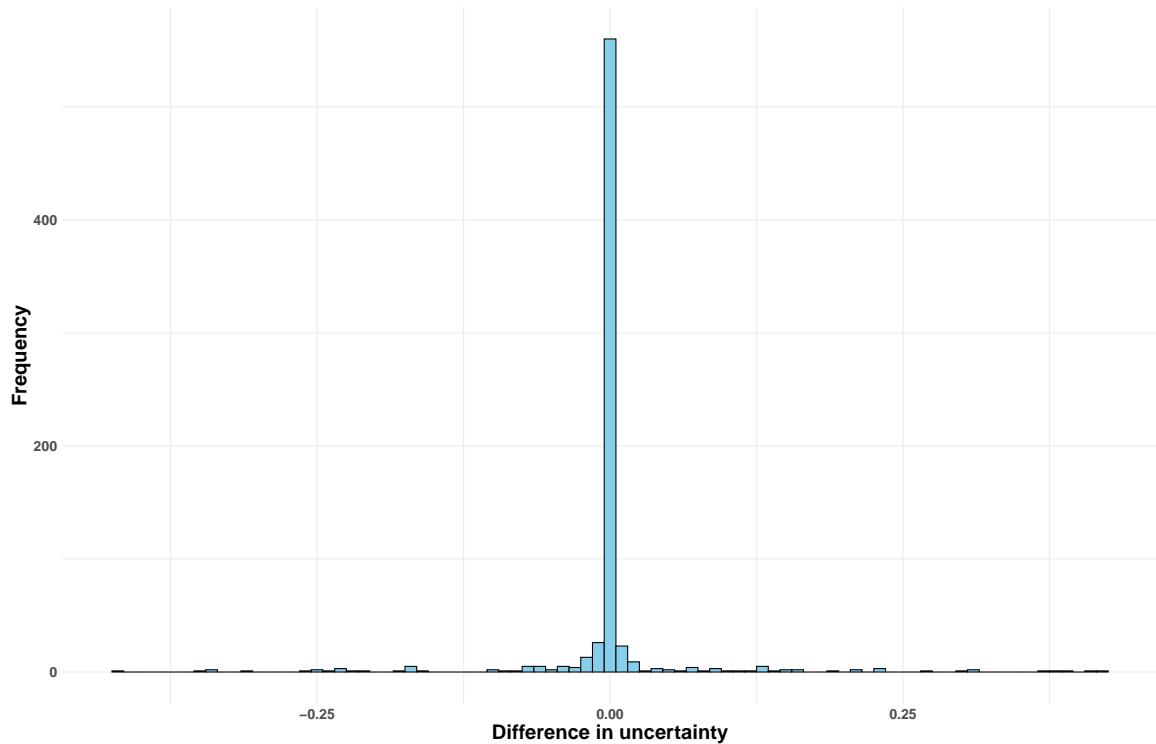


**Figure 27:** Idiffomix applied to chromosome 21 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .



**Figure 28:** Idiffomix applied to chromosome 22 of TCGA breast cancer data: (A) estimated cluster membership probabilities  $\hat{\tau}$ , (B) the estimated matrix  $\hat{\pi}$  of conditional probabilities of CpG site methylation status given gene cluster membership, (c) conditional probabilities of genes belonging to cluster  $k$  given a single CpG site associated with the gene belongs to cluster  $l$ .

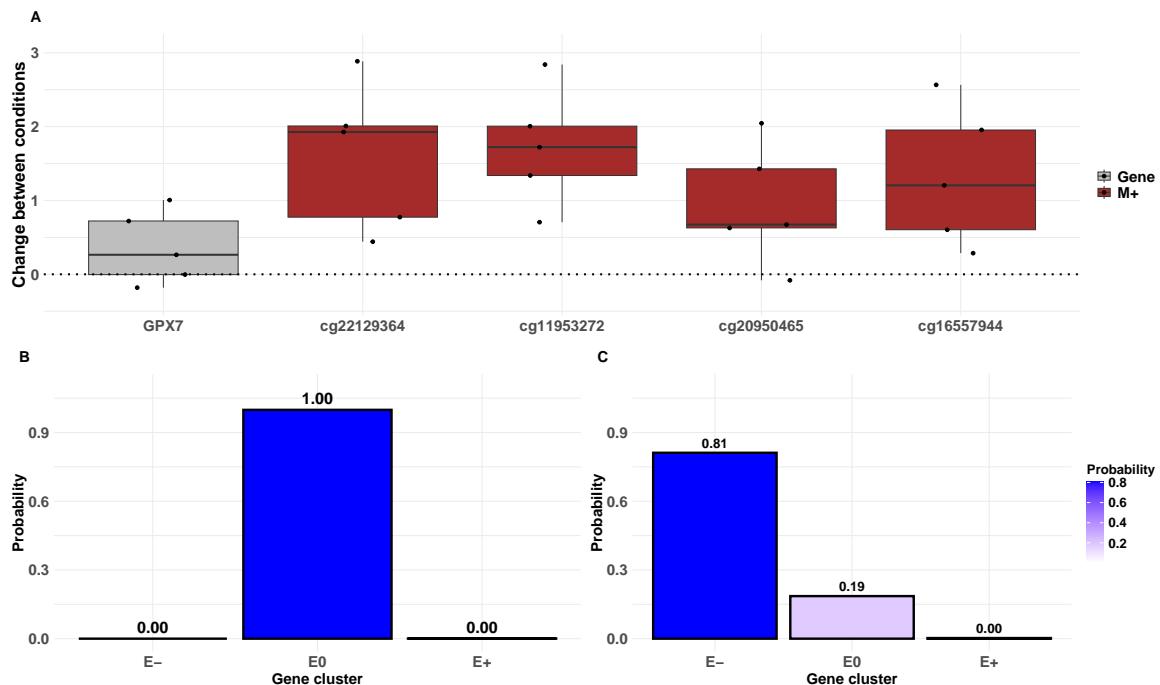
## A.5 Clustering uncertainty



**Figure 29:** Difference in uncertainties for genes not changing clustering between `idiffomix` and `mclust`.

## A.6 Other genes of interest

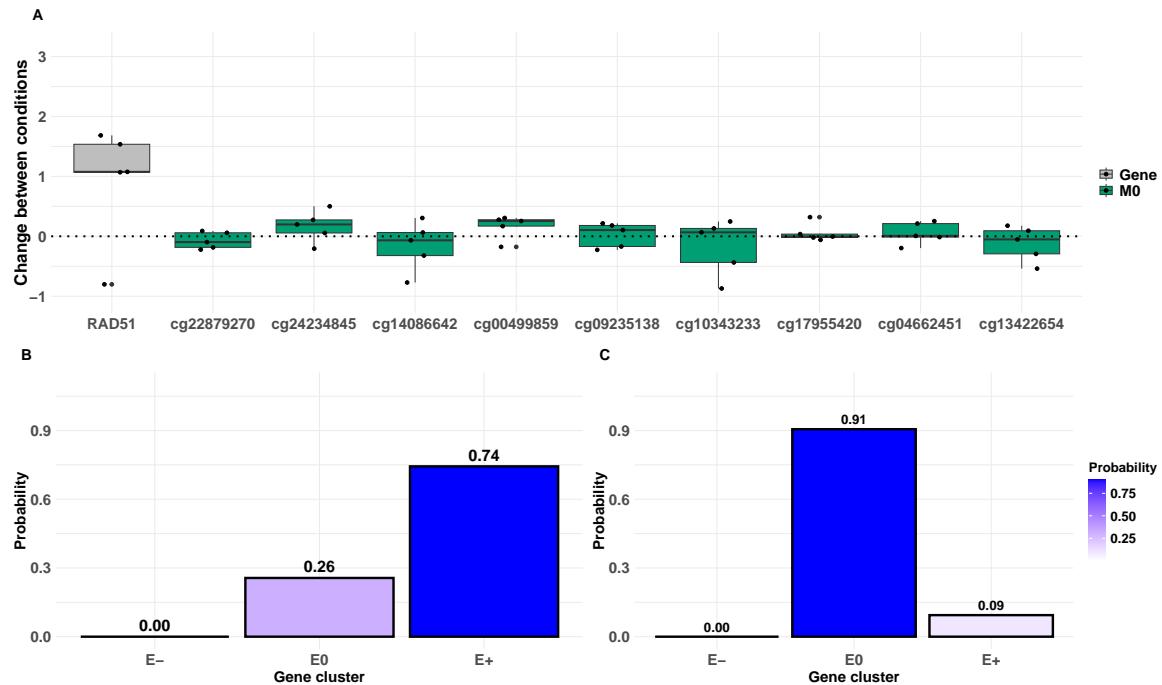
Genes implicated in breast cancer, such as *TNFRSF18*, *GPX7*, and *RAD51* play important roles in the development and progression of the disease. Panel A in Figure 30 illustrates that the log-fold change of the gene expression for *GPX7* and the differences in *M*-values for the associated CpG sites, located on chromosome 1. When the gene expression data is modelled independently Panel B in Figure 30 suggested *GPX7* to be non-differential (E0). However, the differences in *M*-values of the CpG sites linked to this gene are suggested to be hypermethylated (M+) when modelled jointly. Therefore, when the two data types are modelled jointly, Panel C in Figure 30 suggests the gene to be downregulated (E-) with a probability of 0.81 and to be non-differential (E0) with a probability of 0.19.



**Figure 30:** Comparison of results for independent and integrated analyses for *GPX7* on chromosome 1: (A) log-fold change in gene expression levels (grey) and differences in *M*-values between tumour and normal samples, coloured by inferred idiffomix cluster (hypermethylated CpG sites, M+ in brown); (B) posterior probability of *GPX7* belonging to the E-, E0 and E+ clusters under *mclust*, (C) posterior probability of *GPX7* belonging to the E-, E0 and E+ clusters when jointly modelled with methylation data under *idiffomix*. Larger posterior probabilities are represented by increasingly dark shades of blue

Panel A in Figure 31 illustrates the log-fold change for *RAD51* gene, located on chromosome 15, and the differences in *M*-values for the associated CpG sites. Panel B in Figure 31 suggests the gene to be upregulated when modelled independently and is estimated to be in cluster E+ with probability of 0.74. However, the CpG sites linked to this gene are all found to be

non-differential (M0) when the two data types are modelled jointly. Thus, Panel C in Figure 31 suggests the gene to be in E0 cluster with a probability of 0.9 and in E+ cluster with a probability of 0.1 when the gene expression and methylation data are modelled jointly.



**Figure 31:** Comparison of results for independent and integrated analyses for *RAD51* on chromosome 15: (A) log-fold change in gene expression levels (grey) and differences in *M*-values between tumour and normal samples, coloured by inferred idiffomix cluster (non-differentially methylated CpG sites, M0 in green); (B) posterior probability of *RAD51* belonging to the E-, E0 and E+ clusters under *mclust*, (C) posterior probability of *RAD51* belonging to the E-, E0 and E+ clusters when jointly modelled with methylation data under *idiffomix*. Larger posterior probabilities are represented by increasingly dark shades of blue.

## A.7 Top GO and KEGG terms associated with identified DEGs and DMCs

**Table 6:** Top 10 GO processes linked to DMCs identified by `idiffomix` and not `limma`.

GO Process	ONTOLOGY	TERM	FDR
GO:0062023	CC	collagen-containing extracellular matrix	$4.91 \times 10^{-13}$
GO:0051094	BP	positive regulation of developmental process	$2.68 \times 10^{-11}$
GO:0045229	BP	external encapsulating structure organization	$3.28 \times 10^{-11}$
GO:0007166	BP	cell surface receptor signaling pathway	$4.56 \times 10^{-11}$
GO:0030198	BP	extracellular matrix organization	$8.04 \times 10^{-11}$
GO:0043062	BP	extracellular structure organization	$1.31 \times 10^{-10}$
GO:0048513	BP	animal organ development	$3.65 \times 10^{-10}$
GO:0048646	BP	anatomical structure formation involved in morphogenesis	$5.432 \times 10^{-10}$
GO:0005201	MF	extracellular matrix structural constituent	$1.07 \times 10^{-9}$
GO:0030545	MF	signaling receptor regulator activity	$1.5 \times 10^{-9}$

**Table 7:** Top 10 GO processes linked to DMCs identified by `idiffomix` and not `mclust`.

GO Process	ONTOLOGY	TERM	FDR
GO:0030509	BP	BMP signaling pathway	0.0023
GO:0018146	BP	keratan sulfate biosynthetic process	0.0040
GO:0042339	BP	keratan sulfate metabolic process	0.0042
GO:0035282	BP	segmentation	0.0056
GO:0010720	BP	positive regulation of cell development	0.0060
GO:0043950	BP	positive regulation of cAMP-mediated signaling	0.0081
GO:0000165	BP	MAPK cascade	0.0082
GO:0070371	BP	ERK1 and ERK2 cascade	0.0128
GO:0001707	BP	mesoderm formation	0.0134
GO:0035418	BP	protein localization to synapse	0.0144

**Table 8:** Top 10 KEGG pathways linked to DMCs identified by `idiffomix` and not `limma`.

KEGG Pathway	TERM	FDR
hsa04820	Cytoskeleton in muscle cells	$1.41 \times 10^{-05}$
hsa04060	Cytokine-cytokine receptor interaction	$3.42 \times 10^{-05}$
hsa05033	Nicotine addiction	$3.42 \times 10^{-05}$
hsa04974	Protein digestion and absorption	$2.55 \times 10^{-03}$
hsa04512	ECM-receptor interaction	$3.44 \times 10^{-03}$
hsa04514	Cell adhesion molecules	$1.24 \times 10^{-02}$
hsa05146	Amoebiasis	$1.53 \times 10^{-02}$
hsa00512	Mucin type O-glycan biosynthesis	$1.89 \times 10^{-02}$
hsa04061	Viral protein interaction with cytokine and cytokine receptor	$2.09 \times 10^{-02}$
hsa05032	Morphine addiction	$2.09 \times 10^{-02}$

**Table 9:** Top KEGG pathways linked to DMCs identified by `idiffomix` and not `mclust`.

KEGG Pathway	TERM	FDR
hsa04640	Hematopoietic cell lineage	0.027
hsa04724	Glutamatergic synapse	0.035

**Table 10:** Top 10 GO processes associated with DEGs identified by `idiffomix` and not by `limma`.

GO Process	TERM	Adjusted p-value
GO:0046942	carboxylic acid transport	$9.60 \times 10^{-08}$
GO:0015849	organic acid transport	$1.034 \times 10^{-07}$
GO:0015711	organic anion transport	$2.58 \times 10^{-07}$
GO:0009954	proximal/distal pattern formation	$3.32 \times 10^{-07}$
GO:0071805	potassium ion transmembrane transport	$1.01 \times 10^{-06}$
GO:0042698	ovulation cycle	$1.31 \times 10^{-06}$
GO:0098657	import into cell	$2.42 \times 10^{-06}$
GO:0048645	animal organ formation	$2.53 \times 10^{-06}$
GO:0071772	response to BMP	$4.48 \times 10^{-06}$
GO:0071773	cellular response to BMP stimulus	$4.48 \times 10^{-06}$

**Table 11:** Top 10 GO processes associated with DEGs identified by `idiffomix` and not by `mclust`.

GO Process	TERM	Adjusted p-value
GO:0007215	glutamate receptor signaling pathway	$1.66 \times 10^{-04}$
GO:0050808	synapse organization	$2.58 \times 10^{-04}$
GO:0010092	specification of animal organ identity	$4.13 \times 10^{-04}$
GO:0035136	forelimb morphogenesis	$9.273 \times 10^{-04}$
GO:0021545	cranial nerve development	$9.33 \times 10^{-04}$
GO:0048934	peripheral nervous system neuron differentiation	$1.03 \times 10^{-03}$
GO:0048935	peripheral nervous system neuron development	$1.03 \times 10^{-03}$
GO:0033555	multicellular organismal response to stress	$1.44 \times 10^{-03}$
GO:0021983	pituitary gland development	$1.52 \times 10^{-03}$
GO:0002920	regulation of humoral immune response	$1.53 \times 10^{-03}$

**Table 12:** Top 10 KEGG processes associated with DEGs identified by `idiffomix` and not by `limma`.

KEGG Pathways	Description	Adjusted p-value
hsa04974	Protein digestion and absorption	$1.87 \times 10^{-08}$
hsa04610	Complement and coagulation cascades	$1.42 \times 10^{-06}$
hsa03320	PPAR signaling pathway	$8.51 \times 10^{-05}$
hsa05144	Malaria	$1.14 \times 10^{-04}$
hsa04390	Hippo signaling pathway	$1.36 \times 10^{-03}$
hsa04724	Glutamatergic synapse	$2.25 \times 10^{-03}$
hsa04061	Viral protein interaction with cytokine and cytokine receptor	$2.77 \times 10^{-03}$
hsa04024	cAMP signaling pathway	$6.29 \times 10^{-03}$
hsa05323	Rheumatoid arthritis	$1.41 \times 10^{-02}$
hsa05033	Nicotine addiction	$1.48 \times 10^{-02}$

**Table 13:** Top KEGG processes associated with DEGs identified by `idiffomix` and not by `mclust`.

KEGG Pathways	Description	Adjusted p-value
hsa04724	Glutamatergic synapse	$2.25 \times 10^{-03}$
hsa05033	Nicotine addiction	$1.49 \times 10^{-02}$
hsa04080	Neuroactive ligand-receptor interaction	$1.91 \times 10^{-02}$
hsa02010	ABC transporters	$2.91 \times 10^{-02}$
hsa04713	Circadian entrainment	$3.35 \times 10^{-02}$
hsa05412	Arrhythmogenic right ventricular cardiomyopathy	$3.54 \times 10^{-02}$

## References

- Chamroukhi, F., Huynh, B. T.: Regularized maximum-likelihood estimation of mixture-of-experts for regression and clustering. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2018).
- Salter-Townshend, M., Murphy, T. B.: Variational Bayesian inference for the latent position cluster model for network data. Computational Statistics & Data Analysis **57**(1), 661–671 (2013).

## Bettina Grun

### *Material list:*

Sablica, L., Hornik, K., & Grün, B. (2025). circlus: An R Package for Circular and Spherical Clustering Using Poisson Kernel-Based and Spherical Cauchy Distributions. *Austrian Journal of Statistics*, 54(3), 27–42.

# circlus: An R Package for Circular and Spherical Clustering Using Poisson Kernel-Based and Spherical Cauchy Distributions

Lukas Sablica 

WU Wien

Kurt Hornik 

WU Wien

Bettina Grün 

WU Wien

---

## Abstract

This paper introduces **circlus**, an R package designed for clustering circular and spherical data using Poisson kernel-based (PKB) distributions and spherical Cauchy distributions. The package leverages the general framework for Expectation-Maximization (EM) estimation implemented by package **flexmix** and provides model drivers for estimating PKB and spherical Cauchy distributions in the components. The drivers implement two approaches for the M-step. The first is a direct maximization approach implemented in C++ via **Rcpp**, while the second incorporates covariates by solving the M-step using neural networks with the **torch** package. The package is particularly suited for high-dimensional clustering tasks, such as text embeddings on a spherical space, and supports models both with and without covariates. As a case study, we apply **circlus** to cluster the abstracts of papers co-authored by Fritz Leisch and demonstrate the use with and without the inclusion of co-author count as a covariate.

*Keywords:* spherical data, model-based clustering, embeddings, **flexmix**, R.

---

## 1. Introduction

Clustering is a fundamental technique in data analysis and machine learning, commonly used to uncover underlying patterns in data by grouping similar items. Traditional clustering methods, such as  $k$ -means (Macqueen 1967) and Gaussian mixture models (Dempster, Laird, and Rubin 1977), assume that data lie in Euclidean space, which works well for many applications. However, certain types of data, such as directional data, biological data, and text data, are often more appropriately modeled on spherical or circular spaces. In such cases, applying Euclidean-based methods can lead to suboptimal or misleading results.

The extension of  $k$ -means clustering to spherical data uses the cosine similarity as distance (Maitra and Ramler 2010) and an implementation for the R environment for statistical computing and graphics (R Core Team 2024) is available in package **skmeans** (Hornik, Feinerer, Kober, and Buchta 2012). Similar to how Gaussian model-based clustering is the generalization of  $k$ -means to a model-based approach (see, for example Grün 2019), mixtures of von Mises-Fisher distributions (Banerjee, Dhillon, Ghosh, and Sra 2005) have been proposed as generalization of spherical  $k$ -means and an R implementation is available in package **mvMF**

(Hornik and Grün 2014).

Model-based clustering of spherical data based on finite mixtures is provided for specific component distributions by separate R packages. E.g., package **movMF** provides fitting of finite mixtures of von Mises-Fisher distributions and package **QuadratiK** (Saraceno, Markatou, Mukhopadhyay, and Golzy 2024) covers finite mixtures of Poisson-kernel-based distributions. However, neither of these packages allows the inclusion of covariates to control for differences in the component-specific parameters in dependence of covariates. A more general implementation within the **flexmix** framework (Leisch 2004), allowing for different component distributions and the inclusion of covariates, is thus warranted. This gap is filled by **circlus** (Sablica, Hornik, Gruen, and Leydold 2024), the R package we introduce in this paper and that is freely available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=circlus>.

Building on the solid foundation provided by **flexmix**, **circlus** extends its capabilities to circular and spherical clustering by allowing to specify as component distributions the Poisson kernel-based distribution (Golzy and Markatou 2020) and the spherical Cauchy distribution (Kato and McCullagh 2020). Package **circlus** contributes new models that enable the clustering of data on the surface of a sphere. Two estimation methods for the M-step are offered for each of the two distributions: one implemented in C++ for direct and efficient calculation, and another using neural networks via the **torch** package (Falbel and Luraschi 2024), which allows for the incorporation of covariates into the clustering process. This neural network approach maps the covariate space to clustering parameters, facilitating the inclusion of additional data, such as metadata or context, in the clustering model.



The rest of this paper is organized as follows: in the next section, we define the Poisson kernel-based and spherical Cauchy distributions that underlie the models in package **circlus**. In Section 3, we discuss the strengths and advantages of clustering on the sphere compared to Euclidean methods. Section 4 introduces the **circlus** software package, detailing its architecture and implementation. This is followed by an application section, where we demonstrate package **circlus** in action by clustering abstracts written by Fritz Leisch, with and without the inclusion of co-author count as a covariate. Finally, we conclude by summarizing the key contributions of package **circlus**.

## 2. Spherical distributions for clustering

Various rotationally symmetric distributions have been developed for modeling data on the unit sphere, including the von Mises-Fisher (vMF) distribution (Khatri and Mardia 1977), Poisson kernel-based distribution (Golzy and Markatou 2020), and spherical Cauchy distribution (Kato and McCullagh 2020). In more detail, these are as follows.

**von Mises-Fisher (vMF) distribution.** Let  $S^{d-1} = \{x \in \mathbb{R}^d : \|x\| = 1\}$  represent the unit sphere in  $\mathbb{R}^d$ . A random vector  $x \in S^{d-1}$  has a von Mises-Fisher (vMF) distribution with parameters  $\kappa \geq 0$  and  $\mu \in S^{d-1}$  if its probability density function is given by

$$f_{\text{vMF}}(x|\kappa, \mu) = \frac{e^{\kappa\mu'x}}{H_{d/2-1}(\kappa)},$$

where  $H_\nu(\kappa) = {}_0F_1(\nu+1; \kappa^2/4) = \frac{\Gamma(\nu+1)}{(\kappa/2)^\nu} I_\nu(\kappa)$  with  ${}_0F_1$  and  $I_\nu$  being the confluent hypergeometric limit function (e.g., Mardia and Jupp 2009, page 352) and modified Bessel function of the first kind (DLMF 2024, Eq. 10.25.2), respectively. The vMF distribution is widely used for spherical data due to its simplicity, with a concentration parameter  $\kappa$  determining the level of clustering and  $\mu$  as the location.

However, due to the exponential decay of its density function, the vMF distribution can struggle in scenarios with outliers or broader variability, as it tends to sharply cluster data around the mean direction. This makes it less suitable for datasets that require more flexibility in capturing heavy-tailed structures. Additionally, while a closed-form expression for the normalizing constant exists, its computation can be numerically demanding and can easily overflow for large parameter values (Hornik and Grün 2014).

**Poisson kernel-based (PKB) distribution.** The Poisson kernel-based (PKB) distribution provides an alternative with better stability and computational efficiency. The PKB distribution with parameters  $0 \leq \rho < 1$  and  $\mu \in S^{d-1}$  has the following density function with respect to the uniform distribution on the unit sphere:

$$f_{\text{PKB}}(x|\rho, \mu) = \frac{1 - \rho^2}{\|x - \rho\mu\|^d}, \quad x \in S^{d-1}.$$

For  $\rho = 0$ , this distribution reduces to the uniform distribution on the sphere, and as  $\rho \rightarrow 1^-$ , it tends toward the Dirac distribution centered at  $\mu$ . The PKB distribution belongs to a family of densities of the form:

$$f(x) \propto \|x - \rho\mu\|^{-\xi}, \quad x \in S^{d-1}, \quad \xi > 0.$$

The PKB distribution arises for  $\xi = d$ , making it particularly useful for modeling spherical data. One key advantage is that the PKB distribution allows for straightforward density evaluation without the need for complex special functions, unlike the vMF or Watson distributions (Sablica and Hornik 2023).

**Spherical Cauchy distribution.** The spherical Cauchy distribution is another member of the same family of distributions. Its density is closely related to that of the PKB distribution, and the two distributions coincide when  $d = 2$ . The density of the spherical Cauchy distribution with parameters  $0 \leq \rho < 1$  and  $\mu \in S^{d-1}$  with respect to the uniform distribution on the unit sphere is given by:

$$f_{\text{Cauchy}}(x|\rho, \mu) = \left( \frac{1 - \rho^2}{\|x - \rho\mu\|^2} \right)^{d-1}, \quad x \in S^{d-1}.$$

Similar to the PKB distribution, when  $\rho = 0$ , the distribution reduces to the uniform distribution on the sphere, and as  $\rho \rightarrow 1^-$ , it tends toward the Dirac distribution centered at  $\mu$ .

The spherical Cauchy and PKB distributions have the following advantages compared to the vMF distribution: (1) They have heavier tails, making them ideal for capturing large deviations and outliers, much like the role of the Cauchy and Student- $t$  distributions in Euclidean space. (2) They are much simpler and computationally more efficient to evaluate on modern accelerators such as GPUs. Both distributions avoid the need for computing complex normalizing constants that must be sequentially evaluated, which would otherwise hinder parallel processing on GPUs. The density evaluation for both PKB and spherical Cauchy distributions essentially reduces to matrix operations, such as computing norms and matrix multiplications, which are highly optimized for GPU architectures. This allows for efficient and scalable implementation of spherical clustering, making these distributions particularly well-suited for modern, large-scale data processing tasks.

The PKB has slightly heavier tails than the spherical Cauchy, offering a good option when dealing with data containing more extreme outliers. The spherical Cauchy distribution provides a balance between the traditional von Mises-Fisher and the extremely heavy-tailed PKB,

making it suitable for data with moderate outliers or when a balance between robustness and computational efficiency is desired.

### 3. Clustering on the sphere

#### 3.1. Spherical clustering for high-dimensional data

Clustering data on the sphere has become increasingly important with the rise of high-dimensional data representations, particularly in natural language processing and machine learning. Embeddings, such as those derived from models like BERT (Devlin, Chang, Lee, and Toutanova 2019) or other transformer-based architectures, are often normalized to lie on the surface of a unit sphere. This normalization occurs because the magnitude of the embeddings, which represent the strength or scale of the data points, is irrelevant in most contexts, what matters is their direction. Clustering on the sphere allows for a better understanding of relationships between data points, as it operates in the correct geometric space, making the results more accurate and meaningful.

#### 3.2. Regression-based clustering with covariates

While spherical clustering is powerful on its own, adding the ability to incorporate covariates into the clustering model further enhances its usefulness. Covariates provide a way to control for known factors in the data that could influence clustering, allowing the model to focus on discovering more subtle or latent patterns.

In many practical applications, the data being clustered comes with associated metadata or known characteristics that can be incorporated into the model. For instance, imagine clustering the text embeddings of financial reports from various companies. Without any additional information, the clustering algorithm might primarily group companies based on their industry or market segment, which is an obvious correlation often reflected in the text of such reports.

By incorporating covariates, we can “control” for this known information and allow the model to focus on other latent patterns within the data. For example, if we input the market segment as a covariate into the clustering model, the algorithm can focus on distinguishing companies within the same market segment based on their risk exposure, financial strategy, or other factors that are not immediately obvious from surface-level industry groupings.

Another example is the clustering of patient health records, incorporating age, gender, or pre-existing conditions as covariates. This approach could allow the model to focus on discovering patterns related to treatment effectiveness, lifestyle impacts, or specific health risks that are not simply reducible to demographic categories.

More formally, this approach can be viewed within the framework of model-based clustering. In this context, we aim to maximize the likelihood of a multivariate mixture model, given by:

$$f(y_1, \dots, y_n | x_1, \dots, x_n, \pi_1, \dots, \pi_K, A_1, \dots, A_K) = \prod_{i=1}^n \sum_{j=1}^K \pi_j f_j(y_i | A_j, x_i),$$

where  $y_i \in S^{d-1}$  is the observed spherical response and  $x_i \in \mathbb{R}^m$  the covariate vector for observation  $i$ ,  $\pi_j$  is the prior probability of belonging to cluster  $j$  (with  $\sum_{j=1}^K \pi_j = 1$ ),  $f_j(y_i | A_j, x_i)$  is the spherical density function for cluster  $j$ , and  $K$  is the number of clusters. In our case,  $f_j$  could be the PKB distribution or the spherical Cauchy distribution.

To incorporate covariates into this framework, we link the parameters of the spherical distribution,  $\mu$  (location) and  $\rho$  (concentration), to the covariates  $x_i$  through a cluster-specific linear map represented by the matrix  $A_j \in \mathbb{R}^{m \times d}$ . This mapping is given by:

$$\theta_{j,i} = A'_j x_i,$$

where  $\theta_{j,i} \in \mathbb{R}^d$  is an unrestricted parameter vector characterizing the spherical distribution for cluster  $j$ . The matrices  $A_j$  contain the learnable parameters and can be estimated using optimization techniques such as those employed in neural networks.

Since  $\mu$  must lie on the unit sphere and  $\rho$  must remain positive and bounded, a direct linear mapping is insufficient. For that reason we then link  $\theta_{j,i}$  to the parameters  $\mu$  and  $\rho$  through suitable transformations

$$\mu(\theta_{j,i}) = \frac{\theta_{j,i}}{\|\theta_{j,i}\|}, \quad \text{and} \quad \rho(\theta_{j,i}) = \frac{\|\theta_{j,i}\|}{1 + \|\theta_{j,i}\|}.$$

This mapping provides a 1-to-1 correspondence between  $\mathbb{R}^d$  and the set of parameters  $(\mu, \rho)$ , ensuring that  $\mu$  is always a unit vector and that  $\rho$  remains within the valid range  $(0, 1)$ , thereby guaranteeing well-defined model parameters. While this parametrization does not allow for completely independent modeling of location and concentration, in our experiments, it has proven general enough to provide robust estimates while offering the advantages of computational speed and simplicity. More flexible parametrizations, allowing for more nuanced relationships between covariates and concentration, could be explored in future research.

This model-based clustering framework, with the inclusion of covariates, allows us to group together observations that exhibit similar relationships between covariates and the parameters of the spherical response distribution. Essentially, we are clustering observations based on the similarity of their covariate-response mappings.

In the case of only one mixture component ( $K = 1$ ), this simplifies to a simple regression task, as we are effectively estimating a single mapping from covariates to the response distribution. Conversely, with only an intercept as a covariate, we recover standard clustering of the responses, as the model focuses solely on grouping similar responses without considering any additional covariate information. The mixture likelihood can be estimated using various methods, including the Expectation-Maximization (EM) algorithm (Dempster *et al.* 1977).

## 4. Software

The **circlus** package extends the flexibility of the EM framework implemented in package **flexmix** to handle spherical clustering using Poisson kernel-based and spherical Cauchy distributions. The package implements four M-step drivers, which provide the two distributions (PKB and spherical Cauchy distribution), with and without covariates. Each model is designed to integrate seamlessly into the **flexmix** implementation, providing methods for maximum likelihood estimation through the EM algorithm (Dempster *et al.* 1977). Below, we introduce the key functions for clustering and explain their functionality and parameters.

### 4.1. Clustering with PKB distributions

The **circlus** package provides two main functions for performing the M-step of clustering using the Poisson kernel-based distribution:

```
FLXMCpkb(formula = .~.)
```

and

```
FLXMRpkb(formula = .~., EPOCHS = 100, LR = 0.1,
           max_iter = 200, adam_iter = 5, free_iter = adam_iter,
           line_search_fn = "strong_wolfe")
```

The first function, **FLXMCpkb**, uses C++ code to perform the M-step, making it highly efficient and well-suited for tasks where speed is crucial. The “C” in **FLXMCpkb** stands for “clustering”, indicating that this function focuses purely on clustering without the inclusion of covariates. The second function, **FLXMRpkb**, leverages neural networks through the **torch** package to incorporate covariates into the estimation process, with the “R” standing for “regression”, signifying the function’s ability to handle covariates in the clustering model.

The **FLXMCpkb** function performs the M-step using a direct C++ implementation via **Rcpp** (Eddelbuettel and François 2011) based on Golzy and Markatou (2020), which is designed to handle the estimation process efficiently without the need for any additional optimization frameworks. This approach excels in both speed and simplicity, making it ideal for scenarios where covariates are not needed. We note that this algorithm has also been implemented in pure R within the **QuadratiK** package (Saraceno *et al.* 2024), which offers a robust suite of methods for working with spherical data, including tests for multivariate normality, tests for uniformity on the sphere, and clustering algorithms, among other valuable tools. To extend the possible applications, particularly for models that incorporate covariates, enhance performance through C++, and leverage the wide range of functions offered by the **flexmix** framework, we developed **circlus**. By building on the strengths of existing tools, **circlus** offers users additional flexibility and scalability, particularly for larger datasets and more complex clustering tasks involving covariates.

The **FLXMRpkb** function, on the other hand, uses a neural network to perform the M-step. Function **FLXMRpkb** can be used with or without inclusion of covariates into the clustering model. More specifically, in case covariates are included, the algorithm maps the covariate space to the space of response variables using a simple linear transformation network without a bias term and links the mapped vector to parameters  $\mu$  and  $\rho$  exactly as discussed in Section 3.2.

The optimization process in **FLXMRpkb** starts with the robust Adam optimizer (Kingma and Ba 2014) and resets the weights of the neural network at every iteration to prevent local minima and ensure robustness in the early stages of training. After an initial phase controlled by the **adam\_iter** parameter, the algorithm switches to the quasi-Newton L-BFGS method, which is better suited for fast convergence in the later stages of optimization. The number of epochs for the Adam optimizer and the maximum iterations for L-BFGS are controlled by the **EPOCHS** and **max\_iter** parameters, respectively. The learning rate for both optimizers is set by the **LR** parameter. Additionally, the **line\_search\_fn** parameter specifies the line search function used in the L-BFGS optimizer, with the "strong\_wolfe" method being the default. For more details on this parameter, see the documentation of package **torch**.

## 4.2. Clustering with spherical Cauchy distributions

The **circlus** package also provides two key functions for clustering based on the spherical Cauchy distribution:

```
FLXMCspcauchy(formula = . ~ .)
```

and

```
FLXMRspcauchy(formula = . ~ ., EPOCHS = 100, LR = 0.1,
                max_iter = 200, adam_iter = 5, free_iter = adam_iter,
                line_search_fn = "strong_wolfe")
```

These functions follow a similar design to the PKB distribution functions, leveraging the **flexmix** framework to perform model-based clustering with and without covariates.

The **FLXMCspcauchy** function provides a direct, fast, and efficient solution for spherical Cauchy clustering without covariates, utilizing C++ for the M-step. It uses the method of Algorithm 4.1 in Kato and McCullagh (2020). This algorithm is extended to cover the mixture model

case by weighting the sum by the posterior probabilities of the data points belonging to each cluster, with the weights normalized to sum to 1 for each cluster. The initial  $\psi_0$  for the algorithm is estimated using the method of moments, as outlined in Subsection 4.1 of the reference, where  $\bar{Y}$  in Equation 4.1 is also weighted according to the posterior probabilities. In addition, **FLXMRspcauchy** incorporates covariates into the clustering process using a neural network based on the **torch** package. The neural network maps covariates to the parameters of the spherical Cauchy distribution in the same manner as the PKB distribution model.

To our knowledge, there are currently no other implementations available that provide spherical Cauchy-based clustering with or without covariates, making **circlus** the first package to offer this capability within the **flexmix** framework.

### 4.3. Simulation methods

In addition to the clustering functions, the **circlus** package also provides random sampling methods for both the Poisson kernel-based and spherical Cauchy distributions. These random sampling methods are valuable because they allow for further analysis of clusters through techniques like the parametric bootstrap, where one can assess the variability or stability of the identified clusters by resampling from the fitted model. For instance, in applications such as text analysis or financial modeling, simulated data can be used to validate model performance or to test the sensitivity of clustering results. See for example McLachlan (1987) and O'Hagan, Murphy, Scrucca, and Gormley (2019).

Function `rpkb(n, rho, mu, method = "ACG")` generates random samples from the PKB distribution. The user can specify the number of random draws `n` and the desired parameters `rho` (the concentration) and `mu` (the location). The `method` argument allows the user to choose between two sampling approaches: the first uses the Angular Central Gaussian (ACG) distribution as the envelope in a rejection sampling scheme, while the second method is based on the projected Saw distribution. Both methods are efficient and follow the approach described in Sablica, Hornik, and Leydold (2023). The ability to switch between these methods offers flexibility depending on the specific use case or computational requirements. We note that PKB random sample generation is also available in the packages **QuadratiK** and **Directional** (Tsagris, Athineou, Adam, and Yu 2024).

For the spherical Cauchy distribution, function `rspcauchy(n, rho, mu)` provides a direct method for generating random samples. The number of samples `n`, concentration `rho`, and location `mu` can be specified. This method is based on the Möbius transformation of uniform samples on the sphere, as detailed in Kato and McCullagh (2020).

## 5. Case study: Clustering Fritz Leisch's work

Friedrich “Fritz” Leisch was a highly respected figure in the field of statistical computing, known for his broad contributions that spanned multiple domains. His work on flexible clustering models has had a lasting impact on the world of data analysis. Leisch’s research was not only theoretically innovative but also highly practical, enabling users across disciplines to apply sophisticated clustering techniques to real-world problems.

Throughout his prolific career, Leisch was involved in a wide range of research topics, contributing to areas such as benchmarking, computational statistics, and reproducible research. His work was characterized by a collaborative spirit, with nearly 300 unique co-authors, reflecting his strong belief in interdisciplinary research and the importance of working with others to advance science. Leisch’s collaborations and innovations have helped shape modern statistical methodology, making his work essential reading for statisticians and data scientists alike.

For this analysis, we compiled a dataset of 129 abstracts from the works of Fritz Leisch, which were verified through the Crossref API (CrossRef 2024). Alongside the abstracts, we

collected important metadata, including the number of pages, digital object identifier (DOI), journal name, names of the co-authors, and the year of publication. The dataset is made available as `Abstracts` within the *circlus* package. Co-author information has been encoded using 272 dummy variables, where each co-author is represented as a binary variable. To numerically represent the textual data, we transformed the abstracts into embeddings using four different models. The first method employed the `gte-large-en-v1.5` embedding model from Alibaba (Zhang, Zhang, Long, Xie, Dai, Tang, Lin, Yang, Xie, Huang *et al.* 2024), which produced embeddings with 1024 dimensions. The remaining three methods used OpenAI's `text-embedding-3-large` model (OpenAI 2024), with output dimensions of 3072, 512, and 256, respectively. These embeddings outputs are available as the last four columns of the `Abstracts` dataset. Overall this results in a dataset with 129 rows and 283 columns, offering a comprehensive view of Leisch's collaborations across different publications.

Given the relatively small number of abstracts and their thematic similarity, we selected the 256-dimensional embeddings for our analysis. We found that this dimensionality was sufficient to capture the essential semantic relationships between the abstracts without overcomplicating the clustering process. In general, the choice of dimensionality should be guided by a combination of factors such as sample size, data complexity, and computational constraints. For larger or more complex datasets, higher-dimensional embeddings might be necessary, but it is important to balance the need for detail with the risk of overfitting and increased computational burden. It is recommended to explore different dimensionalities and embedding models to find the best fit for the specific data and task.

### 5.1. Mixtures of distributions

In the first stage of our analysis, we clustered the dataset without incorporating any covariates. Assuming that the word usage distribution differs by research area, this approach aims to cluster the abstracts such that the clusters correspond to research areas. The estimation of the mixture model was carried out using the default parameters of `flexmix` for the EM algorithm. This implies that the EM algorithm is randomly initialized by assigning a-posteriori probabilities to observations and then continuing with an M-step. When specifying only the number of clusters, observations are randomly assigned to clusters with equal probability and weights of 0.9 assigned to these clusters and weights of 0.1 for the remaining ones. In addition the parameter `minprior` is set to 0.05. This parameter ensures that any cluster representing less than 5% of the data is eliminated from the estimation process during the EM iterations. This provides a more stable estimation avoiding estimation issues in the M-step in case of very small components which could result in degenerate solutions where only observations with identical values have positive weights for this component. In addition, a minimum cluster size is usually of interest for an interpretable solution. In our experiments where we fitted the spherical cluster model with different, higher numbers of components, the model consistently reduced to eight clusters during the EM algorithm. We adopted this as the final number of clusters for our analysis. This number of clusters provided a meaningful balance between interpretability and capturing the diversity of research topics within the dataset.

We chose the spherical Cauchy distribution for this case study, as the even heavier tails of the PKB distribution were not necessary for data that is relatively similar. All abstracts represent scientific contributions in neighboring disciplines. The clustering was performed using both available models in the *circlus* package: `FLXMCspcauchy`, which leverages the direct estimation, and `FLXMRspcauchy`, which incorporates a neural network to estimate the M-step. The code used for this analysis is shown below, with the clustering results following.

First we loaded the necessary packages and the dataset as well as extracted the embedding obtained for OpenAI with dimension 256:

```
R> library("flexmix")
R> library("circlus")
```

```
R> data("Abstracts", package = "circlus")
R> OAI256 <- do.call(rbind, Abstracts[, "OpenAI_embeddings256"])
```

We applied the `FLXMCspcauchy()` model:

```
R> set.seed(1)
R> (SC_abstract_8 <- flexmix(OAI256 ~ 1, k = 8,
+     model = FLXMCspcauchy()))
Call:
flexmix(formula = OAI256 ~ 1, k = 8, model = FLXMCspcauchy())

Cluster sizes:
 1 2 3 4 5 6 7 8
15 9 16 19 9 16 19 26

convergence after 16 iterations
```

The default `print()` method for objects fitted with `flexmix()` indicates how the object was created by showing the call as well as the cluster sizes of the partition obtained by assigning observations to the cluster where their a-posteriori probability is maximum. The output shows that the clusters vary in size and have between 9 and 26 abstracts assigned.

Next, we used the neural network-based model for comparison:

```
R> set.seed(1)
R> torch::torch_manual_seed(1)
R> (SCNN_abstract_8 <- flexmix(OAI256 ~ 1, k = 8,
+     model = FLXMRspcauchy(LR = 0.02, adam_iter = 0, free_iter = 5)))
Call:
flexmix(formula = OAI256 ~ 1, k = 8, model = FLXMRspcauchy(LR = 0.02,
adaman_iter = 0, free_iter = 5))

Cluster sizes:
 1 2 3 4 5 6 7 8
15 9 16 19 9 16 19 26

convergence after 18 iterations
```

Both models resulted in identical cluster allocations, with each cluster containing the same number of abstracts. Setting the random seed to the same value ensures that the EM algorithm is initialized using the same a-posteriori probabilities for the first M-step. Hence, only the optimization step in the M-step differs for these model fits. This congruence in results shows that the M-step implementations are robust and produce consistent clustering outcomes, even though they rely on different optimization techniques. While it is common for estimates obtained with different optimization algorithms to yield slight different results, in this case, the resulting cluster allocation sizes were fully aligned, highlighting the reliability of the estimation for this particular dataset.

In terms of log-likelihood, `FLXMCspcauchy` achieved a final log-likelihood of 12674.6, while `FLXMRspcauchy` reached 12674.61, confirming that both methods converged to virtually identical solutions.

We analyzed the content of the abstracts within each cluster to verify that indeed spherical clustering identified eight distinct clusters that represent different areas of Fritz Leisch's research. Based on this inspection of the content, we classified the clusters and assigned titles as shown in Table 1 which capture the primary scientific focus of each group.

Table 1: Cluster titles based on abstract content

Cluster No.	Cluster Title
1	Genetic Influences on Psychiatric and Behavioral Disorders
2	Advancements in Model Validation and Benchmarking Techniques
3	Travel Behavior and Environmental Impact in Transportation and Tourism
4	Environmental and Biological Effects of Agrochemicals
5	Market Segmentation Techniques and Applications
6	Biopharmaceutical Production Through Data-Driven Approaches
7	Finite Mixture Models and Their Applications
8	Clustering Techniques in Data Analysis



Figure 1: Word cloud visualization of the most frequently occurring terms (left) and co-author networks (right) across the clusters

Inspecting the concentration parameters of the clusters indicates how compact or spread out the identified clusters are. Clusters 4 and 8 have the smallest concentration parameter values, with  $\rho = 0.346$  for Cluster 4 and  $\rho = 0.397$  for Cluster 8, indicating that they represent the least compact clusters. These clusters include observations that are not easily assigned to the more specialized clusters, serving as broader, less-defined groups that capture data points with weaker associations to the other, more focused clusters. Cluster 8 (titled “Clustering Techniques in Data Analysis”), in particular, acts as a background cluster for the more data-driven clusters such as Clusters 2, 6, and 7, as can be seen by comparing the cosine distances between the location parameters  $\mu$  of the individual clusters.

To further visualize the content of these clusters, we calculated a term frequency matrix for the dataset as a whole with respect to the individual clusters. This enabled us to identify the most frequently occurring terms within each cluster. Using the **wordcloud** package (Fellows 2018), we created visual representations of the key terms for each cluster (see the left sub-image of Figure 1).

In addition to the thematic analysis, we aimed to highlight the extensive network of collaborations that Fritz Leisch fostered throughout his career. For each of the eight clusters, we generated a co-author frequency matrix, which quantified the presence of each co-author within the publications assigned to that cluster. Using the `comparison.cloud()` function from the `wordcloud` package, we visualized the co-author networks for each cluster in the right sub-image of Figure 1, showcasing the diverse and widespread collaborations across different fields of research.

One key insight from examining the dataset is the variation in the number of co-authors across

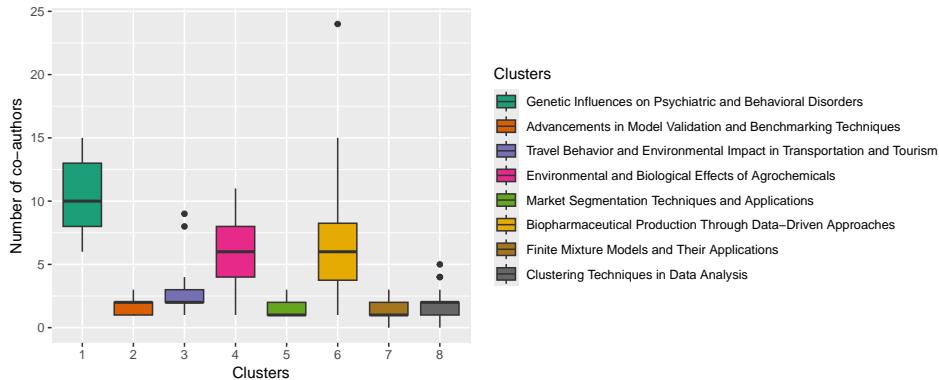


Figure 2: Boxplot of the number of co-authors across clusters

different scientific disciplines. Certain fields tend to have more co-authors per publication, reflecting the collaborative nature of these research areas. This variation is also evident in the clusters identified which correspond to different research areas where Fritz Leisch has contributed to. Figure 2 visualizes the number of co-authors for each of the eight clusters in a parallel boxplot.

### 5.2. Mixtures with covariates

As illustrated in Figure 2, clusters corresponding to genetics, biology and biopharmaceutics generally exhibit a higher number of co-authors compared to other clusters. This is valuable information that can be explicitly included in our clustering model as a covariate, allowing the model to account for the number of co-authors while focusing more on other abstract features such as the content and semantic relationships within the embeddings.

To incorporate this into our analysis, we re-estimated the model using the number of co-authors as a covariate and the spherical Cauchy distribution as the component distribution. The following code shows the estimation process:

```
R> (SCNN_abstract_8b <- flexmix(OAI256 ~ 1 + num_of_coauthors, k = 8,
+   model = FLXMRspcauchy(EPOCHS = 200, LR = 0.02, adam_iter = 10)))

Call:
flexmix(formula = OAI256 ~ 1 + num_of_coauthors, k = 8, model =
FLXMRspcauchy(EPOCHS = 200, LR = 0.02, adam_iter = 10))

Cluster sizes:
 1  2  3  4  5  6 
15 30 27 15 19 23 

convergence after 28 iterations
```

As seen from the output, while we initially started with eight clusters, the automatic removal of clusters with a small component size during the iterative procedure of the EM algorithm reduced the number of clusters to six clusters. Our experiments revealed that incorporating the number of co-authors as a covariate often leads to fewer clusters, as the model uses the covariate information to account for variation between abstracts within a cluster depending on the number of co-authors. In this case, the achieved log-likelihood was 14192.16, which indicates that the model found a better fit with six clusters, compared to the previous clustering solution that used eight clusters without covariates.

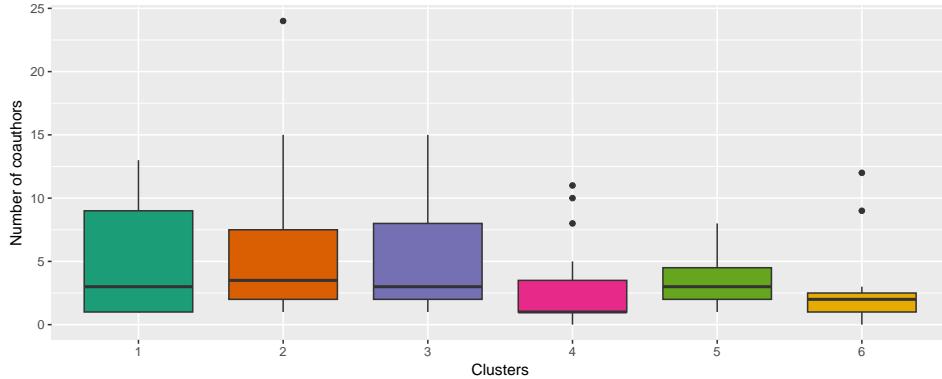


Figure 3: Boxplot of the number of co-authors in the cluster solution with covariates

By including the number of co-authors, the model successfully captures the variation in the number of collaborators across different research areas, without needing to rediscover this pattern. This allows the model to focus on other important relationships in the data, such as thematic content or research methodologies. To illustrate this further, we can once again plot the number of co-authors across the newly formed six clusters using a parallel boxplot. As shown in Figure 3, the number of co-authors across the six clusters has become more uniform, which demonstrates that the covariate was successfully controlled for in the model. This approach shows the power of incorporating known metadata into clustering models to reduce the number of clusters and account for within-cluster heterogeneity due to this covariate in an efficient way.

When we compare the clustering results before and after including the number of co-authors as a covariate, we can observe notable shifts in how the abstracts are assigned to clusters. The addition of this covariate introduces a significant factor in determining cluster membership, leading to observable movements of certain abstracts between clusters. The table below shows the comparison, where the rows represent the assignments with covariate and the columns represent the cluster assignments without covariate:

```
R> table(with_num_of_coauthors = clusters(SCNN_abstract_8b),
+       without_num_of_coauthors = clusters(SC_abstract_8))

           without_num_of_coauthors
with_num_of_coauthors 1 2 3 4 5 6 7 8
1                   6 0 0 2 7 0 0 0
2                   0 0 16 11 0 3 0 0
3                   9 9 0 2 0 1 0 6
4                   0 0 0 4 2 0 0 9
5                   0 0 0 0 0 10 0 9
6                   0 0 0 0 0 2 19 2
```

We observe that Clusters 2, 3, 5, and 7 of the clustering solution without covariates exhibit little to no movement (with respective changes of 0, 0, 2, and 0 abstracts). In contrast, the other clusters get moved considerably. In particular, Cluster 8, which had previously been identified as the background cluster for the more statistically oriented clusters, shows the strongest decomposition when the number of co-authors is included. This is likely due to the catch-all nature of this cluster in the solution without covariates, which captured abstracts that did not fit neatly into more specific research categories. Cluster 4, which has an even smaller concentration parameter than Cluster 8, also shows significant decomposition, further underscoring its role as one of the least compact clusters in the clustering without covariates.

By incorporating co-author information, we decompose these background clusters, leading to a clearer separation of abstracts and improving the overall clustering model.

One of the strengths of the **flexmix** package is the suite of high-level functions it offers for investigating clustering results, such as the **summary** and **plot** methods for objects returned by **flexmix**. These methods focus on providing valuable insights into the quality of the clustering, i.e., indicate how close the a-posteriori probabilities are to a 0-1 distribution. The **summary** function shows key metrics like the prior probabilities (the overall proportion of data belonging to each cluster), the number of assigned observations per cluster, and the ratio of assigned observations to those with a positive posterior probability (by default using a threshold of `eps = 1e-04`). This ratio is particularly useful for understanding how well-separated each cluster is from the others. A high ratio (close to 1) indicates that most assigned observations have a strong affinity for their assigned cluster and are unlikely to belong to other clusters. A lower ratio suggests some observations might have comparable probabilities for multiple clusters, indicating potential overlap.

```
R> summary(SCNN_abstract_8b)

Call:
flexmix(formula = OAI256 ~ 1 + num_of_coauthors, k = 8,
model = FLXMRspcauchy(EPOCHS = 200, LR = 0.02, adam_iter = 10))

      prior size post>0 ratio
Comp.1 0.116   15     15 1.000
Comp.2 0.233   30     34 0.882
Comp.3 0.209   27     29 0.931
Comp.4 0.116   15     15 1.000
Comp.5 0.147   19     19 1.000
Comp.6 0.178   23     23 1.000

'log Lik.' 14192.16 (df=3077)
AIC: -22230.32   BIC: -13430.67
```

In our results, four clusters (1, 4, 5, and 6) are perfectly separated with a ratio of 1.000, while Clusters 2 and 3 show some overlap, with ratios of 0.882 and 0.931 respectively. This indicates that there are observations which have some posterior probability to be from Clusters 2 and 3 but are eventually assigned to a different cluster. But overall, the clusters are well separated, reflecting a strong clustering fit with a clear assignment of observations to clusters.

## 6. Conclusion

In this paper, we introduced the **circlus** package, which extends the EM framework implemented by package **flexmix** to perform spherical clustering using the Poisson kernel-based and spherical Cauchy distributions. By providing efficient C++ implementations and flexible neural network-based models, package **circlus** allows for clustering on the sphere, with and without covariates. The inclusion of covariate-based models adds a significant advantage, enabling users to account for known metadata and focus the clustering process on uncovering deeper relationships within the data.

Our case study of Fritz Leisch's published works demonstrates the practical application of spherical clustering, highlighting how different research areas can be clustered based on their abstract embeddings. By incorporating metadata, such as the number of co-authors, we further showed how covariates can enhance the clustering model's interpretability and accuracy. This approach allows the model to focus on more nuanced aspects of the data, leading to alternative clustering solutions.

The results of our analysis reveal that spherical clustering, combined with covariate information, offers a powerful tool for handling high-dimensional and complex datasets, such as text embeddings. The flexibility of the **circlus** package makes it suitable for a wide range of applications, from natural language processing to biological and social sciences, where data naturally lie on a sphere.

Overall, the **circlus** package builds upon the legacy of Fritz Leisch's contributions to statistical computing, offering modern and scalable tools for model-based clustering on spherical spaces. We hope that this work will continue to support research in these areas and inspire further advancements in clustering methodology and statistical modeling.

## References

- Banerjee A, Dhillon IS, Ghosh J, Sra S (2005). “Clustering on the Unit Hypersphere Using von Mises-Fisher Distributions.” *Journal of Machine Learning Research*, **6**(September), 1345–1382. URL <https://jmlr.csail.mit.edu/papers/v6/banerjee05a.html>.
- CrossRef (2024). “CrossRef REST API.” Available at <https://api.crossref.org>.
- Dempster AP, Laird NM, Rubin DB (1977). “Maximum Likelihood from Incomplete Data via the EM Algorithm.” *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1), 1–22. doi:[10.1111/j.2517-6161.1977.tb01600.x](https://doi.org/10.1111/j.2517-6161.1977.tb01600.x).
- Devlin J, Chang MW, Lee K, Toutanova K (2019). “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” In *Proceedings of NAACL-HLT 2019*, pp. 4171–4186. URL <https://aclanthology.org/N19-1423.pdf>.
- DLMF (2024). “*NIST Digital Library of Mathematical Functions*.” Release 1.0.19 of 2018-06-22. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds., URL <https://dlmf.nist.gov/>.
- Eddelbuettel D, François R (2011). “**Rcpp**: Seamless R and C++ Integration.” *Journal of Statistical Software*, **40**(8), 1–18. doi:[10.18637/jss.v040.i08](https://doi.org/10.18637/jss.v040.i08).
- Falbel D, Luraschi J (2024). **torch**: Tensors and Neural Networks with GPU Acceleration. R package version 0.13.0, URL <https://torch.mlverse.org/docs>.
- Fellows I (2018). **wordcloud**: Word Clouds. R package version 2.6, URL <https://CRAN.R-project.org/package=wordcloud>.
- Golzy M, Markatou M (2020). “Poisson Kernel-Based Clustering on the Sphere: Convergence Properties, Identifiability, and a Method of Sampling.” *Journal of Computational and Graphical Statistics*, **29**(4), 758–770. doi:[10.1080/10618600.2020.1740713](https://doi.org/10.1080/10618600.2020.1740713).
- Grün B (2019). “Model-based Clustering.” In S Frühwirth-Schnatter, G Celeux, CP Robert (eds.), *Handbook of Mixture Analysis*, Chapman & Hall/CRC Handbooks of Modern Statistical Methods, pp. 157–192. Chapman and Hall/CRC.
- Hornik K, Feinerer I, Kober M, Buchta C (2012). “Spherical k-Means Clustering.” *Journal of Statistical Software*, **50**(10), 1–22. doi:[10.18637/jss.v050.i10](https://doi.org/10.18637/jss.v050.i10).
- Hornik K, Grün B (2014). “**movMF**: An R Package for Fitting Mixtures of Von Mises-Fisher Distributions.” *Journal of Statistical Software*, **58**(10), 1–31. doi:[10.18637/jss.v058.i10](https://doi.org/10.18637/jss.v058.i10).
- Kato S, McCullagh P (2020). “Some Properties of a Cauchy Family on the Sphere Derived from the Möbius Transformations.” *Bernoulli*, **26**(4). doi:[10.3150/20-bej1222](https://doi.org/10.3150/20-bej1222).

- Khatri CG, Mardia KV (1977). “The von Mises-Fisher Matrix Distribution in Orientation Statistics.” *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1), 95–106. [doi:10.1111/j.2517-6161.1977.tb01610.x](https://doi.org/10.1111/j.2517-6161.1977.tb01610.x).
- Kingma DP, Ba J (2014). “Adam: A Method for Stochastic Optimization.” [doi:10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980). ArXiv Preprint arXiv:1412.6980.
- Leisch F (2004). “FlexMix: A General Framework for Finite Mixture Models and Latent Class Regression in R.” *Journal of Statistical Software*, **11**(8), 1–18. [doi:10.18637/jss.v011.i08](https://doi.org/10.18637/jss.v011.i08).
- Macqueen J (1967). “Some Methods for Classification and Analysis of Multivariate Observations.” In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press.
- Maitra R, Ramler IP (2010). “A  $k$ -Mean-Directions Algorithm for Fast Clustering of Data on the Sphere.” *Journal of Computational and Graphical Statistics*, **19**(2), 377–396. [doi:10.1198/jcgs.2009.08054](https://doi.org/10.1198/jcgs.2009.08054).
- Mardia KV, Jupp PE (2009). *Directional Statistics*, volume 494. John Wiley & Sons.
- McLachlan GJ (1987). “On Bootstrapping the Likelihood Ratio Test Statistic for the Number of Components in a Normal Mixture.” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **36**(3), 318–324. [doi:10.2307/2347790](https://doi.org/10.2307/2347790).
- OpenAI (2024). “Text-Embedding-3-Large Model.” URL <https://openai.com/index-new-embedding-models-and-api-updates/>.
- O’Hagan A, Murphy TB, Scrucca L, Gormley IC (2019). “Investigation of Parameter Uncertainty in Clustering Using a Gaussian Mixture Model via Jackknife, Bootstrap and Weighted Likelihood Bootstrap.” *Computational Statistics*, **34**(4), 1779–1813. [doi:10.1007/s00180-019-00897-9](https://doi.org/10.1007/s00180-019-00897-9).
- R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Sablica L, Hornik K (2023). “Family of Integrable Bounds for the Logarithmic Derivative of Kummer’s Functions.” *Journal of Mathematical Analysis and Applications*, **537**(1), 128262. [doi:10.1016/j.jmaa.2024.128262](https://doi.org/10.1016/j.jmaa.2024.128262).
- Sablica L, Hornik K, Gruen B, Leydold J (2024). *circlus: Clustering and Simulation of Spherical Cauchy and PKBD Models*. R package version 0.0.1, URL <https://CRAN.R-project.org/package=circlus>.
- Sablica L, Hornik K, Leydold J (2023). “Efficient Sampling from the PKBD Distribution.” *Electronic Journal of Statistics*, **17**(2), 2180–2209. [doi:10.1214/23-ejs2149](https://doi.org/10.1214/23-ejs2149).
- Saraceno G, Markatou M, Mukhopadhyay R, Golzy M (2024). “Goodness-of-Fit and Clustering of Spherical Data: The **QuadratiK** Package in R and Python.” [doi:10.48550/arXiv.2402.02290](https://doi.org/10.48550/arXiv.2402.02290). ArXiv Preprint arXiv:2402.02290.
- Tsagris M, Athineou G, Adam C, Yu Z (2024). *Directional: A Collection of Functions for Directional Data Analysis*. R package version 7.0, URL <https://CRAN.R-project.org/package=Directional>.
- Zhang X, Zhang Y, Long D, Xie W, Dai Z, Tang J, Lin H, Yang B, Xie P, Huang F, et al. (2024). “mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval.” [doi:10.48550/arXiv.2407.19669](https://doi.org/10.48550/arXiv.2407.19669). ArXiv Preprint arXiv:2407.19669.

**Affiliation:**

Lukas Sablica  
Institute for Statistics and Mathematics  
Vienna University of Economics and Business  
Welthandelsplatz 1, 1020 Wien  
Telephone: +43 1 31336-5058  
E-mail: [lukas.sablica@wu.ac.at](mailto:lukas.sablica@wu.ac.at)

Kurt Hornik  
Institute for Statistics and Mathematics  
Vienna University of Economics and Business  
Welthandelsplatz 1, 1020 Wien  
Telephone: +43 1 31336-4756  
E-mail: [kurt.hornik@wu.ac.at](mailto:kurt.hornik@wu.ac.at)

Bettina Grün  
Institute for Statistics and Mathematics  
Vienna University of Economics and Business  
Welthandelsplatz 1, 1020 Wien  
Telephone: +43 1 31336-5286  
E-mail: [bettina.gruen@wu.ac.at](mailto:bettina.gruen@wu.ac.at)

## Brendan Murphy

### *Material list:*

Jacques J. and Murphy T.B. (2025) Model-based clustering and variable selection for multivariate count data, *Computo*.



# Model-Based Clustering and Variable Selection for Multivariate Count Data

COMPUTO

ISSN 2824-7795

Julien Jacques<sup>1</sup> Université Lumière Lyon 2, Université Claude Bernard Lyon 1, ERIC, 69007, Lyon,  
France

Thomas Brendan Murphy School of Mathematics & Statistics, University College Dublin  
Institut d'Études Avancées, Université de Lyon

Date published: 2025-07-01 Last modified: 2025-07-01

## Abstract

Model-based clustering provides a principled way of developing clustering methods. We develop a new model-based clustering methods for count data. The method combines clustering and variable selection for improved clustering. The method is based on conditionally independent Poisson mixture models and Poisson generalized linear models. The method is demonstrated on simulated data and data from an ultra running race, where the method yields excellent clustering and variable selection performance.

*Keywords:* Count data, Model-based clustering, Variable selection

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Motivating Example</b>	<b>2</b>
<b>3</b>	<b>Independent Poisson Mixture</b>	<b>2</b>
<b>4</b>	<b>Variable selection</b>	<b>4</b>
4.1	Model setup . . . . .	4
4.2	Interpretation . . . . .	5
4.3	Stepwise selection algorithm . . . . .	6
4.3.1	Screening variables: Initialization . . . . .	6
4.3.2	Stepwise algorithm: Updating . . . . .	6
<b>5</b>	<b>Simulation study</b>	<b>7</b>
5.1	Illustrative example . . . . .	7
5.2	Scenarios of simulation . . . . .	9
5.3	Results . . . . .	9
<b>6</b>	<b>International Ultrarunning Association Data</b>	<b>10</b>
<b>7</b>	<b>Discussion</b>	<b>13</b>
<b>8</b>	<b>Acknowledgements</b>	<b>14</b>

<sup>1</sup>Corresponding author: [julien.jacques@univ-lyon2.fr](mailto:julien.jacques@univ-lyon2.fr)

## 1 Introduction

Multivariate count data is ubiquitous in statistical applications, as ecology (Chiquet, Mariadassou, and Robin 2021), genomics (Rau et al. 2015; Silva et al. 2019). These data arise when each observation consists of a vector of count values. Count data are often treated as continuous data and therefore modeled by a Gaussian distribution, this assumption is particularly poor when the measured counts are low. Instead, we use the reference distribution for count data which is the Poisson distribution (Agresti 2013; Inouye et al. 2017).

When a data set is heterogeneous, clustering allows to extract homogeneous subsets from the whole data set. Many clustering methods, such as  $k$ -means (Hartigan and Wong 1979), are geometric in nature, whereas many modern clustering approaches are based on probabilistic models. In this work, we use model-based clustering which has been developed for many types of data (Bouveyron et al. 2019; McLachlan and Peel 2000; Frühwirth-Schnatter, Celeux, and Robert 2018).

Modern data are often high-dimensional, that is the number of variables is often large. Among these variables, some are useful for the task of interest, some are useless for the task of interest and some others are useful but redundant. There is a need to select only the relevant variables, and that, whatever is the task. Variable selection methods are widespread for supervised learning tasks, in particular to avoid overfitting. However, variable selection methods are less well developed for unsupervised learning tasks, such as clustering. Recently, several methods have been proposed for selecting the relevant variables in model-based clustering; we refer to Fop and Murphy (2018) and McParland and Murphy (2018) for recent detailed surveys.

The goal of the present work is to provide a clustering and variable selection method for multivariate count data, which, to the best of our knowledge, has not yet been studied in depth. A methodology based on a conditionally independent Poisson mixture is developed to achieve this goal. The method yields a final clustering model which is a conditionally independent Poisson mixture model for a subset of the variables.

## 2 Motivating Example

The International Association of Ultrarunners (IAU) 24 hour World Championships were held in Katowice, Poland from September 8th to 9th, 2012. Two hundred and sixty athletes representing twenty four countries entered the race, which was held on a course consisting of a 1.554 km looped route. An update of the number of laps covered by each athlete was recorded approximately every hour (White and Murphy 2016). Figure 1 plots the number of loops recorded each hour for the three medalists.

We can see among these three runners different strategies, the second placed runner lapped at a regular rate, the first placed runner had a fast start but slowed later, and the third placed runner also started fast but slowed more than the first place runner.

Our first goal will be, to analyze the whole data set to identify the different running strategies and to evaluate which strategies are the best ones. The second goal is to identify which variables allows to distinguish between the clusters, in order to identify which hour is essential in the management of this endurance race.

## 3 Independent Poisson Mixture

Let  $X_n = (X_{n1}, X_{n2}, \dots, X_{nM})$  be a random vector of counts for  $n = 1, 2, \dots, N$ . The goal is to cluster these  $N$  observations into  $G$  clusters. Let  $Z_n = (Z_{n1}, Z_{n2}, \dots, Z_{nG})$  be the latent cluster indicator

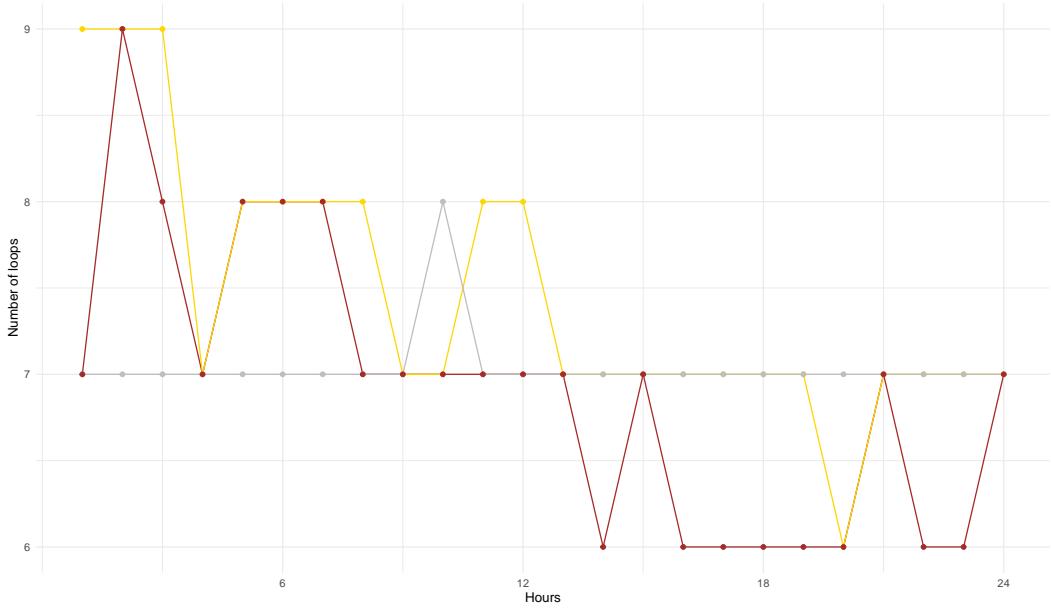


Figure 1: Number of loops per hour for the three medalists.

vector, where  $Z_{ng} = 1$  if observation  $n$  belongs to cluster  $g$  and  $Z_{ng} = 0$  otherwise. We assume that  $\mathbb{P}\{Z_{ng} = 1\} = \tau_g$  for  $g = 1, 2, \dots, G$ . Let denote  $\tau = (\tau_1, \dots, \tau_G)$ . The conditionally independent Poisson mixture model (Karlis 2018, sec. 9.4.2.1) assumes that the elements of  $X_n$  are independent Poisson distributed random variables, conditional on  $Z_n$ . That is,

$$Z_n \sim \text{Multinomial}(1, \tau)$$

$$X_{nm}|(Z_{ng} = 1) \sim \text{Poisson}(\lambda_{gm}), \text{ for } m = 1, 2, \dots, M.$$

Alternative modelling frameworks exist, either to introduce some dependence between variables or to normalize the variables. We refer the interested reader to (Karlis 2018; Bouveyron et al. 2019, chap. 6) for more details.

Denoting the model parameters by  $\theta = (\tau, \lambda)$  where  $\lambda = (\lambda_{gm})_{1 \leq g \leq G, 1 \leq m \leq M}$ , and where  $X = (x_n)_{1 \leq n \leq N}$  denotes the observations, the observed likelihood is

$$L(\theta) = \sum_{n=1}^N \sum_{g=1}^G \tau_g \prod_{m=1}^M \phi(x_{nm}, \lambda_{gm}),$$

where  $\phi(x, \lambda) = \exp(-\lambda)\lambda^x/x!$ , the Poisson probability mass function.

Due to form of the mixture distribution, there are no closed form for the maximum likelihood estimators, and an iterative EM algorithm needs to be used (Dempster, Laird, and Rubin 1977) to maximize the likelihood. The EM algorithm starts from an initial value  $\theta^{(0)}$  for the model parameter, and alternates the two following steps until convergence of the likelihood.

At the  $q$ th iteration of the EM algorithm, the E-step consists of computing for all  $1 \leq n \leq N$  and  $1 \leq g \leq G$ :

$$t_{ng}^{(q)} = \frac{\tau_g^{(q)} \prod_{m=1}^M \phi(x_{nm}, \lambda_{gm})}{\sum_{h=1}^G \tau_h^{(q)} \prod_{m=1}^M \phi(x_{nm}, \lambda_{hm})}.$$

In the M-step, the model parameters are updated as follows:

$$\tau_g^{(q+1)} = \frac{\sum_{n=1}^N t_{ng}^{(q)}}{N} \quad \text{and} \quad \lambda_{gm}^{(q+1)} = \frac{\sum_{n=1}^N t_{ng}^{(q)} x_{nm}}{\sum_{n=1}^N t_{ng}^{(q)}}.$$

The EM algorithm steps are iterated until convergence, where convergence is determined when  $\log L(\theta^{(q+1)}) - \log L(\theta^{(q)}) < \epsilon$ .

The number of clusters  $G$  is selected using the Bayesian information criterion (BIC) (Schwarz 1978),

$$BIC = 2 \log L(\hat{\theta}) - \{(G - 1) + GM\} \log(N),$$

where  $\hat{\theta}$  is the maximum likelihood estimate of the model parameters; models with higher BIC are preferred to models with lower BIC.

## 4 Variable selection

We develop a model-based clustering method with variable selection for multivariate count data. The method follows the approach of Raftery and Dean (2006) and Maugis, Celeux, and Martin-Magniette (2009) for continuous data and Dean and Raftery (2010) and Fop, Smart, and Murphy (2017) for categorical data. It consists in a stepwise model comparison approach where variables are added and removed from a set of clustering variables.

### 4.1 Model setup

The clustering and variable selection approach is based around partitioning  $X_n = (X_n^C, X_n^P, X_n^O)$  into three parts:

- $X_n^C$ : The current clustering variables,
- $X_n^P$ : The proposed variable to add to the clustering variables,
- $X_n^O$ : The other variables.

For simplicity of notation,  $C$  will be used to denote the set of indices of the current clustering variables,  $P$  the indices of the proposed variable and  $O$  the indices of the other one. Then  $(C, P, O)$  is a partition of  $\{1, \dots, M\}$ .

The decision on whether to add the proposed variable to the clustering variables is based on comparing two models:

$M_1$  (Clustering Model), which assumes that the proposed variable is useful for clustering:

$$(X_n^C, X_n^P) \sim \sum_{g=1}^G \tau_g \prod_{m \in \{C, P\}} \text{Poisson}(\lambda_{gm}).$$

The  $M_1$  model is fitted for different values of  $G$  between 1 and  $G_{max}$  to achieve the best clustering model.

$M_2$  (Non-Clustering Model) which assumes that the proposed variable is not useful for clustering, but is potentially linked to the clustering variables through a Poisson GLM, that is,

$$X_n^C \sim \sum_{g=1}^G \tau_g \prod_{m \in C} \text{Poisson}(\lambda_{gm})$$

$$X_n^P | (X_n^C = x_n^C, Z_{ng} = 1) \sim \text{PoissonGLM}(x_n^C),$$

where Poisson GLM states that

$$\log \mathbb{E}[X_n^P | X_n^C = x_n^C, Z_{ng} = 1] = \alpha + \beta^\top x_n^C.$$

In order to avoid non significant terms in the Poisson GLM model, a standard stepwise variable selection approach (using BIC as the variable selection criterion) is considered. Thus, the proposed variable  $X_n^P$  will be dependent on only a subset  $X_n^R$  of the clustering variables  $X_n^C$ . We note that  $G$  is fixed in the non-clustering model, because an optimal value for  $G$  is previously chosen. The other variables  $X_n^O$  are assumed to be conditionally independent of  $Z_n$  given  $X_n^C$  and  $X_n^P$ .

The clustering and non-clustering models are represented as graphical models in Figure 2.

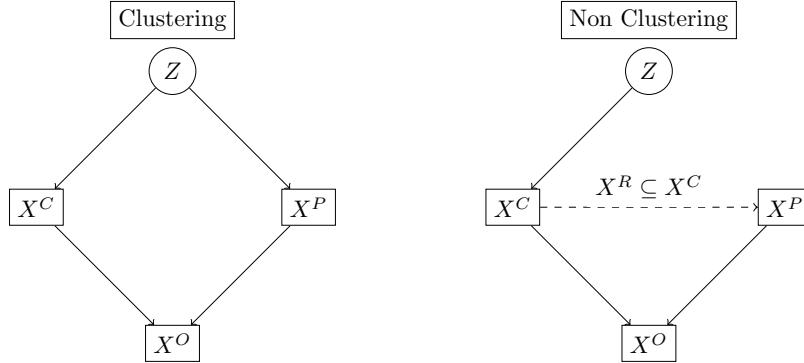


Figure 2: Graphical model representations of the clustering and non-clustering models.

Thus, there are two reasons for which  $M_2$  can be preferred to  $M_1$ : either  $X_n^P$  does not contain information about the latent clustering variable at all (ie.  $X_n^R = \emptyset$ ), or  $X_n^P$  does not add further useful information about the clustering given the information already contained in the current clustering variables. In the first situation, we say that  $X_n^P$  is an irrelevant variable, because it contains no clustering information. In the second situation, we say that  $X_n^P$  is a redundant variable because it contains no extra information about the clustering beyond the current clustering variables ( $X_n^C$ ).

Additionally, both models assume the same form for the conditional distribution for  $X_n^O | (X_n^C, X_n^P)$  and whose form doesn't need to be explicitly specified because it doesn't affect the model choice.

Variable  $P$  is added to  $C$  if the clustering model ( $M_1$ ) is preferred to the non-clustering model ( $M_2$ ). In order to compare  $M_1$  and  $M_2$ , following (Dean and Raftery 2010), we consider the Bayes Factor:

$$B_{1,2} = \frac{p(X|M_1)}{p(X|M_2)}$$

which is asymptotically approximated (Fop, Smart, and Murphy 2017; Kass and Raftery 1995) using the difference of the BIC criteria for both models:

$$2 \log B_{1,2} \simeq BIC_{M_1} - BIC_{M_2}.$$

The same modelling framework can be used for removing variables from the current set of clustering variables.

## 4.2 Interpretation

Comparing  $M_1$  and  $M_2$  is equivalent to comparing the following  $X_n^P | (X_n^C = x_n^C)$  structures.

The  $M_1$  (Clustering Model) assumes that,

$$X_n^P | (X_n^C = x_n^C) \sim \sum_{g=1}^G \mathbb{P}\{Z_{ng} = 1 | X_n^C = x_n^C\} \text{Poisson}(\lambda_{gm}),$$

where

$$\mathbb{P}\{Z_{ng} = 1 | X_n^C = x_n^C\} = \frac{\tau_g \prod_{m=1}^M \phi(x_{nm}, \lambda_{gm})}{\sum_{h=1}^G \tau_h \prod_{m=1}^M \phi(x_{nm}, \lambda_{hm})}.$$

Whereas, the  $M_2$  (Non-Clustering Model) assumes that,

$$X_n^P | (X_n^C = x_n^C) = \text{PoissonGLM}(x_n^C).$$

The method contrasts which of conditional model structures is better describing the distribution of the proposed variable  $X^P$ . The clustering model ( $M_1$ ) uses a mixture model, with covariate dependent weights, for the conditional model whereas the non-clustering model ( $M_2$ ) is a Poisson generalized linear model. The model selection criterion chooses the model that best models this conditional distribution.

### 4.3 Stepwise selection algorithm

#### 4.3.1 Screening variables: Initialization

We start with an initial choice of  $C$  by first screening each individual variable by fitting a mixture of univariate Poisson distributions (eg. Everitt and Hand 1981, chap. 4.3),

$$X_{nm} \sim \sum_{g=1}^G \tau_g \text{Poisson}(\lambda_{gm}), \text{ for } G = 1, 2, \dots, G_{max}.$$

The initial set of variables is set to be those variables where any model with  $G > 1$  is preferred to the  $G = 1$  model.

#### 4.3.2 Stepwise algorithm: Updating

We consider a stepwise algorithm which alternates between adding and removing steps. In the removal step, all the variables in  $X^C$  are examined in turn to be removed from the set. In the adding step, all the variables in  $X^O$  are examined in turn to be added to the clustering set.

The algorithm also performs the selection of the number  $G$  of clusters finding at each stage the optimal combination of clustering variables and number of clusters. The procedure stops when no change has been made to the set  $X^C$  after consecutive exclusion and inclusion steps.

With the present stepwise selection algorithm, it can occur that during the process, we get back on a solution (a set of clustering variable) already explored. Since our algorithm is not stochastic, we fall into an infinite cycle. In this situation the algorithm is stopped, and the best solution according to BIC among the solution of the cycle is kept.

The following pseudo-code summarizes our stepwise algorithm:

```

ALGORITHM Stepwise
BEGIN
initialize  $X^C$ 
WHILE  $X^C$  changes:
```

```

- for all variable  $X_j$  which are not in  $X^C$ 
- estimate  $M_1$  on  $X^C \cup X_j$  and select the best  $G$ 
- estimate  $M_2$  with the model for  $X^C$  (with  $G$  selected at the previous step) and a Poisson regression for  $X_j$  given  $X^C$ 
- add  $X_j$  in  $X^C$  if  $BIC_{M_1} > BIC_{M_2}$ 
- for each  $X_j$  in  $X^C$ 
- estimate  $M_2$  on  $X^C \setminus X_j$ , select the best  $G$  and use a Poisson regression for  $X_j$  given  $X^C \setminus X_j$ 
- estimate  $M_1$  on  $X^C$  (with  $G$  selected at the previous step)
- remove  $X_j$  from  $X^C$  if  $BIC_{M_2} > BIC_{M_1}$ 
- test for infinite loop
ENDWHILE
return  $X^C$  and  $M_1$  estimate
END

```

## 5 Simulation study

In this section, we evaluate the proposed variable selection method through three different simulation scenarios. We start with an illustrative example in which, using a data set simulated according to the proposed model, we show how to perform the variable selection.

Then, simulation studies are performed to evaluate the behavior of the proposed selection method, when the data are simulated according to the proposed model (Scenario1) and when the model assumptions are violated. In Scenario2, the link between  $X^R$  and  $X^C$  is no longer a Poisson GLM but a more complex model. In Scenario3, the clustering variables are no longer conditionally independent.

### 5.1 Illustrative example

In the first simulation setting we consider 10 Poisson random variables. Variables  $X_1, X_2, X_3$  and  $X_4$  are the clustering variables, distributed according to a mixture of  $G = 3$  independent Poisson mixture distributions with mixing proportions 0.4, 0.3, 0.3. Variables  $X_5, X_6$  and  $X_7$  are redundant variables, each one generated dependent on the clustering variables. These three variables are linked to the four first ones through a Poisson GLM. The last three variables,  $X_8, X_9$  and  $X_{10}$  are irrelevant variables not related to the previous ones. Table 1 shows the parameter of the Poisson distribution for each variable and each cluster.

Table 1: True values of component parameters (Scenario 1)

	$\lambda_{g1}$	$\lambda_{g2}$	$\lambda_{g3}$	$\lambda_{g4}$	$\lambda_{g5}$	$\lambda_{g6}$	$\lambda_{g7}$	$\lambda_{g8}$	$\lambda_{g9}$	$\lambda_{g10}$
$g = 1$	1	1	1	1	$\lambda_{g5}$	$\lambda_{g6}$	$\lambda_{g7}$	4	2	1
$g = 2$	2	2	1	4	$\lambda_{g5}$	$\lambda_{g6}$	$\lambda_{g7}$	4	2	1
$g = 3$	4	4	4	4	$\lambda_{g5}$	$\lambda_{g6}$	$\lambda_{g7}$	4	2	1

with  $\lambda_{g5} = \exp(0.2X_2)$ ,  $\lambda_{g6} = \exp(0.2X_1 - 0.1X_2)$  and  $\lambda_{g7} = \exp(0.1(X_1 + X_3 + X_4))$ .

Below is the result obtained for one data set of size  $N = 400$ . The evaluation criteria is the selected features (true one are  $X_1$  to  $X_4$ ) and the Adjusted Rand Index (Rand 1971; Hubert and Arable 1985)

obtained with the selected variables in comparison to those obtained with the full set of variables and with the true clustering variables.

The independent Poisson mixture model was fitted to the simulated data with  $N = 400$  rows and  $P = 10$  columns. Models with  $G = 1$  to  $G = 10$  were fitted using the EM algorithm.

The values of BIC for the independent Poisson mixture model are plotted in Figure 3.

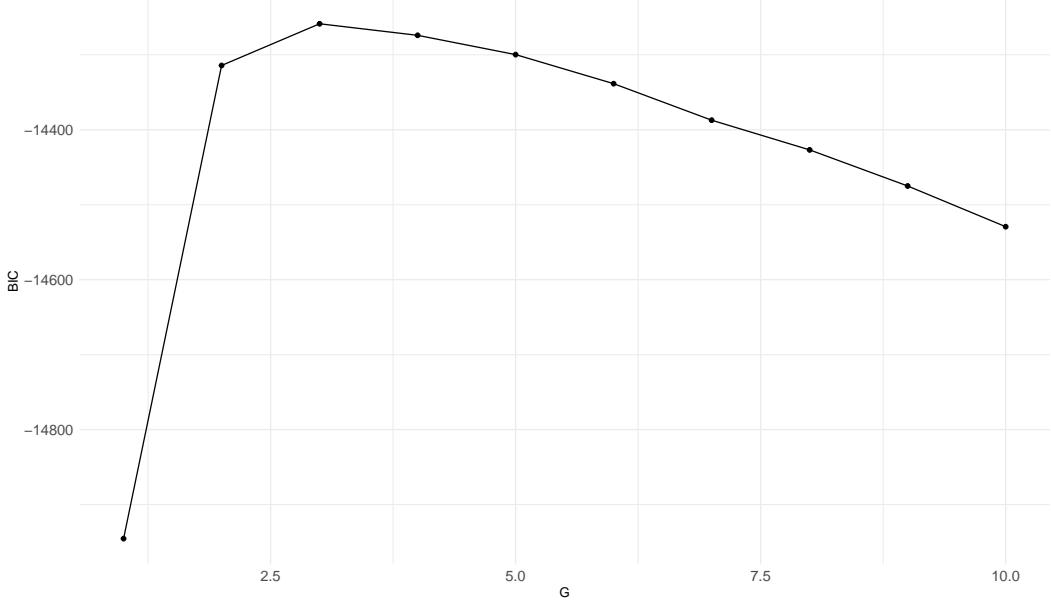


Figure 3: Bayesian Information Criterion (BIC) for the independent Poisson mixture model.

The model with the highest BIC has  $G = 3$  components and the resulting estimates of  $\tau$  and  $\lambda$  are given as:

Table 2: Estimates of the mixing proportions and component parameters.

$\tau_g$	$\lambda_{g1}$	$\lambda_{g2}$	$\lambda_{g3}$	$\lambda_{g4}$	$\lambda_{g5}$	$\lambda_{g6}$	$\lambda_{g7}$	$\lambda_{g8}$	$\lambda_{g9}$	$\lambda_{g10}$
$g = 1$	0.29	4.09	4.00	4.15	4.34	2.51	1.87	3.95	4.04	1.85
$g = 2$	0.42	2.04	2.11	1.34	3.74	1.64	1.27	2.00	3.91	2.06
$g = 3$	0.29	0.93	0.88	1.08	0.96	1.13	1.01	1.16	3.82	2.02

A look at Table 1 of true values allows us to say that these estimates are correct (except for label switching).

Let start by initializing the stepwise algorithm.

```
fit_screen <- poissonmix_screen(x, G = 1:Gmax)
jchosen <- fit_screen$jchosen
```

The variables selected by the screening procedure are  $\{1, 2, 3, 4, 6, 7\}$ .

Now, we execute the stepwise selection algorithm:

```
fit <- poissonmix_varsel(x, jchosen=jchosen, G = 1:Gmax)

[1] "Initial Selected Variables: 1,2,3,4,6,7"
[1] "Iteration: 1"
[1] "Add Variable: NONE 10 BIC Difference: -13.2"
[1] "Remove Variable: 6 BIC Difference: 83.7"
[1] "Current Selected Variables: 1,2,3,4,7"
[1] "Iteration: 2"
[1] "Add Variable: NONE 9 BIC Difference: -10.6"
[1] "Remove Variable: 7 BIC Difference: 50.1"
[1] "Current Selected Variables: 1,2,3,4"
[1] "Iteration: 3"
[1] "Add Variable: NONE 10 BIC Difference: -10.5"
[1] "Remove Variable: NONE 3 BIC Difference: -26.8"
[1] "Current Selected Variables: 1,2,3,4"
```

Note that the computing time is about 5 minutes on a laptop with 2.3 GHz Intel Core i7 processor and 32Go of RAM.

The final chosen variables are {1, 2, 3, 4}.

Finally, the ARI obtained with the selected variables, which turn out to be the true clustering variable, is 0.594 whereas it is 0.432 with all the variables.

## 5.2 Scenarios of simulation

In this section the three scenario of simulation are described. The first scenario is similar to the previous illustrative example.

The second scenario is similar to the first one, except for variables  $X_5$ ,  $X_6$  and  $X_7$  which are still redundant but linked to the true clustering variables through linear, quadratic and exponential term in an identity link function, respectively, and not a Poisson GLM with logarithm link function. More precisely,  $X_5$ ,  $X_6$  and  $X_7$  have Poisson distribution of respective parameter  $\lambda_{g5} = \exp(2X_2)$ ,  $\lambda_{g6} = \exp(X_1^2 + X_3)$  and  $\lambda_{g7} = \exp(\exp(0.1(X_1 + X_3 + X_4)))$ . Thus, the data are simulated from a model which does not satisfy assumptions of model  $M_2$ .

The third scenario is similar to the second one, but some dependence between the clustering variables  $X_1$  and  $X_2$  is introduced, in order to create some redundancy among the true clustering variables. For this,  $X_1$  and  $X_2$  are simulated as in the previous setting, and a same term is added to both of these variables (simulated according a Poisson distribution of parameter 2).

## 5.3 Results

Table 3 shows the number of times, among the 100 simulated data sets, that each variable is selected. For Scenario 1, the model selection procedure perform perfectly, selecting each time only the true clustering variables. For Scenario 2, due to the fact the link between the redundant and the true clustering variables is not a standard Poisson GLM, the variable selection is perturbed and variables  $X_5$  is sometimes selected. For Scenario 3, the results is that the dependency between  $X_1$  and  $X_2$  perturb the variable selection, and only one of them is selected (and even sometimes none of them). Redundant variables  $X_5$  and  $X_6$ , which are linked to the clustering variables but with a linear link, are also sometimes selected.

Table 3: Number of selection for each variable, simulation setting number 3.

	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$	$X_9$	$X_{10}$
Scenario 1	100	100	100	100	0	0	0	0	0	0
Scenario 2	97	100	90	98	44	0	0	0	0	0
Scenario 3	48	35	89	88	65	34	3	0	0	0

Figure 4 plots the distribution of the ARI differences between the model with either the selected variables or all the variables, and the one with the true clustering variables. These plots shows that for all scenarios, the ARI of the model with the selected variables (left boxplot of each plot) are always closest to the optimal ARI (obtained with the true clustering variables).

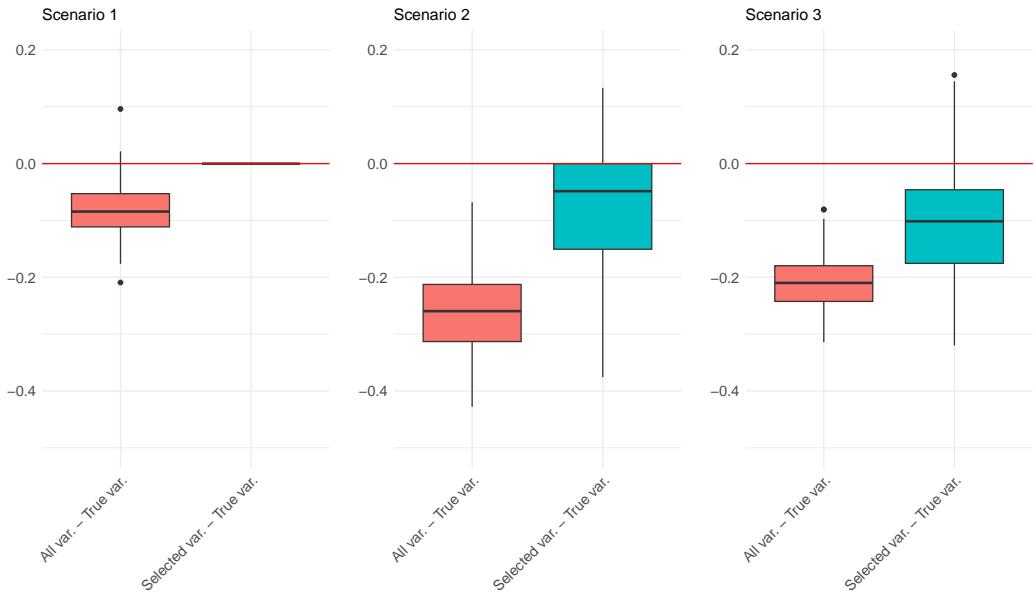


Figure 4: Distribution of the ARI differences with the model with the true clustering variables, for the model with the selected variables and the model with all variables.

Finally Figure 5 plots the histogram of the difference of ARI with the selected variables and with all the variables. This plot illustrates the interest of variable selection on the clustering results, and indeed, for all the scenarios, the ARI is better with the selected variables than when using all the variables.

## 6 International Ultrarunning Association Data

We apply the proposed procedure to the data from the 2012 International Ultrarunning Association World 24H Championships.

We start by initializing the stepwise algorithm, and find the variables selected by the screening procedure:

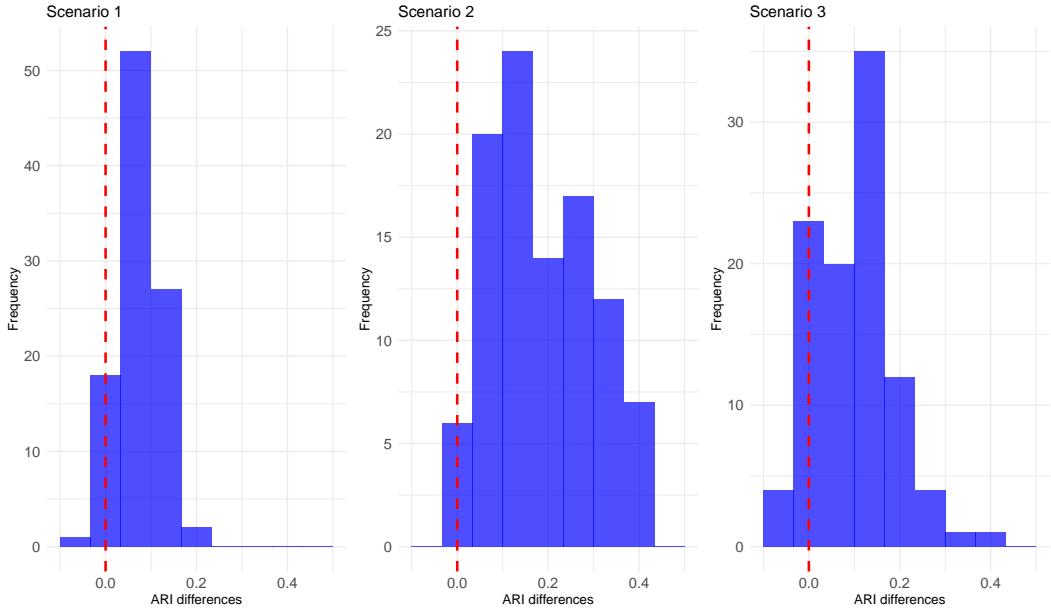


Figure 5: Distribution of the ARI differences for the model with the selected variables and the model with all variables.

```
fit_screen <- poissonmix_screen(x, G = 1:Gmax)
jchosen <- fit_screen$jchosen
jchosen
[1] 3 5 6 7 8 9 11 12 13 14 15 16 17 18 19 20 21 22 23 24
```

We then execute the proposed stepwise selection algorithm (the computing time is about 26 minutes on a laptop with 2.3 GHz Intel Core i7 processor and 32Go of RAM):

```
fit <- poissonmix_varsel(x, jchosen = jchosen, G = 1:Gmax)
```

The final chosen variables found by the algorithm are:

```
[1] 9 10 11 12 14 15 16 17 18 19 20 21 22 24
```

The optimal number of clusters 6 has been chosen inside the stepwise selection algorithm. The same choice is obtained when looking for the best  $G$  with the conditionally independent Poisson mixture on the selected variables (Figure 6).

In order to illustrate the results, we plot the cluster means according to the 24 variable mean parameters per cluster. For each variable not in the chosen variable set, a Poisson regression model is fitted with the chosen variables as predictors. Forward and backwards variable selection is conducted on this regression, if the regression model has any predictor variables, then the variable is called “redundant” and if the regression model has no predictor variables, then the variable is called “irrelevant”. Figure 7 shows the cluster mean for each variable, where the label indicates if the variable is irrelevant for clustering (“I”), redundant (“R”) or useful (then the point is unlabelled).

The variables discriminate the clusters pacing strategies of the runners are the number of laps covered during the last two thirds of the race (except during the 13th and 23rd hours). The number of laps covered during the first eight hours does not provide any additional clustering information, and even no information at all for the number of laps covered during the first hour.

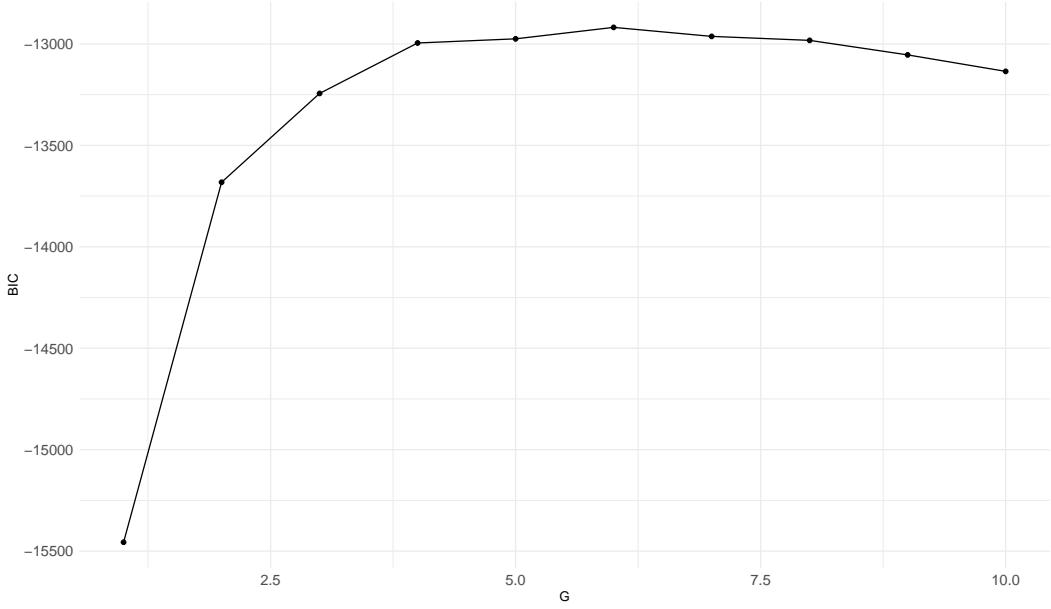


Figure 6: Bayesian Information Criterion (BIC) for the independent Poisson mixture model with the selected variables.

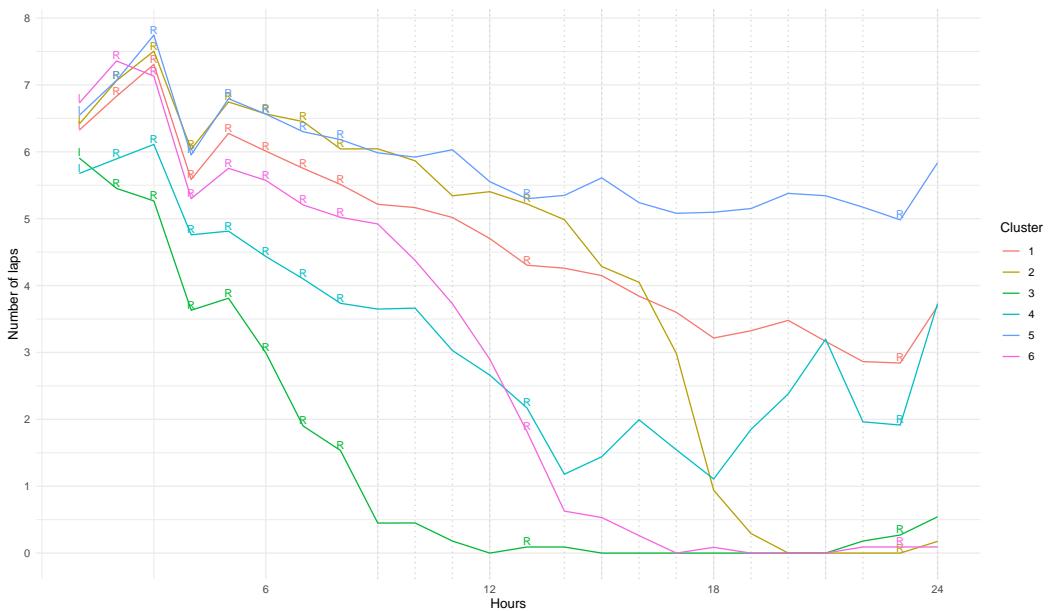


Figure 7: Cluster means and usefulness of the variables.

Figure 8 plots the density map per clusters. Area of high density (red) indicates the hours and the corresponding average number of laps specific of each cluster.

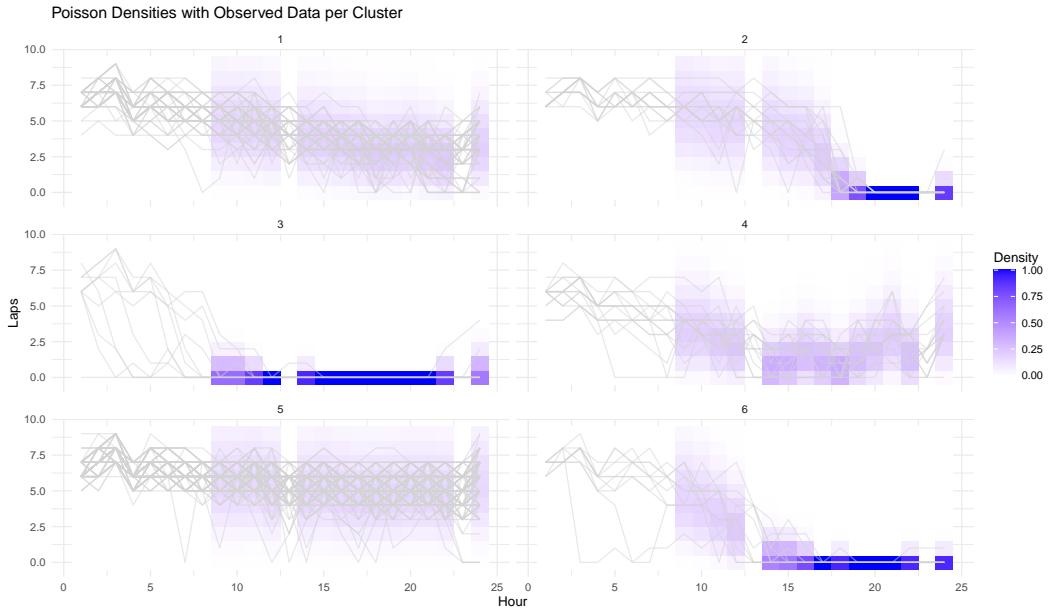


Figure 8: Density maps per cluster

Cluster 5 are clearly the most efficient runners. Looking at the running strategy in Figure 7 and Figure 8, we can see that they start as runners of Cluster 1 and Cluster 2, but they managed to keep a constant pace on the second part of the race, unlike those of the other two clusters which faltered. Runners of Cluster 3 has covered the fewest number of laps. Indeed, looking at their running strategy, we can see that most of these runners stop after the first third of the race. Cluster 6 is relatively similar to Cluster 3, but runners manage to continue running until half of the race is completed. Finally, Cluster 4 obtains slightly better results than Cluster 6, starting more carefully, and managing to run until the end of the race, even if the pace of the last hours is not very constant.

Finally, Figure 9 shows boxplots of the total number of loops covered by the runners of each of the clusters.

## 7 Discussion

A method for clustering and variable selection for multivariate count data has been proposed. The method is shown to give excellent performance on both simulated and real data examples. The method selects set of relevant variables for clustering and other variables are not selected if they are irrelevant or redundant for clustering purposes.

The proposed method is shown to give interesting insights in the application domain, where some clusters members are shown to perform better overall to others and the benefits of constant (or near constant pacing) are shown.

The level of variable selection is determined by the relative performance of the two models (as shown in Section 4.2) is compared. Alternative models to the Poisson GLM model which have greater flexibility could lead to a smaller set of selected variables than the proposed method achieves. This is a topic for future research.

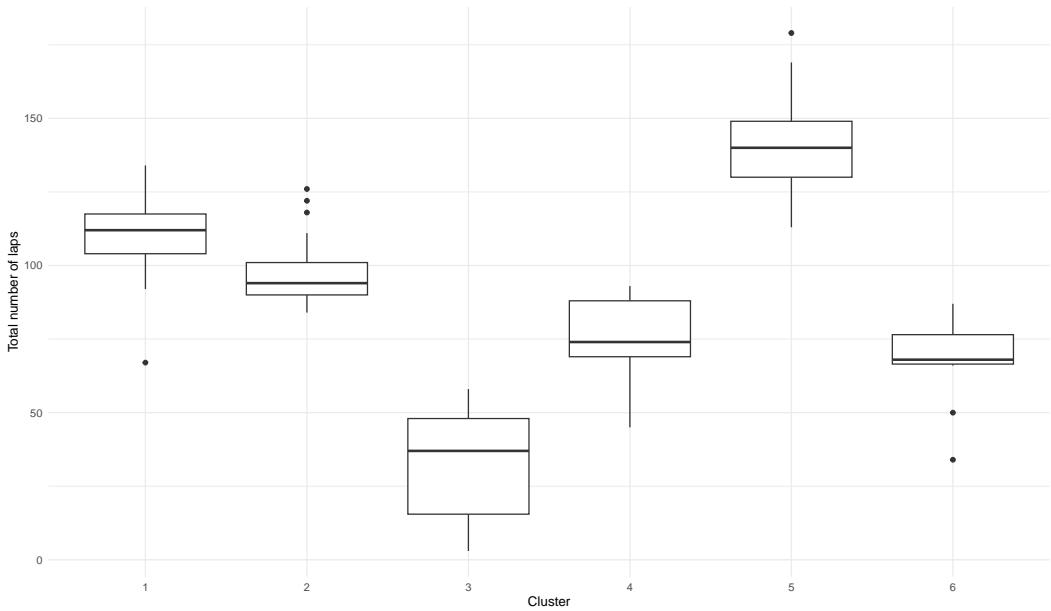


Figure 9: Number of loops covered by the runners of each of the clusters.

The proposed method is based on a conditionally independent Poisson mixture model for the selected variables. It could be argued that the conditional independence assumption is unrealistic in the application. Hand and Yu (2001) consider the implication of incorrectly assuming conditional independence in a classification setting and show that it can make the group membership probabilities over confident. Furthermore, in the conditional independent Poisson mixture model, the number of clusters can be upwardly biased, where extra clusters are included to model dependence in the data. The approach taken in the paper could be extended to use other multivariate count distributions, including multivariate distributions without the conditional independence assumption (eg. Karlis 2018; Karlis and Meligkotsidou 2007; Inouye et al. 2017).

The code for the proposed approach is available as an R package [poissonmix\\_0.1.tar.gz](#).

## 8 Acknowledgements

This work was supported by the Science Foundation Ireland Insight Research Centre (SFI/12/RC/2289\_P2) and a visit to the Collégium – Institut d’Études Avancées de Lyon.

- Agresti, Alan. 2013. *Categorical Data Analysis*. Third. Wiley Series in Probability and Statistics. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ.
- Bouveyron, Charles, Gilles Celeux, T. Brendan Murphy, and Adrian E. Raftery. 2019. *Model-Based Clustering and Classification for Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge. <https://doi.org/10.1017/9781108644181>.
- Chiquet, Julien, Mahendra Mariadassou, and Stéphane Robin. 2021. “The Poisson-Lognormal Model as a Versatile Framework for the Joint Analysis of Species Abundances.” *Frontiers in Ecology and Evolution* 9: 188. <https://doi.org/10.3389/fevo.2021.588292>.
- Dean, Nema, and Adrian E. Raftery. 2010. “Latent Class Analysis Variable Selection.” *Annals of the Institute of Statistical Mathematics* 62 (1): 11–35. <https://doi.org/10.1007/s10463-009-0258-9>.
- Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. 1977. “Maximum Likelihood from Incomplete Data via the EM Algorithm.” *Journal of the Royal Statistical Society: Series B* 39: 1–38.

- <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>.
- Everitt, Brian S., and David J. Hand. 1981. *Finite Mixture Distributions*. Chapman & Hall.
- Fop, Michael, and Thomas Brendan Murphy. 2018. “Variable Selection Methods for Model-Based Clustering.” *Statistics Surveys* 12: 18–65. <https://doi.org/10.1214/18-SS119>.
- Fop, Michael, Keith Smart, and Thomas Brendan Murphy. 2017. “Variable Selection for Latent Class Analysis with Application to Low Back Pain Diagnosis.” *Annals of Applied Statistics*. 11: 2085–115.
- Frühwirth-Schnatter, Sylvia, Gilles Celeux, and Christian P. Robert. 2018. *Handbook of Mixture Analysis*. Chapman; Hall/CRC. <https://doi.org/10.1201/9780429055911>.
- Hand, David J., and Keming Yu. 2001. “Idiot’s Bayes—Not so Stupid After All?” *International Statistical Review* 69 (3): 385–98. <https://doi.org/10.1111/j.1751-5823.2001.tb00465.x>.
- Hartigan, John A., and M. Anthony Wong. 1979. “A k-Means Clustering Algorithm.” *Applied Statistics* 28 (1): 100–108.
- Hubert, Lawrence, and Phipps Arable. 1985. “Comparing Partitions.” *Journal of Classification* 2 (1): 193–218. <https://doi.org/10.1007/BF01908075>.
- Inouye, David I., Eunho Yang, Genevera I. Allen, and Pradeep Ravikumar. 2017. “A Review of Multivariate Distributions for Count Data Derived from the Poisson Distribution.” *WIREs Computational Statistics* 9 (3): e1398. <https://doi.org/10.1002/wics.1398>.
- Karlis, Dimitris. 2018. “Mixture Modelling of Discrete Data.” In *Handbook of Mixture Analysis*, edited by Sylvia Frühwirth-Schnatter, Gilles Celeux, and Christian P. Robert, 193–218. CRC Press.
- Karlis, Dimitris, and Loukia Meligkotsidou. 2007. “Finite Mixtures of Multivariate Poisson Distributions with Application.” *Journal of Statistical Planning and Inference* 137 (6): 1942–60. <https://doi.org/10.1016/j.jspi.2006.07.001>.
- Kass, Robert E., and Adrian E. Raftery. 1995. “Bayes Factors.” *Journal of the American Statistical Association* 90 (430): 773–95. <https://doi.org/10.1080/01621459.1995.10476572>.
- Maugis, Cathy, Gilles Celeux, and Marie-Laure Martin-Magniette. 2009. “Variable Selection in Model-Based Clustering: A General Variable Role Modeling.” *Computational Statistics & Data Analysis* 53 (11): 3872–82. <https://doi.org/10.1016/j.csda.2009.04.013>.
- McLachlan, Geoffrey, and David Peel. 2000. *Finite Mixture Models*. New York: Wiley. <https://doi.org/10.1002/0471721182>.
- McParland, Damien, and Thomas Brendan Murphy. 2018. “Mixture Modelling of High-Dimensional Data.” In *Handbook of Mixture Analysis*, edited by Sylvia Frühwirth-Schnatter, Gilles Celeux, and Christian P. Robert, 247–80. CRC Press.
- Raftery, Adrian E., and Nema Dean. 2006. “Variable Selection for Model-Based Clustering.” *Journal of the American Statistical Association* 101 (473): 168–78. <https://doi.org/10.1198/016214506000000113>.
- Rand, William M. 1971. “Objective Criteria for the Evaluation of Clustering Methods.” *Journal of the American Statistical Association* 66 (336): 846–50.
- Rau, Andrea, Cathy Maugis-Rabasseau, Marie-Laure Martin-Magniette, and Gilles Celeux. 2015. “Co-expression analysis of high-throughput transcriptome sequencing data with Poisson mixture models.” *Bioinformatics* 31 (9): 1420–27. <https://doi.org/10.1093/bioinformatics/btu845>.
- Schwarz, Gideon. 1978. “Estimating the Dimension of a Model.” *The Annals of Statistics* 6 (2): 461–64. <https://doi.org/10.1214/aos/1176344136>.
- Silva, Anjali, Steven J. Rothstein, Paul D. McNicholas, and Sanjeena Subedi. 2019. “A Multivariate Poisson-Log Normal Mixture Model for Clustering Transcriptome Sequencing Data.” *BMC Bioinformatics* 20 (1): 394. <https://doi.org/10.1186/s12859-019-2916-0>.
- White, Arthur, and Thomas Brendan Murphy. 2016. “Exponential Family Mixed Membership Models for Soft Clustering of Multivariate Data.” *Advances in Data Analysis and Classification* 10: 521–40.

## Marco Cornelì

### *Material list:*

Corneli M., Marchello G. & Bouveyron C. (2025) A deep dynamic latent block model for co-clustering of zero-inflated data matrices. WGMBC 2025 slides.

# A Deep Dynamic Latent Block Model for Co-clustering of Zero-Inflated Data Matrices

Marco CORNELI

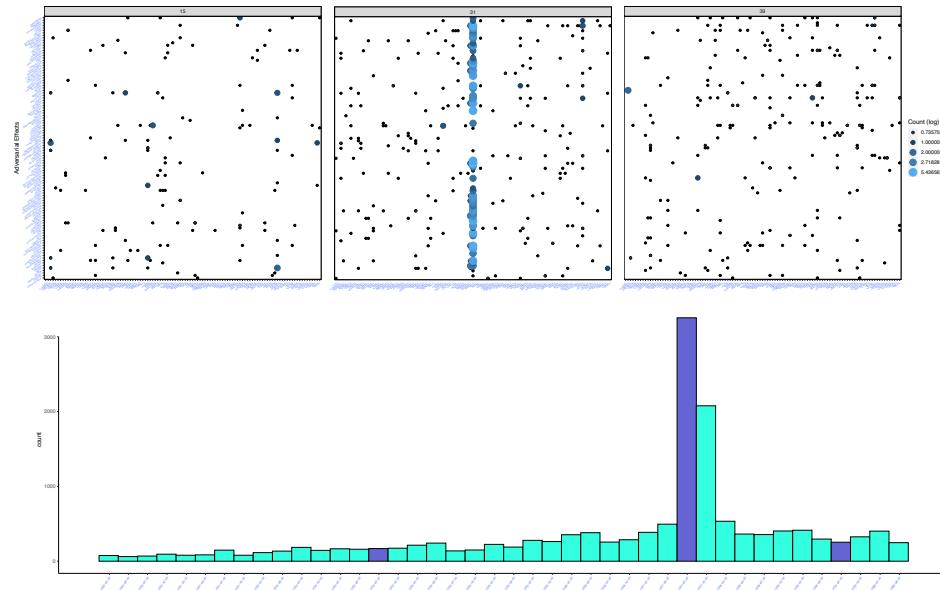
Joint work with  
**G. Marchello & C. Bouveyron**

July 23, 2025

## Dynamic block modelling of counting data

- ▶ PhD thesis of [Giulia Marchello](#) (defended in 2023)
- ▶ main ideas from Marchello, Cornel, and Bouveyron, 2023, 2024:
- ▶ collaboration with the Regional Center of **Pharmacovigilance** (RCPV) of the hospital university center (CHU) in Nice
- ▶ adverse drug reaction (ADR) data set: co-clustering *and* change-point detection
- ▶ sequence of count matrices whose entry  $(i, j)$  at time  $t$  counts the number of times the AE  $j$  is reported for drug  $i$  at time  $t$  (e.g. trimester)

## Data and objectives



3

## Data and objectives

### Motivation:

- summarize massive dynamic datasets
- detect model changes in cluster memberships
- sparsity modeling

### Main features:

- exploit systems of ODEs to model cluster membership over time
- enhance sparsity with mixtures of ZIP distributions
- mixed inference procedure EM-SGD ([neural nets](#) are in)

**Figure:** Example of dynamic co-clustering

4

## Data and Objectives

The data we consider are organized as follows

- rows are indexed by  $i = 1, \dots, N$
- columns are indexed by  $j = 1, \dots, M$
- time instants  $t \in [0, T]$  during which  $N$  and  $M$  are fixed
- the  $N \times M \times T$  tensor  $X := \{X_{ij}(t)\}$  contains the number of interactions between any observation and feature pair at any given  $t$

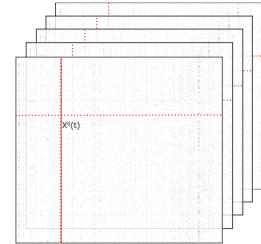


Figure: Data structure.

5

## Data and Objectives

The data we consider are organized as follows

- rows are indexed by  $i = 1, \dots, N$
- columns are indexed by  $j = 1, \dots, M$
- time instants  $t \in [0, T]$  during which  $N$  and  $M$  are fixed
- the  $N \times M \times T$  tensor  $X := \{X_{ij}(t)\}$  contains the number of interactions between any observation and feature pair at any given  $t$

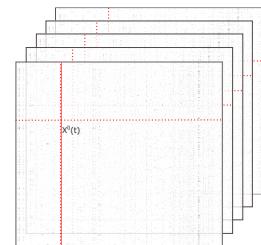


Figure: Data structure.

We aim at estimating:

- the latent variables for the clustering of rows and columns into  $Q$  and  $L$  groups
- a latent variable for modeling the evolving sparsity of the data

5

## The ZIP-dLBM

- **Multinomial** random variables to represent the membership to clusters:

- $Z_i(t) \sim \mathcal{M}(1, \alpha(t)) := (\alpha_1(t), \dots, \alpha_Q(t))$ ,
  - $W_j(t) \sim \mathcal{M}(1, \beta(t)) := (\beta_1(t), \dots, \beta_L(t))$ .

- **Zero-Inflated Poisson** distribution to model the data:

- $X_{ij}(t)|Z_i(t), W_j(t) \sim ZIP(\Lambda_{Z_i(t), W_j(t)}, \pi(t))$ .

where:

- $\Lambda$ : block-dependent Poisson intensity function,
    - $\pi(t)$ : sparsity at any given time period.

$$\begin{cases} X_{ij}(t)|Z_i(t), W_j(t) \sim 0 & \text{with probability } \pi(t) \\ X_{ij}(t)|Z_i(t), W_j(t) \sim \mathcal{P}(\Lambda_{Z_i(t), W_j(t)}) & \text{with probability } 1 - \pi(t) \end{cases} \quad (1)$$

6

## The ZIP-dLBM

- **Multinomial** random variables to represent the membership to clusters:

- $Z_i(t) \sim \mathcal{M}(1, \alpha(t)) := (\alpha_1(t), \dots, \alpha_Q(t))$ ,
  - $W_j(t) \sim \mathcal{M}(1, \beta(t)) := (\beta_1(t), \dots, \beta_L(t))$ .

- **Zero-Inflated Poisson** distribution to model the data:

- $X_{ij}(t)|Z_i(t), W_j(t) \sim ZIP(\Lambda_{Z_i(t), W_j(t)}, \pi(t))$ .

where:

- $\Lambda$ : block-dependent Poisson intensity function,
    - $\pi(t)$ : sparsity at any given time period.

$$\begin{cases} X_{ij}(t)|Z_i(t), W_j(t) \sim 0 & \text{with probability } \pi(t) \\ X_{ij}(t)|Z_i(t), W_j(t) \sim \mathcal{P}(\Lambda_{Z_i(t), W_j(t)}) & \text{with probability } 1 - \pi(t) \end{cases} \quad (1)$$

- To model the **data sparsity** we introduce:  $A_{ij}(t) \sim \mathcal{B}(\pi(t))$ :

$$\begin{cases} X_{ij}(t)|Z_i(t), W_j(t) \sim 0 & \text{if } A_{ij}(t) = 1 \\ X_{ij}(t)|Z_i(t), W_j(t) \sim \mathcal{P}(\Lambda_{Z_i(t), W_j(t)}) & \text{if } A_{ij}(t) = 0 \end{cases} \quad (2)$$

6

## The ZIP-dLBM

- The **evolving mixing proportion** and the **sparsity** parameter are assumed to be generated by three **systems of ODEs**, respectively:

$$\begin{aligned} \square \dot{a}(t) &= f_Z(a(t)), & \text{with } \alpha_q(t) = \frac{e^{a_q(t)}}{\sum_{q=1}^Q e^{a_q(t)}}, \\ \square \dot{b}(t) &= f_W(b(t)), & \text{with } \beta_\ell(t) = \frac{e^{b_\ell(t)}}{\sum_{\ell=1}^L e^{b_\ell(t)}}, \\ \square \dot{c}(t) &= f_A(c(t)), & \text{with } \pi(t) = \frac{e^{c(t)}}{e^{c(t)} + e^{(1-c(t))}}. \end{aligned}$$

7

## The ZIP-dLBM

- The **evolving mixing proportion** and the **sparsity** parameter are assumed to be generated by three **systems of ODEs**, respectively:

$$\begin{aligned} \square \dot{a}(t) &= f_Z(a(t)), & \text{with } \alpha_q(t) = \frac{e^{a_q(t)}}{\sum_{q=1}^Q e^{a_q(t)}}, \\ \square \dot{b}(t) &= f_W(b(t)), & \text{with } \beta_\ell(t) = \frac{e^{b_\ell(t)}}{\sum_{\ell=1}^L e^{b_\ell(t)}}, \\ \square \dot{c}(t) &= f_A(c(t)), & \text{with } \pi(t) = \frac{e^{c(t)}}{e^{c(t)} + e^{(1-c(t))}}. \end{aligned}$$

- $f_Z$ ,  $f_W$  and  $f_A$  are modelled via three **fully connected neural networks**.

7

## The ZIP-dLBM

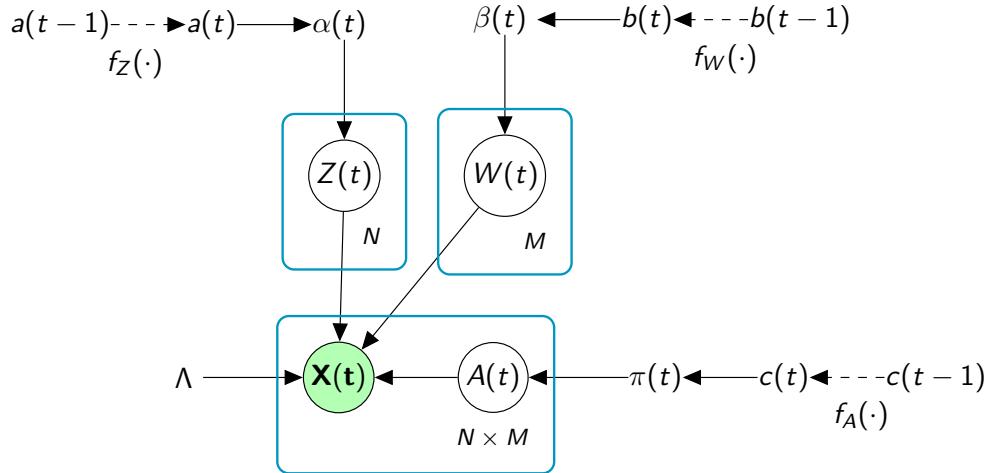


Figure: Graphical representation of ZIP-dLBM.

8

## The joint distribution

Given  $\theta = (\Lambda, \alpha(t), \beta(t), \pi(t))$ , we can compute the likelihood of the complete data:

$$p(X, Z, W, A|\theta) = p(X|Z, W, A, \Lambda, \pi)p(A|\pi)p(Z|\alpha)p(W|\beta) \quad (3)$$

where:

$$p(X|A, Z, W, \Lambda, \pi) = \prod_{i=1}^N \prod_{j=1}^M \prod_{t=1}^T \mathbf{1}_{\{X_{ij}(t)=0\}} \left\{ \left( \frac{\Lambda_{Z_i(t)W_j(t)}^{X_{ij}(t)}}{X_{ij}(t)!} \exp(-\Lambda_{Z_i(t)W_j(t)}) \right)^{(1-A_{ij}(t))} \right\}, \quad (4)$$

$$p(A|\pi) = \prod_{i=1}^N \prod_{j=1}^M \prod_{t=1}^T \pi(t)^{A_{ij}(t)} (1 - \pi(t))^{(1-A_{ij}(t))}, \quad (5)$$

$$p(Z|\alpha) = \prod_{i=1}^N \prod_{q=1}^Q \prod_{t=1}^T \alpha_q(t)^{Z_{iq}(t)}, \quad (6)$$

$$p(W|\beta) = \prod_{j=1}^M \prod_{\ell=1}^L \prod_{t=1}^T \beta_\ell(t)^{W_{j\ell}(t)}. \quad (7)$$

9

## The inference: Variational assumptions

**Goal:** maximization of the log-likelihood with respect to the model parameters.

- We rely on the [Variational-EM algorithm](#) (VEM).

Given a variational distribution  $q(\cdot)$ :

$$\log p(X|\theta) = \mathcal{L}(q; \theta) + KL(q(\cdot)||p(\cdot|X, \theta)),$$

where:

$$\begin{aligned}\mathcal{L}(q, \theta) &= \sum_Z \sum_W \sum_A q(Z, W, A) \log \frac{p(X, A, Z, W|\theta)}{q(Z, W, A)} \\ &= \mathbb{E}_{q(A, Z, W)} \left[ \log \frac{p(X, A, Z, W|\theta)}{q(A, Z, W)} \right].\end{aligned}$$

$$KL(q(\cdot)||p(\cdot|X, \theta)) = - \sum_Z \sum_W \sum_A q(Z, W, A) \log \frac{p(Z, W, A|X, \theta)}{q(Z, W, A)}.$$

10

## The inference: Variational assumptions

**Goal:** maximization of the log-likelihood with respect to the model parameters.

- We rely on the [Variational-EM algorithm](#) (VEM).

Given a variational distribution  $q(\cdot)$ :

$$\log p(X|\theta) = \mathcal{L}(q; \theta) + KL(q(\cdot)||p(\cdot|X, \theta)),$$

where:

$$\begin{aligned}\mathcal{L}(q, \theta) &= \sum_Z \sum_W \sum_A q(Z, W, A) \log \frac{p(X, A, Z, W|\theta)}{q(Z, W, A)} \\ &= \mathbb{E}_{q(A, Z, W)} \left[ \log \frac{p(X, A, Z, W|\theta)}{q(A, Z, W)} \right].\end{aligned}$$

$$KL(q(\cdot)||p(\cdot|X, \theta)) = - \sum_Z \sum_W \sum_A q(Z, W, A) \log \frac{p(Z, W, A|X, \theta)}{q(Z, W, A)}.$$

In order to optimize this lower bound  $\mathcal{L}(q, \theta)$  we assume that  $q(A, Z, W)$  fully factorizes over the latent variables ([mean field](#))

10

## The inference: Lower Bound

$\mathcal{L}(q, \theta)$  can be finally expressed as:

$$\begin{aligned}
& \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^M \left\{ \delta_{ij}(t) \log(\pi(t) \mathbf{1}_{\{X_{ij}(t)=0\}}) + (1 - \delta_{ij}(t)) \left[ \log(1 - \pi(t)) + \right. \right. \\
& \quad \left. \left. + \sum_{q=1}^Q \sum_{\ell=1}^L \left\{ \tau_{iq}(t) \eta_{j\ell}(t) X_{ij}(t) \log \Lambda_{q\ell} - \tau_{iq}(t) \eta_{j\ell}(t) \Lambda_{q\ell} \right\} \right] - (1 - \delta_{ij}(t)) \log(X_{ij}(t)!) \right\} + \\
& \quad + \sum_{t=1}^T \sum_{i=1}^N \sum_{q=1}^Q \tau_{iq}(t) \log(\alpha_q(t)) + \sum_{t=1}^T \sum_{j=1}^M \sum_{\ell=1}^L \eta_{j\ell}(t) \log(\beta_\ell(t)) - \sum_{t=1}^T \sum_{i=1}^N \sum_{q=1}^Q \tau_{iq}(t) \log \tau_{iq}(t) + \\
& \quad - \sum_{t=1}^T \sum_{j=1}^M \sum_{\ell=1}^L \eta_{j\ell}(t) \log(\eta_{j\ell}(t)) - \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^M \left( \delta_{ij}(t) \log(\delta_{ij}(t)) + (1 - \delta_{ij}(t)) \log(1 - \delta_{ij}(t)) \right).
\end{aligned}$$

11

## The inference: VEM Algorithm

### ■ VE-Step: Lower bound maximization with respect to $q(A, Z, W)$ .

The optimal sequential updates of the variational distributions are computed through:

- $\log q^*(A) = E_{W,Z}[\log p(X, A, Z, W | \theta)]$
- $\log q^*(Z) = E_{W,A}[\log p(X, A, Z, W | \theta)]$
- $\log q^*(W) = E_{A,Z}[\log p(X, A, Z, W | \theta)]$

### ■ M-Step: Lower bound maximization with respect to

$\theta = (\alpha(t), \beta(t), \pi(t), \Lambda)$ .

- The derived optimal update of  $\Lambda$  is:

$$\hat{\Lambda}_{q\ell} = \frac{\sum_{i=1}^N \sum_{j=1}^M \sum_{t=1}^T \left\{ \tau_{iq}(t) \eta_{j\ell}(t) (X_{ij}(t) - \delta_{ij}(t) X_{ij}(t)) \right\}}{\sum_{i=1}^N \sum_{j=1}^M \sum_{t=1}^T \left\{ \tau_{iq}(t) \eta_{j\ell}(t) (1 - \delta_{ij}(t)) \right\}}$$

- The optimal updates of  $\alpha(t), \beta(t)$  and  $\pi(t)$  are obtained through a stochastic gradient descent optimization process.

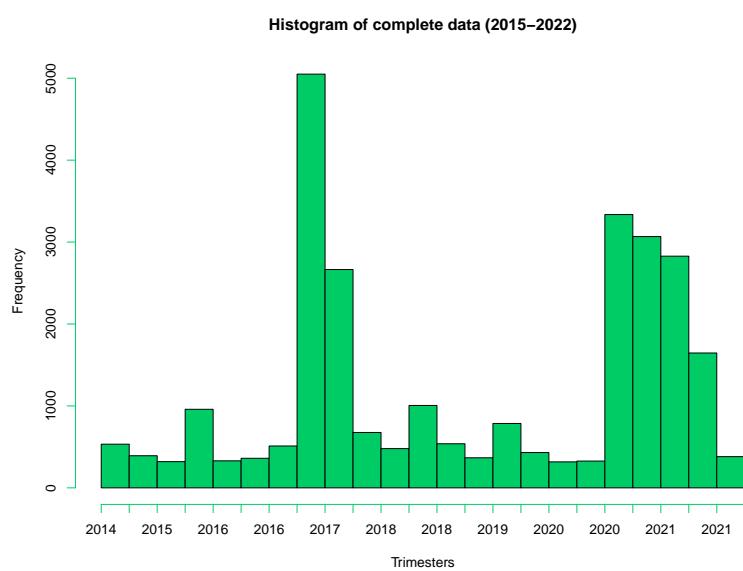
12

## Analysis of the adverse drug reaction dataset

We consider adverse drug reaction (ADR) data collected by the Regional Center of Pharmacovigilance (RCPV), located in the University Hospital of Nice:

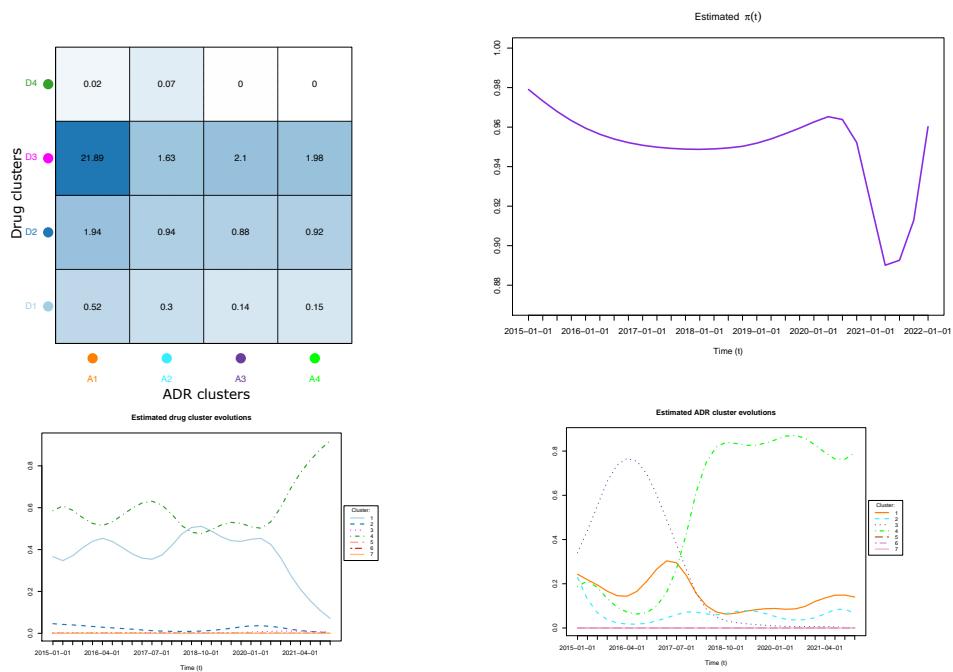
- 2.3 million inhabitants
- several channels (e.g. website form, email, etc)
- time horizon of 7 years (trimester as unity measure)
- 27 754 notifications in the dataset
- only drugs and ADRs notified more than 20 times are considered
- the resulting dataset contains 236 drugs, 324 ADRs and 29 trimesters

## Analysis of the adverse drug reaction data set



**Figure:** Frequency of declarations received by the pharmacovigilance center from January 2015 to March 2022, sorted by month.

## Analysis of the adverse drug reaction data set



15

## Analysis of the adverse drug reaction data set

Major updates in Marchello et al., 2024: i) LSTM neural nets ii) Bayesian change point detection

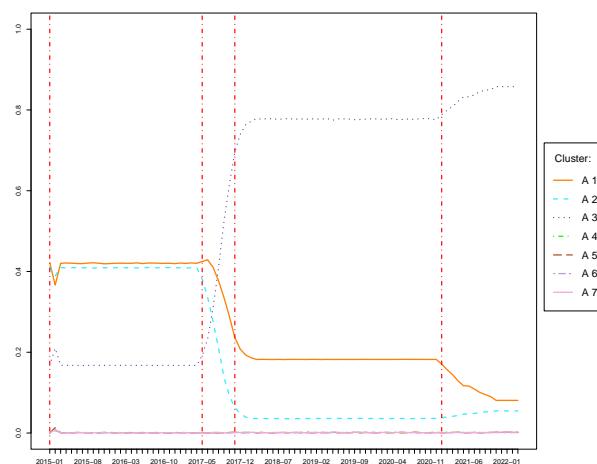


Figure: Clusters of AEs. Evolutions and change points.

16



Thank you for your attention



## Vincent Vandewalle

*Material list:*

## Adrian Raftery

*Material list:*

## **Michael Fop**

### *Material list:*

Clarke C.J., Fop M. (2025) A Latent Position Co-Clustering Model for Multiplex Networks. Unpublished manuscript.

# A Latent Position Co-Clustering Model for Multiplex Networks

C.J. Clarke<sup>\*1, 2</sup> and Michael Fop<sup>†1</sup>

<sup>1</sup>School of Mathematics and Statistics, University College Dublin, Ireland

<sup>2</sup>Taighde Éireann Centre for Research Training in Foundations of Data Science

July 14, 2025

## Abstract

Multiplex networks are increasingly common across diverse domains, motivating the development of clustering methods that uncover patterns at multiple levels. Existing approaches typically focus on clustering either entire networks or nodes within a single network. We address the lack of a unified latent space framework for simultaneous network- and node-level clustering by proposing a latent position co-clustering model (LaPCoM), based on a hierarchical mixture-of-mixtures formulation. LaPCoM enables co-clustering of networks and their constituent nodes, providing joint dimension reduction and two-level cluster detection. At the network level, it identifies global homogeneity in topological patterns by grouping networks that share similar latent representations. At the node level, it captures local connectivity and community patterns. The model adopts a Bayesian nonparametric framework using a mixture of finite mixtures, which places priors on the number of clusters at both levels and incorporates sparse priors to encourage parsimonious clustering. Inference is performed via Markov chain Monte Carlo with automatic selection of the number of clusters. LaPCoM accommodates both binary and count-valued multiplex data. Simulation studies and comparisons with existing methods demonstrate accurate recovery of latent structure and clusters. Applications to real-world social multiplexes reveal interpretable network-level clusters aligned with context-specific patterns, and node-level clusters reflecting social patterns and roles.

---

<sup>\*</sup>courtney.clarke@ucdconnect.ie  
<sup>†</sup>michael.fop@ucd.ie

## 1 Introduction

Networks are widely used to model pairwise relationships among individuals, with applications spanning the social, biological, and physical sciences. A multiplex is a collection of networks (or views) that share a common set of nodes but have different types of relationships between the nodes in each view. Natural examples of multiplex networks arise in a variety of domains, including social networks (D’Angelo et al., 2019; Ren et al., 2023), longitudinal social networks such as co-voting networks (Yin et al., 2022), brain connectivity networks (Taya et al., 2016; Lunagómez et al., 2021; Yin et al., 2022), and systems biology, particularly in the study of cancer gene interactions (Lunagómez et al., 2021).

Clustering and community detection are common tasks in the analysis of network data. Traditional approaches to network and multiplex clustering focus on either clustering entire networks or clustering nodes within a single network. However, in some applications, such as multiplex social networks capturing different types of relationships among the same individuals (e.g., Magnani et al., 2013; D’Angelo et al., 2023), it may be desirable to perform *co-clustering*, inspired by classical data analysis methods that simultaneously cluster rows (observations) and columns (variables) in a data matrix (e.g., Biernacki et al., 2023; Battaglia et al., 2024). We extend this idea to multiplex networks by proposing a model to jointly cluster both the networks and their constituent nodes. This approach enables multilevel insights into the underlying structure of multiplex data. Indeed, clustering at both levels can yield a deeper understanding of complex systems by revealing patterns across and within networks.

With a focus on modelling relations between nodes, a common model for analysing network data is the latent position model (LPM, Hoff et al., 2002), which embeds nodes in a low-dimensional latent space and posits that the probability of an edge between two nodes depends on their relative positions. The LPM has become a foundational tool in network analysis and has inspired numerous extensions. An important extension of the LPM is the latent position cluster model (LPCM, Handcock et al., 2007), which incorporates a Gaussian mixture model to induce clustering of the nodes in a network. Several extensions of the LPM and LPCM have been proposed to analyse multiplex networks. For instance, Gollini and Murphy (2016) introduced the latent space joint model, where a single latent space is assumed to have generated the multiplex. However, inference is carried out in a variational framework where a collection of variational latent spaces, one for each network, are averaged to estimate the generating latent space. Similarly, D’Angelo et al. (2019) assumed a common latent space shared across networks, which may overlook network-specific variation, while D’Angelo et al. (2023) extended this approach to clustering the nodes of a multiplex by adapting the LPCM within a Bayesian nonparametric (BNP) framework. Salter-Townshend and McCormick (2017) proposed a multivariate Bernoulli model, using a conditional LPM to model dyadic dependence within each layer. Sweet et al. (2013) introduced a hierarchical framework that extends single-network models, including the LPM, to multiplex data. However, this requires estimating a separate latent space for each network, which becomes computationally intensive as

the number of networks increases.

Several network-clustering methods have been developed, where entire networks are grouped based on structural characteristics. These approaches typically use distance- or summary-statistic-based representations of networks, followed by standard clustering algorithms. However, they often rely on additional information, such as nodal class labels (Sweet et al., 2019), node attributes (Brandes et al., 2011), or network-level labels (Lunagómez et al., 2021). Building on the stochastic block model (SBM, Lee and Wilkinson, 2019), recent work has proposed co-clustering frameworks that simultaneously cluster both nodes and networks. These include methods based on spectral clustering (Pensky and Wang, 2024), sparse subspace techniques (Noroozi and Pensky, 2024), tensor decompositions (Jing et al., 2021), minimisation algorithms (Fan et al., 2022), and hierarchical or multilayer SBM extensions (Chabert-Liddell et al., 2024; Stanley et al., 2016; Josephs et al., 2025).

In addition to latent space and SBM approaches, alternative approaches based on mixture models offer a flexible framework for capturing heterogeneity in multiplex networks by representing observed networks as draws from a mixture of latent structures. Within this class, both finite and nonparametric mixtures have been explored. For example, mixtures of exponential random graph models (ERGMs) have been applied using Dirichlet process mixtures (DPMs) (Ren et al., 2023) and finite mixtures (Yin et al., 2022). Rebaafka (2024) proposed a finite mixture of SBMs for networks with varying node sets, using hierarchical clustering to infer the number of components. Mantziou et al. (2024) introduced a mixture of measurement error models with latent representative networks, extending to node-level clustering via block structures. This was further developed by Barile et al. (2024) into a BNP approach using a location-scale DPM. Other approaches include mixtures of network-clustering probability models, such as generalised linear (mixed) models (Signorelli and Wit, 2020), and latent space mixtures with component-specific embeddings and nonparametric priors (Durante et al., 2017).

In this paper, we propose a *latent position co-clustering model* (*LaPCoM*) for multiplex network data. Existing methods largely focus on node-clustering or network-clustering only, often require a fixed number of clusters, and offer limited interpretability. The main contribution of *LaPCoM* is to provide a co-clustering framework that simultaneously identifies network-level clusters and node-level clusters within each network-level cluster, leveraging the interpretability of latent space models. Each network-level cluster is represented by a distinct latent space, providing a visual and low dimensional summary of the shared structural features of those networks. Within each latent space, a second mixture model, based on the LPCM, is used to uncover communities of nodes. Importantly, *LaPCoM* employs a BNP approach via a mixture of finite mixtures (MFM) model (Frühwirth-Schnatter et al., 2021) at both levels, enabling automatic inference of the number of clusters from the data and principled uncertainty quantification. Informed by foundational work in latent position modelling (Hoff et al., 2002) and its extensions to clustering and multiplex settings (Handcock et al., 2007; Durante et al., 2017; D’Angelo et al., 2019, 2023), *LaPCoM* unifies these ideas into a cohesive framework for simultaneous

co-clustering and low-dimensional representation of multiplex networks.

The paper is structured as follows: Section 2 describes the LPM, the LPCM and the proposed *LaPCoM*; Section 3 outlines the inferential procedure; Section 4 explores the performance of *LaPCoM* on simulated data; Section 5 applies the proposed *LaPCoM* to a selection of illustrative datasets; and Section 6 concludes with a discussion and potential avenues for future research.

The R (R Core Team, 2023) code to implement *LaPCoM* is freely available at <https://github.com/cjclarke258/LaPCoM>.

## 2 Latent Position Co-Clustering Model

### 2.1 The Latent Position Model

A *network* is a data structure which records relations among a set of  $N$  nodes, describing how these nodes are connected to one another. A network is typically represented by an  $N \times N$  square adjacency matrix  $\mathbf{Y}$  with element  $y_{ij}$  equal to 1 if nodes  $i$  and  $j$  are connected, and 0 otherwise. This particular representation is specific to binary networks; however, other types of networks exist, such as weighted networks, where the ties between the nodes of the network have weights assigned to them. For example, one can consider count networks in which the elements of the adjacency matrix can take non-negative integer values.

The latent position model (LPM, Hoff et al., 2002) models the probability of an edge between nodes  $i$  and  $j$  based on their unobserved positions  $\mathbf{z}_i$  and  $\mathbf{z}_j$  in a  $p$ -dimensional latent space. Edges are assumed independent given the latent positions, with nodes closer in the latent space more likely to connect. Self-loops are excluded ( $y_{ii} = 0 \forall i$ ). The model assumes that  $y_{ij} \sim \mathcal{P}(\lambda_{ij})$ , where  $\mathcal{P}$  is an appropriate distribution, with  $\lambda_{ij} = \mathbb{E}[y_{ij}]$  and  $f(\lambda_{ij}) = \alpha - \|\mathbf{z}_i - \mathbf{z}_j\|_2^2$ , where  $f$  is an appropriate link function. For example, the logit link function and the Bernoulli distribution are used when modelling binary networks (Hoff et al., 2002), while a log link and the Poisson distribution may be used when the edge weights are counts (Gwee et al., 2025). Additionally,  $\alpha$  is an intercept controlling overall connectivity, and the squared Euclidean distance penalises long-range connections (Gollini and Murphy, 2016).

The latent position cluster model (LPCM, Handcock et al., 2007) extends the LPM by assuming that the latent positions arise from a finite mixture of  $K$  multivariate normals:  $\mathbf{z}_i \sim \sum_{k=1}^K \pi_k \mathcal{MVN}_p(\boldsymbol{\mu}_k, \sigma_k^2 \mathbb{I}_p)$ , where  $\pi_k$  is the cluster probability,  $\boldsymbol{\mu}_k$  the cluster mean, and  $\sigma_k^2 \mathbb{I}_p$  a spherical covariance matrix reflecting rotational invariance. This enables community detection while preserving latent space interpretability. Extensions to multiplex networks include Gollini and Murphy (2016), D'Angelo et al. (2019, 2023), and Durante et al. (2017).

### 2.2 A Latent Position Co-Clustering Model for Multiplex Networks (*LaPCoM*)

*LaPCoM* is a mixture-of-mixtures model that clusters networks (views) within a

multiplex while simultaneously clustering the nodes within each network-level group. The model consists of a two-level mixture formulation: a top-level mixture captures structural similarities across networks, while a lower-level mixture clusters nodes within each network-level group based on their connectivity patterns. This hierarchical formulation enables joint modelling of global (network-level) structure and local (node-level) variation. Global homogeneity reflects shared topological features across views, where nodes exhibit similar connectivity profiles along certain relational dimensions. Local heterogeneity captures within-network features, such as community structure or node-specific roles.

In the following, we consider generic valued multiplex networks, represented by a collection of adjacency matrices  $\mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(m)}, \dots, \mathbf{Y}^{(M)}$ , where each entry  $y_{ij}^{(m)}$  represents the relationship between nodes  $i$  and  $j$  in view  $m$ ,  $m = 1, \dots, M$ . This relationship can be binary or weighted, depending on the nature of the data. We model each network using an appropriate distribution  $\mathcal{P}$  with associated link function  $f$ , consistent with the general latent position formulation introduced above.

Our model assumes that each network  $\mathbf{Y}^{(m)}$  is generated from a network-level finite mixture distribution:

$$\mathbf{Y}^{(m)} \sim \sum_{g=1}^G \tau_g \prod_{\substack{i,j \\ i \neq j}} \mathcal{P}(\lambda_{g,ij}), \quad m = 1, \dots, M,$$

where  $G$  is the number of components in the network-level mixture,  $\boldsymbol{\tau}$  are the mixing proportions such that  $\tau_g \geq 0$  for all  $g$  and  $\sum_{g=1}^G \tau_g = 1$ . The parameter  $\lambda_{g,ij}$  denotes the expected value  $\mathbb{E}[y_{ij}^{(m)}]$ , and has the associated link function  $f(\lambda_{g,ij}) = \alpha - \|\mathbf{z}_{g,i} - \mathbf{z}_{g,j}\|_2^2$ . For example, if the multiplex consists of count-valued adjacency matrices, then  $\lambda_{g,ij}$  represents the rate parameter corresponding to component  $g$ ,  $\mathcal{P}$  is the Poisson distribution, and the log link function implies  $\log \lambda_{g,ij} = \alpha - \|\mathbf{z}_{g,i} - \mathbf{z}_{g,j}\|_2^2$ . The latent positions  $\mathbf{z}_{g,i}$  capture the connectivity patterns of the nodes across all network views arising from component  $g$ .

To induce clustering of the nodes in addition to clustering of the networks, *LaPCoM* assumes that each component-specific latent space has positions originating from a node-level finite mixture distribution:

$$\mathbf{z}_{g,i} \sim \sum_{k=1}^{K_g} \pi_{gk} \mathcal{MVN}_2(\boldsymbol{\mu}_{gk}, \boldsymbol{\Sigma}_{gk}), \quad i = 1, \dots, N, \quad (1)$$

where  $K_g$  is the number of components in the node-level mixture within the  $g^{\text{th}}$  network-level component,  $\boldsymbol{\pi}_g$  represents the mixing proportions, defined such that  $\pi_{gk} \geq 0$  for all  $k$  and  $\sum_{k=1}^{K_g} \pi_{gk} = 1$ , and  $\boldsymbol{\Sigma}_{gk} = \text{diag}(\sigma_{gk,1}^2, \sigma_{gk,2}^2)$ . Note that, for visualisation purposes and ease of interpretability, we will focus on latent spaces of dimension  $p = 2$ . *LaPCoM* incorporates the following prior distributions for the mixture components:

$$\begin{aligned} \boldsymbol{\mu}_{gk} &\sim \mathcal{MVN}_2(\mathbf{0}, \mathbb{I}_2), \quad k = 1, \dots, K_g, \text{ and } g = 1, \dots, G, \\ \sigma_{gk,q}^2 &\sim \mathcal{IG}(u_{\sigma^2}, v_{\sigma^2}), \quad q = 1, 2, \quad k = 1, \dots, K_g, \text{ and } g = 1, \dots, G, \end{aligned}$$

where  $\mathcal{IG}$  denotes the Inverse-Gamma distribution.

At both network and node level, *LaPCoM* employs a mixture of finite mixtures (MFMs) framework. Unlike standard finite mixtures that fix the number of components  $K$ , MFMs adopt a Bayesian nonparametric (BNP) approach by placing a prior on  $K$ , allowing for uncertainty while retaining a finite mixture structure (Miller and Harrison, 2018). A key development in this framework is the extension of Frühwirth-Schnatter et al. (2021), who introduced a dynamic MFM, where the concentration parameter of the symmetric Dirichlet prior on the mixture weights depends on  $K$ . This leads to a richer, generalised class of BNP models. Crucially, they highlight the important distinction between the number of components  $K$  and the number of clusters (active components),  $K_+$ . The distribution of  $K_+$  is influenced by both the prior on  $K$  and the Dirichlet hyperparameter, allowing for greater modelling flexibility. In contrast, Dirichlet process mixtures (DPMs) assume an infinite number of components and focus on estimating the number of clusters  $K_+$ . While widely used, DPMs lack the finite tractability of MFMs, which can approximate DPMs asymptotically when the prior on  $K$  is diffuse. Notably, Frühwirth-Schnatter et al. (2021) employ a three-parameter translated beta-negative-binomial (BNB) prior distribution on  $K$ , which provides flexible control over both the expected number of components and the tail behaviour of the induced prior on  $K_+$ . A detailed justification for the use of this MFM framework over alternatives, such as overfitted finite mixtures (Malsiner-Walli et al., 2016), is provided in Section 10 of the Supplementary Material.

Following Frühwirth-Schnatter et al. (2021), *LaPCoM* places a translated BNB prior on the number of components at both network and node level:

$$G - 1 \sim \mathcal{BNB}(a_G, b_G, c_G) \quad K_g - 1 \sim \mathcal{BNB}(a_K, b_K, c_K) \quad g = 1, \dots, G,$$

where  $\mathcal{BNB}$  denotes the beta-negative-binomial (BNB) distribution. Additionally, we employ a dynamic MFM formulation in which the hyperparameter of the Dirichlet prior on the mixture weights depends on the number of components, inducing a shrinkage that encourages sparsity in the mixture weights, leading to the emptying of superfluous components. Specifically:

$$\tau \sim \mathcal{D}\left(\frac{e}{G}\right), \quad \pi_g \sim \mathcal{D}\left(\frac{w_g}{K_g}\right), \quad g = 1, \dots, G,$$

where  $\mathcal{D}$  denotes the Dirichlet distribution.

To implement both network-level and node-level clustering within our model, we introduce a set of latent allocation variables. At the network level, we define the  $M \times G$  binary matrix  $\mathbf{C}$ , such that  $C_g^{(m)} = 1$  if network  $m$  is allocated to component  $g$ . Within each network-level cluster  $g$ , we define the  $N \times K_g$  binary matrix  $\mathbf{S}_g$ , such that  $S_{gk}^{(i)} = 1$  if node  $i$  is allocated to component  $k$ .

The remaining priors of the model are assumed to be as follows:

$$\begin{aligned} e &\sim \mathcal{F}(l_G, r_G), \\ \mathbf{C}^{(m)} &\sim \mathcal{MN}(\boldsymbol{\tau}) \quad \forall m = 1, \dots, M, \\ w_g &\sim \mathcal{F}(l_K, r_K) \quad \forall g = 1, \dots, G, \\ \mathbf{S}_g^{(i)} &\sim \mathcal{MN}(\boldsymbol{\pi}_g) \quad \forall g = 1, \dots, G \text{ and } \forall i = 1, \dots, N, \\ \alpha &\sim \mathcal{N}(m_\alpha, s_\alpha^2), \end{aligned}$$

where,  $\mathcal{F}$  denotes the Fisher-Snedecor distribution,  $\mathcal{MN}$  denotes the multinomial distribution and  $\mathcal{N}$  denotes the univariate normal distribution.

### 3 Inference

#### 3.1 MCMC Algorithm

With hyperparameters  $\mathbf{H} = \{a_G, b_G, c_G, l_G, r_G, m_\alpha, s_\alpha, a_K, b_K, c_K, l_K, r_K, u_{\sigma^2}, v_{\sigma^2}\}$ , the joint posterior distribution of *LaPCoM* is as follows:

$$\begin{aligned} &\mathbb{P}\left(G, e, \boldsymbol{\tau}, \mathbf{C}, \alpha, \left\{K_g, w_g, \boldsymbol{\pi}_g, \mathbf{S}_g, \{\boldsymbol{\mu}_{gk}, \boldsymbol{\Sigma}_{gk}\}_{k=1}^{K_g}\right\}_{g=1}^G, \{\mathbf{Z}_g\}_{g=1}^G \mid \mathbf{Y}\right) \\ &\propto \mathbb{P}(\mathbf{Y} \mid \mathbf{C}, \alpha, \{\mathbf{Z}_g\}_{g=1}^G) \times \mathbb{P}(\mathbf{C} \mid \boldsymbol{\tau}) \times \mathbb{P}(\boldsymbol{\tau} \mid G, e) \times \mathbb{P}(G \mid \mathbf{H}) \times \mathbb{P}(e \mid \mathbf{H}) \times \mathbb{P}(\alpha \mid \mathbf{H}) \\ &\quad \times \prod_{g=1}^G \mathbb{P}(\mathbf{Z}_g \mid \mathbf{S}_g, \{\boldsymbol{\mu}_{gk}, \boldsymbol{\Sigma}_{gk}\}_{k=1}^{K_g}) \times \prod_{g=1}^G \prod_{k=1}^{K_g} \mathbb{P}(\boldsymbol{\mu}_{gk} \mid \mathbf{H}) \times \prod_{g=1}^G \prod_{k=1}^{K_g} \mathbb{P}(\boldsymbol{\sigma}_{gk}^2 \mid \mathbf{H}) \\ &\quad \times \prod_{g=1}^G \mathbb{P}(\mathbf{S}_g \mid \boldsymbol{\pi}_g) \times \prod_{g=1}^G \mathbb{P}(\boldsymbol{\pi}_{gk} \mid K_g, w_g) \times \prod_{g=1}^G \mathbb{P}(K_g \mid \mathbf{H}) \times \prod_{g=1}^G \mathbb{P}(w_g \mid \mathbf{H}). \end{aligned}$$

Inference is carried out using a Metropolis-within-Gibbs MCMC algorithm. Additionally, the algorithm incorporates the telescoping sampling procedure of [Frühwirth-Schnatter et al. \(2021\)](#) to adaptively sample mixture components and cluster allocations. A brief description of the MCMC algorithm is described below. The Supplementary Material includes a complete list of notation (Section 1), details on our choice of hyperparameters (Section 2), detailed derivations of all full conditionals (Section 3), a pseudocode description of the MCMC algorithm (Section 4), and details of the algorithm initialisation (Section 5).

The MCMC algorithm iterates through the following steps, for  $t = 1, \dots, T$ , where  $t$  denotes the current iteration and  $T$  is the total number of iterations:

1. Sample  $C_g^{(m)[t+1]}$  for each network  $m = 1, \dots, M$  from a multinomial distribution such that  $Pr(C^{(m)[t+1]} = g \mid \dots) \propto \tau_g^{[t]} \prod_{i \neq j} p(y_{ij}^{(m)}; \lambda_{g,ij}^{[t]})$ , where  $p$  is the pdf of distribution  $\mathcal{P}$  and  $f(\lambda_{g,ij}^{[t]}) = \alpha^{[t]} - \|\mathbf{z}_{g,i}^{[t]} - \mathbf{z}_{g,j}^{[t]}\|_2^2$ .

2. Compute  $M_g^{[t+1]}$  for each  $g = 1, \dots, G^{[t]}$ , the number of networks allocated to each network-level component; determine  $G_+^{[t+1]}$ , the number of active (non-empty) components; relabel the network-level mixture parameters so that the first  $G_+^{[t+1]}$  components are non-empty.
3. For  $g = 1, \dots, G_+^{[t+1]}$ :
  - (a) Propose a block update  $\hat{\mathbf{Z}}_g$ , conditioned on  $\mathbf{S}_g^{[t]}$ , with proposal distribution  $\hat{\mathbf{z}}_{g,i} \sim \mathcal{MVN}\left(\mathbf{z}_{g,i}^{[t]}, \delta_Z^2 \Sigma_{gk}^{[t]}\right)$ ,  $i = 1, \dots, N$ , where  $\delta_Z$  is a scaling factor. Accept  $\hat{\mathbf{Z}}_g$  as  $\mathbf{Z}_g^{[t+1]}$  with the appropriate probability; otherwise, set  $\mathbf{Z}_g^{[t+1]} = \mathbf{Z}_g^{[t]}$ .
  - (b) For each  $i = 1, \dots, N$ , sample  $S_{gk}^{(i)[t+1]}$  from a multinomial distribution with  $Pr(S_g^{(i)[t+1]} | \dots) \propto \pi_{gk}^{[t]} |\Sigma_{gk}^{[t]}|^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (\mathbf{z}_{g,i}^{[t+1]} - \boldsymbol{\mu}_{gk}^{[t]})^T \Sigma_{gk}^{[t]-1} (\mathbf{z}_{g,i}^{[t+1]} - \boldsymbol{\mu}_{gk}^{[t]}) \right]$ .
  - (c) Calculate  $N_{gk}^{[t+1]}$  for each  $k = 1, \dots, K_g^{[t]}$ , the number of nodes allocated to each node-level component; determine  $K_+^{[t+1]}$ , the number of active components; and relabel the node-level mixture parameters so that the first  $K_+^{[t+1]}$  components are non-empty.
  - (d) For each  $k = 1, \dots, K_+^{[t+1]}$ , sample  $\boldsymbol{\mu}_{gk}^{[t+1]}$  from  $\mathcal{MVN}(\boldsymbol{\mu}_{gk}^*, \Sigma_{gk}^*)$  and  $\sigma_{gk,q}^{2[t+1]}$  for  $q = 1, 2$  from  $\mathcal{IG}(u_{\sigma^2}^*, v_{\sigma^2}^*)$ , where  $\Sigma_{gk}^*$ ,  $\boldsymbol{\mu}_{gk}^*$ ,  $u_{\sigma^2}^*$ , and  $v_{\sigma^2}^*$  are updated based on the current model parameters, as derived in Section 3 of the Supplementary Material.
  - (e) Sample  $K_g^{[t+1]}$  from a multinomial distribution with probabilities  $\mathbb{P}(k^* | \mathbf{S}_g^{[t+1]}, w_g^{[t]}) \propto \mathbb{P}(k^*) \frac{(w_g^{[t]})^{K_g^{[t+1]}} k^*!}{(k^*)^{K_g^{[t+1]}} (k^* - K_g^{[t+1]})!} \prod_{k=1}^{K_g^{[t+1]}} \frac{\Gamma(N_{gk}^{[t+1]} + \frac{w_g^{[t]}}{k^*})}{\Gamma(1 + \frac{w_g^{[t]}}{k^*})}$ , where  $\mathbb{P}(k^*)$  is the prior distribution,  $k^*$  ranges from  $K_+^{[t]}$  to  $K_{\max}$ , and  $K_{\max}$  is the maximum number of node-level components considered;  $\Gamma(\cdot)$  is the gamma function.
  - (f) Propose  $\log(\hat{w}_g)$  from the proposal distribution  $\mathcal{N}(\log(w_g^{[t]}), s_w^2)$ . Accept  $\hat{w}_g$  as  $w_g^{[t+1]}$  with the appropriate probability; otherwise, set  $w_g^{[t+1]} = w_g^{[t]}$ .
  - (g) If  $K_g^{[t+1]} > K_+^{[t+1]}$ , add  $K_g^{[t+1]} - K_+^{[t+1]}$  empty components by sampling  $\boldsymbol{\mu}_{gk}^{[t+1]}$  and  $\sigma_{gk}^{2[t+1]}$  from their respective priors, for  $k = K_+^{[t+1]} + 1, \dots, K_g^{[t+1]}$ .
  - (h) Sample  $\boldsymbol{\pi}_g^{[t+1]}$  from  $\mathcal{D}(\boldsymbol{\psi}_g^{[t+1]})$ , where  $\boldsymbol{\psi}_g^{[t+1]} = \frac{w_g^{[t+1]}}{K_g^{[t+1]}} + N_{gk}^{[t+1]}$ ,  $k = 1, \dots, K_g^{[t+1]}$ .
4. Propose  $\hat{\alpha}$  using the proposal distribution  $\mathcal{N}(\alpha^{[t]}, \delta_\alpha^2 s_\alpha^{*2})$ , where  $\delta_\alpha$  is a scaling factor. Accept  $\hat{\alpha}$  as  $\alpha^{[t+1]}$  with the appropriate probability; otherwise, set  $\alpha^{[t+1]} = \alpha^{[t]}$ .

5. Sample  $G^{[t+1]}$  from a multinomial distribution with probabilities  $\mathbb{P}(g^* \mid \mathbf{C}^{[t+1]}, e^{[t]}) \propto \mathbb{P}(g^*) \frac{(e^{[t]})^{G_+^{[t+1]}} g^*!}{(g^*)^{G_+^{[t+1]}} (g^* - G_+^{[t+1]})!} \prod_{g=1}^{G_+^{[t+1]}} \frac{\Gamma(M_g^{[t+1]} + \frac{e^{[t]}}{g^*})}{\Gamma(1 + \frac{e^{[t]}}{g^*})}$ , where  $\mathbb{P}(g^*)$  is the prior distribution, and  $g^* = G_+^{[t]}, G_+^{[t]} + 1, \dots, G_{\max}$ , with  $G_{\max}$  the maximum number of network-level components considered.
6. Propose  $\log(\hat{e})$  from the proposal distribution  $\mathcal{N}(\log(e^{[t]}), s_e^2)$ . Accept  $\hat{e}$  as  $e^{[t+1]}$  with the appropriate probability; otherwise, set  $e^{[t+1]} = e^{[t]}$ .
7. If  $G^{[t+1]} > G_+^{[t+1]}$ , add  $G^{[t+1]} - G_+^{[t+1]}$  empty components by sampling the corresponding hyperparameters from their priors, for  $g = G_+^{[t+1]} + 1, \dots, G^{[t+1]}$ , assuming no clustering structure in these additional latent spaces.
8. Sample  $\boldsymbol{\tau}^{[t+1]}$  from  $\mathcal{D}(\zeta_1^{[t+1]}, \dots, \zeta_{G^{[t+1]}}^{[t+1]})$ , where  $\zeta_g^{[t+1]} = \frac{e^{[t+1]}}{G^{[t+1]}} + M_g^{[t+1]}$ .

### 3.2 Hyperparameter Choices

Table 1 summarises our hyperparameter choices; we provide some details below. We use non-informative priors for the intercept  $\alpha$  and node-level cluster means  $\boldsymbol{\mu}_{gk}$ . Cluster variances  $\sigma_{gk,q}^2$  follow an  $\mathcal{IG}(u_{\sigma^2}, v_{\sigma^2})$  prior, with parameters chosen to keep clusters tight, yielding expected variances around 0.2 for small to moderate networks and 0.1 for larger ones. We adopt a  $\mathcal{BNB}(8, 18, 10)$  prior on the number of mixture components  $G$  and  $K_g$ , that favours a moderate number of clusters but allows a heavy tail for flexibility and avoids excessive shrinkage. Initial values  $G_0$  and  $K_0$  are set to 2, balancing flexibility and shrinkage. Coupled with a Dirichlet shrinkage prior on mixing proportions, this enables adaptive growth of the number of components. Upper bounds  $G_{\max}$  and  $K_{\max}$  are set lower than [Frühwirth-Schnatter and Malsiner-Walli \(2019\)](#) recommend, to reduce computational cost, with  $K_{\max}$  tailored to network size and  $G_{\max}$  chosen mainly for computational efficiency. Finally, node-level Dirichlet concentration parameters have an  $\mathcal{F}(6, 3)$  prior as in [Frühwirth-Schnatter et al. \(2021\)](#), reflecting moderate homogeneity. Preliminary tests validate these settings as appropriate for the data's complexity and structure. Full details of these choices can be found in Section 2 of the Supplementary Material.

### 3.3 Post-Processing

To ensure valid inference, we address the identifiability issue due to label switching, which arises because the mixture model likelihood is invariant to component label permutations. Following [D'Angelo et al. \(2023\)](#), we combine strategies from [Frühwirth-Schnatter \(2011\)](#) and [Wade and Ghahramani \(2018\)](#). This section describe this process for the network-level mixture; given the co-clustering nature of *LaPCoM*, the same procedure is applied analogously at the node level. As a first step, we apply the method of [Wade and Ghahramani \(2018\)](#) to the posterior samples of the clustering vector  $\mathbf{C}$ , using the variation of information and posterior similarity matrix to obtain an optimal partition,  $\tilde{\mathbf{C}}$ . This is implemented via the `mcclust.ext`

package in R (Wade, 2015). From  $\hat{\mathbf{C}}$ , we obtain a posterior point estimate of the number of network-level clusters,  $\hat{G}_+$ , as the number of unique cluster labels present in the optimal partition.

Next, we cluster the vectorised latent spaces  $\mathbf{Z}_g$  (at the network level) into  $\hat{G}_+$  groups using  $K$ -means, following Frühwirth-Schnatter (2011). This provides a classification sequence for each MCMC iteration, assigning consistent labels to components. A permutation check ensures each sequence is a valid relabelling of  $1, \dots, \hat{G}_+$ ; only iterations satisfying this condition are retained. Valid iterations are then used to relabel all component-specific parameters and the cluster allocation vector  $\mathbf{C}$  for coherence. This relabelling procedure is applied at the node-level using the cluster means  $\boldsymbol{\mu}_{gk}$  as the component-specific parameter.

To ensure label consistency across parameters, we align the optimal clustering vector  $\hat{\mathbf{C}}$  with its relabelled counterpart  $\hat{\mathbf{C}}_{\text{relabelled}}$  by comparing each  $\mathbf{C}^{[t]}$  and  $\mathbf{C}_{\text{relabelled}}^{[t]}$  via cross-tabulation. The most frequent label-matching pattern across iterations is used to permute  $\hat{\mathbf{C}}$ , yielding the final aligned clustering  $\hat{\mathbf{C}}^*$ . This step ensures coherence between  $\hat{\mathbf{C}}$  and the labels of cluster-specific parameters such as  $\{\mathbf{Z}_g\}_{g=1}^{\hat{G}_+}$ , avoiding mismatches that can arise from applying independent post-processing procedures.

The posterior mode of the number of clusters from Wade and Ghahramani (2018) may differ from that implied by the relabelled parameters using Frühwirth-Schnatter (2011), as the two methods target different aspects of the posterior and operate independently. We adopt the mode from the former as our primary estimate and use it as a reference in the Frühwirth-Schnatter (2011) relabelling step. Although reversing this order is possible, it can yield different results, highlighting the impact of methodological choices on posterior summaries. When the posterior is multimodal,

Table 1: Hyperparameter choices.

Parameter	Prior Distribution	Hyperparameters	Recommended Values
$\alpha$	$\alpha \sim \mathcal{N}(m_\alpha, s_\alpha)$	$m_\alpha, s_\alpha$	$m_\alpha = 0, s_\alpha = 1$
$\boldsymbol{\mu}_{gk}$	$\boldsymbol{\mu}_{gk} \sim \mathcal{MVN}_2(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_\mu)$	$\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_\mu$	$\boldsymbol{\mu}_0 = \mathbf{0}, \boldsymbol{\Sigma}_\mu = \mathbb{I}_2$
$\sigma_{gk,q}^2$	$\sigma_{gk,q}^2 \sim \mathcal{IG}(u_{\sigma^2}, v_{\sigma^2})$	$u_{\sigma^2}, v_{\sigma^2}$	$u_{\sigma^2} = \begin{cases} 11 & \text{if } N < 60 \\ 21 & \text{if } N \geq 60 \end{cases}, v_{\sigma^2} = 2$
$G$	$G - 1 \sim \mathcal{BNB}(a_G, b_G, c_G)$	$a_G, b_G, c_G$	$a_G = 8, b_G = 18, c_G = 10$
$G_+$	Induced by $\mathbb{P}(G)$	-	-
$G_0$	-	-	$G_0 = 2$
$G_{\max}$	-	-	$G_{\max} = \begin{cases} 5 & \text{if } M < 60 \\ 10 & \text{if } M \geq 60 \end{cases}$
$e$	$e \sim \mathcal{F}(l_G, r_G)$	$l_G, r_G$	$l_G = 6, r_G = 3$
$K_g$	$K_g - 1 \sim \mathcal{BNB}(a_K, b_K, c_K)$	$a_K, b_K, c_K$	$a_K = 8, b_K = 18, c_K = 10$
$K_{g+}$	Induced by $\mathbb{P}(K_g)$	-	-
$K_0$	-	-	$K_0 = 2$
$K_{\max}$	-	-	$K_{\max} = \begin{cases} \frac{N}{5} + 2 & \text{if } N < 60 \\ \frac{N}{10} + 2 & \text{if } N \geq 60 \end{cases}$
$w_g$	$w_g \sim \mathcal{F}(l_K, r_K)$	$l_K, r_K$	$l_K = 6, r_K = 3$

we select the smaller mode to favour parsimony.

As in standard LPMs, the use of Euclidean distances leads to identifiability issues due to invariance under rotation, reflection, and translation. Following Hoff et al. (2002), we apply a Procrustes transformation to each sampled latent space to resolve this. In *LaPCoM*, this is done “offline”: after MCMC sampling and label correction, each cluster-specific latent space is aligned to the first retained sample for that cluster. The same transformation (scaling, rotation, and translation) is applied to the corresponding cluster means  $\mu_g$ , with only scaling applied to the variances  $\Sigma_g$ , since they are diagonal matrices.

## 4 Simulation Studies

To assess the performance of *LaPCoM*, we conduct two simulation studies: Section 4.1 assesses clustering performance across various scenarios, and Section 4.2 compares *LaPCoM* with four relevant models. In all settings, we align estimated and true cluster labels using the `matchClasses` function from the R package `e1071` (Meyer et al., 2023), which maximises the number of correctly matched cases.

### 4.1 Study 1: Clustering Performance

This simulation study evaluates the clustering performance of *LaPCoM* across eight scenarios, in which different multiplex networks are generated under varying combinations of the number of nodes, number of networks, and number of clusters at both levels. The full details of these scenarios are provided in Table 3 in Section 6 of the Supplementary Material. For each scenario, 10 multiplexes are generated using the specified parameters. Two models are fitted: *LaPCoM* and a simplified variant, *mono-LaPCM*, which clusters the networks only, omitting the node-level mixture as in (1) and instead assumes  $z_{g,i} \sim \mathcal{MVN}_2(\mathbf{0}, \mathbb{I}_2)$ . This comparison enables an assessment of the contribution of the node-level mixture, allowing us to evaluate whether incorporating the additional clustering structure in the latent positions improves model performance. Scenario-specific scaling factors ensure suitable acceptance rates. MCMC chains run for 300,000 iterations, with an additional 90,000 as burn-in and thinning every 300<sup>th</sup> sample to yield 1,000 posterior draws. Network-level allocations,  $\mathbf{C}$ , are initialised via  $K$ -means.

Posterior summaries of the number of network-level clusters,  $G_+$ , inferred from the 10 simulated multiplexes across eight scenarios show that both *LaPCoM* and *mono-LaPCM* accurately recovered  $\hat{G}_+ = 2$  in 100% of draws for all but scenario B. In scenario B, *mono-LaPCM* assigned 80% posterior mass to  $\hat{G}_+ = 2$  and 20% to  $\hat{G}_+ = 1$ , while *LaPCoM* split mass between  $\hat{G}_+ = 2$  (80%) and  $\hat{G}_+ = 4$  (20%). Both models consistently recovered the latent space, each with a mean Procrustes correlation (PC) of 0.99 across all scenarios. PC was calculated using the `protest` function from the `vegan` package (Oksanen et al., 2024). the interquartile range (IQR) was 0.01, indicating low variability in performance. Clustering performance, assessed by the adjusted Rand index (ARI; Hubert and Arabie (1985)), was perfect (mean = 1, standard deviation (SD) = 0) in all but scenario B, where *mono-LaPCM*

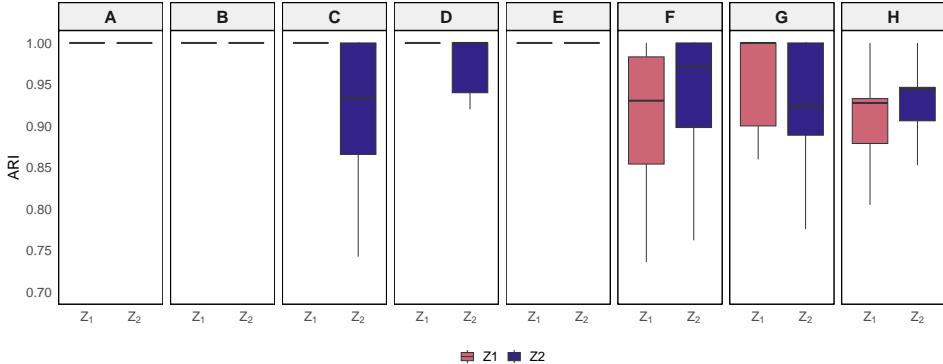


Figure 1: Boxplots comparing the Adjusted Rand Index (ARI) across 10 simulations for each scenario, between the estimated clustering solution to the true clustering solution, illustrating the clustering performance of *LaPCoM*.

achieved a mean ARI of 0.80 (SD = 0.42) and *LaPCoM* 0.97 (SD = 0.09). These results demonstrate comparable performance overall, with improved clustering performance for *LaPCoM* in some settings due to its additional flexibility given by the node-level mixture.

Figure 1 shows node-level clustering performance, assessed via ARI, across 10 simulations per scenario using *LaPCoM* (not applicable to *mono-LaPCM*). Median ARIs exceed 0.92 with IQRs below 0.14. Scenarios A and B, each with one node-level cluster per  $G^* = 2$  latent spaces, achieved perfect median ARIs of 1.00 (IQR = 0.00). In scenarios C and D ( $K_g = \{1, 2\}$ ), clustering improved in D (more nodes), with median ARI for the second latent space increasing from 0.93 (IQR = 0.13) to 1.00 (IQR = 0.06), while the first latent space remained perfect. More complex scenarios G and H ( $K_g = \{2, 3\}$ ) showed a slight drop in median ARI for the first latent space (1.00 to 0.93) and a minor increase for the second (0.92 to 0.95), with reduced IQRs in H indicating greater stability. These results confirm that *LaPCoM* achieves good node-level clustering performance across diverse multiplex settings.

## 4.2 Study 2: Comparison with Other Methods

This simulation study compares the clustering performance of *LaPCoM* against four competing models across five scenarios, in which different multiplex networks are generated under varying combinations of the number of nodes, number of networks, and number of clusters at both levels. For each scenario, 10 multiplexes are generated, using the parameters provided in Table 4 in Section 6 of the Supplementary Material.

While many relevant methods exist (see Section 1), we select one representative from each major category for comparison: a mixture of LPMs (*PopNet*, [Durante et al., 2017](#)), a finite mixture of SBMs ([Rebafka, 2024](#)) via the *graphclust* R package

(Rebafka, 2023), a mixture of measurement error models (Mantziou et al., 2024), and a mixture of generalised linear (mixed) models (Signorelli and Wit, 2020). Code for all models is publicly available. Each model represents a distinct latent structure approach.

As in the first simulation study, *LaPCoM* adjusts scaling factors for the intercept and latent positions in each scenario to maintain suitable acceptance rates. Mantziou et al. (2024)'s model uses  $C_{\max} = 10$  (maximum number of network-level clusters) and  $B = 2$  (SBM blocks), with other hyperparameters set as recommended or slightly adapted per scenario. Unlike *LaPCoM*, it employs a  $\mathcal{G}(1, 400)$  prior on the Dirichlet concentration parameter to enforce cluster shrinkage. The other three competing models use default hyperparameters. *LaPCoM*, *PopNet*, and Mantziou et al. (2024)'s model run for 100,000 iterations including 30,000 burn-in, and thinning every 100<sup>th</sup> iteration for 700 samples. For *LaPCoM*, network-level allocations,  $\mathbf{C}$ , are initialised via  $K$ -means clustering.

Table 2 summarises a simulation comparing *LaPCoM* with four competing models over five scenarios (10 replicates each). For each simulation, the posterior mode of the number of clusters,  $\hat{G}_+$ , with 80% credible intervals is reported. The Mantziou et al., Signorelli and Wit, and *graphclust* models provided poor performance, generally overestimating or underestimating the number of clusters. *PopNet* reliably recovered the true  $\hat{G}_+$  with narrow intervals. *LaPCoM* matched this accuracy, consistently identifying the correct  $\hat{G}_+$ , though with slightly wider intervals due to its added flexibility.

Table 3 reports mean ARI values concerning network clustering performance (SDs in brackets). *PopNet* achieved perfect ARIs in all five scenarios, followed by *LaPCoM*, which consistently attained mean ARIs above 0.86 with SDs  $\leq 0.17$ . Both models substantially outperformed the other three across all scenarios, highlighting good network clustering performance. These results highlight *LaPCoM*'s performance and suitability for clustering a multiplex.

Of the five models compared, only *LaPCoM*, *graphclust* (Rebafka, 2023), and Mantziou et al.'s model support nodal clustering. *LaPCoM* models networks within a cluster as dependent, sharing a cluster-specific latent space. In a similar manner, Mantziou et al. model networks as noisy observations of a latent representative network. In contrast, Rebafka treat networks as independent realisations of a cluster-

Table 2: Posterior mode of  $G_+$ , denoted  $\hat{G}_+$ , with 80% credible intervals in brackets, for *LaPCoM* and the four competing models under the five simulation scenarios. The true generating number of network-level clusters is denoted by  $G^*$ .

Scenario	$G^*$	Mantziou et al.	Signorelli and Wit	<i>graphclust</i>	<i>PopNet</i>	<i>LaPCoM</i>
I	2	10 (10, 10)	2 (2, 3)	1 (1, 3)	2 (2, 2)	2 (2, 3)
II	2	10 (9, 10)	2 (2, 7)	3 (3, 5)	2 (2, 2)	2 (2, 2)
III	3	10 (10, 10)	2 (2, 2)	5 (4, 6)	3 (3, 3)	3 (3, 3)
IV	4	10 (10, 10)	2 (2, 2)	8 (6, 9)	4 (4, 4)	4 (3, 4)
V	4	10 (10, 10)	2 (2, 2)	10 (5, 11)	4 (4, 4)	4 (3, 4)

specific SBM with network-specific latent variables. Thus, *LaPCoM* and [Mantziou et al.](#) are comparable with a shared focus on interdependence, while the assumptions of *graphclust* differ substantially.

Figure 2 compares node clustering performance via ARI across 10 simulations per scenario for [Mantziou et al.](#)'s method (2a) and *LaPCoM* (2b). Latent spaces differ across scenarios and are not comparable between panels. For example,  $Z_1$  in Scenario I differs from  $Z_1$  in Scenario II. [Mantziou et al.](#)'s method shows poor performance, with median ARIs near 0.00 and an average IQR of 0.25. In contrast, *LaPCoM* achieves median ARIs above 0.90 and an average IQR of 0.07, demonstrating superior node-level clustering performance.

Tables 2 and 3 show that only *LaPCoM* and *PopNet* perform competitively. Our *LaPCoM* offers several advantages over *PopNet*. The average runtime for *LaPCoM* ranged from 3 hours (Scenario I) to 15 hours (Scenario V), compared to 6 and 25 hours respectively for *PopNet*, making *LaPCoM* substantially faster. Additionally, although we ran 100,000 MCMC iterations for convergence, fewer iterations may suffice in practice. Moreover, *LaPCoM* supports clustering for both binary and weighted multiplexes, while *PopNet* handles only binary data. Additionally, *LaPCoM* offers node-level clustering, which *PopNet* lacks.

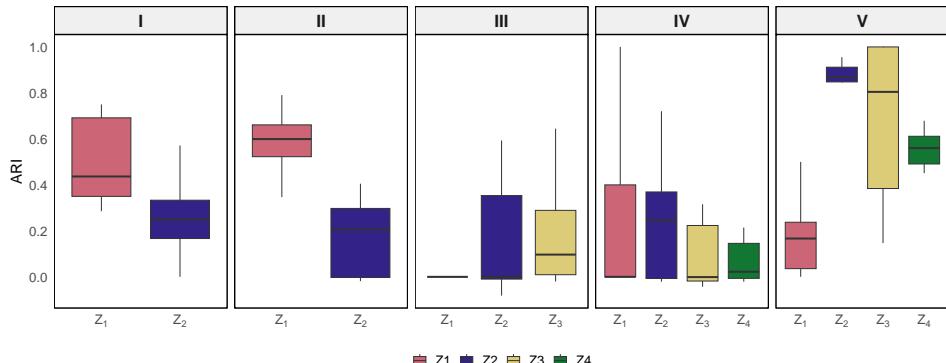
## 5 Illustrative Applications

This section illustrates the application of *LaPCoM* to a variety of multiplex datasets arising in different contexts.

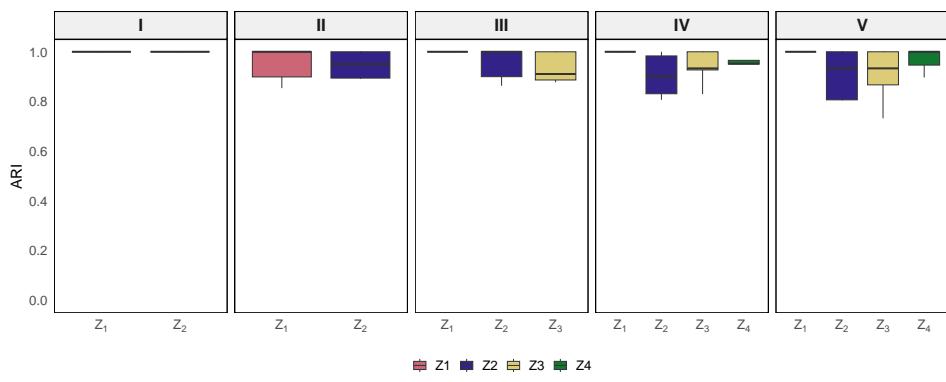
In these illustrative data applications, we run the MCMC algorithm for multiple chains. To reconcile cluster-specific parameters across chains we post-process all chains and identify the overall mode of the number of active clusters,  $\hat{G}_+^*$ . Only chains matching this mode ( $\hat{G}_+^{\text{chain}} = \hat{G}_+^*$ ) are retained. Among these, we compute the log-posterior value for each chain by evaluating the log-posterior at that chain's parameter estimates. We then select the chain with the highest log-posterior value and use its parameter estimates for all subsequent analyses. Robustness is evaluated by computing the ARI between its cluster allocations and those from the other retained chains.

Table 3: Mean network clustering ARI (standard deviations in brackets) for the proposed model and the four competing models across five distinct scenarios.

Scenario	<a href="#">Mantziou et al.</a>	<a href="#">Signorelli and Wit</a>	<i>graphclust</i>	<i>PopNet</i>	<i>LaPCoM</i>
I	0.16 (0.10)	0.67 (0.26)	0.05 (0.09)	1.00 (0.00)	0.95 (0.12)
II	0.39 (0.14)	0.76 (0.31)	0.59 (0.30)	1.00 (0.00)	1.00 (0.00)
III	0.52 (0.14)	0.58 (0.08)	0.44 (0.20)	1.00 (0.00)	0.98 (0.07)
IV	0.57 (0.04)	0.38 (0.06)	0.48 (0.14)	1.00 (0.00)	0.89 (0.17)
V	0.62 (0.12)	0.40 (0.07)	0.47 (0.18)	1.00 (0.00)	0.86 (0.16)



(a) Mantziou et al.'s model.



(b) LaPCoM.

Figure 2: Comparison of nodal clustering performance measured by Adjusted Rand Index (ARI) across 10 simulations for each scenario. Results for (a) Mantziou et al.'s model and (b) LaPCoM.

### 5.1 Krackhardt Advice Networks

We analyse the advice network dataset of Krackhardt (1987), consisting of  $M = 21$  directed binary networks among  $N = 21$  employees at a United States high-tech firm. Each network reflects one employee's perception of who seeks advice from whom, with  $y_{ij}^{(m)} = 1$  indicating that employee  $m$  believes employee  $i$  seeks advice from employee  $j$ . While the networks are directed, LaPCoM does not explicitly model directionality through sender or receiver effects. Instead, it implements node clustering without distinguishing advice-givers from recipients, capturing the overall structure of advice ties rather than their directional nuances. As noted in Signorelli and Wit (2020), some employees may share similar perceptions of the advice network, suggesting clusters of networks with common structure. Our LaPCoM model

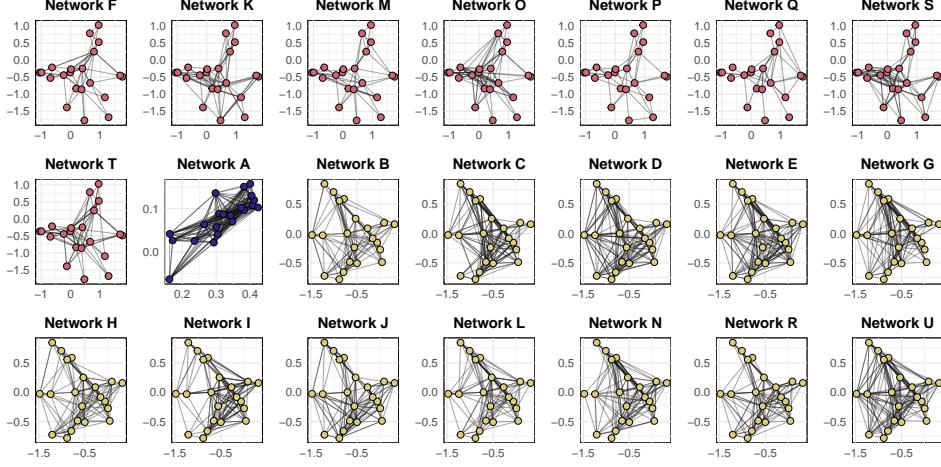


Figure 3: Latent space representation of the Krackhardt advice multiplex obtained from *LaPCoM*. The networks are reordered and coloured according to the clusters identified by *LaPCoM*.

captures this by supporting both network- and node-level clustering.

Scaling factors for the intercept and latent positions are tuned to ensure suitable acceptance rates. To assess robustness, we run 40 MCMC chains: one using the initialisation described in Section 5 of the Supplementary Material, and 39 with random perturbations. Each chain runs for 300,000 iterations, discarding an additional 90,000 as burn-in and thinning every 300<sup>th</sup> draw, resulting in 1,000 posterior samples per chain. Network-level allocations,  $\mathbf{C}$ , are initialised via  $K$ -means clustering.

Post-processing (Section 3.3) revealed that the optimal number of clusters was  $\hat{G}_+ = 3$ , appearing in 34 of 40 chains. Analyses were restricted to these chains, and the chain with the highest log-posterior value was selected for inference. Its estimated clustering partition matched those from all other retained chains (ARI = 1), confirming stability.

Figure 3 visualises employee positions in the estimated latent spaces by cluster. Edges are projected into the cluster-specific latent space, with proximate nodes indicating a higher likelihood of advice-seeking. Panels are ordered by cluster. The three clusters differed in perceived advice density: Cluster 1 (red, density 0.17) has the most diffuse configuration. Cluster 2 (blue, density 0.66) has a tightly packed latent space, indicating frequent advice-seeking and cluster 3 (yellow, density 0.38) shows moderate compactness. These spatial patterns align with the varying levels of advice-seeking across clusters.

It is worth noting that *LaPCoM* did not provide evidence of any node-level clustering within the  $\hat{G}_+ = 3$  latent spaces. This outcome appears reasonable, as the networks projected onto their respective latent spaces in Figure 3 shows no indication of distinct node-level groupings.

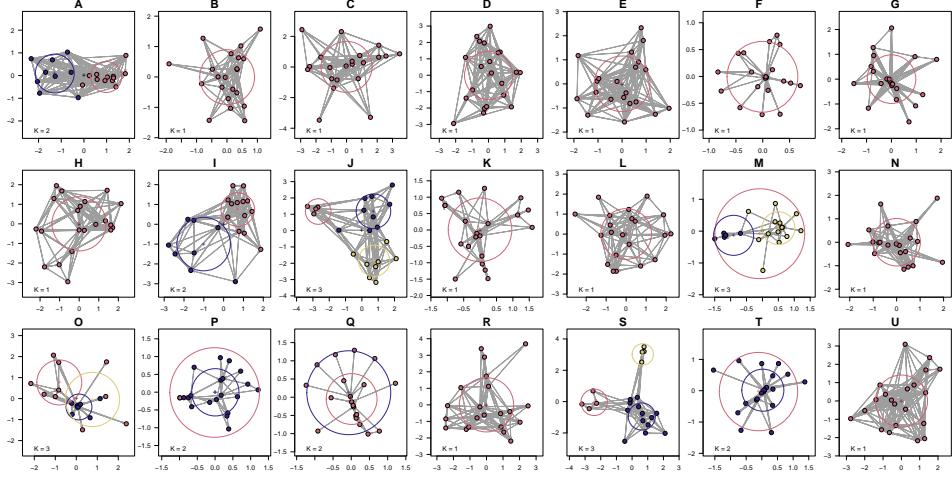


Figure 4: Latent space representation of the Krackhardt advice multiplex obtained from fitting a LPCM with `latentnet` to each network. The nodes of the networks are coloured according to the node-level clusters identified by `latentnet`.

### 5.1.1 Posterior Predictive Checks

Without a reference partition of the [Krackhardt \(1987\)](#) dataset, we assess model fit using posterior predictive checks (PPCs), which compare observed and simulated multiplexes to evaluate adequacy of the obtained clustering. Using the final 500 samples of  $\alpha$ ,  $C$ , and  $\{\mathbf{Z}_g\}_{g=1}^G$  from the selected chain, we generate 500 simulated multiplexes. Fit is evaluated using AUC of the precision-recall curve,  $F_1$ -score, density, [Schieber et al. \(2017\)](#) network distance, and Hamming distance, reflecting the binary nature of the data.

For comparison, we applied the LPCM to each of the 21 networks using the `latentnet` package in **R** ([Krivitsky and Handcock, 2024, 2008](#)), fitting models with one to three node-level clusters. The `ergm` function was run with 5,000 retained samples, a burn-in of 1,500,000 iterations, and thinning every 50 iterations. The optimal number of clusters per network was selected via the Bayesian information criterion (BIC). Unlike *LaPCoM*, which shares three latent spaces across networks for parsimony, `latentnet` fits a separate latent space for each network.

The `ergm` function provided evidence for node-level clustering in 9 of the 21 networks: five were best fit by two clusters, four by three, and the remaining 12 by a single component. Figure 4 shows the estimated latent spaces with nodes coloured by cluster using `plot.ergm`. Some partitions appear spurious. Unlike *LaPCoM*, which did not indicate evidence of node-level clustering, `latentnet` fits separate latent spaces per network, allowing it to capture network-specific patterns. In contrast, *LaPCoM* uses shared cluster-specific latent spaces across networks, leading to more parsimonious representations.

We report the [Schieber et al. \(2017\)](#) dissimilarity metric, which ranges from 0 to

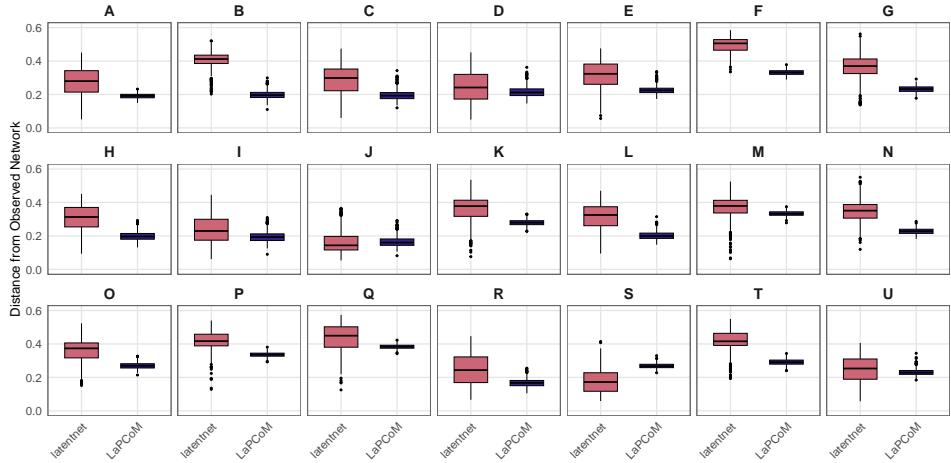


Figure 5: Boxplots showing the distribution of Schieber et al. dissimilarities across 500 multiplexes generated from the posterior predictive distribution, using both `latentnet` and *LaPCoM*.

1, with lower values indicating better fit. Distances were computed for each of the  $M$  networks across 500 simulated multiplexes. Figure 5 shows boxplots comparing `latentnet` and *LaPCoM*. Posterior simulations for `latentnet` generated networks with no edges in 13 instances. Excluding these cases, the median network distance was 0.33 for `latentnet` and 0.24 for *LaPCoM*, with interquartile ranges of 0.10 and 0.03, respectively, indicating that *LaPCoM* achieves a better and more precise fit.

The other PPC metrics (AUC,  $F_1$ -score, density, and Hamming distance) are reported in Section 7 of the Supplementary Material and show similar results for both models, with a slight advantage for *LaPCoM*. It is important to note that `latentnet` fits a separate latent space for each network, resulting in a total of 21 latent spaces. Consequently, some network-specific PPC metrics may favour `latentnet` compared to *LaPCoM*, which models the entire multiplex using only three latent spaces. Nonetheless, *LaPCoM* achieves comparable, and in some cases better, results across PPC metrics while offering a more parsimonious representation of the multiplex.

## 5.2 Aarhus Computer Science Department Social Networks

This multiplex network comprises  $M = 5$  undirected binary networks representing different types of interactions among  $N = 61$  members of Aarhus University’s Department of Computer Science. Member roles include administrative staff, PhD students, postdocs, associate professors, and professors. Each network captures a distinct type of relationships: co-authorship, Facebook friendship, co-participation in leisure activities, having lunch together, and working together. See Magnani et al. (2013) for further details. Given that the data represent multiple interaction types

among the same set of individuals, it is expected that some members will exhibit similar interaction patterns across networks, indicating the presence of network-level clusters. Simultaneously, individuals may form subgroups within each network based on social interactions and roles, suggesting the presence of node-level clusters. Consequently, *LaPCoM* appears to be particularly well suited for modelling these data.

Scaling factors for the intercept and latent positions were tuned to achieve good acceptance rates. The settings for the number of MCMC chains, iterations, burn-in, thinning, and initialisation of network-level cluster allocations followed those described in Section 5.1.1.

The optimal number of network clusters,  $\hat{G}_+$ , was consistently estimated as 2 across all 40 chains. Two chains returned node-level cluster estimates of  $K_{g+} = 6, 5$ ; however, the estimate  $\hat{K}_{2+} = 5$  was deemed spurious upon further inspection, as the resulting node partitions lacked interpretability. These chains were therefore excluded, leaving 38 for analysis. The chain with the highest log-posterior was selected for analysis of results. In this chain, the Facebook friendship network was assigned to its own cluster, while the remaining four networks were grouped together. Stability was confirmed with an ARI of 1 between the estimated clustering of the selected chain and those of the remaining chains.

Figure 6 presents the posterior mean latent spaces for the Aarhus multiplex. Network-level clusters are indicated by the colour of the surrounding box, while node-level clusters are represented by the colour of the nodes. For these data, *LaPCoM* provided indication of node-level clustering. The posterior mode of the number of node-level clusters,  $K_{2+}$ , within the latent space of the second network-level cluster (denoted  $Z_2$ , which corresponds solely to the Facebook network) was estimated as 2. Further examination showed that these two clusters corresponded to connected versus unconnected nodes. However, after applying post-processing (specifically the label-switching correction procedure of Frühwirth-Schnatter (2011)) these clusters merged into a single cluster. This merging likely occurs because the unconnected nodes are difficult to position in the latent space, resulting in a high node-cluster variance that supports their merging. Despite this, we consider the  $\hat{K}_{2+} = 2$  solution more interpretable, as the unconnected nodes likely represent individuals without Facebook accounts, thus justifying separate clusters. The posterior distribution of  $K_{2+}$  exhibited considerable spread, reflecting uncertainty in this partition. In contrast, the latent space of the first network-level cluster,  $Z_1$ , clearly exhibits six stable node-level clusters, each containing between six and fourteen nodes.

Node-level role data for department members are available, grouped here as administrative staff (6), professors (8, combining associate and full professors), PhD students and postdocs (43), and one unknown. Figure 6 shows the latent spaces with node shapes representing roles. In the second cluster's latent space,  $Z_2$  (Facebook network), no clear role-related patterns appear, suggesting no association between roles and node clusters. In contrast, the first cluster's latent space,  $Z_1$ , reveals role-related structure: five of the six node-level clusters include at least one professor alongside PhD students and postdoctoral researchers, reflecting a typical aca-

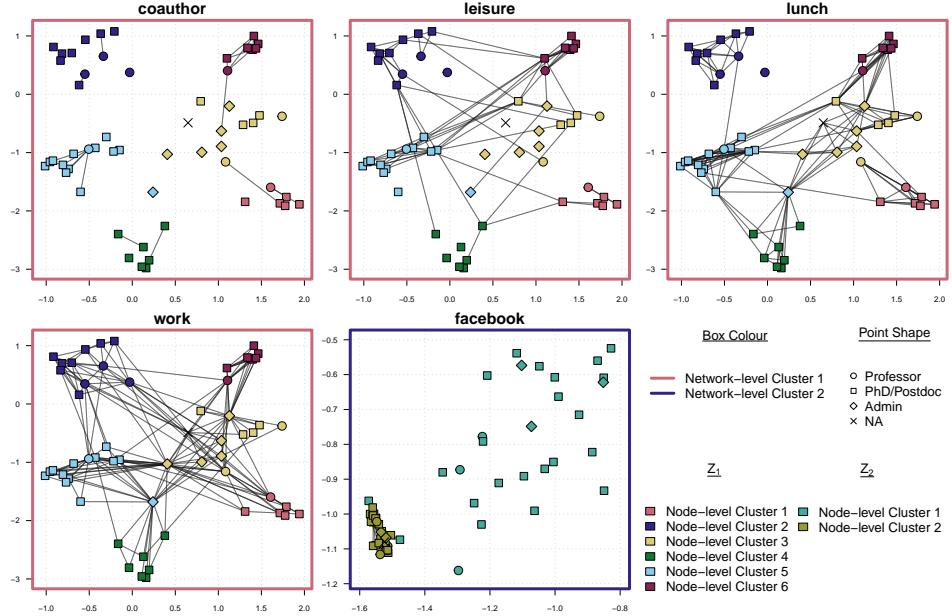


Figure 6: Latent space representation of the Aarhus multiplex from *LaPCoM*. Nodes coloured according to their node-level clusters; network-level cluster indicated by the outline colour of the plotting box. Shape of the points indicates the role of the department members.

demic group structure with a principal investigator (PI) supervising early-career researchers. Leisure ties mostly form among PhDs and postdocs, consistent with their similar career stage and social activities. In the lunch network, one node-level cluster (green) appears relatively isolated, possibly indicating a separate lunch space, though overall, there is substantial inter-cluster mixing, as expected in a shared departmental environment. The work network shows strong intra- and inter-cluster ties, with professors collaborating among themselves and with supervisees, PhDs and postdocs interacting frequently, and administrative staff centrally positioned, highlighting their key support role.

### 5.2.1 Posterior Predictive Checks

As in Section 5.1, model fit is assessed using PPCs. We generate 500 replicate multiplexes from the posterior predictive distribution using the final 500 samples of  $\alpha$ ,  $\mathbf{C}$ , and  $\{\mathbf{Z}_g\}_{g=1}^G$  from the selected chain. Given the binary networks, we use the same PPC metrics as before: AUC of the precision-recall curve,  $F_1$ -score, density, network distance, and Hamming distance.

To validate our findings, we also fitted LPCMs to each of the  $M = 5$  networks individually using the R package `latentnet` (Krivitsky and Handcock, 2024, 2008). Models with 1–7 node-level clusters were estimated via the `ergm` function, using

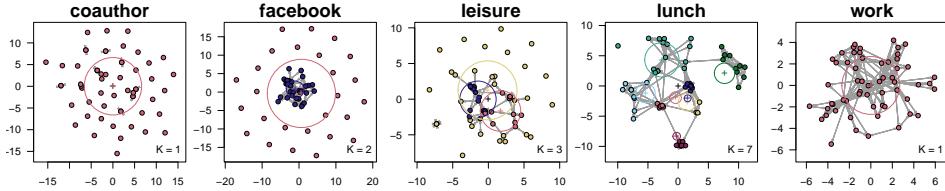


Figure 7: Latent space representation of the Aarhus multiplex obtained from fitting a LPCM with `latentnet` to each network; nodes coloured according to the node-level clusters.

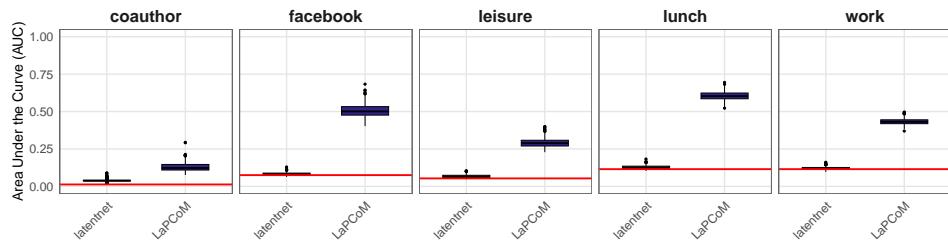


Figure 8: Boxplots showing the distribution of area under the curve (AUC) values across 500 multiplexes generated from the posterior predictive distribution, using both `latentnet` (pink) and `LaPCoM` (blue). Observed network density marked by a red horizontal line.

5,000 retained samples, a burn-in of 1,500,000 iterations, and thinning every 50 iterations, as previously justified. The BIC guided model selection. We then computed the same five PPC metrics for comparison. The best-fitting models identified one, two, three, or seven clusters across the networks. Figure 7 displays the resulting latent spaces, with node colours indicating cluster membership, as visualised using `plot.ergm`. Notably, the “lunch” and “Facebook” networks show node-level clusters that closely align with those from *LaPCoM*, reinforcing the validity of our model. In contrast, the “co-authorship”, “work”, and “leisure” networks show more pronounced differences. While both approaches aim to uncover latent clustering, `latentnet` fits a separate latent space to each network. In comparison, *LaPCoM* uses only  $\hat{G}_+ = 2$  latent spaces to model the association structure in the multiplex, offering a more parsimonious representation.

Figure 8 shows boxplots of AUC values from 500 posterior predictive multiplexes for both `latentnet` and *LaPCoM* at the network level. The observed network density is marked by a red horizontal line in each panel. AUC values above this line indicate good fit. For `latentnet`, median AUCs generally align with the density, suggesting average fit. In contrast, *LaPCoM* consistently has all values, including the lower whisker, above the density line, indicating a better fit across networks.

The other four PPC metrics are reported in Section 8 of the Supplementary Material. Two show similar performance for both models, while the other two

slightly favour *LaPCoM*. Overall, these results indicate that *LaPCoM* provides a more accurate and parsimonious fit than `latentnet`.

### 5.3 Primary School Face-to-face Interaction Networks

This multiplex captures the temporal network of face-to-face interactions between students and teachers at a primary school in France. Originally collected for an epidemiological study (Gemmetto et al., 2014), it has since been used in network analysis research (Stehlé et al., 2011). The dataset includes  $N_S = 232$  students and  $N_T = 10$  teachers over two school days (Thursday, October 1st, and Friday, October 2nd, 2009), spanning five grades divided into 10 classes. For analysis, we construct a multiplex in which each layer corresponds to a one hour snapshot of interactions. While *LaPCoM* does not directly model temporal dependence, the shared latent spaces at the network-level are capable of capturing similar features across the one-hour slots. We represent the data as a multiplex with  $M = 16$  layers, each corresponding to an hour from 9 AM to 5 PM on each day. Nodes represent students and teachers; edges  $y_{ij}^{(m)}$  count interactions between individuals  $i$  and  $j$  during hour  $m$ .

We apply our mixture of mixtures model, *LaPCoM*, to identify clusters of networks and nodes in this dataset. As before, scaling factors for the intercept and latent positions are tuned to ensure appropriate acceptance rates. Note that due to the large number of nodes in this application, we increased the value of  $n_{\min}$  used to determine  $K_{\max}$  to  $n_{\min} = 25$ . To assess robustness and stability, we run 20 independent MCMC chains. One chain is initialised as described in Section 5 of the Supplementary Material; the other 19 use the same method with added noise. Each chain runs for 250,000 iterations, discarding an additional 75,000 as burn-in, and is thinned by keeping every 250<sup>th</sup> iteration, yielding 1,000 posterior samples per chain.

We initialise the network-level allocations,  $\mathbf{C}$ , using model-based clustering via the `mclust` package with the flexible VVV model, yielding  $G_0 = 3$ . This replaces the previously used  $K$ -means initialisation, which often led to poor posterior exploration. A sensitivity analysis varying  $G_0 \in \{1, \dots, 8\}$  and the  $\mathcal{BNB}$  hyperparameters showed that  $K$ -means frequently produced singleton clusters for larger values of  $G_0$ . Based on these findings, the `mclust` initialisation was used for final model fitting.

The mode of the optimal number of clusters,  $\hat{G}_+ = 2$  across all 20 chains. Three chains failed the permutation test at the node level and were discarded, leaving 17 chains for analysis. The chain with the highest log-posterior value was selected for inference. Comparison with the remaining chains showed perfect agreement (ARI = 1), confirming the stability of the solution.

Figure 9 shows the estimated latent spaces for the two inferred network-level clusters, with each network's connections projected onto the shared space. Networks are indicated using the convention OctD.H.h, where D is the day (1 or 2) and h the hour (9 to 16). As described previously, node-level metadata includes each individual's class designation (e.g., 1A, 1B, ..., 5A, 5B) or teacher status. The inferred network-level partition groups networks Oct1.H.12, Oct1.H.13, Oct2.H.12, and Oct2.H.13 together, with the remaining 12 networks in a second cluster. In the

latent space of the first network-level cluster,  $\hat{\mathbf{Z}}_1$ , the model found evidence of two node-level clusters: one comprising the large connected component of the network, and the other consisting of the remaining disconnected nodes. In the latent space of the second network-level cluster,  $\hat{\mathbf{Z}}_2$ , the model inferred 13 node-level clusters, which align strongly with class/status labels, yielding an ARI of 0.86.

The clustering and low-dimensional representation aligns well with key aspects of the social interactions in the data. During lessons, students interact mostly within their class, producing distinct class-based patterns. Around lunchtime (12:00–14:00), cross-class interactions increase, reflecting freer movement in shared areas like the playground, a feature well-captured by the network-level clustering. Node-level clustering also reflects this structure: most links occur within classes, with some limited inter-class ties, consistent with occasional joint activities.

### 5.3.1 Posterior Predictive Checks

As in Section 5.1 and Section 5.2, model fit is assessed using posterior predictive checks (PPCs) based on the final 500 samples from the posterior predictive distribution from the selected chain, in order to generate simulated multiplexes under the fitted model. Since this dataset comprises count-valued networks, unlike the binary-valued networks in the previous examples, we adopt alternative PPC metrics. Specifically, to compare observed and predicted networks, we evaluate: (i) the mean absolute difference in edge counts, (ii) the network distance proposed by [Schieber et al. \(2017\)](#), (iii) the true negative rate, and (iv) the log-empirical cumulative distribution function (ECDF) of counts greater than zero.

In previous applications, we used `latentnet` for comparison. However, due to the size of the networks in this multiplex ( $N = 242$ ), it was computationally infeasible to run `latentnet` across all 16 networks and a range of cluster values. The required runtime for networks of this size was prohibitive, and thus `latentnet` was not used in this analysis.

Table 4 presents the mean values (standard deviations, SD, in brackets) for three key PPC metrics across all networks: mean absolute difference in counts (MAD), [Schieber et al. \(2017\)](#) network distance, and true negative rate (TNR). The overall average MAD is 0.4977 (SD = 0.0032), indicating that predicted edge counts deviate by less than one count from observed values on average, suggesting a good fit. The average network distance is 0.4423 (SD = 0.0033), which, given the metric is bounded between 0 and 1, reflects a reasonable structural similarity between the observed and predicted networks. The average TNR of 0.8746 (SD = 0.0013) demonstrates that the model is conservative in predicting edges, favouring the correct identification of non-edges and avoiding the introduction of spurious edges.

Additional posterior predictive checks based on the ECDFs of positive counts (see Section 9 of the Supplementary Material) show that the model struggles to capture the distribution of non-zero edge counts, particularly underestimating large values. This suggests that the Poisson distribution may be too restrictive. Nonetheless, the model captures the overall network structure well and recovers clusters that align with students' class memberships and time-of-day patterns.

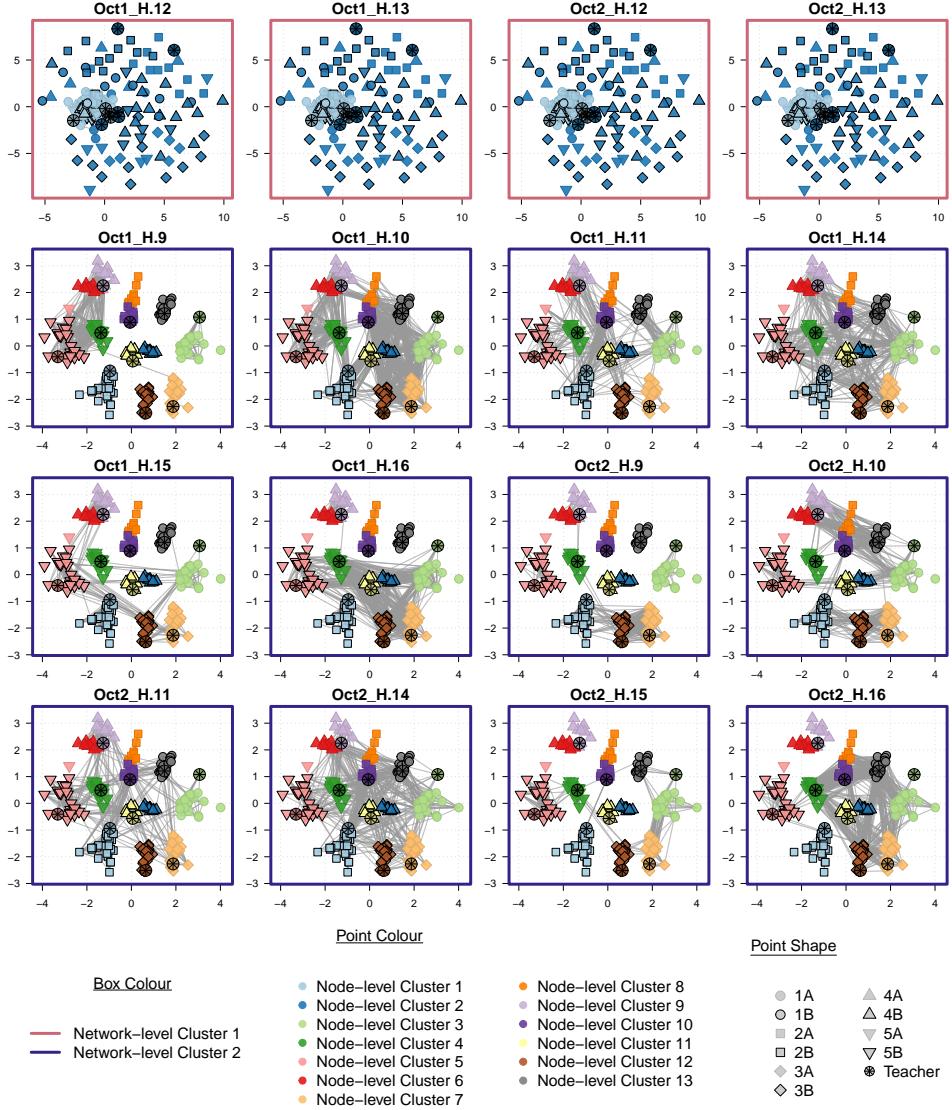


Figure 9: Latent space representations of the networks. Point colours indicate the class membership of each student or whether the individual is a teacher. The outline colour of each plotting panel denotes the network-level cluster assignment.

## 6 Discussion

The proposed latent position co-clustering model (*LaPCoM*) introduces a novel framework for jointly clustering networks within a multiplex alongside their con-

Table 4: Mean (standard deviation) values of the mean absolute difference (MAD) in counts, network distance, and true negative rate (TNR) across all networks.

Network	MAD	Distance	TNR
Oct1_H.9	0.4438 (0.0032)	0.4401 (0.0029)	0.8744 (0.0013)
Oct1_H.10	0.5888 (0.0032)	0.3396 (0.0043)	0.8749 (0.0014)
Oct1_H.11	0.4654 (0.0031)	0.4158 (0.0037)	0.8749 (0.0014)
Oct1_H.12	0.5990 (0.0032)	0.4814 (0.0022)	0.8741 (0.0014)
Oct1_H.13	0.4938 (0.0032)	0.4928 (0.0022)	0.8743 (0.0013)
Oct1_H.14	0.4059 (0.0033)	0.4165 (0.0040)	0.8749 (0.0013)
Oct1_H.15	0.4510 (0.0033)	0.4833 (0.0031)	0.8748 (0.0013)
Oct1_H.16	0.4732 (0.0032)	0.4220 (0.0036)	0.8744 (0.0014)
Oct2_H.9	0.4685 (0.0033)	0.4616 (0.0035)	0.8744 (0.0014)
Oct2_H.10	0.5660 (0.0031)	0.4090 (0.0038)	0.8750 (0.0014)
Oct2_H.11	0.4913 (0.0033)	0.4035 (0.0039)	0.8749 (0.0013)
Oct2_H.12	0.6079 (0.0033)	0.4822 (0.0023)	0.8739 (0.0012)
Oct2_H.13	0.5465 (0.0034)	0.4961 (0.0025)	0.8742 (0.0012)
Oct2_H.14	0.4204 (0.0033)	0.4154 (0.0040)	0.8744 (0.0013)
Oct2_H.15	0.4497 (0.0033)	0.4900 (0.0032)	0.8748 (0.0014)
Oct2_H.16	0.4926 (0.0032)	0.4274 (0.0033)	0.8746 (0.0013)

stituent nodes. By leveraging a mixture-of-mixtures model, *LaPCoM* simultaneously achieves dimension reduction and two-level clustering of a multiplex network. The model is formulated under a Bayesian nonparametric framework using a mixture of finite mixtures (MFM), which places priors on the number of components at both the network and node levels. A sparse prior on the mixing proportions encourages the discarding of empty components and enables data-driven identification of the number of clusters. The performance of *LaPCoM* was evaluated through simulation studies and illustrative examples, where it successfully recovered interpretable clustering structures, demonstrating its practical utility across diverse multiplex datasets.

For interpretability and ease of visualisation, we fixed the dimension of the latent spaces to two in all analyses. However, this constraint may restrict the model’s capacity to capture higher-dimensional latent structures in some data. A potential avenue of future work would be to introduce automatic latent dimension selection using BNP priors. Examples include the multiplicative gamma process shrinkage prior ([Bhattacharya and Dunson, 2011](#)), which has shown success in similar contexts [Gwee et al. \(2023\)](#), or the cumulative shrinkage process prior ([Legramanti et al., 2020](#)), which has been extended to various factor analysis models [Kowal and Canale \(2023\); Frühwirth-Schnatter \(2023\)](#).

In Section 5.3, we applied *LaPCoM* to a time-varying multiplex. However, the model does not explicitly account for temporal dependencies between networks. This is a potential limitation, and a natural extension would involve the incorporation of a Markov process into the prior of the latent positions, an approach successfully

employed in dynamic network modelling ([Sewell and Chen, 2016, 2017](#)).

Future work could also explore more flexible modelling of edge formation in multiplex networks. The current model does not account for edge directionality. Incorporating sender-receiver effects into the latent space, as proposed by [Liu and Liu \(2025\)](#), offers a valuable extension. Additionally, the application in Section 5.3 suggests that the Poisson distribution may not adequately capture the observed edge count distribution. While clustering remains the primary goal of *LaPCoM*, adopting a zero-inflated Poisson model ([Lu et al., 2025](#)), or other flexible alternatives, could better accommodate under- or overdispersion and improve fit for count-valued multiplex data.

In summary, while *LaPCoM* offers a flexible and interpretable framework for the co-clustering of multiplex networks, these directions highlight potential methodological enhancements. Addressing these limitations would further improve the model's applicability across a broader range of multiplex network settings.

## Acknowledgements

This publication has emanated from research conducted with the financial support of Taighde Éireann (Research Ireland) under grant number 18/CRT/6049. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

## Supplementary Material for “A Latent Position Co-Clustering Model for Multiplex Networks”

In this document, we provide supplementary material to the article “*A Latent Position Co-Clustering Model for Multiplex Networks*”.

- **Section A:** A comprehensive list of all notation used in the main article.
- **Section B:** The rationale behind the choice of hyperparameters in the prior distributions.
- **Section C:** Derivations of the full conditional distributions of the parameters.
- **Section D:** Pseudocode for the MCMC algorithm discussed in Section 3 of the main article.
- **Section E:** Initialisation details for the MCMC algorithm discussed in Section 3 of the main article.
- **Section F:** Design details of the simulation studies discussed in Section 4 of the main article.
- **Section G:** Additional posterior predictive checks related to the Krackhardt ([Krackhardt, 1987](#)) application discussed in Section 5.1 of the main article.
- **Section H:** Additional posterior predictive checks related to the Aarhus ([Magagnani et al., 2013](#)) application discussed in Section 5.2 of the main article.
- **Section I:** Additional posterior predictive checks related to the primary school ([Gemmetto et al., 2014](#)) application discussed in Section 5.3 of the main article.
- **Section J:** Comparison between a mixture of finite mixtures (MFM; [Frühwirth-Schnatter et al. \(2021\)](#)) model and an overfitted mixture model.

## A Notation

$\mathcal{Y}$ : The collection of  $M$  networks.

$\mathbf{Y}^{(m)}$ : The  $N \times N$  network adjacency matrix.

$y_{ij}^{(m)}$ : The value of the edge between nodes  $i$  and  $j$  in network  $\mathbf{Y}^{(m)}$ .

$M$ : Number of networks.

$N$ : Number of nodes.

$\alpha$ : The intercept parameter of the latent position model (LPM) that describes the overall level of connectivity in the network.

$\delta_\alpha$ : The scaling factor used in the standard deviation of the proposal distribution for the intercept parameter,  $\alpha$ .

$G$ : The number of network-level mixture components.

$G_+$ : The number of network-level clusters (active/non-empty mixture components).

$\mathcal{D}$ : Denotes the Dirichlet distribution.

$e$ : The Dirichlet concentration hyperparameter in the network-level mixing proportions prior distribution.

$\tau$ : The network-level mixing proportions.

$\tau_g$ : The probability that a network belongs to network-level component  $g$ .

$C$ : The  $M \times G$  binary matrix indicating the membership of networks to network-level components.

$C_g$ : The binary vector of length  $M$  indicating which networks belong to network-level component  $g$ .

$C_g^{(m)}$ : The binary indicator of membership of network  $\mathbf{Y}^{(m)}$  to network-level component  $g$ .

$K_g$ : The number of node-level mixture components within the  $g^{\text{th}}$  latent space.

$K_{g+}$ : The number of node-level clusters (active/non-empty mixture components) within the  $g^{\text{th}}$  latent space.

$w_g$ : The Dirichlet concentration hyperparameter in the node-level mixing proportions prior distribution.

$\pi_g$ : The node-level mixing proportions within the  $g^{\text{th}}$  latent space.

$\pi_{gk}$ : The probability that a node belongs to node-level component  $k$  within the  $g^{\text{th}}$  latent space.

$S_g$ : The  $N \times K_g$  binary matrix indicating the membership of nodes to node-level components within the  $g^{\text{th}}$  latent space.

$S_{gk}$ : The binary vector of length  $N$  indicating which nodes belong to node-level component  $k$  within the  $g^{\text{th}}$  latent space.

$S_{gk}^{(i)}$ : The binary indicator of membership of node  $i$  to network-level component  $k$  within the  $g^{\text{th}}$  latent space.

$\mathcal{MVN}_p$ : Denotes the multivariate Normal distribution of dimension  $p$ .

$\mu_{gk}$ : The (2-dimensional) mean vector of the multivariate Normal distribution describing the  $k^{\text{th}}$  node-level component within the  $g^{\text{th}}$  latent space.

$\Sigma_{gk}$ : The (diagonal) variance-covariance matrix of the multivariate Normal distribution describing the  $k^{\text{th}}$  node-level component within the  $g^{\text{th}}$  latent space.

$\sigma_{gk,q}^2$ : The covariance parameter of the multivariate Normal distribution in the  $q^{\text{th}}$

dimension, describing the  $k^{\text{th}}$  node-level component within the  $g^{\text{th}}$  latent space.

$\mathbf{Z}_g$ : The  $N \times 2$  matrix of latent positions of the  $g^{\text{th}}$  network-level component.

$\mathbf{z}_{g,i}$ : The (2-dimensional) latent position of node  $i$  in the latent space of the  $g^{\text{th}}$  network-level component.

$\delta_Z$ : The scaling factor used in the standard deviation of the proposal distribution for the latent positions.

$\mathbf{H}$ : A collective term for the hyperparameters of the model.

$G_0$ : The initial number of network-level mixture components.

$G_{\max}$ : The maximum number of network-level mixture components considered.

$\mathcal{N}$ : Denotes the univariate Normal distribution.

$m_\alpha$ : The mean of the Normal prior on the intercept parameter,  $\alpha$ .

$s_\alpha$ : The standard deviation of the Normal prior on the intercept parameter,  $\alpha$ .

$\mathcal{BNB}$ : Denotes the Beta-Negative-Binomial distribution.

$a_G$ : The “number of successes until the experiment is stopped” parameter of the  $\mathcal{BNB}$  prior distribution on the number of network-level mixture components.

$b_G$ : The first shape parameter of the  $\mathcal{BNB}$  prior distribution on the number of network-level mixture components.

$c_G$ : The second shape parameter of the  $\mathcal{BNB}$  prior distribution on the number of network-level mixture components.

$\mathcal{F}$ : Denotes the Fisher-Snedecor distribution.

$l_G$ : The first degrees of freedom parameter of the  $\mathcal{F}$  prior distribution on the Dirichlet concentration hyperparameter of the network-level mixing proportions prior distribution.

$r_G$ : The second degrees of freedom parameter of the  $\mathcal{F}$  prior distribution on the Dirichlet concentration hyperparameter of the network-level mixing proportions prior distribution.

$s_e$ : The standard deviation of the proposal distribution for the Dirichlet concentration hyperparameter of the network-level mixing proportions prior distribution.

$K_0$ : The initial number of node-level mixture components.

$K_{\max}$ : The maximum number of node-level mixture components considered.

$a_K$ : The “number of successes until the experiment is stopped” parameter of the  $\mathcal{BNB}$  prior distribution on the number of node-level mixture components.

$b_K$ : The first shape parameter of the  $\mathcal{BNB}$  prior distribution on the number of node-level mixture components.

$c_K$ : The second shape parameter of the  $\mathcal{BNB}$  prior distribution on the number of node-level mixture components.

$l_K$ : The first degrees of freedom parameter of the  $\mathcal{F}$  prior distribution on the Dirichlet concentration hyperparameter of the node-level mixing proportions prior distribution.

$r_K$ : The first degrees of freedom parameter of the  $\mathcal{F}$  prior distribution on the Dirichlet concentration hyperparameter of the node-level mixing proportions prior distribution.

$s_w$ : The standard deviation of the proposal distribution for the Dirichlet concentration hyperparameter of the node-level mixing proportions prior distribution.

$\mathcal{G}$ : Denotes the Gamma distribution.

$\mathcal{IG}$ : Denotes the Inverse-Gamma distribution.

$u_{\sigma^2}$ : The shape parameter of the  $\mathcal{IG}$  prior distribution for any  $\sigma_{gkq}^2$ ,  $q = 1, 2$ .

$v_{\sigma^2}$ : The scale parameter of the  $\mathcal{IG}$  prior distribution for any  $\sigma_{gkq}^2$ ,  $q = 1, 2$ .

$T_S$ : The total number of posterior samples after thinning and burn-in of the resulting MCMC chain.

$T_B$ : The number of burn-in iterations to be discarded from the resulting MCMC chain.

$T_{\text{thin}}$ : The number of iterations to thin the resulting MCMC chain by.

$T$ : The total number of iterations the MCMC algorithm runs for,  $T = (T_S * T_T) + T_B$ .

$B$ : Beta function.

$\Gamma$ : Gamma function.

## B Hyperparameter Choices

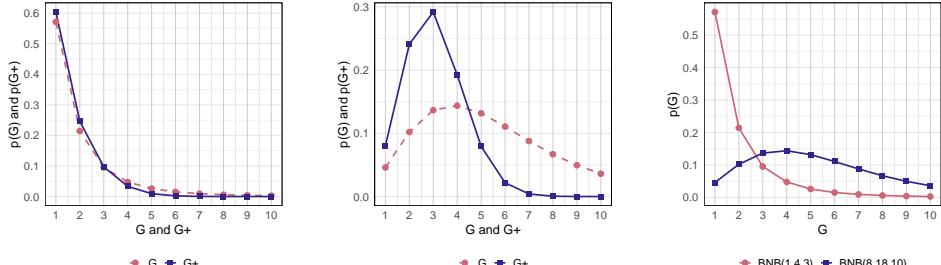
The choice of a  $\mathcal{N}(0, 1)$  prior for the intercept parameter  $\alpha$  is motivated by the role of the intercept in latent position models (LPMs), where it represents the baseline propensity for edge formation across the entire network. Centring the prior around zero allows the data to primarily influence the intercept estimation, while a prior standard deviation of 1 maintains a relatively uninformative stance on  $\alpha$  in the absence of strong prior knowledge.

The  $\mathcal{MVN}(\mathbf{0}, \mathbb{I}_2)$  prior for the node-level cluster means  $\boldsymbol{\mu}_{gk}$  was chosen to provide a non-informative basis, letting the data drive the estimation of cluster means in the latent spaces. Setting a mean of zero is appropriate, as the latent space is translation-invariant, while a diagonal covariance matrix respects the rotation and scaling invariance within the space.

The choice of hyperparameters  $u_{\sigma^2}$  and  $v_{\sigma^2}$  in the  $\mathcal{IG}(u_{\sigma^2}, v_{\sigma^2})$  prior on the node-level cluster variances  $\sigma_{gk,q}^2$  is guided by the need to control the spatial dispersion of nodes within each cluster in the latent space  $Z_g$ . These parameters influence the expected within-cluster variance, ensuring that node positions remain sufficiently concentrated in a manner consistent with the assumptions of the latent position cluster model (LPCM), where clusters correspond to nodes located close together in the latent space with higher probabilities of connection. Given that latent positions drawn from  $\mathcal{MVN}_2(\mathbf{0}, \mathbb{I}_2)$  typically fall within the range  $(-3, 3)$ , the scale of the latent space implies that clusters should become increasingly tight as the network size  $N$  grows to preserve interpretability. In networks with fewer than 60 nodes, it is reasonable to assume that a typical cluster contains at least  $n_{\min} = 5$  nodes, while for networks with  $N \geq 60$ , an average of at least  $n_{\min} = 10$  nodes per cluster is assumed. Accordingly, the expected variance  $\mathbb{V}(N)$  within a cluster  $k$  is defined as  $\mathbb{V}(N) = \frac{1}{n_{\min}(1 - \frac{n_{\min}}{N})}$ , where  $n_{\min} = 5$  if  $N < 60$ , and  $n_{\min} = 5$  if  $N \geq 60$ .

This function yields a variance around 0.2 for  $N \in \{20, 30, 40, 50\}$  and approximately 0.1 for  $N \in \{60, 70, 80, 90, 100\}$ . To achieve these expected values, we place an  $\mathcal{IG}(u_{\sigma^2}, v_{\sigma^2})$  prior on  $\sigma_{gk,q}^2$ , where  $\mathbb{E}[\sigma_{gk,q}^2] = \frac{v_{\sigma^2}}{u_{\sigma^2}-1}$  (for  $u_{\sigma^2} > 1$ ) and  $\mathbb{V}[\sigma_{gk,q}^2] = \frac{v_{\sigma^2}^2}{(u_{\sigma^2}-1)^2(u_{\sigma^2}-2)}$  (for  $u_{\sigma^2} > 2$ ). With  $v_{\sigma^2} = 2$ , the expected value of  $\sigma_{gk,q}^2$  is 0.2 when  $u_{\sigma^2} = 11$  (with variance 0.004) and 0.1 when  $u_{\sigma^2} = 21$  (with variance 0.001). Thus, the priors  $\mathcal{IG}(u_{\sigma^2} = 11, v_{\sigma^2} = 2)$  and  $\mathcal{IG}(u_{\sigma^2} = 21, v_{\sigma^2} = 2)$  are reasonable choices for networks with  $20 \leq N < 60$  and  $60 \leq N \leq 100$ , respectively, providing sufficient flexibility for variance while concentrating the probability mass close to smaller values.

[Frühwirth-Schnatter et al. \(2021\)](#), recommend to use a  $\mathcal{BNB}(1, 4, 3) + 1$  translated prior distribution for the number of mixture components. This prior is weakly informative, concentrating on a small number of clusters while allowing for a broader range through its fat tail if the data support additional clusters. As shown in Figure 10c, the  $\mathcal{BNB}(1, 4, 3)$  prior places most of its probability mass on  $G = 1$  or  $G = 2$  components, naturally inducing the heavy shrinkage exploited in [Frühwirth-Schnatter et al. \(2021\)](#). For comparison, a  $\mathcal{BNB}(8, 18, 10)$  prior distribution is also



(a) Comparison of the prior on  $G$  (red) under a  $\text{BNB}(1, 4, 3)$  prior and the corresponding induced prior on  $G_+$  (blue).

(b) Comparison of the prior on  $G$  (red) under a  $\text{BNB}(8, 18, 10)$  prior and the corresponding induced prior on  $G_+$  (blue).

(c) Comparison of the prior on  $G$  under a  $\text{BNB}(1, 4, 3)$  prior (red) and under a  $\text{BNB}(8, 18, 10)$  prior (blue).

Figure 10: Illustration of how different  $\text{BNB}$  prior settings affect the prior distribution on the number of components  $G$  and the corresponding induced prior on the number of clusters  $G_+$ .

displayed.

The expected value and variance of  $G$  under a  $\text{BNB}(a_G, b_G, c_G)$  prior distribution are given by  $\mathbb{E}[G] = 1 + \frac{a_G c_G}{b_G - 1}$  and  $\mathbb{V}[G] = 1 + \frac{a_G c_G (a_G + b_G - 1)(c_G + b_G - 1)}{(b_G - 2)(b_G - 1)^2}$ . For a  $\text{BNB}(1, 4, 3)$  prior, this yields an expected value of 2 with a variance of 5; however, we found this expectation too small, as it could introduce unnecessary shrinkage of the mixture components, especially given that  $G_+ \leq G$ . Furthermore, a variance of 5 does not allow for sufficient prior mass on moderate to large values of  $G$ , potentially limiting their occurrence in the model and further promoting undue shrinkage. To address these concerns, we selected the  $\text{BNB}(8, 18, 10)$  prior, resulting in an expected  $G$  of approximately 6 with a variance of about 12. This prior distribution encourages more moderate values of  $G$  while still allowing for flexibility, making it a reasonable choice given our objectives and assumptions for  $G - 1$  and  $K_g - 1 \forall g$ .

[Frühwirth-Schnatter et al. \(2021\)](#) discuss and derive the prior distribution induced on  $G_+$  from the  $\text{BNB}$  prior distribution placed on  $G$ . In a dynamic mixture of finite mixtures (MFM) model, the priors  $\mathbb{P}(G)$  and  $\mathbb{P}(G_+)$  align closely only when the  $\text{BNB}$  prior has a small expected value. For priors  $\mathbb{P}(G)$  with larger expected values,  $\mathbb{P}(G_+)$  deviates notably, with its mass pulled toward smaller values of  $G_+$ . The  $\text{BNB}(1, 4, 3)$  prior, as suggested in [Frühwirth-Schnatter et al. \(2021\)](#), leads to a weakly informative prior on  $G_+$  that mirrors the prior on  $G$ , concentrating on a small number of clusters, predominantly  $G_+ = 1$ , but retaining a fat tail to allow estimation of a larger number of clusters, as shown in Figure 10a. By contrast, the  $\text{BNB}(8, 18, 10)$  prior we propose results in a more informative prior on  $G_+$ , as shown in Figure 10b. This prior concentrates on a moderate number of clusters, primarily  $G_+ \in \{2, 3, 4\}$ , while still preserving the fat tail needed to account for potentially larger numbers of clusters. A direct comparison of the two priors on  $G$  can be seen

in Figure 10c.

[Frühwirth-Schnatter et al. \(2021\)](#) suggest to set the initial number of clusters  $G_0$  to overfit by a factor of two to three times the expected number of clusters. However, our choice of the  $\mathcal{BNB}(8, 18, 10)$  prior over the  $\mathcal{BNB}(1, 4, 3)$  for the number of mixture components  $G$  was informed by a desire to avoid excessively low  $G$  values, which might otherwise impose unwarranted shrinkage on the mixture components. By selecting a more moderate expected value for  $G$  and allowing  $G_+$  to adjust downward as needed, this prior enables the model to capture sufficient complexity without being overly restrictive. Additionally, we employ the same Dirichlet shrinkage prior on the mixing proportions as in [Frühwirth-Schnatter et al. \(2021\)](#), which allows growth or shrinkage in accordance with the data. Unlike [Frühwirth-Schnatter et al. \(2021\)](#), which recommends setting  $G_0$  higher, we set  $G_0 = 2$  (or  $K_0 = 2$  for the node-level mixture) to leverage the adaptive growth permitted by our  $\mathcal{BNB}(8, 18, 10)$  prior. Starting with a lower  $G_0$  improves computational efficiency by reducing the number of components fitted in each iteration while remaining aligned with the structure of our multiplex data. Empirical results validate these hyperparameter choices, emphasising the need to tailor settings to dataset-specific characteristics, as [Frühwirth-Schnatter et al. \(2021\)](#) primarily addressed multivariate observations, in contrast to our focus on multiplex network data.

In the telescoping sampler (TS) procedure, the Markov chain Monte Carlo (MCMC) algorithm updates the value of  $G$  by considering an upper bound  $G_{\max}$  and sampling from a multinomial distribution over the set  $\{G_+, \dots, G_{\max}\}$  with probabilities proportional to the unnormalised posterior of  $G$ . While [Frühwirth-Schnatter et al. \(2021\)](#) set  $G_{\max} = 100$ , this is unrealistic for our setting, where the maximum number of networks considered is  $M = 100$ . Since networks are clustered based on their structural similarity, it is plausible that a single network could form its own cluster. This stands in contrast to the node-level clustering, where we can reasonably assume a minimum average number of nodes per cluster,  $n_{\min}$ , as above. At the network level, however, no such realistic lower bound exists, as assuming an expected minimum network-level cluster size would risk excluding structurally distinct networks that merit separation. However, permitting all  $M$  networks to form their own clusters would defeat the purpose of clustering. To strike a balance, we define  $G_{\max}$  values that scale with the number of networks while still enabling meaningful clustering, as summarised in Table 5.

Similarly,  $K_g$  is updated by sampling from  $K_{g+}, \dots, K_{\max}$ , using weights based on the unnormalised posterior. Again, setting  $K_{\max} = 100$  is impractical in our setting. To enable flexible yet interpretable clustering, we define  $K_{\max}$  as a function of the number of nodes  $N$  and a minimum average cluster size  $n_{\min}$ , specifically,  $K_{\max} = \frac{N}{n_{\min}} + 2$  (rounded to the nearest integer), where  $n_{\min} = 5$  for  $N < 60$  and  $n_{\min} = 10$  for  $N \geq 60$ . Values of  $K_{\max}$  for different values of  $N$  are summarised in Table 6. In addition to interpretability and practicality, the specifications of  $G_{\max}$  and  $K_{\max}$  were also chosen to reduce computational burden while preserving model flexibility and avoiding the overhead of unnecessarily large upper bounds.

Lastly, the  $\mathcal{F}(6, 3)$  prior distribution on the Dirichlet concentration hyperpa-

Table 5: Value of  $G_{\max}$  for various  $M$ .

$M$	$G_{\max}$
20	5
30	5
40	5
50	5
60	10
70	10
80	10
90	10
100	10

Table 6: Value of  $K_{\max}$  for various  $N$ .

$N$	$K_{\max}$
20	6
30	8
40	10
50	12
60	8
70	9
80	10
90	11
100	12

rameters, as suggested in [Frühwirth-Schnatter et al. \(2021\)](#), provides flexibility in the prior probability of homogeneity. We utilise this  $\mathcal{F}(6, 3)$  prior distribution on the Dirichlet concentration hyperparameters  $e$  and  $w_g$ ,  $g = 1, \dots, G_+$  within the node-level mixture distribution, encoding an expectation of three for  $e$  and  $w_g$ ,  $g = 1, \dots, G_+$ , with infinite variance.

We conducted preliminary tests to compare our selected hyperparameters with those recommended by [Frühwirth-Schnatter et al. \(2021\)](#), in order to validate our decision to diverge from their guidelines. The results, presented in Section [J](#) of this Supplementary Material, support the suitability of our choices for the specific network data and co-clustering model used.

## C Derivation of Full Conditional Distributions

As discussed in Section 3 of the main article, the joint posterior distribution of *LaPCoM* is as follows:

$$\begin{aligned} & \mathbb{P}\left(G, e, \boldsymbol{\tau}, \mathbf{C}, \alpha, \left\{K_g, w_g, \boldsymbol{\pi}_g, \mathbf{S}_g, \{\boldsymbol{\mu}_{gk}, \boldsymbol{\Sigma}_{gk}\}_{k=1}^{K_g}\right\}_{g=1}^G, \{\mathbf{Z}_g\}_{g=1}^G \mid \mathbf{y}\right) \\ & \propto \mathbb{P}(\mathbf{y} \mid \mathbf{C}, \alpha, \{\mathbf{Z}_g\}_{g=1}^G) \times \mathbb{P}(\mathbf{C} \mid \boldsymbol{\tau}) \times \mathbb{P}(\boldsymbol{\tau} \mid G, e) \times \mathbb{P}(G \mid \mathbf{H}) \times \mathbb{P}(e \mid \mathbf{H}) \times \mathbb{P}(\alpha \mid \mathbf{H}) \\ & \quad \times \prod_{g=1}^G \mathbb{P}(\mathbf{Z}_g \mid \mathbf{S}_g, \{\boldsymbol{\mu}_{gk}, \boldsymbol{\Sigma}_{gk}\}_{k=1}^{K_g}) \times \prod_{g=1}^G \prod_{k=1}^{K_g} \mathbb{P}(\boldsymbol{\mu}_{gk} \mid \mathbf{H}) \times \prod_{g=1}^G \prod_{k=1}^{K_g} \mathbb{P}(\boldsymbol{\sigma}_{gk}^2 \mid \mathbf{H}) \\ & \quad \times \prod_{g=1}^G \mathbb{P}(\mathbf{S}_g \mid \boldsymbol{\pi}_g) \times \prod_{g=1}^G \mathbb{P}(\boldsymbol{\pi}_{gk} \mid K_g, w_g) \times \prod_{g=1}^G \mathbb{P}(K_g \mid \mathbf{H}) \times \prod_{g=1}^G \mathbb{P}(w_g \mid \mathbf{H}). \end{aligned}$$

As an example, in the following derivations we consider a count-valued weighted multiplex, where each entry  $y_{ij}^{(m)} \in \mathbb{N}$  represents the count shared between nodes  $i$  and  $j$  in view  $m$ . Accordingly, we model the edges using a Poisson distribution with rate parameter  $\lambda_{g,ij}$  and apply the log function as the link function (see Section 2.2 of the main text). Therefore, each network view is modelled

$$\mathbf{Y}^{(m)} \sim \sum_{g=1}^G \tau_g \prod_{i \neq j} \frac{\lambda_{g,ij}^{y_{ij}^{(m)}} \exp(-\lambda_{g,ij})}{y_{ij}^{(m)}!}, \quad m = 1, \dots, M.$$

Throughout the derivation of the full conditional distributions, we adopt the standard convention that the conditioning set, denoted by  $\dots$ , includes the observed data and all other parameters not currently being updated. Analogous derivations apply in the case of a binary multiplex, where edges are modelled using a Bernoulli distribution and the logit function is used as the link function.

The full conditional distribution of the probability of network  $\mathbf{Y}^{(m)}$  belonging to network-level cluster  $g$  is derived as follows:

$$\begin{aligned} Pr\left(C^{(m)} = g \mid \dots\right) & \propto \mathbb{P}(\mathbf{Y}^{(m)} \mid C_g^{(m)}, \alpha, \mathbf{Z}_g) \times \mathbb{P}(C_g^{(m)} \mid \tau_g) \\ & \propto \tau_g \prod_{i \neq j} \frac{\lambda_{g,ij}^{y_{ij}^{(m)}} \exp(-\lambda_{g,ij})}{y_{ij}^{(m)}!} \\ & = \frac{\tau_g \prod_{i \neq j} \frac{\lambda_{g,ij}^{y_{ij}^{(m)}} \exp(-\lambda_{g,ij})}{y_{ij}^{(m)}!}}{\sum_{h=1}^G \tau_h \prod_{i \neq j} \frac{\lambda_{h,ij}^{y_{ij}^{(m)}} \exp(-\lambda_{h,ij})}{y_{ij}^{(m)}!}}. \end{aligned}$$

The full conditional distribution of the network-level mixing proportions  $\boldsymbol{\tau}$  is

derived as follows:

$$\begin{aligned}
\mathbb{P}(\boldsymbol{\tau} \mid \dots) &\propto \mathbb{P}(\mathbf{C} \mid \boldsymbol{\tau}) \times \mathbb{P}(\boldsymbol{\tau} \mid G, e) \\
&\propto \prod_{g=1}^G \prod_{m=1}^M \tau_g^{C^{(m)}} \times \prod_{g=1}^G \tau_g^{\frac{e}{G}-1} \\
&= \prod_{g=1}^G \tau_g^{M_g + \frac{e}{G}-1} \\
&\sim \mathcal{D}(\zeta_1, \dots, \zeta_G),
\end{aligned}$$

where  $\zeta_g = M_g + \frac{e}{G}$  and  $M_g = \sum_{m=1}^M \mathbb{I}\{C_g^{(m)} = 1\}$ .

The full conditional distribution of the intercept  $\alpha$  is derived as follows:

$$\begin{aligned}
\mathbb{P}(\alpha \mid \dots) &\propto \mathbb{P}(\mathbf{Y} \mid \mathbf{C}, \alpha, \{\mathbf{Z}_g\}_{g=1}^G) \times \mathbb{P}(\alpha \mid \mathbf{H}) \\
&\propto \prod_{m=1}^M \prod_{g=1}^G \left[ \prod_{i \neq j} \frac{\lambda_{g,ij}^{y_{ij}^{(m)}} \exp(-\lambda_{g,ij})}{y_{ij}^{(m)}!} \right]^{C_g^{(m)}} \\
&\quad \times \frac{1}{\sqrt{2\pi}s_\alpha} \exp \left( -\frac{1}{2} \frac{(\alpha - m_\alpha)^2}{s_\alpha^2} \right),
\end{aligned}$$

which is sampled using a Metropolis-Hastings (MH) update, as it is not of recognisable/closed form.

The full conditional distribution of the latent space corresponding to network-level cluster  $g$ ,  $\mathbf{Z}_g$ , is derived as follows:

$$\begin{aligned}
\mathbb{P}(\mathbf{Z}_g \mid \dots) &\propto \mathbb{P}(\mathbf{Y} \mid \mathbf{C}, \alpha, \mathbf{Z}_g) \times \mathbb{P}(\mathbf{Z}_g \mid \mathbf{S}_g, \{\boldsymbol{\mu}_{gk}, \boldsymbol{\Sigma}_{gk}\}_{k=1}^{K_g}) \\
&\propto \prod_{m:C_g^{(m)}=1} \prod_{i \neq j} \frac{\lambda_{g,ij}^{y_{ij}^{(m)}} \exp(-\lambda_{g,ij})}{y_{ij}^{(m)}!} \\
&\quad \times \prod_{i=1}^N \prod_{k=1}^{K_g} \left\{ |\boldsymbol{\Sigma}_{gk}|^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk})^T \boldsymbol{\Sigma}_{gk}^{-1} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk}) \right] \right\}^{S_{gk}^i},
\end{aligned}$$

which is sampled using a Metropolis-Hastings update, as it is not of recognisable/closed form.

The number of network-level components  $G$  is sampled from a multinomial distribution, considering options from  $G_+, \dots, G_{\max}$ , with the following probabilities:

$$Pr(G = g \mid \dots) \propto \mathbb{P}(g) \frac{e^{G_+} g!}{g^{G_+} (g - G_+)!} \prod_{h=1}^{G_+} \frac{\Gamma(M_h + \frac{e}{g})}{\Gamma(1 + \frac{e}{g})},$$

where  $\mathbb{P}(g) = \frac{\Gamma(a_G + g - 1) B(a_G + b_G, g - 1 + c_G)}{\Gamma(a_G) \Gamma(g) B(b_G, c_G)}$ , as  $\mathbb{P}(g - 1) \sim \mathcal{B}NB(a_G, b_G, c_G)$ .

The full conditional distribution of the network-level Dirichlet concentration hyperparameter,  $e$ , is derived as follows:

$$\begin{aligned}\mathbb{P}(e | \dots) &\propto \mathbb{P}(e) \frac{e^{G+} \Gamma(e)}{\Gamma(M+e)} \prod_{h=1}^{G+} \frac{\Gamma(M_h + \frac{e}{G})}{\Gamma(1 + \frac{e}{G})} \\ &\propto \frac{\sqrt{\frac{(l_G e)^{l_G} r_G^{r_G}}{(l_G e + r_G)^{l_G + r_G}}}}{e B(\frac{l_G}{2}, \frac{r_G}{2})} \frac{e^{G+} \Gamma(e)}{\Gamma(M+e)} \prod_{h=1}^{G+} \frac{\Gamma(M_h + \frac{e}{G})}{\Gamma(1 + \frac{e}{G})},\end{aligned}$$

which is sampled using a Metropolis-Hastings update, as it is not of recognisable/closed form.

The full conditional distribution of the assignment of node  $i$  to node-level cluster  $k$  within network-level cluster  $g$  is a multinomial with probability derived as follows:

$$\begin{aligned}Pr(S_g^{(i)} = k | \dots) &\propto \mathbb{P}(\mathbf{Z}_g | \mathbf{S}_g, \boldsymbol{\mu}_{gk}, \boldsymbol{\Sigma}_{gk}, \pi_{gk}) \times \mathbb{P}(\mathbf{S}_g | \pi_{gk}) \\ &\propto \pi_{gk} | \boldsymbol{\Sigma}_{gk} |^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk})^T \boldsymbol{\Sigma}_{gk}^{-1} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk}) \right] \\ &= \frac{\pi_{gk} | \boldsymbol{\Sigma}_{gk} |^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk})^T \boldsymbol{\Sigma}_{gk}^{-1} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk}) \right]}{\sum_{h=1}^K \pi_{gh} | \boldsymbol{\Sigma}_{gh} |^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gh})^T \boldsymbol{\Sigma}_{gh}^{-1} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gh}) \right]}.\end{aligned}$$

The full conditional distribution of the mean of node-level cluster  $k$  within network-level cluster  $g$ ,  $\boldsymbol{\mu}_{gk}$  is derived as follows:

$$\begin{aligned}\mathbb{P}(\boldsymbol{\mu}_{gk} | \dots) &\propto \mathbb{P}(\mathbf{Z}_g | \mathbf{S}_g, \boldsymbol{\mu}_{gk}, \boldsymbol{\Sigma}_{gk}, \pi_{gk}) \times \mathbb{P}(\boldsymbol{\mu}_{gk} | \mathbf{H}) \\ &\propto \prod_{i: S_g^{(i)} = k} | \boldsymbol{\Sigma}_{gk} |^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk})^T \boldsymbol{\Sigma}_{gk}^{-1} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk}) \right] \\ &\quad \times | \mathbb{I}_2 |^{-\frac{1}{2}} \exp \left[ (\boldsymbol{\mu}_{gk} - \mathbf{0})^T \mathbb{I}_2^{-1} (\boldsymbol{\mu}_{gk} - \mathbf{0}) \right] \\ &\sim \mathcal{MVN}_2 \left( \boldsymbol{\Sigma}_{gk}^* \left[ \boldsymbol{\Sigma}_{gk}^{-1} \left( \sum_{i: S_g^{(i)} = k} \mathbf{z}_{g,i} \right) + \mathbb{I}_2^{-1} \mathbf{0} \right], [N_{gk} \boldsymbol{\Sigma}_{gk}^{-1} + \mathbb{I}_2^{-1}]^{-1} \right),\end{aligned}$$

where  $N_{gk} = \sum_{i=1}^N \mathbb{I}\{S_g^{(i)} = k\}$ .

The full conditional distribution of the variance of dimension  $q$ ,  $q = 1, 2$ , of node-level cluster  $k$  within network-level cluster  $g$ ,  $\sigma_{gkq}^2$  is derived as follows:

$$\begin{aligned}\mathbb{P}(\sigma_{gkq}^2 | \dots) &\propto \mathbb{P}(\mathbf{Z}_g | \mathbf{S}_g, \boldsymbol{\mu}_{gk}, \sigma_{gkq}^2) \times \mathbb{P}(\sigma_{gkq}^2 | \mathbf{H}) \\ &\propto \prod_{i: S_g^{(i)} = k} (\sigma_{gkq}^2)^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk})^T \frac{1}{\sigma_{gkq}^2} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk}) \right] \\ &\quad \times (\sigma_{gkq}^2)^{-u_{\sigma^2}-1} \exp \left[ -\frac{v_{\sigma^2}}{\sigma_{gkq}^2} \right] \\ &\sim \mathcal{IG}(u_{\sigma^2}^*, v_{\sigma^2}^*),\end{aligned}$$

where  $u_{\sigma^2}^* = \frac{N_{gk}}{2} + u_{\sigma^2}$  and  $v_{\sigma^2}^* = v_{\sigma^2} + \frac{1}{2} \sum_{i:S_g^{(i)}=k} (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk})^T (\mathbf{z}_{g,i} - \boldsymbol{\mu}_{gk})$ .

The full conditional distribution of the node-level mixing proportions  $\boldsymbol{\pi}_g$  within network-level cluster  $g$  is derived as follows:

$$\begin{aligned}\mathbb{P}(\boldsymbol{\pi}_g | \dots) &\propto \mathbb{P}(\mathbf{S}_g | \boldsymbol{\pi}_g) \times \mathbb{P}(\boldsymbol{\pi}_g | K_g, w_g) \\ &\propto \prod_{k=1}^{K_g} \prod_{i=1}^N \pi_{gk}^{S_g^{(i)}} \times \prod_{k=1}^{K_g} \pi_{gk}^{\frac{w_g}{K_g} - 1} \\ &\propto \prod_{k=1}^{K_g} \pi_{gk}^{N_{gk} + \frac{w_g}{K_g} - 1} \\ &\sim \mathcal{D}(\psi_1, \dots, \psi_{K_g}),\end{aligned}$$

where  $\psi = N_{gk} + \frac{w_g}{K_g}$  and  $N_{gk}$  is defined as before.

The number of node-level components  $K_g$ , in network-level component  $g$ , is sampled from a multinomial distribution, considering options from  $K_{g+}, \dots, K_{\max}$ , with the following probabilities:

$$Pr(K_g = k | \dots) \propto \mathbb{P}(k) \frac{w_g^{K_g} k!}{k^{K_g} (k - K_g)!} \prod_{k^*=1}^{K_g} \frac{\Gamma(N_{gk^*} + \frac{w_g}{k})}{\Gamma(1 + \frac{w_g}{k})},$$

where  $\mathbb{P}(k) = \frac{\Gamma(a_K + k - 1) B(a_K + b_K, k - 1 + c_K)}{\Gamma(a_K) \Gamma(k) B(b_K, c_K)}$ , as  $\mathbb{P}(k - 1) \sim \mathcal{B}\mathcal{N}\mathcal{B}(a_K, b_K, c_K)$ .

The full conditional distribution of the node-level Dirichlet concentration hyper-parameter,  $w_g$ , in network-level component  $g$ , is derived as follows:

$$\begin{aligned}\mathbb{P}(w_g | \dots) &\propto \mathbb{P}(w_g) \frac{w_g^{K_g} \Gamma(w_g)}{\Gamma(N + w_g)} \prod_{k^*=1}^{K_g} \frac{\Gamma(N_{gk^*} + \frac{w_g}{K_g})}{\Gamma(1 + \frac{w_g}{K_g})} \\ &\propto \frac{\sqrt{\frac{(l_K w_g)^{l_K} r_K^{r_K}}{(l_K w_g + r_K)^{l_K + r_K}}}}{w_g B(\frac{l_K}{2}, \frac{r_K}{2})} \frac{w_g^{K_g} \Gamma(w_g)}{\Gamma(N + w_g)} \prod_{k^*=1}^{K_g} \frac{\Gamma(N_{gk^*} + \frac{w_g}{K_g})}{\Gamma(1 + \frac{w_g}{K_g})},\end{aligned}$$

which is sampled using a Metropolis-Hastings update, as it is not of recognisable/closed form.

## D Pseudocode of the MCMC algorithm

---

**Algorithm 1** MCMC Algorithm

---

Initialise  $G^{[1]}, \boldsymbol{\tau}^{[1]}, e^{[1]}, \mathbf{C}^{[1]}, \{\mathbf{Z}_g^{[1]}\}_{g=1}^{G^{[1]}}, \alpha^{[1]}, \left\{K_g^{[1]}, \boldsymbol{\pi}^{[1]}, w_g^{[1]}, \mathbf{S}_g^{[1]}, \{\boldsymbol{\mu}_{gk}^{[1]}, \boldsymbol{\Sigma}_{gk}^{[1]}\}_{k=1}^{K_g^{[1]}}\right\}_{g=1}^{G^{[1]}}$

**for**  $t = 1, \dots, T$  **do**

**for**  $m = 1, \dots, M$  **do**

        Sample  $C_g^{(m)[t+1]}$  from a multinomial distribution with  $\Pr(C^{(m)[t+1]} = g | \dots)$  as derived in Section C

**end for**

    Compute  $M_g^{[t+1]}$  for  $g = 1, \dots, G^{[t]}$ , determine  $G_+^{[t+1]}$ , and relabel to ensure the first  $G_+^{[t+1]}$  (network) mixture components are non-empty

**for**  $g = 1, \dots, G_+^{[t+1]}$  **do**

        Sample  $\mathbf{Z}_g^{[t+1]}$  using a block MH step, with  $\hat{z}_{g,i} \sim \mathcal{MVN}_2(\mathbf{z}_{g,i}^{[t]}, \delta_Z^2 \boldsymbol{\Sigma}_{gk}^{[t]})$

**for**  $i = 1, \dots, N$  **do**

            Sample  $S_{gk}^{(i)[t+1]}$  from a multinomial distribution with  $\Pr(S_g^{(i)[t+1]} | \dots)$  as derived in Section C

**end for**

        Compute  $N_{gk}^{[t+1]}$  for  $k = 1, \dots, K_g^{[t]}$ , determine  $K_{g+}^{[t+1]}$ , and relabel to ensure the first  $K_{g+}^{[t+1]}$  (network) mixture components are non-empty

**for**  $k = 1, \dots, K_{g+}^{[t+1]}$  **do**

            Sample  $\boldsymbol{\mu}_{gk}^{[t+1]} \sim \mathcal{MVN}_2(\boldsymbol{\mu}_{gk}^*, \boldsymbol{\Sigma}_{gk}^*)$  and  $\sigma_{gk,q}^{2[t+1]} \sim \mathcal{IG}(u_{\sigma^2}, v_{\sigma^2})$ ,  $q = 1, 2$

**end for**

        Sample  $K_g^{[t+1]}$  from a multinomial distribution with  $\mathbb{P}(k^* | \dots)$  as derived in Section C, considering  $k^* = K_{g+}^{[t]}, \dots, K_{\max}$

        Sample  $w_g^{[t+1]}$  from a MH step, with proposal  $\log(\hat{w}_g) \sim \mathcal{N}(\log(w_g^{[t]}), s_w^2)$

**if**  $K_g^{[t+1]} > K_{g+}^{[t+1]}$  **then**

**for**  $k = K_{g+}^{[t+1]} + 1, \dots, K_g^{[t+1]}$  **do**

                Add an empty node-level mixture component, sampling  $\boldsymbol{\mu}_{gk}^{[t+1]}$  and  $\sigma_{gk,q}^{2[t+1]}$  from the appropriate prior distributions

**end for**

**end if**

        Sample  $\boldsymbol{\pi}^{[t+1]} \sim \mathcal{D}(\psi_{g1}^{[t+1]}, \dots, \psi_{gK_g^{[t+1]}}^{[t+1]})$ , where  $\psi_{gk}^{[t+1]} = \frac{w_g^{[t+1]}}{K_g^{[t+1]}} + N_{gk}^{[t+1]}$

**end for**

    Sample  $\alpha^{[t+1]}$  from a Metropolis-Hastings step, with proposal  $\hat{\alpha} \sim \mathcal{N}(\alpha^{[t]}, \delta_\alpha^2 s_\alpha^2)$

    Sample  $G^{[t+1]}$  from a multinomial distribution with  $\mathbb{P}(g^* | \dots)$  as derived in Section C, considering  $g^* = G_+^{[t]}, \dots, G_{\max}$

    Sample  $e^{[t+1]}$  from a MH step, with proposal  $\log(\hat{e}) \sim \mathcal{N}(\log(e^{[t]}), s_e^2)$

**if**  $G^{[t+1]} > G_+^{[t+1]}$  **then**

**for**  $g = G_+^{[t+1]} + 1, \dots, G^{[t+1]}$  **do**

            Add an empty node-level mixture component, sampling all related parameters from the appropriate prior distribution, assuming no clustering structure in the additional latent spaces

**end for**

**end if**

    Sample  $\boldsymbol{\tau}^{[t+1]} \sim \mathcal{D}(\zeta_1^{[t+1]}, \dots, \zeta_{G^{[t+1]}}^{[t+1]})$ , where  $\zeta_g = \frac{e^{[t+1]}}{G^{[t+1]}} + M_g^{[t+1]}$

**end for**

---

## E Initialisation of the Model Parameters

We initialise the model parameters in the following manner, for  $t = 1$ :

1. Initialise the number of network-level mixture components as  $G^{[1]} = G_0$  (Section B), with all components active.
2. Set network-level mixing proportions uniformly as  $\boldsymbol{\tau}^{[1]} = (\frac{1}{G^{[1]}}, \dots, \frac{1}{G^{[1]}})$ .
3. Set the network-level Dirichlet hyperparameter to a small value  $e^{[1]} = 10^{-5}$  to encourage shrinkage (Malsiner-Walli et al., 2016; Frühwirth-Schnatter and Malsiner-Walli, 2019).
4. Initialise network-level allocations  $\mathbf{C}^{[1]}$  by computing pairwise network distances (Schieber et al., 2017), applying multidimensional scaling (MDS) to obtain a 2-dimensional representation (Cox and Cox, 2008), then clustering (using K-means (Wu, 2012) or mclust (Scrucca et al., 2016)) into  $G^{[1]}$  groups. The initial  $\mathbf{C}^{[1]}$  is set to the partition obtained.
5. Initialise network-level latent spaces  $\{\mathbf{Z}_g^{[1]}\}_{g=1}^{G^{[1]}}$  by averaging geodesic distances within each cluster  $g$  and applying MDS to obtain 2D representations.
6. Initialise the intercept  $\alpha^{[1]}$  by calculating  $\hat{\alpha}_g^{(m)[1]} = \log\left(\frac{1}{N} \sum_{i,j} y_{ij}^{(m)}\right) + \frac{1}{N} \sum_{i \leq j} \|\mathbf{z}_{g,i}^{[1]} - \mathbf{z}_{g,j}^{[1]}\|_2^2$  (Hoff et al., 2002) and averaging across networks  $m$  and clusters  $g$ .
7. For each  $g = 1, \dots, G^{[1]}$ , initialise node-level parameters as follows:
  - (a) Set the number of node-level components  $K_g^{[1]} = K_0$  (Section B).
  - (b) Initialise mixing proportions uniformly:  $\boldsymbol{\pi}_g^{[1]} = (\frac{1}{K_g^{[1]}}, \dots, \frac{1}{K_g^{[1]}})$ .
  - (c) Draw the Dirichlet hyperparameter  $w_g^{[1]}$  from the  $\mathcal{F}(l_K, r_K)$  prior.
  - (d) Initialise node-level allocations  $\mathbf{S}_g^{[1]}$  and means  $\boldsymbol{\mu}_g^{[1]}$  via K-means clustering of  $\mathbf{Z}_g^{[1]}$  into  $K_g^{[1]}$  clusters; set  $\boldsymbol{\mu}_g^{[1]}$  to cluster centres.
  - (e) For each  $k = 1, \dots, K_g^{[1]}$ , sample diagonal covariance elements of  $\boldsymbol{\Sigma}_{gk}^{[1]}$  independently from  $\mathcal{IG}(u_{\sigma^2}, v_{\sigma^2})$  with zero off-diagonal entries.

## F Simulation Study Designs

This section provides the details of the data-generating mechanisms used in Simulation Studies 1 and 2, discussed in Section 4 of the main article. Each study investigates clustering performance under varying structural properties of the generated multiplex.

The node-level cluster means  $\mu_{gk}$  and variances  $\Sigma_{gk}$  were kept consistent across both simulation studies. Specifically, when the number of node-level clusters within the  $g^{\text{th}}$  network-level cluster was one, the cluster mean was set to  $\mu_{g1} = (0, 0)$ . For two node-level clusters, the means were  $\mu_{g1} = (-0.8, 0.8)$  and  $\mu_{g2} = (0.8, -0.8)$ . For three node-level clusters, the means were  $\mu_{g1} = (-0.9, -0.9)$ ,  $\mu_{g2} = (1.4, 0.4)$  and  $\mu_{g3} = (-0.9, 1.4)$ . The node-level cluster variance was fixed to  $\sigma_{gk,q}^2 = 0.25$  for all dimensions  $q = 1, 2$ , for all node-level clusters  $k$  and for all network-level clusters  $g$ .

In each scenario, simulated network multiplexes comprise  $M$  networks, each containing  $N$  nodes. The networks are generated from  $G^*$  network-level clusters with proportions defined by  $\tau$ , and each network-level cluster relates to a corresponding latent space generated with  $K_g$  node-level clusters, characterised by proportions  $\pi_g$ .

Table 7 summarises the parameter configurations for Simulation Study 1. In this study, we generated undirected, count-valued networks where edges follow a Poisson distribution with a log link function. For smaller networks ( $N = 30$ ),  $\alpha = 0.6$  was used, whereas larger networks ( $N > 30$ ) used  $\alpha = -0.4$ .

Table 8 summarises the parameter configurations for Simulation Study 2. In this study, we generated undirected, binary-valued networks where edges follow a Bernoulli distribution with a logit link function. For smaller networks ( $N = 30$ ),  $\alpha = 0.6$  was used, whereas larger networks ( $N > 30$ ) used  $\alpha = -0.4$ .

Table 7: Settings for simulation study 1.

Scenario	$M$	$N$	$G^*$	$\tau$	$K_g$	$\pi$
A	20	30	2	{0.6, 0.4}	{1, 1}	{\{1\}, \{1\}}
B	20	50	2	{0.6, 0.4}	{1, 1}	{\{1\}, \{1\}}
C	20	30	2	{0.6, 0.4}	{1, 2}	{\{1\}, \{0.5, 0.5\}}
D	20	50	2	{0.6, 0.4}	{1, 2}	{\{1\}, \{0.5, 0.5\}}
E	20	30	2	{0.6, 0.4}	{2, 3}	{\{0.7, 0.3\}, \{0.4, 0.3, 0.3\}}
F	20	60	2	{0.6, 0.4}	{2, 3}	{\{0.7, 0.3\}, \{0.4, 0.3, 0.3\}}
G	50	30	2	{0.6, 0.4}	{2, 3}	{\{0.7, 0.3\}, \{0.4, 0.3, 0.3\}}
H	50	60	2	{0.6, 0.4}	{2, 3}	{\{0.7, 0.3\}, \{0.4, 0.3, 0.3\}}

Table 8: Settings for simulation study 2.

Scenario	$M$	$N$	$G^*$	$\tau$	$K_g$	$\pi_g$
I	20	50	2	{3/5, 2/5}	{1, 1}	{\{1\}, \{1\}}
II	50	30	2	{3/5, 2/5}	{2, 3}	{\{7/10, 3/10\}, \{2/5, 3/10, 3/10\}}
III	50	30	3	{2/5, 3/10, 3/10}	{1, 2, 3}	{\{1\}, \{7/10, 3/10\}, \{2/5, 3/10, 3/10\}}
IV	50	60	4	{3/10, 3/10, 1/5, 1/5}	{1, 2, 2, 3}	{\{1\}, \{1/2, 1/2\}, \{7/10, 3/10\}, \{2/5, 3/10, 3/10\}}
V	100	60	4	{3/10, 3/10, 1/5, 1/5}	{1, 2, 2, 3}	{\{1\}, \{1/2, 1/2\}, \{7/10, 3/10\}, \{2/5, 3/10, 3/10\}}

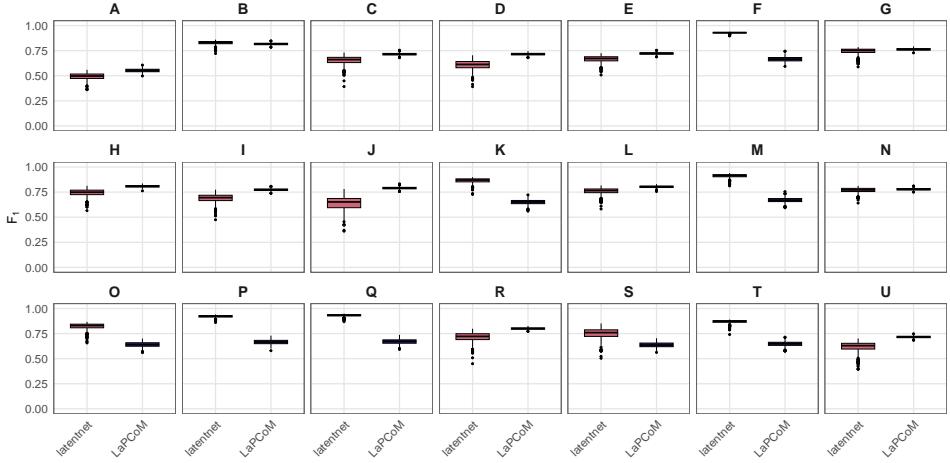


Figure 11: Boxplots showing the distribution of  $F_1$ -scores across 500 multiplexes generated from the posterior predictive distribution, using both `latentnet` and `LaPCoM`, for the Krackhardt application.

## G Posterior Predictive Checks for the Krackhardt Multiplex

In this section, we present the results for the remaining four posterior predictive check (PPC) metrics not included in Section 5.1 of the main article. These metrics further assess model fit by comparing the observed data to the data simulated from the posterior predictive distributions. While the primary metric highlighted in the main article showed a clear advantage for `LaPCoM`, the results for these additional metrics are either comparable across both `LaPCoM` and `latentnet` or show a slight preference for `LaPCoM`. The details of these metrics and their interpretations are provided below.

The  $F_1$ -score, defined as the harmonic mean of precision and recall, is calculated as follows:  $F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$ . Precision is the proportion of true positive ties among all predicted ties, while recall is the proportion of true positive ties among all actual ties. For any network, we calculate the  $F_1$ -score using the simulated network and the observed network across all  $M$  networks in the multiplex and the 500 simulated multiplexes. The  $F_1$ -score ranges from 0 to 1 with higher values representing a better fit. Figure 11 presents a boxplot of the 500  $F_1$ -scores for each network, using both `latentnet` and `LaPCoM`. For `latentnet`, the average median  $F_1$ -score was 0.76. In comparison, `LaPCoM` achieved a comparable but slightly lower average median of 0.72.

Network properties such as density are also considered in our analysis. A network's density is the ratio of the number of observed dyadic connections to the number of possible dyadic connections. For any network, we calculate the squared differ-

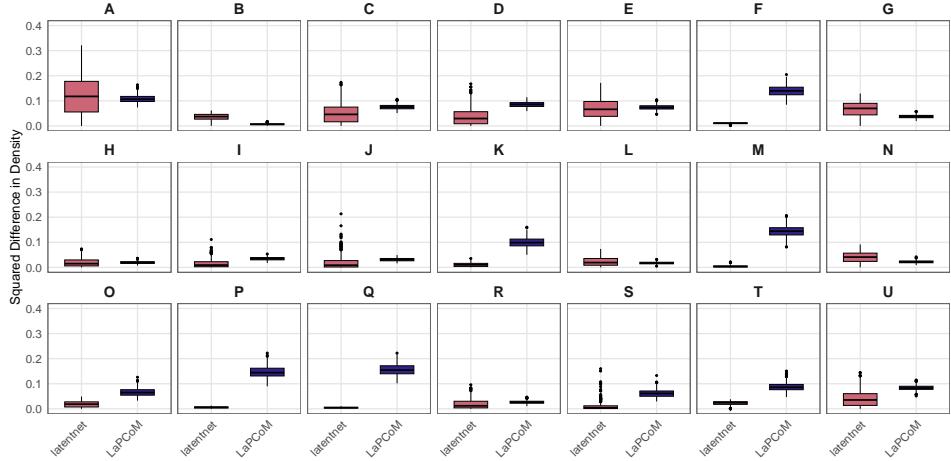


Figure 12: Boxplots showing the distribution of squared differences (in density) across 500 multiplexes generated from the posterior predictive distribution, using both `latentnet` and `LaPCoM`, for the Krackhardt application.

ence between the density of the simulated network and the density of the observed network across all  $M$  networks in the multiplex and the 500 simulated multiplexes. A lower squared difference indicates a better model fit. Figure 12 presents a boxplot of the 500 squared density differences for each network, using both `latentnet` and `LaPCoM`. For `latentnet`, the average median squared difference was 0.028. `LaPCoM` showed a marginally higher median squared difference of 0.072, still indicating a good fit overall.

Figure 13 presents a boxplot of the 500 Hamming distances for each network, using both `latentnet` and `LaPCoM`. The `latentnet` results had an average median of 0.04, compared to 0.06 for `LaPCoM`.

For each network, we compute the precision-recall curve using the simulated tie probabilities and the vectorised observed adjacency matrices across all  $M$  networks and 500 simulated multiplexes. This curve evaluates the trade-off between precision (true positives among predicted ties) and recall (true positives among actual ties) without relying on a specific threshold. We summarise performance by the area under the curve (AUC), a scalar measure of model performance. Figure 14 shows boxplots of AUC distributions across the 500 posterior predictive multiplexes, comparing `latentnet` and `LaPCoM`. The observed network density is marked by a red horizontal line in each panel. For `latentnet`, the median AUC aligns with the observed density, indicating an average fit, whereas for `LaPCoM`, the entire distribution lies above the observed density, demonstrating a markedly better fit.

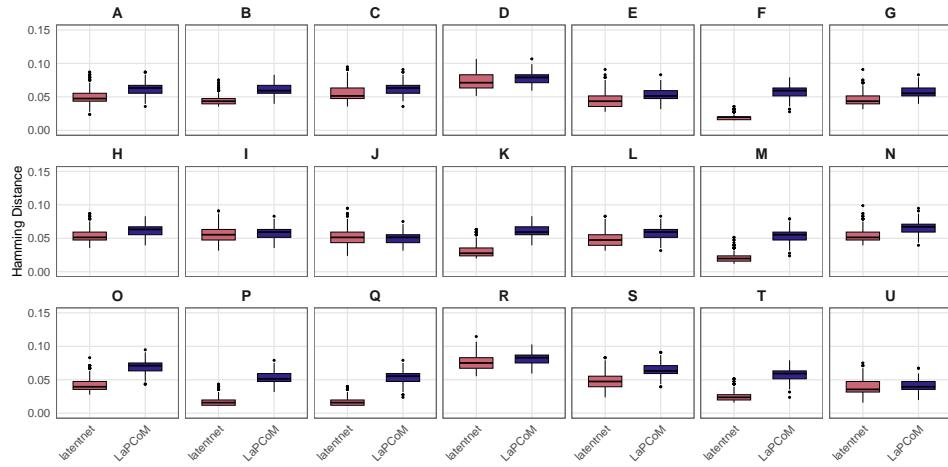


Figure 13: Boxplots showing the distribution of Hamming distances across 500 multiplexes generated from the posterior predictive distribution, using both `latentnet` and `LaPCoM`, for the Krackhardt application.

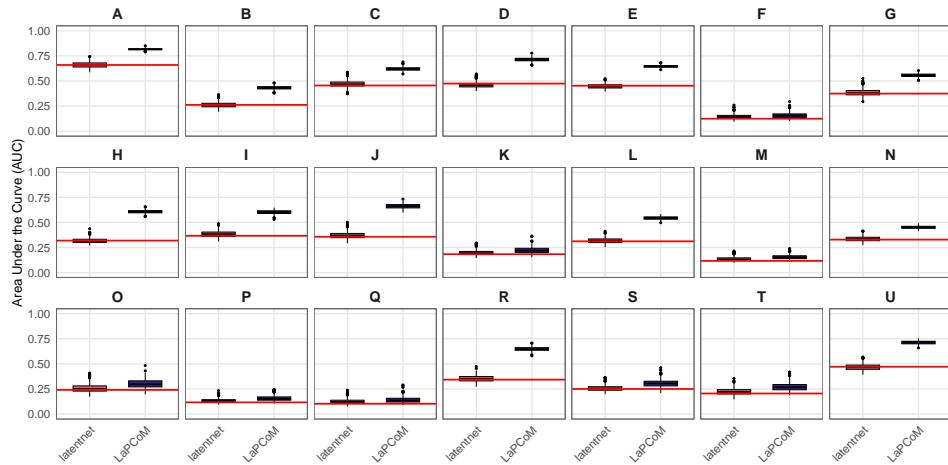


Figure 14: Boxplots showing the distribution of area under the curve (AUC) values across 500 multiplexes generated from the posterior predictive distribution, using both `latentnet` and `LaPCoM`.

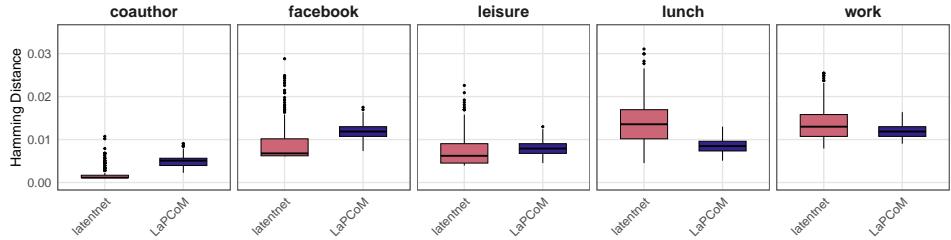


Figure 15: Boxplots showing the distribution of the Hamming distances across 500 multiplexes generated from the posterior predictive distribution, using both `latentnet` and `LaPCoM`, for the Aarhus application.

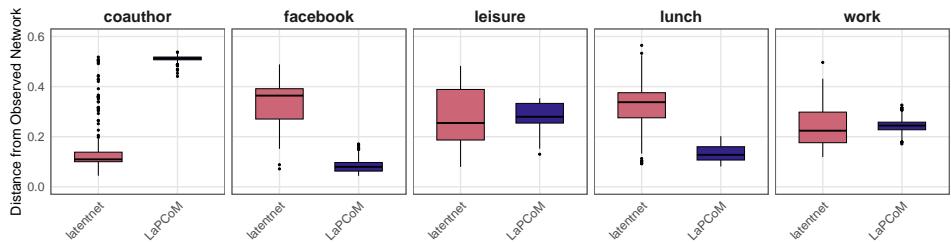


Figure 16: Boxplots showing the distribution of the network distances across 500 multiplexes generated from the posterior predictive distribution, using both `latentnet` and `LaPCoM`, for the Aarhus application.

## H Posterior Predictive Checks for the Aarhus Multiplex

In this section, we present additional PPC metric results for the application presented in Section 5.2 of the main article. As in the previous section, `LaPCoM` generally demonstrates better or comparable performance relative to `latentnet`.

Figure 15 presents a boxplot of the 500 Hamming distances for each network, using both `latentnet` and `LaPCoM`. `LaPCoM` has an average median Hamming distance of 0.009 with an interquartile range (IQR) of 0.002, while `latentnet` has an average median of 0.008 with a median IQR of 0.004. Although the median Hamming distances for `latentnet` tend to be, on average, slightly smaller, `LaPCoM` is generally more precise, with less deviation. This indicates that both models perform at a comparable level with respect to Hamming distance.

Figure 16 presents a boxplot of the 500 Schieber et al. (2017) network distances for each network, using both `latentnet` and `LaPCoM` (lower values are better). The average median network distance for `LaPCoM` was 0.25, slightly lower than the `latentnet` average median of 0.27. Additionally, the average IQR for `latentnet` was 0.12, compared to 0.04 for `LaPCoM`. These results suggest that `LaPCoM` provides a better fit, on average, to the data based on this metric.

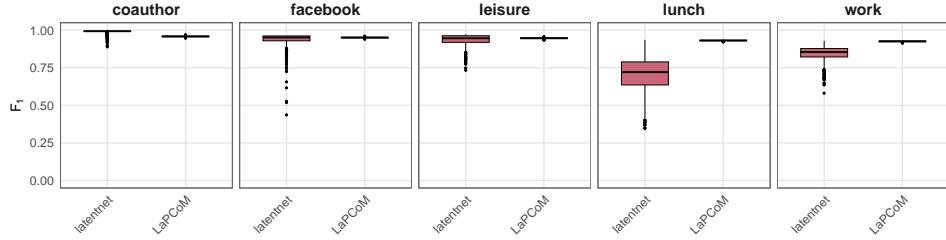


Figure 17: Boxplots showing the distribution of  $F_1$ -scores across 500 multiplexes generated from the posterior predictive distribution, using both `latentnet` and `LaPCoM`.

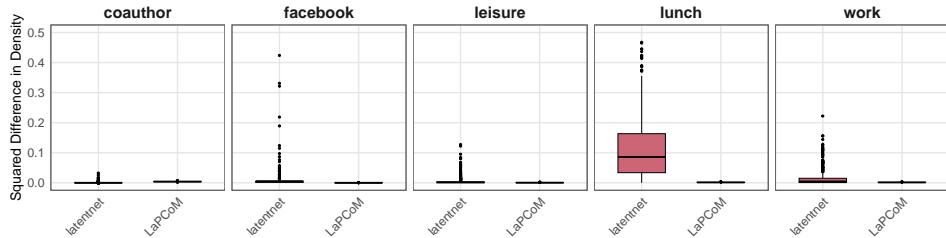


Figure 18: Boxplots showing the distribution of squared differences (in density) across 500 multiplexes generated from the posterior predictive distribution, using both `latentnet` and `LaPCoM`.

Figure 17 shows boxplots of the 500  $F_1$ -scores for each network under both `latentnet` and `LaPCoM`. The models perform similarly for three of the five networks: `LaPCoM` yields an average median  $F_1$ -score of 0.95 with an average IQR of 0.005, while `latentnet` gives a slightly higher median of 0.96 but with a wider IQR of 0.03, indicating lower precision. More pronounced differences appear in the “lunch” and “work” networks. For “work”, `LaPCoM` achieves a median  $F_1$ -score of 0.93 (IQR 0.004), outperforming `latentnet` (median 0.86, IQR 0.06). For “lunch”, `LaPCoM` again excels with a median of 0.93 and IQR of 0.005, compared to `latentnet`’s median of 0.72 and IQR of 0.15, indicating both higher accuracy and lower variability.

Figure 18 presents boxplots of the 500 squared differences in density for each network under both `latentnet` and `LaPCoM`. Across four networks, the models are broadly comparable: `LaPCoM` yields an average median of 0.002 with an average IQR of 0.0006, while `latentnet` has a slightly higher average median of 0.003 and a wider IQR of 0.005. Although the differences are modest, `LaPCoM` demonstrates greater precision, with notably fewer outliers. For the “lunch” network, `LaPCoM` performs substantially better, with a median of 0.002 and IQR of 0.0007, in contrast to `latentnet`, which shows a median of 0.09 and IQR of 0.13, indicating poorer fit.

## I Posterior Predictive Checks for the Primary School Interactions Multiplex

In this section, we present results for an additional PPC metric in relation to the primary school network data described in Section 5.3 of the main article. This metric further evaluates model fit by comparing the observed data to data simulated from the posterior predictive distribution. Specifically, we focus on the empirical cumulative distribution function (ECDF) of the positive edge counts (on the log scale), which provides insight into how well the model captures the distributional structure of non-zero edge weights across the network.

Figure 19 shows the ECDF of the positive counts (on a log scale) for each network. The true ECDF is indicated by the pink line, while the black lines represent 500 ECDFs from posterior predictive samples overlaid for each network. The deviation from the observed ECDF indicates that the model tends to underestimate the occurrence of large edge counts. The edge count distribution in the multiplex data is characterised by overdispersion. Therefore, the Poisson distribution may be too restrictive to capture this aspect of the data. Nonetheless, the model still captures the overall structure of the network reasonably well, and it provides a clustering which aligns with the membership of the students to the different classes and to specific times of the day.

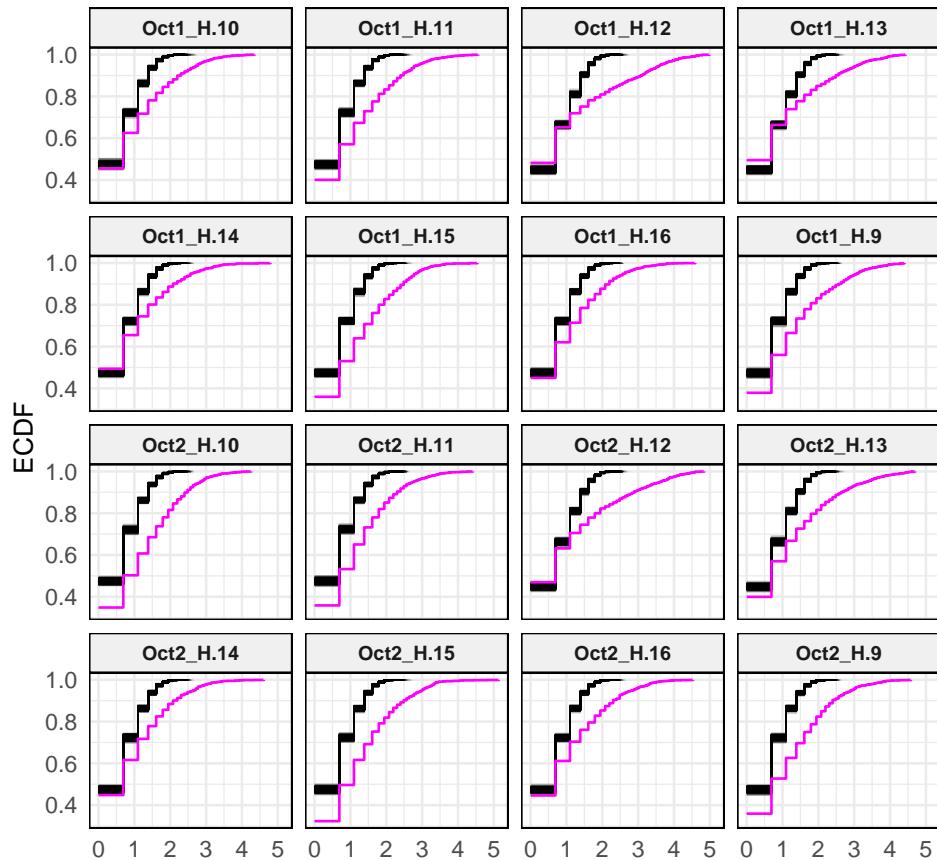


Figure 19: Empirical cumulative distribution functions of log positive counts for each network. The pink line shows the observed ECDF, and black lines show 500 posterior predictive replicates.

## J A Mixture of Finite Mixtures Model versus an Overfitted Mixture Model

As discussed in Section 2.4 of the main article, we chose to implement a mixture of finite mixtures (MFM) model with TS (Miller and Harrison, 2018; Frühwirth-Schnatter et al., 2021), rather than an overfitted mixture (OM) approach (Malsiner-Walli et al., 2016; Frühwirth-Schnatter and Malsiner-Walli, 2019). The OM model fixes the number of components to a large value and assumes a sparse prior on the mixing proportions to recover the underlying number of clusters. In contrast, the MFM places a prior on the number of components (specifically, a translated beta-negative-binomial distribution) and differentiates between total and active (non-empty) components, which are dynamically updated through the TS procedure. This flexibility made the MFM more suitable for our setting, which involves simultaneous inference of network- and node-level clusters.

In the following, we compare three different LaPCoM formulations: the MFM formulation proposed in the main text (MFMTS1), an MFM variant with hyperparameters set following Frühwirth-Schnatter et al. (2021) (MFMTS2), and a formulation employing the OM approach (OM). The aim is to determine the effect of the prior specification for the mixture structure of the model at both network and node level, especially in regards to clustering performance at the network level.

In MFMTS1, we employed our specified hyperparameters, where at the network-level we specify  $G - 1 \sim \mathcal{BNB}(8, 18, 10)$  with  $G_0 = 2$  and  $G_{\max}$  is as in Table 5; at the node-level we specify  $K_g - 1 \sim \mathcal{BNB}(8, 18, 10)$  with  $K_0 = 2$  and  $K_{\max}$  is as in Table 6. Formulation MFMTS2, followed the hyperparameter choices recommended in Frühwirth-Schnatter et al. (2021), where at the network-level we specify  $G - 1 \sim \mathcal{BNB}(1, 4, 3)$  with  $G_0 = 6$  and  $G_{\max} = 10$ , while at the node-level we specify  $K_g - 1 \sim \mathcal{BNB}(1, 4, 3)$  with  $K_0 = 3$  and  $K_{\max} = 15$ . Lastly, we used an OM approach with  $G_{\max} = 10$  and  $K_{\max} = 10$ . The MFMTS2 configuration was based on recommendations in Frühwirth-Schnatter et al. (2021), where we selected  $G_0 = 6$ , approximately 2-3 times the expected true value  $G^* = 2$  used in our simulated data experiments. For  $K_0 = 3$ , we aimed to reflect the range of values observed in the simulations,  $K_g^* = \{1, 2, 3\}$ . For the OM configuration, we chose  $G_{\max} = 10$ , a large yet practical choice inspired by Frühwirth-Schnatter et al. (2021), where they empirically tested values as high as  $G_{\max} = 100$ . However, such a large value was not feasible in our setting. Similarly, we set  $K_{\max} = 10$ .

We implemented simulation studies using the same setup followed in both simulation study 1 (SS1) and simulation study 2 (SS2) of the main text (details are in Section 4 of the main article and in Section F here). In both cases, MCMC was run for 10 replications of 13,000 iterations, including a burn-in period of 3,000 iterations, which was discarded. Samples were thinned at an interval of 10, resulting in a total of 1,000 samples.

The following results focus solely on network-level clustering solutions to assess each method's efficiency in discerning network-level structure. Table 9 summarises the posterior estimates of the optimal number of network-level clusters,  $\hat{G}$ , as de-

terminated by each of the three approaches tested over 10 replications for each of the eight scenarios considered in SS1. Our configuration, MFMTS1, consistently identified the correct  $\hat{G} = 2$  across all scenarios, supporting the suitability of our chosen hyperparameters. In contrast, MFMTS2 underperformed, identifying the correct number of clusters in none of the scenarios. The OM model showed competitive performance to MFMTS1, estimating the correct  $\hat{G}$  in seven out of eight scenarios. Table 10 presents the adjusted rand index (ARI, [Hubert and Arabie \(1985\)](#)) values for the network-level clustering of each approach. The proposed MFMTS1 approach achieved an ARI of one in all eight scenarios. The MFMTS2 configuration displayed lower ARI values. The OM approach performed comparably to MFMTS1, achieving an ARI of one in four out of eight scenarios. These findings substantiate our choice to use a MFM model with the TS procedure over the OM approach, and further validate our specific hyperparameter settings over those recommended in [Frühwirth-Schnatter et al. \(2021\)](#). The MFM configuration proposed in the main text consistently achieved greater accuracy in identifying network-level clusters across diverse scenarios.

Table 11 summarises the posterior estimates of the optimal number of network-level clusters,  $\hat{G}$ , as determined by each of the three model specifications tested over 10 replications for each of the five scenarios considered in SS2. The proposed MFMTS1, consistently identified the correct  $\hat{G}$  across all scenarios. In contrast, MFMTS2 slightly underperformed, accurately capturing  $\hat{G}$  in three of the five scenarios. The model based on the OM approach showed poor performance in this setting, underestimating the number of network-level clusters in four scenarios. Table 12 presents the ARI values for network-level clustering under each approach. Our proposed MFMTS1 approach achieved a perfect or near to perfect ARI. The MFMTS2 configuration displayed slightly lower ARI values, with an ARI of 0.53 in one scenario. The OM approach showed the weakest performance, achieving an ARI of one in only one scenario, and an average ARI of 0.45 across the remaining four scenarios. These results further support our choice to use a MFM model with TS, particularly with our chosen hyperparameters, which outperformed both the OM method and the parameter settings recommended in [Frühwirth-Schnatter et al.](#)

Table 9: Posterior mode of network-level clusters  $\hat{G}_+$ .

Scenario	OM	MFMTS1	MFMTS2
A	8	2	4
B	2	2	4
C	2	2	3
D	2	2	6
E	2	2	3
F	2	2	4
G	2	2	3
H	2	2	4

Table 10: Adjusted Rand index (ARI) of network-level clustering.

Scenario	OM	MFMTS1	MFMTS2
A	0.56	1.00	0.66
B	0.63	1.00	0.69
C	1.00	1.00	0.70
D	1.00	1.00	0.42
E	0.92	1.00	0.77
F	0.93	1.00	0.72
G	1.00	1.00	0.95
H	1.00	1.00	0.82

Table 11: Posterior mode of network-level clusters  $\hat{G}_+$ .

Scenario	$G^*$	OM	MFMTS1	MFMTS2
I	2	9	2	4
II	2	2	2	2
III	3	2	3	3
IV	4	2	4	5
V	4	2	4	4

Table 12: Adjusted Rand index (ARI) of network-level clustering.

Scenario	OM	MFMTS1	MFMTS2
A	0.32	0.98	0.53
B	1.00	1.00	0.99
C	0.56	1.00	1.00
D	0.47	1.00	0.97
E	0.43	0.99	1.00

(2021).

Because the MCMC algorithms in this comparison were run for fewer iterations than in the main simulations in the main text, in some instances the permutation test failed to identify valid classification sequences of  $1, \dots, \hat{G}_+$  (see Section 3.3 of the main article). This occurred in the SS1 set-up for MFMTS1 (one simulation, Scenario F), MFMTS2 (two simulations, Scenario F), and OM (one simulation each in Scenarios A and F). These instances reflected random variation or insufficient number of iterations and were discarded.

## References

- Barile, F., Lunagómez, S., and Nipoti, B. (2024). Bayesian nonparametric modeling of heterogeneous populations of networks. arXiv 2410.10354. [3](#)
- Battaglia, E., Peiretti, F., and Pensa, R. G. (2024). Co-clustering: A survey of the main methods, recent trends, and open problems. *ACM Comput. Surv.*, 57(2). [2](#)
- Bhattacharya, A. and Dunson, D. B. (2011). Sparse Bayesian infinite factor models. *Biometrika*, 98(2):291–306. [25](#)
- Biernacki, C., Jacques, J., and Keribin, C. (2023). A survey on model-based co-clustering: High dimension and estimation challenges. *Journal of Classification*, 40:332–381. [2](#)
- Brandes, U., Lerner, J., and Nagel, U. (2011). Network ensemble clustering using latent roles. *Advances in Data Analysis and Classification*, 5:81–94. [3](#)
- Chabert-Liddell, S.-C., Barbillon, P., and Donnet, S. (2024). Learning common structures in a collection of networks. an application to food webs. *The Annals of Applied Statistics*, 18(2):1213 – 1235. [3](#)
- Cox, M. A. A. and Cox, T. F. (2008). *Multidimensional scaling*, pages 315–347. Springer Berlin Heidelberg, Berlin, Heidelberg. [40](#)
- Durante, D., Dunson, D. B., and Vogelstein, J. T. (2017). Nonparametric Bayes modeling of populations of networks. *Journal of the American Statistical Association*, 112(520):1516–1530. [3](#), [4](#), [12](#)
- D’Angelo, S., Alfó, M., and Fop, M. (2023). Model-based clustering for multidimensional social networks. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 186(3):481–507. [2](#), [3](#), [4](#), [9](#)
- D’Angelo, S., Murphy, T. B., and Alfó, M. (2019). Latent space modelling of multidimensional networks with application to the exchange of votes in Eurovision song contest. *The Annals of Applied Statistics*, 13(2):900 – 930. [2](#), [3](#), [4](#)
- Fan, X., Pensky, M., Yu, F., and Zhang, T. (2022). Alma: Alternating minimization algorithm for clustering mixture multilayer network. *Journal of Machine Learning Research*, 23(330):1–46. [3](#)
- Frühwirth-Schnatter, S. (2011). *Dealing with label switching under model uncertainty*, chapter 10, pages 213–239. John Wiley & Sons, Ltd. [9](#), [10](#), [19](#)
- Frühwirth-Schnatter, S. (2023). Generalized cumulative shrinkage process priors with applications to sparse Bayesian factor analysis. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 381(2247):20220148. [25](#)

- Frühwirth-Schnatter, S. and Malsiner-Walli, G. (2019). From here to infinity: Sparse finite versus Dirichlet process mixtures in model-based clustering. *Advances in Data Analysis and Classification*, 13(1):33–64. [9](#), [40](#), [49](#)
- Frühwirth-Schnatter, S., Malsiner-Walli, G., and Grün, B. (2021). Generalized mixtures of finite mixtures and telescoping sampling. *Bayesian Analysis*, 16(4):1279 – 1307. [3](#), [6](#), [7](#), [9](#), [27](#), [31](#), [32](#), [33](#), [34](#), [49](#), [50](#)
- Gemmetto, V., Barrat, A., and Cattuto, C. (2014). Mitigation of infectious disease at school: Targeted class closure vs school closure. *BMC infectious diseases*, 14(1):695. [22](#), [27](#)
- Gollini, I. and Murphy, T. B. (2016). Joint modeling of multiple network views. *Journal of Computational and Graphical Statistics*, 25(1):246–265. [2](#), [4](#)
- Gwee, X. Y., Gormley, I. C., and Fop, M. (2023). Model-based clustering for network data via a latent shrinkage position cluster model. arXiv 2310.03630. [25](#)
- Gwee, X. Y., Gormley, I. C., and Fop, M. (2025). A latent shrinkage position model for binary and count network data. *Bayesian Analysis*, 20(2):405 – 433. [4](#)
- Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 170(2):301–354. [2](#), [3](#), [4](#)
- Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098. [2](#), [3](#), [4](#), [11](#), [40](#)
- Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218. [11](#), [50](#)
- Jing, B.-Y., Li, T., Lyu, Z., and Xia, D. (2021). Community detection on mixture multilayer networks via regularized tensor decomposition. *The Annals of Statistics*, 49(6):3181 – 3205. [3](#)
- Josephs, N., Amini, A. A., Paez, M., and Lin, L. (2025). Nested stochastic block model for simultaneously clustering networks and nodes. arXiv 2307.09210. [3](#)
- Kowal, D. R. and Canale, A. (2023). Semiparametric functional factor models with Bayesian rank selection. *Bayesian Analysis*, 18(4):1161 – 1189. [25](#)
- Krackhardt, D. (1987). Cognitive social structures. *Social Networks*, 9(2):109–134. [15](#), [17](#), [27](#)
- Krivitsky, P. N. and Handcock, M. S. (2008). Fitting latent cluster models for networks with latentnet. *Journal of Statistical Software*, 24(5):1–23. [17](#), [20](#)

- Krivitsky, P. N. and Handcock, M. S. (2024). *latentnet: Latent position and cluster models for statistical networks*. The Statnet Project (<https://statnet.org>). R package version 2.11.0. [17](#), [20](#)
- Lee, C. and Wilkinson, D. J. (2019). A review of stochastic block models and extensions for graph clustering. *Applied Network Science* 4:1, 4:1–50. [3](#)
- Legramanti, S., Durante, D., and Dunson, D. B. (2020). Bayesian cumulative shrinkage for infinite factorizations. *Biometrika*, 107(3):745–752. [25](#)
- Liu, H. and Liu, R. (2025). Latent variable modeling of social networks with directional relations: An exploration of profile similarity of latent factors. *Structural Equation Modeling: A Multidisciplinary Journal*, 32(3):550–564. [26](#)
- Lu, C., Rastelli, R., and Friel, N. (2025). A zero-inflated Poisson latent position cluster model. arXiv 2502.13790. [26](#)
- Lunagómez, S., Olhede, S. C., and Wolfe, P. J. (2021). Modeling network populations via graph distances. *Journal of the American Statistical Association*, 116(536):2023–2040. [2](#), [3](#)
- Magnani, M., Micenkova, B., and Rossi, L. (2013). Combinatorial analysis of multiple networks. arXiv 1303.4986. [2](#), [18](#), [27](#)
- Malsiner-Walli, G., Frühwirth-Schnatter, S., and Grün, B. (2016). Model-based clustering based on sparse finite Gaussian mixtures. *Statistics and Computing*, 26:303–324. [6](#), [40](#), [49](#)
- Mantziou, A., Lunagómez, S., and Mitra, R. (2024). Bayesian model-based clustering for populations of network data. *The Annals of Applied Statistics*, 18(1):266 – 302. [3](#), [13](#), [14](#), [15](#)
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2023). *e1071: Misc functions of the department of statistics, probability theory group (Formerly: E1071)*, TU Wien. R package version 1.7-13. [11](#)
- Miller, J. W. and Harrison, M. T. (2018). Mixture models with a prior on the number of components. *Journal of the American Statistical Association*, 113(521):340–356. [6](#), [49](#)
- Noroozi, M. and Pensky, M. (2024). Sparse subspace clustering in diverse multiplex network model. *Journal of Multivariate Analysis*, 203:105333. [3](#)
- Oksanen, J., Simpson, G. L., Blanchet, F. G., Kindt, R., Legendre, P., Minchin, P. R., O'Hara, R., Solymos, P., Stevens, M. H. H., Szoecs, E., Wagner, H., Barbour, M., Bedward, M., Bolker, B., Borcard, D., Carvalho, G., Chirico, M., De Caceres, M., Durand, S., Evangelista, H. B. A., FitzJohn, R., Friendly, M., Furneaux, B., Hannigan, G., Hill, M. O., Lahti, L., McGlinn, D., Ouellette, M.-H., Ribeiro Cunha, E., Smith, T., Stier, A., Ter Braak, C. J., and Weedon, J. (2024). *vegan: Community ecology package*. R package version 2.6-8. [11](#)

- Pensky, M. and Wang, Y. (2024). Clustering of diverse multiplex networks. *IEEE Transactions on Network Science and Engineering*, 11(4):3441–3454. [3](#)
- R Core Team (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. [4](#)
- Rebafka, T. (2023). *graphclust: Hierarchical graph clustering for a collection of networks*. R package version 1.3. [13](#)
- Rebafka, T. (2024). Model-based clustering of multiple networks with a hierarchical algorithm. *Statistical Computing*, 34(32). [3](#), [12](#), [13](#)
- Ren, S., Wang, X., Liu, P., and Zhang, J. (2023). Bayesian nonparametric mixtures of exponential random graph models for ensembles of networks. *Social Networks*, 74:156–165. [2](#), [3](#)
- Salter-Townshend, M. and McCormick, T. H. (2017). Latent space models for multiview network data. *The Annals of Applied Statistics*, 11(3):1217–1244. [2](#)
- Schieber, T. A., Carpi, L., Díaz-Guilera, A., Pardalos, P. M., Masoller, C., and Ravetti, M. G. (2017). Quantification of network structural dissimilarities. *Nature communications*, 8:13928. [17](#), [18](#), [23](#), [40](#), [45](#)
- Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):289–317. [40](#)
- Sewell, D. K. and Chen, Y. (2016). Latent space models for dynamic networks with weighted edges. *Social Networks*, 44:105–116. [26](#)
- Sewell, D. K. and Chen, Y. (2017). Latent space approaches to community detection in dynamic networks. *Bayesian Analysis*, 12(2):351 – 377. [26](#)
- Signorelli, M. and Wit, E. C. (2020). Model-based clustering for populations of networks. *Statistical Modelling*, 20(1):9–29. [3](#), [13](#), [14](#), [15](#)
- Stanley, N., Shai, S., Taylor, D., and Mucha, P. J. (2016). Clustering network layers with the strata multilayer stochastic block model. *IEEE Transactions on Network Science and Engineering*, 3(2):95–105. [3](#)
- Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J., Quaggiotto, M., Van den Broeck, W., Régis, C., Lina, B., and Vanhems, P. (2011). High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6(8):e23176. [22](#)
- Sweet, T. M., Flynt, A., and Choi, D. (2019). Clustering ensembles of social networks. *Network Science*, 7(2):141–159. [3](#)

- Sweet, T. M., Thomas, A. C., and Junker, B. W. (2013). Hierarchical network models for education research: Hierarchical latent space models. *Journal of Educational and Behavioral Statistics*, 38(3):295–318. [2](#)
- Taya, F., de Souza, J., Thakor, N. V., and Bezerianos, A. (2016). Comparison method for community detection on brain networks from neuroimaging data. *Applied Network Science*, 1(1):8. [2](#)
- Wade, S. (2015). *mcclust.ext: Point estimation and credible balls for Bayesian cluster analysis*. R package version 1.0. [10](#)
- Wade, S. and Ghahramani, Z. (2018). Bayesian cluster analysis: Point estimation and credible balls (with discussion). *Bayesian Analysis*, 13(2). [9](#), [10](#)
- Wu, J. (2012). *Advances in K-means clustering: A data mining thinking*. Springer Theses. Springer Berlin, Heidelberg, 1 edition. [40](#)
- Yin, F., Shen, W., and Butts, C. T. (2022). Finite mixtures of ERGMs for modeling ensembles of networks. *Bayesian Analysis*, 17(4):1153–1191. [2](#), [3](#)

## Christian Hennig

### *Material list:*

Christian Hennig (2025) The role of data visualisation in cluster analysis. Slides.

Ullmann, T., Hennig, C., and Boulesteix, A.-L. (2022) Validation of cluster analysis results on validation data: A systematic framework. *WIREs Data Mining and Knowledge Discovery*, 12(3), e1444.



## The role of data visualisation in cluster analysis

Christian Hennig

University of Bologna

Christian Hennig

The role of data visualisation in cluster analysis

### 1. Introduction

Motivation:

For most clustering problems  
there's a confusing array of methods available  
that operate on different (often incommensurable) bases  
(such as probability mixture models vs. distances).

Even considering the same clustering approach,  
the number of clusters problem is notoriously hard.

Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

An example with Euclidean data

An example with categorical data

There are comparative benchmark studies, but results are inconsistent, generalisability questionable.

*Aim:* Help with making the decisions required for cluster analysis.

Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

An example with Euclidean data

An example with categorical data

I argue that visualisation has a vital role in decision making (“what is the best clustering for my data?”) and interpretation of any clustering, more so than elsewhere in statistics.

Why?

Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

An example with Euclidean data

An example with categorical data

Clustering is unsupervised. This means that  
*there is no obvious formal definition and measure of quality!*

Many clustering methods optimise an objective function  
(e.g., Gaussian mixture likelihood,  
sum of distances to centroid),  
but different clustering methods and objective functions  
*imply different concepts of what a cluster is.*

We can find a formally best clustering  
according to a given objective function,  
but there is no formally optimal objective function.

Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

An example with Euclidean data

An example with categorical data

Conflicting cluster concepts:

- ▶ Clusters defined by density gaps and separation,
- ▶ requiring small within-cluster distances,
- ▶ fitting clusters by certain “homogeneous” distributions such as Gaussian,
- ▶ requiring clusters to be interpretable by marginal distributions,
- ▶ allowing for very small (even one point) clusters or not,
- ▶ require small number of clusters, or higher granularity.

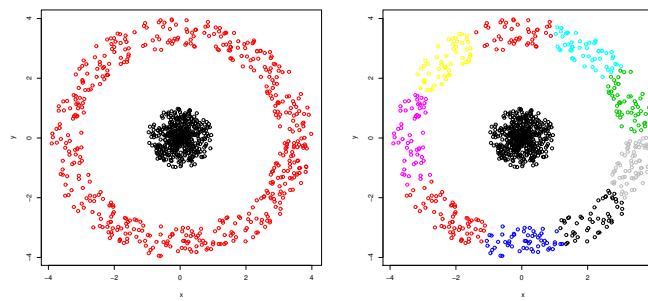
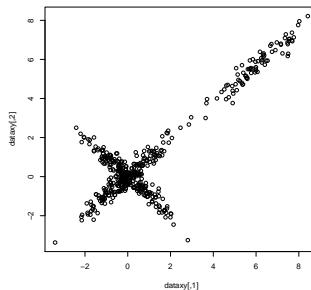
Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

An example with Euclidean data

An example with categorical data



Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

An example with Euclidean data

An example with categorical data

Ideally user would know what cluster concept they need  
*before data* (from background information, aim of clustering).

But in cluster analysis we often don't know enough in advance.

In literature, cluster analysis methods  
are usually motivated by idealised example situations  
that are much simpler than reality.

This is what clustering methods are based on.

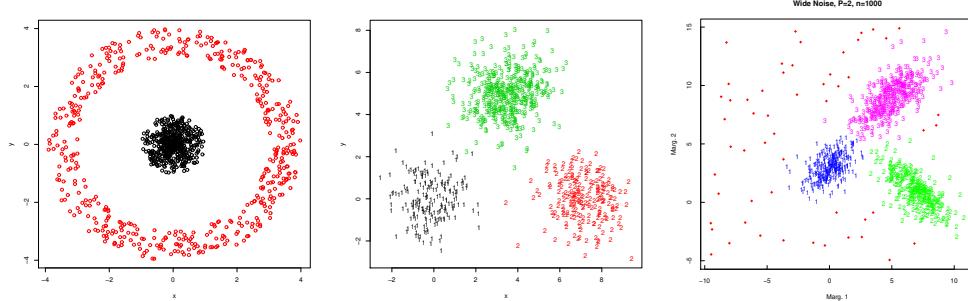
Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

## An example with Euclidean data

## An example with categorical data



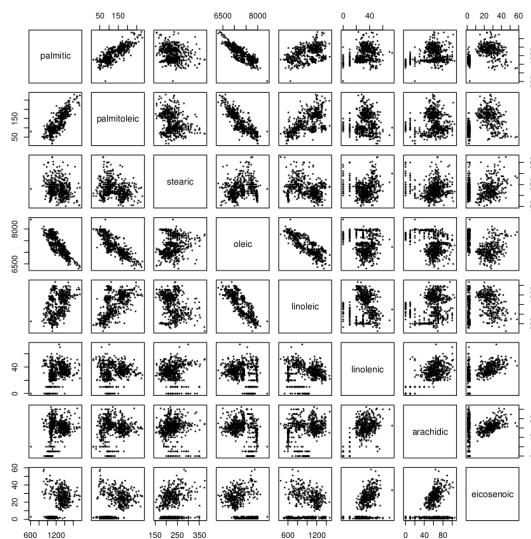
Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

## An example with Euclidean data

## An example with categorical data



Real data can have idiosyncratic features that no standard method has on its radar.

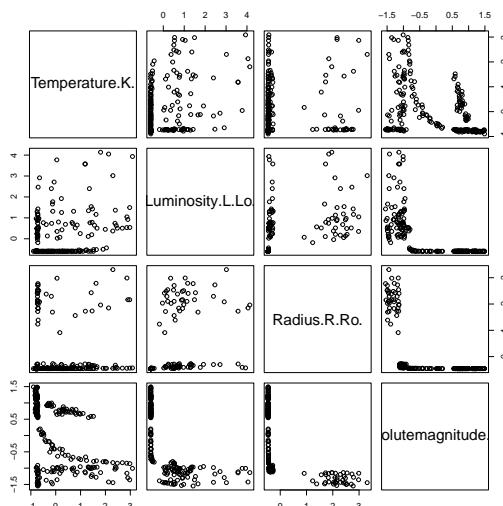
Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

An example with Euclidean data

An example with categorical data



Real data can have groups  
with vastly different within-group distributions.

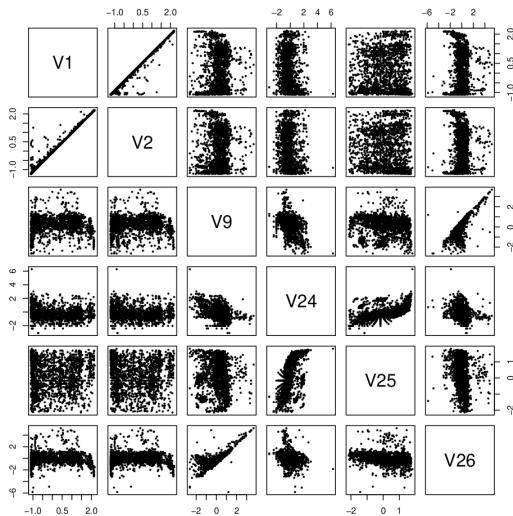
Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

An example with Euclidean data

An example with categorical data



There may be structure without separation,  
also outliers/very small groups.

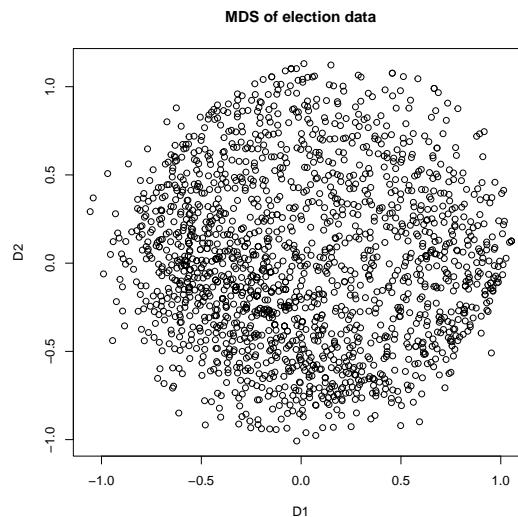
Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

An example with Euclidean data

An example with categorical data



Can any meaningful groups be found?

Christian Hennig

The role of data visualisation in cluster analysis

## Introduction

An example with Euclidean data

An example with categorical data

Don't trust any formal clustering method  
without checking against the data!

Need visualisation to understand interplay between  
method's cluster concept, user's cluster concept,  
and what is in the data.

Also need understand limitations of visualisation.  
May see artifacts, may not see some structure.

Decision from visualisation is informal and subjective,  
may bias inference,  
validation (e.g. on independent data) is crucial.

Christian Hennig

The role of data visualisation in cluster analysis

What can we learn, what may we miss with visualisation?

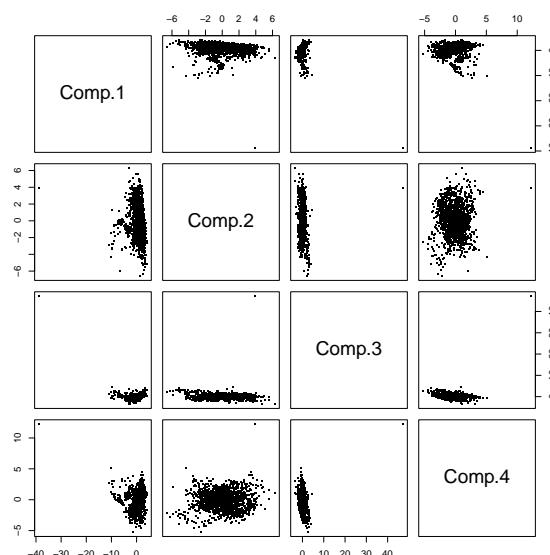
## 2. An example with Euclidean data

### Steel plates data

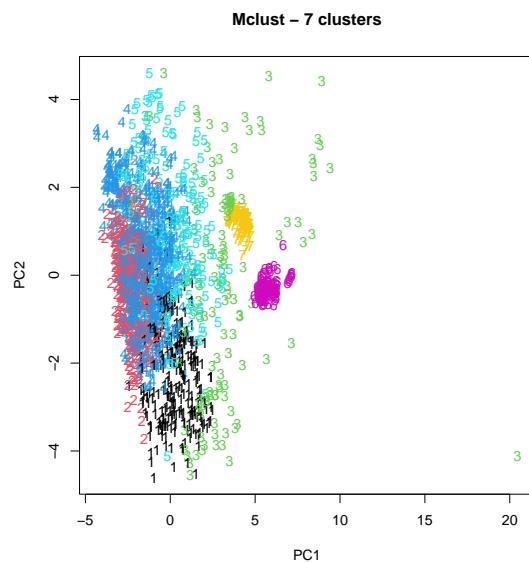
Steel plates data (UCI repository)

1941 steel plates, 22 standardised measurements  
(originally 27)

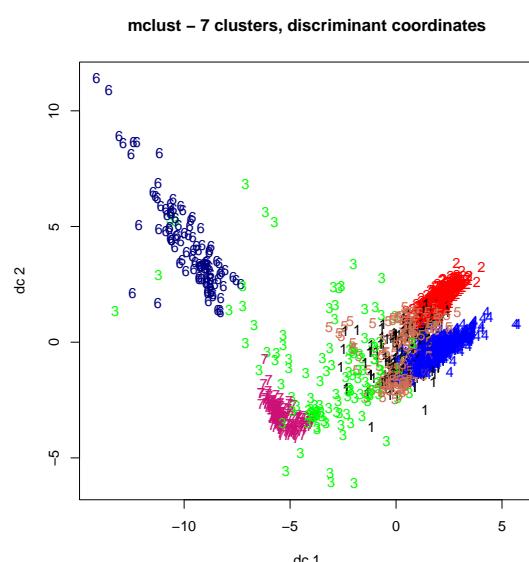
Classification of types of surface defects.



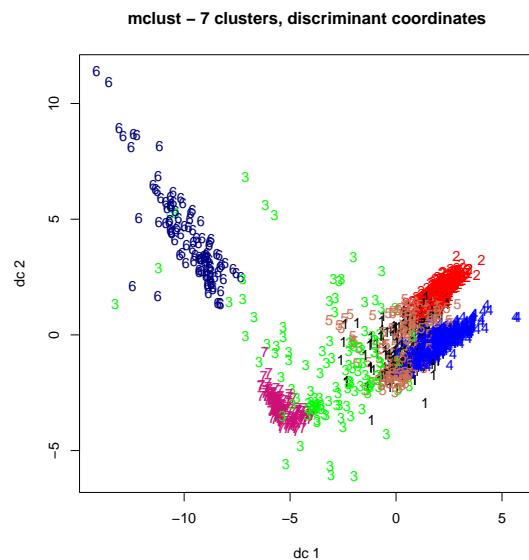
Some patterns, data isn't made up by clusters of same kind;  
outlier issue (use Hubert's robust PCA from now).



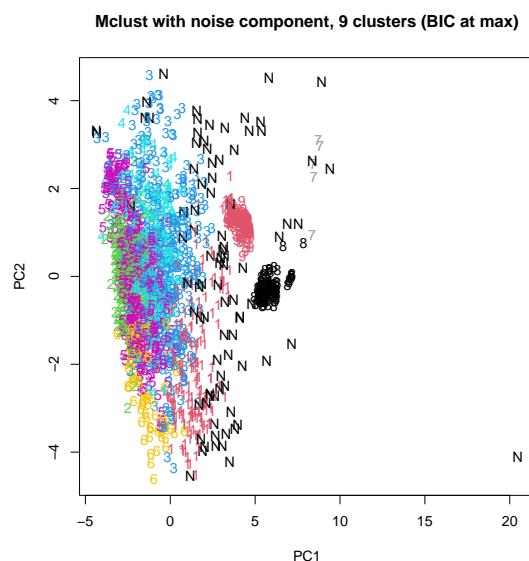
Cluster 3 looks suspicious.  
PCA isn't optimal for showing clusters.



Discriminant coordinates optimise ratio  
between *between* and *within* cluster variation.



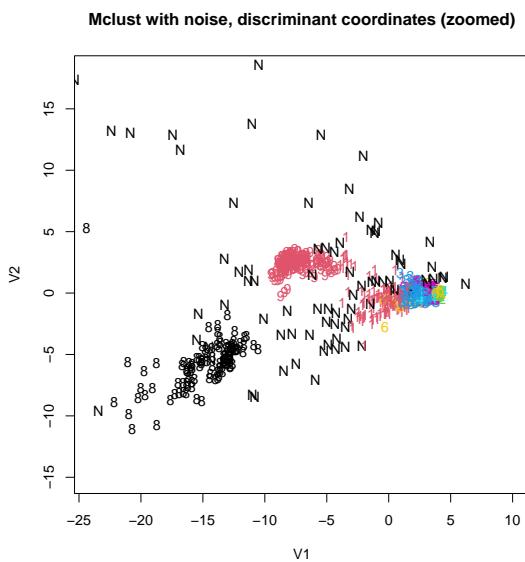
For interpretation understand dimension reduction,  
optimisation.



## Introduction

### An example with Euclidean data

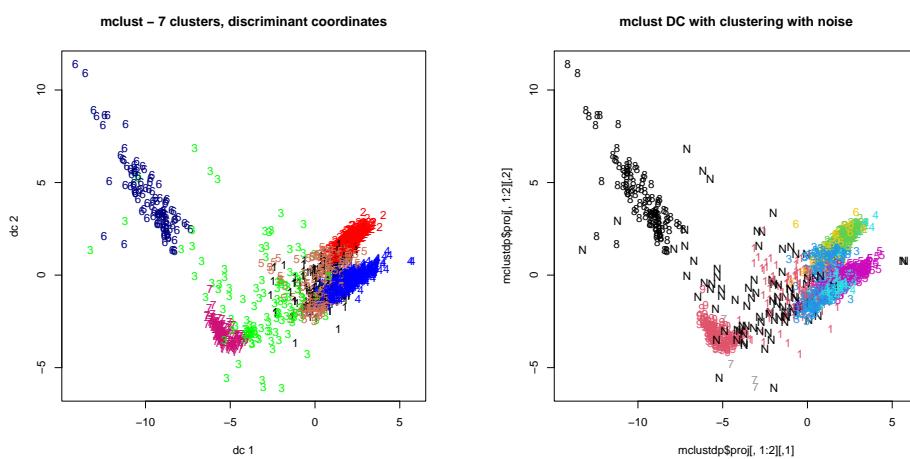
### An example with categorical data



Can compute DCs ignoring noise and then project noise.

Christian Hennig	The role of data visualisation in cluster analysis
Introduction	
An example with Euclidean data	
An example with categorical data	

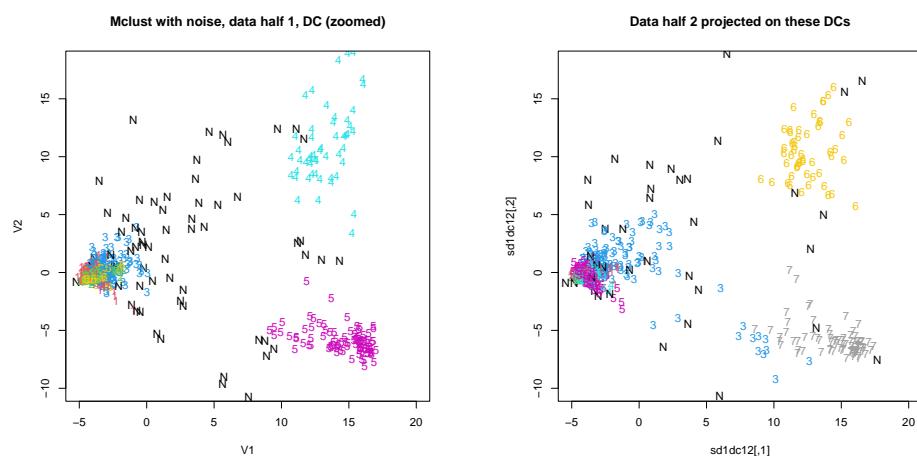
**Idea:** Plot clustering with noise on no-noise DC.



Many similarities, non-noise 3 is classified noise, some no-noise clusters split up.

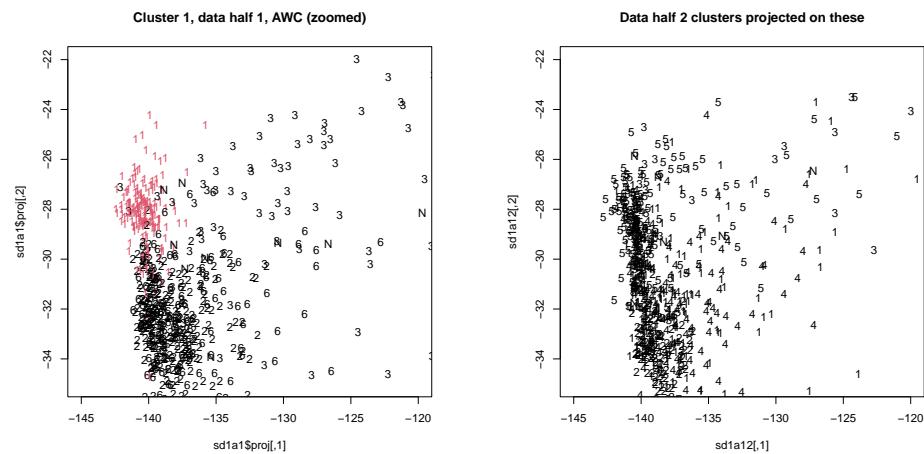
**Further idea:** Split data,  
project data half 2 on DCs of data half 1  
to see whether patterns are stable.  
(Hinted at in Ullmann, Hennig, Boulesteix 2021)

DCs are optimised on half 1 but *not* on half 2,  
removes “optimisation bias”.



**Introduction**  
**An example with Euclidean data**  
**An example with categorical data**

Explore whether clusters can be separated that aren't in DCs.  
 Use Asymmetric Weighted Coordinates (Hennig 2004).  
 Cluster 1 doesn't "cluster" on half 2.

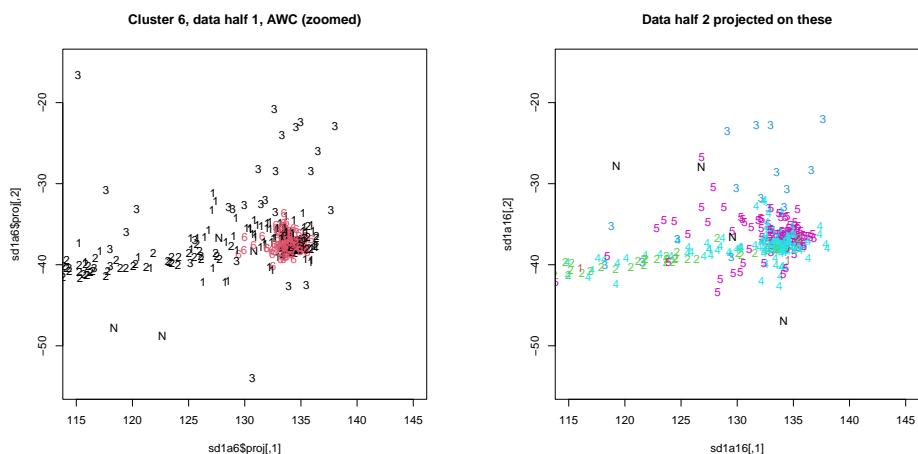


Christian Hennig

The role of data visualisation in cluster analysis

**Introduction**  
**An example with Euclidean data**  
**An example with categorical data**

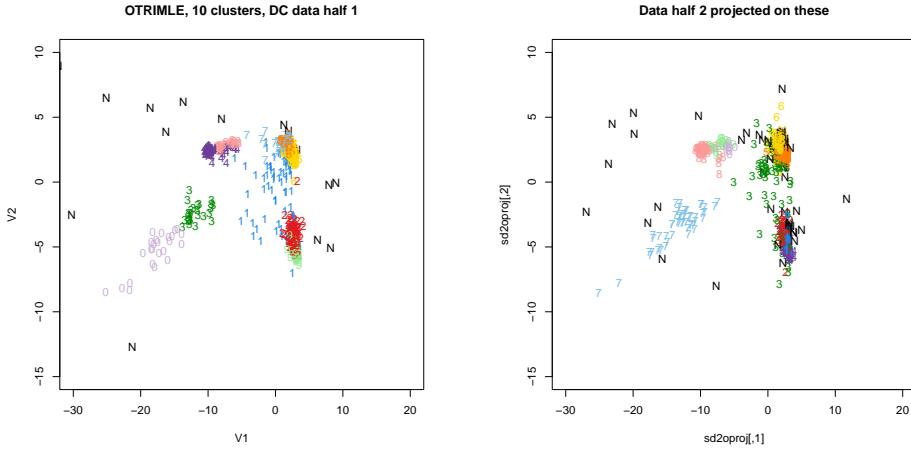
Cluster 6:



Christian Hennig

The role of data visualisation in cluster analysis

**Introduction**  
**An example with Euclidean data**  
**An example with categorical data**

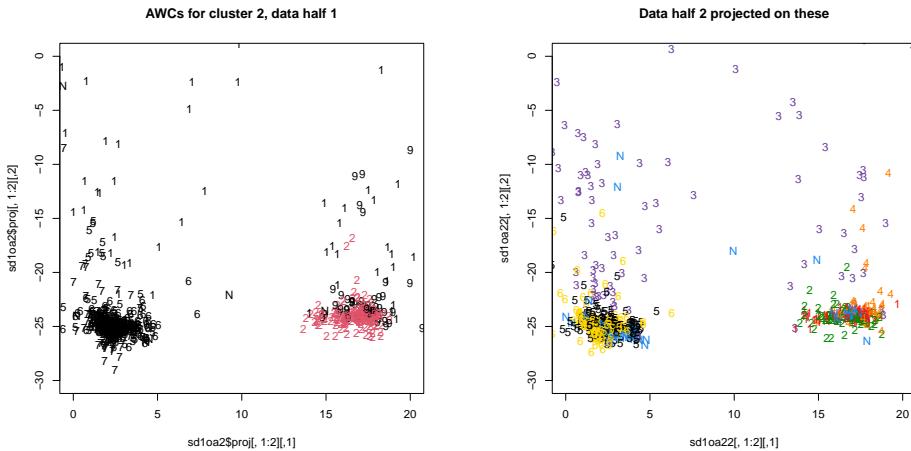


More structure, but maybe “oversplitting” of clusters?  
 Dimension reduction issue? Investigate cluster 2!

Christian Hennig

The role of data visualisation in cluster analysis

**Introduction**  
**An example with Euclidean data**  
**An example with categorical data**



Can't separate cluster 2 from 9.  
 Maybe best clustering but could do with merging.

Christian Hennig

The role of data visualisation in cluster analysis

Generally we can see that there are clear clustering structures in the data,  
and the clustering methods give us these  
*and some more, and may split them up,*  
and some of what we get is quite unstable.

Note that results stem from interaction  
data ↔ clustering method,  
stability problem can well stem from data  
and doesn't indicate "wrong" method  
- unless other method finds clearly better clustering.

Our visual intuition is Euclidean,  
may not apply straight away to non-Euclidean data!

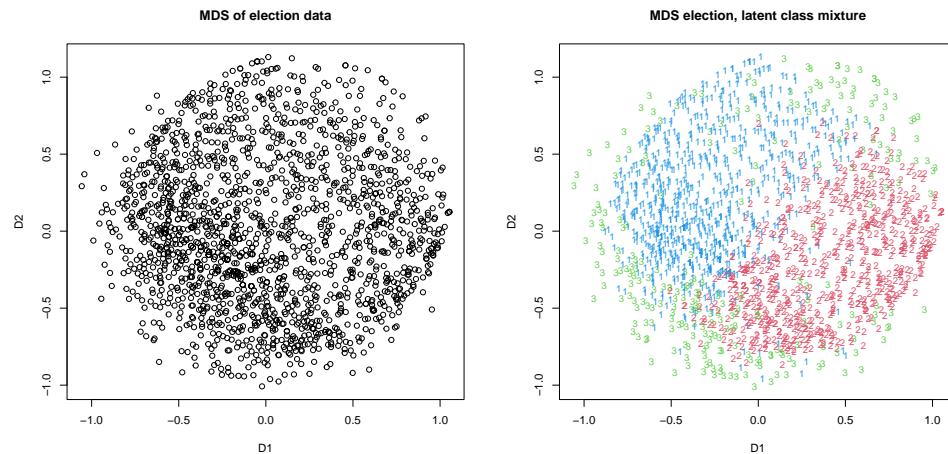
### 3. An example with categorical data

#### US election survey (Gore vs. Bush 2000)

1785 respondents, 12 questions,  
6 each about assessment of Gore and Bush,  
4 categories of agreement plus NA (5th category)

Simple matching distance for MDS,  
Latent class mixture (local independence).

Introduction  
An example with Euclidean data  
An example with categorical data



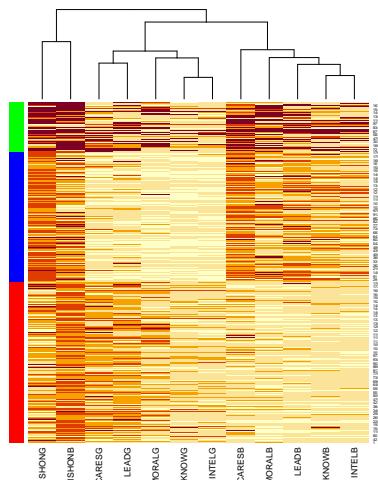
Clustering doesn't look convincing on MDS  
(information loss from dimension reduction; stress 30%!).

Christian Hennig

The role of data visualisation in cluster analysis

Introduction  
An example with Euclidean data  
An example with categorical data

Heatmaps show full information:



Looks strikingly convincing, but bias from order?

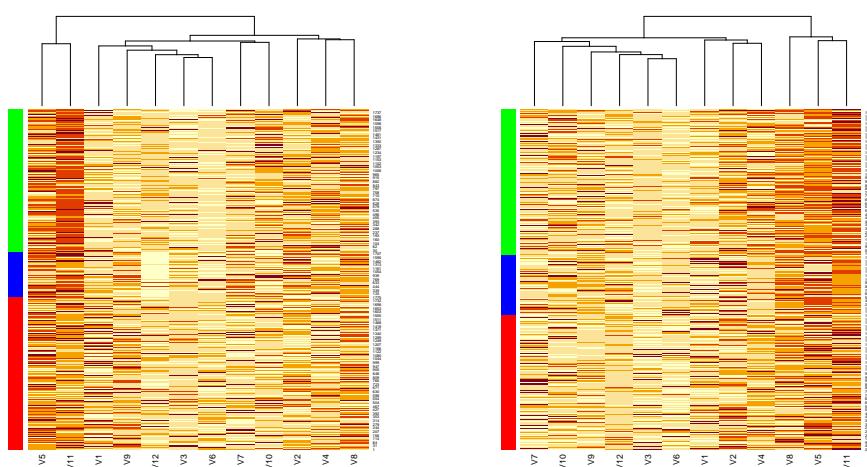
Christian Hennig

The role of data visualisation in cluster analysis

Problem: Heatmaps depend strongly on order;  
order is determined by clusters for making them show up,  
but this can be “overoptimistic”.

**Idea:** Simulate fake data from “null model”  
to see how data heatmaps compare to “no true structure”  
(Hennig & Lin 2015).

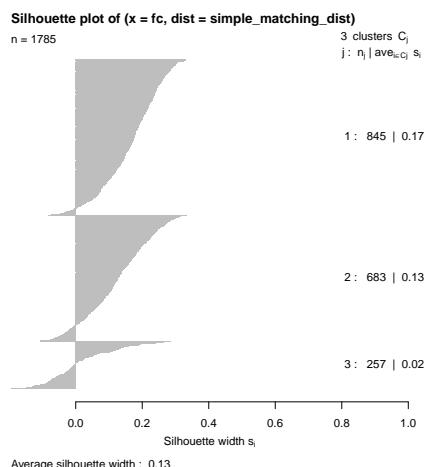
Null model (local independence) homogeneous data  
clearly show less structure.



“Homogeneous” impression from MDS is misleading here!?

**Introduction**  
 An example with Euclidean data  
 An example with categorical data

... but the clustering is not that good  
in terms of distances:

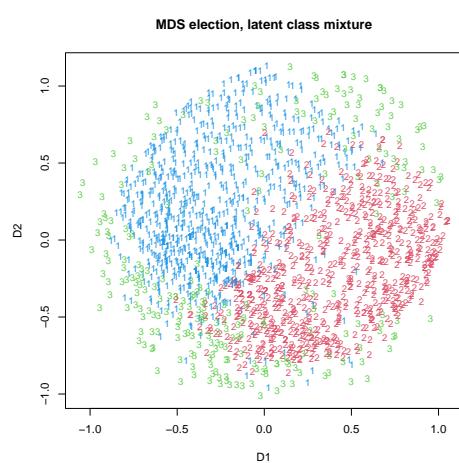
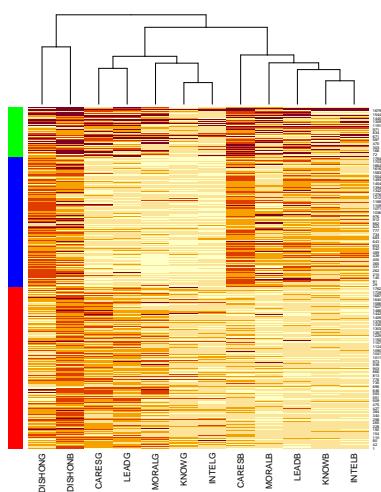


How come?

Christian Hennig

The role of data visualisation in cluster analysis

**Introduction**  
 An example with Euclidean data  
 An example with categorical data

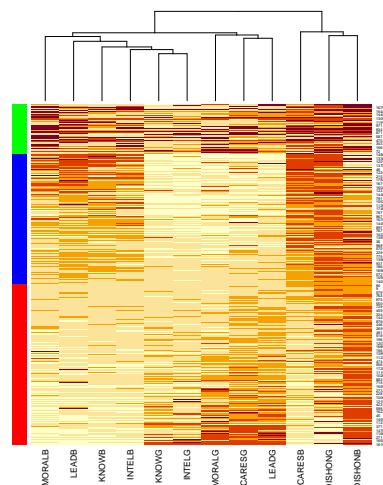


Cluster 3 mops up all kinds of observations  
that don't fit into 1 (pro Bush) and 2 (pro Gore),  
but is quite heterogeneous (bad Silhouette).

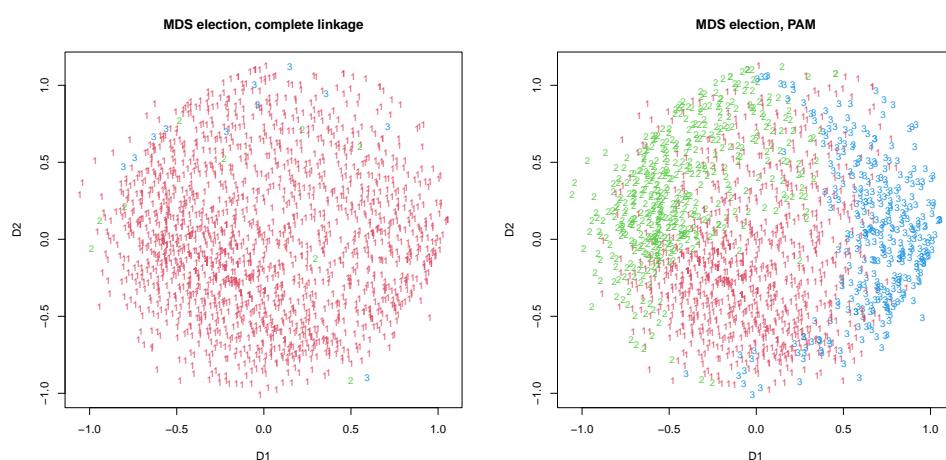
Christian Hennig

The role of data visualisation in cluster analysis

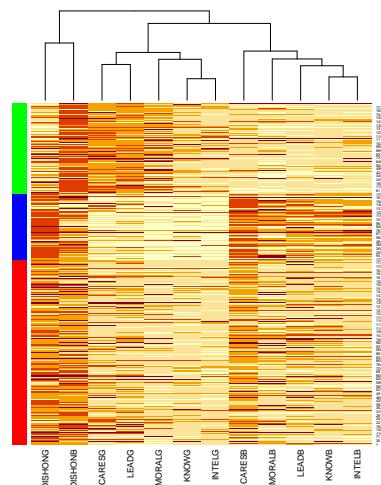
Ordering observations in cluster 1 and 2 according to similarity to other cluster, it turns out they are not that strongly separated.



Some other clustering methods:



Complete doesn't look very useful.  
PAM may have all three clusters rather homogeneous.



Clusters 1 and 2 are not so strikingly different.  
For interpretation/use, is this better,  
or latent class clustering with heterogeneous cluster 3?

## Conclusion

All kinds of considerations and examples showing that algorithms and indexes shouldn't be left on their own.

Some guidance and ideas to use visualisation to assess clusterings.

All this of course requires "subjectivity"  
(actually knowledge and insight,  
but subjectivity won't go away; Gelman & Hennig 2017)

This requires extra validation effort!

**Bibliography:**

- Gelman, A., Hennig, C. (2017) Beyond Subjective and Objective in Statistics, Journal of the Royal Statistical Society Series A 180, 967-1033
- Hennig, C. (2004) Asymmetric linear dimension reduction for classification. Journal of Computational and Graphical Statistics 13, 930-945
- Hennig, C. and Lin, C. (2015) Flexible parametric bootstrap for testing homogeneity against clustering and assessing the number of clusters. Statistics and Computing 25, 821-833
- Hubert, M, Rousseeuw, PJ, Vanden Branden, K (2005) ROBPCA: a new approach to robust principal components analysis, Technometrics, 47, 64-79.
- Scrucca L., Fraley C., Murphy T. B. and Raftery A. E. (2023) Model-Based Clustering, Classification, and Density Estimation. Using mclust in R. Chapman & Hall/CRC
- Ullmann, T., Hennig, C., Boulesteix, A.-L. (2021) Validation of cluster analysis results on validation data: A systematic framework. WIREs Data Mining and Knowledge Discovery, 12(3), e1444.
- Van Mechelen, I., Hennig, C., & Kiers, H. A. L. (2024) Onset of a conceptual outline map to get a hold on the jungle of cluster analysis. WIREs Data Mining and Knowledge Discovery, e1547.

# Validation of cluster analysis results on validation data: A systematic framework

Theresa Ullmann<sup>1</sup>  | Christian Hennig<sup>2</sup>  | Anne-Laure Boulesteix<sup>1</sup> 

<sup>1</sup>Institute for Medical Information Processing, Biometry and Epidemiology, Ludwig-Maximilians-Universität München, Munich, Germany

<sup>2</sup>Dipartimento di Scienze Statistiche "Paolo Fortunati", Università di Bologna, Bologna, Italy

**Correspondence**

Theresa Ullmann, Institute for Medical Information Processing, Biometry and Epidemiology, Ludwig-Maximilians-Universität München, Marchioninistraße 15, 81377, Munich, Germany.  
Email: tullmann@ibe.med.uni-muenchen.de

**Funding information**

Bundesministerium für Bildung und Forschung, Grant/Award Number: 01IS18036A; Deutsche Forschungsgemeinschaft, Grant/Award Number: BO3139/7-1

**Edited by:** Witold Pedrycz, Editor-in-Chief

**Abstract**

Cluster analysis refers to a wide range of data analytic techniques for class discovery and is popular in many application fields. To assess the quality of a clustering result, different cluster validation procedures have been proposed in the literature. While there is extensive work on classical validation techniques, such as internal and external validation, less attention has been given to validating and replicating a clustering result using a validation dataset. Such a dataset may be part of the original dataset, which is separated before analysis begins, or it could be an independently collected dataset. We present a systematic, structured review of the existing literature about this topic. For this purpose, we outline a formal framework that covers most existing approaches for validating clustering results on validation data. In particular, we review classical validation techniques such as internal and external validation, stability analysis, and visual validation, and show how they can be interpreted in terms of our framework. We define and formalize different types of validation of clustering results on a validation dataset, and give examples of how clustering studies from the applied literature that used a validation dataset can be seen as instances of our framework.

This article is categorized under:

Technologies > Structure Discovery and Clustering  
Algorithmic Development > Statistics  
Technologies > Machine Learning

**KEY WORDS**

cluster stability, cluster validation, clustering, independent data, replication

## 1 | INTRODUCTION

Cluster analysis refers to data analytic techniques for structure and class discovery. It is popular in a range of fields, for example, medicine, biology, market research, social science, and data compression. However, when conducting cluster analysis, researchers are confronted with an overwhelming number of existing methods. They must preprocess the data, choose a clustering algorithm, and set parameters, such as the number of clusters (Van Mechelen et al., 2018;

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2021 The Authors. *WIREs Data Mining and Knowledge Discovery* published by Wiley Periodicals LLC.

Zimmermann, 2020). It is often unclear a priori which choice should be made for the analysis, and even once a choice is made, it may remain unclear how good the quality of the resulting clustering is.

These problems have prompted the development of so-called *cluster validation* techniques, see Handl et al. (2005) and Hennig (2015a) for overviews. The literature distinguishes between internal validation (where the clustering is evaluated based on internal properties, such as compactness and separateness of the clusters) and external validation (where the clustering is evaluated by comparing the clusters with respect to one or more variables not used for clustering, e.g., a survival time or a true class membership). Less attention has been given to the validation and replication of clustering results on a *validation dataset*, for which we introduce a structured framework that summarizes the existing literature in a systematic manner. A validation dataset could be part of the original dataset, set apart before the start of the analysis, or it could be a separate dataset, obtained, for example, from a different study centre.

The idea of validating a clustering on another dataset is not new and has appeared in the methodological literature decades ago (Breckenridge, 1989; McIntyre & Blashfield, 1980). In applied literature involving cluster analysis, it is not uncommon for authors to validate their clustering results on new data, be it with the procedure of McIntyre and Blashfield (1980) or another method. To the best of our knowledge, these approaches have never been systematically structured and evaluated, and different validation strategies are scattered across different works and application fields. This contrasts with the abundant methodological literature devoted to validation in the context of *supervised classification* (or more generally, *supervised learning*). This contrast may be partly due to the fact that cluster analysis—as opposed to supervised classification—is often viewed as exploratory research. The validation of clustering results is rightly considered to be less straightforward than the validation of a prediction model because “true labels” are unknown (Von Luxburg et al., 2012). Indeed, it is difficult to define exactly what is meant by validating a clustering on validation data. Answering this question is the key aspect of our framework.

In this article, we aim to give a systematic review of the various strategies used in the literature for validating clustering results on validation data. These existing approaches are combined into a structured framework. In this framework, we define and formalize the concept of validation on a validation dataset. In particular, we demonstrate that many classical validation techniques, such as internal and external validation, stability analysis, and visual validation, can be linked to evaluation on validation data: using validation datasets does not replace these approaches; rather, classical validation can be combined with validation data. Moreover, we show how clustering studies from the applied literature that used a validation dataset can be classified into our framework.

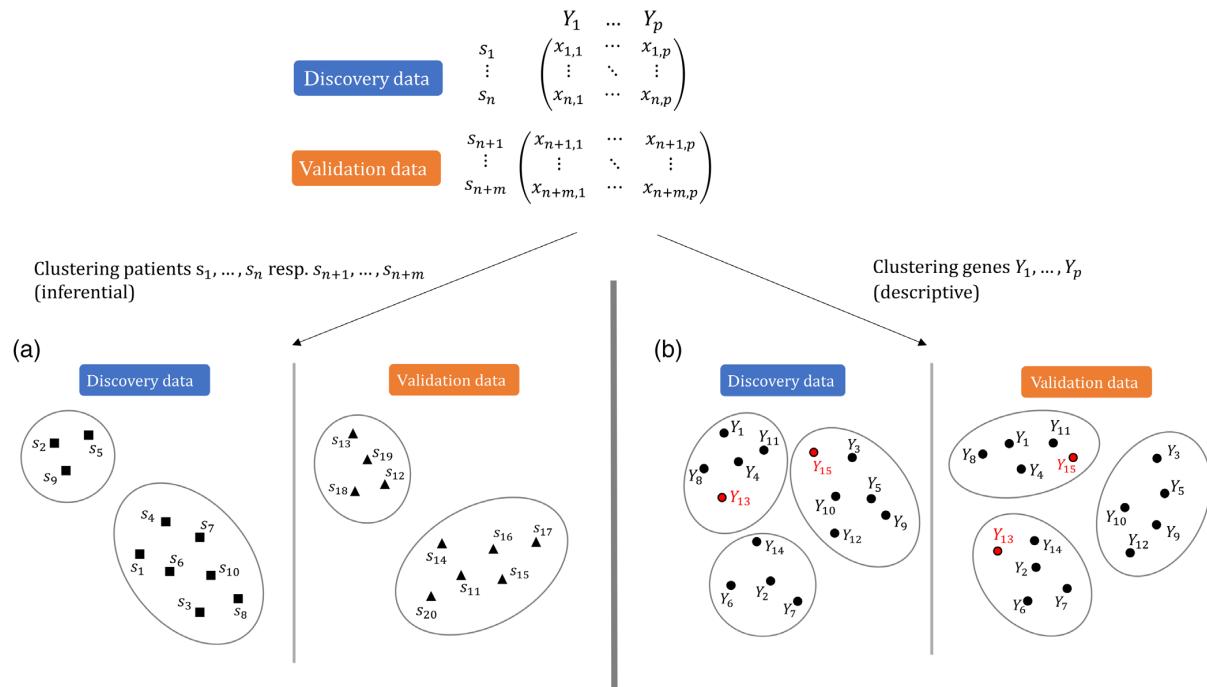
Why do researchers consider validation and replication of clustering results on a validation dataset to be important? The answer is closely tied to the clustering aim, which could either be *inferential* or *descriptive*. We define these terms as follows:

- *Inferential clustering*: The objects being clustered form a sample drawn from an underlying population for which inference is of interest, rather than making statements about the specific objects in the original dataset.
- *Descriptive clustering*: The data form a fixed set of entities of specific interest, and statements such as objects 1, 5, and 99 form a cluster are of interest.

As an example of the difference between inferential and descriptive clustering, consider an  $n \times p$  dataset including the expression levels (continuous values) of  $p$  genes for  $n$  patients suffering from a particular disease, see Figure 1.

On the one hand, it may be of interest to perform clustering analyses of the patients to see if there are subpopulations of patients with systematically different gene expressions. This would be *inferential clustering*. For example, researchers have frequently used gene expression data to detect distinct breast cancer subtypes (Burstein et al., 2015; Curtis et al., 2012; Kapp et al., 2006; Lehmann et al., 2011; Sørlie et al., 2003; Sotiriou et al., 2003). Such subtypes can have clinical implications and may guide targeted treatment (Garrido-Castro et al., 2019; Prat et al., 2015). On the other hand, an  $n \times p$  gene expression dataset could also be used to perform clustering of the (fixed set of)  $p$  genes to see if there are groups of specific genes that behave similarly, which might suggest a similar function or involvement in a common molecular process. This is an example of *descriptive clustering*. For example, researchers have used cluster analysis to find different groups of cancer-related genes (Freudenberg et al., 2009; Yang et al., 2014; Zhang et al., 2014).

For both clustering aims, using validation data is of crucial importance. To illustrate this, we again use the gene expression example, see Figure 1. First, we consider inferential clustering of breast cancer patients. All the papers for breast cancer subtype detection cited above used validation datasets to confirm the results of their own analyses and/or to validate previously reported subtypes. Indeed, a clustering of cancer patients would not be of much use if it only held on a single dataset. Due to the inferential nature of the clustering, the researchers’ aim is to better understand the



**FIGURE 1** Schematic representation of clustering on a gene expression dataset. (a) Inferential clustering of the patients and (b) descriptive clustering of the genes. For illustration purposes, there are 10 patients \$s\_1, \dots, s\_{10}\$ in the discovery data, 10 patients \$s\_{11}, \dots, s\_{20}\$ in the validation data, and 15 genes \$Y\_1, \dots, Y\_{15}\$. For inferential clustering, the objects to cluster (here, patients) are different between discovery and validation data, as indicated by using different symbols (squares vs. triangles). The two resulting clusterings nevertheless look somewhat similar: a smaller cluster on the top left, and a larger cluster on the bottom right. For descriptive clustering, the objects to cluster (here, genes \$Y\_1, \dots, Y\_{15}\$, marked by circles) remain the same across both datasets. However, their positions are slightly shifted in the validation data, because the gene expression values now stem from patients \$s\_{11}, \dots, s\_{20}\$. Consequently, genes \$Y\_{13}\$ and \$Y\_{15}\$ (marked in red) are clustered differently on the validation data

disease and to find options for treatment with respect to the underlying population, for example, the population of *all* breast cancer patients. In particular, the clustering should not only hold for patients from a single hospital or a single country. To make sure that the clustering is not just an artifact of a single dataset, researchers thus use independently collected samples for validation, or at least split their dataset into discovery and validation sets.

Now consider the example of the descriptive clustering of cancer-related genes. While the set of genes is fixed, researchers typically want the gene clustering to hold more generally than only for the  $n$  specific patients. The genes' functions or involvement in molecular processes should reflect biological principles that hold for all patients with the particular cancer type, and researchers thus want to recover the clustering on datasets with other patients having the same disease. Again, validation datasets are used for this purpose. In this sense, descriptive clustering can have an inferential component, with the difference to “inferential clustering” (as defined above) being that the objects to be clustered are fixed and do not represent samples drawn from an underlying population. This has implications for choosing a suitable validation dataset and validation strategy, as will be discussed in more detail in Section 3 below.

Similar arguments about the importance of replicability and generalisability of clusterings results (as given for the example of gene expression data above) hold more generally for most cluster analysis applications, which can typically be classified as either inferential or descriptive clustering. For example, in market segmentation (inferential clustering of customers), the resulting clusters should be replicable such that managers can consistently market their products to the customer groups (Dolnicar & Leisch, 2010; Müller & Hamm, 2014). In text and keyword analysis, where words are clustered to reveal overarching topics (descriptive clustering), it is interesting to see whether topics stay stable on validation data, or whether some changes appear (Ding et al., 2001). Across different application fields, researchers usually want their results to be as generalizable as possible. Interesting properties of a clustering result should hold not only for a single specific dataset, but should also reappear when clustering validation data sampled from the same, or even

different distributions. Validating clusterings on validation data also enables researchers to evaluate results reported by other research teams. The confirmation of results on validation data is a vital part of research in general, and it has received considerable attention in recent years due to the so-called “replication crisis” (Hutson, 2018). In the context of classical hypothesis tests and effect estimates, many published results have turned out to be non-replicable, that is, they could not be confirmed on independent data [e.g., in psychology (Open Science Collaboration, 2015), cancer research (Begley & Ellis, 2012), or economics (Camerer et al., 2016)]. Replication is thus vital for assessing the credibility of scientific claims (Nosek & Errington, 2020). For cluster analysis, our article appears to be the first one to systematically review and discuss this topic.

Our framework, which is described in detail in Section 3, is based on the following two-step cluster analysis procedure (see also Figure 2):

1. *The primary cluster analysis and method selection step:* Using the original dataset or a part of it (in the following called “discovery data”) a single clustering method is selected (where the “method” includes not only the choice of clustering algorithm, but also parameters such as the number of clusters and diverse pre/postprocessing steps), for example, via its performance with respect to internal/external validation indices.
2. *The validation step:* Important aspects of the clustering resulting from this method are validated on another dataset or the rest of the original dataset (in the following denoted by “validation data”). The validation data should be completely hidden from the method selection process of Step 1—analogously to the evaluation of supervised classifiers, where the selected model (including the chosen parameters) must be finally evaluated using validation data that was *not* used in any way for parameter tuning or model selection (Boulesteix et al., 2008; Simon et al., 2003).

The “important aspects” of the clustering that are checked in Step 2 usually depend on the research question and the field of application. Consider again the above example of clustering cancer patients, based on expression levels of cancer-related genes, for the purpose of finding subtypes of that disease. In this context, the following properties might be relevant aspects of the clustering:

- Suppose that Step 1 has resulted in two clusters. One cluster is much larger than the other, with about 80% percent of the patients in this cluster. One might be interested in whether this pattern of one large cluster and one smaller cluster can be replicated in Step 2.
- Assume it is found that the clustering chosen in Step 1 is related to survival time, that is, the patients’ survival times differ depending on which cluster they belong to. Can this finding be replicated in Step 2 for patients in the validation data?

In the literature, the term “cluster validation” is sometimes used to refer to the use of validation techniques as a tool to compare different clusterings and select the most appropriate. This use of terminology would place validation within Step 1. But when validation techniques are used as selection tool, it is still an open issue whether the results generalize to new data, and this is addressed by Step 2.

The phrase “cluster validation” also appears in the literature about *benchmarking* of clustering methods (Boulesteix & Hatz, 2017; Van Mechelen et al., 2018; Zimmermann, 2020). A benchmarking study is a systematic comparison of different clustering *methods* on a class of data distributions or datasets. Validation techniques may be used to compare different methods. Benchmark studies thus analyze the “validity” of clustering methods and provide general guidance on which method to use. In contrast, our review considers the validation of specific results of applied clustering studies.

This article is structured as follows: in Section 2, we give an overview of the different uses of the term “validation” and perspectives on validity found in the clustering literature. We then present our validation framework in detail in Section 3. In Section 4, we demonstrate in an exemplary manner how clustering studies from the applied literature can

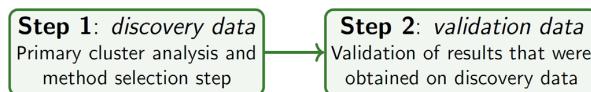


FIGURE 2 Two-step procedure for validating clustering results

be sorted into the framework. Section 5 contains a final discussion. In the Supporting Information, we present an illustration of the discussed validation strategies using openly available real-world data, where the data analysis is performed with thoroughly commented R code.

## 2 | DIFFERENT PERSPECTIVES ON “VALIDITY” IN CLUSTER ANALYSIS

We identified four approaches that address the validity of clusterings in the literature: (1) the comparison of “true” cluster labels with inferred clusters, (2) internal and external validity indices, (3) stability analyses, and (4) visual validation. These four approaches are briefly reviewed in the following subsections. An additional approach, hypothesis testing, is briefly discussed in Section 5. Internal and external validation, stability, and visual validation form the building blocks of our framework, see Section 3.

### 2.1 | Recovery of “true” clusters and analogies to the validity of supervised classification models

According to this perspective, a clustering of a dataset is “valid” if it corresponds to the “true” cluster structure in the data. Correspondingly, a clustering method is called “valid” if it can recover the “true” clusters in the data (Breckenridge, 1989; Milligan & Cooper, 1987). A related view is presented in the paper of Dougherty et al. (2007), which shows a connection to the term “validity” in the context of supervised classification. For supervised classification models, the validation of a classifier relates to estimating the *prediction error* on a test set, that is, how well the classifier can predict the known “true” labels of the instances in the test set. Dougherty et al. (2007) demonstrate that this approach can be transferred to cluster analysis. However, this requires datasets with *known* cluster labels. Yet, in practice, cluster analysis is usually applied to real datasets for which the “true” cluster labels of the data points are unknown. Note that even in the rare case of a cluster analysis performed on a dataset with given “true” cluster labels, these may not be unique, and there might be other equally legitimate cluster structures in the data, which can be even more interesting and useful as a result of the analysis than the one previously known (see Färber et al., 2010; Hennig, 2015b). When validating a clustering on validation data, the validation step used in supervised classification usually cannot be mimicked. The idea of Dougherty et al. (2007) thus mainly makes sense in the context of benchmark studies comparing clustering methods using simulated data with known “true” cluster labels. The ability of the methods to recover the true clusters may then be used as a performance criterion. To evaluate clusterings in applied studies, other options for validation are needed.

### 2.2 | Internal and external validation

In the absence of “true” cluster labels, assessing “cluster validity” often uses so-called internal indices or external information—leading to the terms “internal validation” and “external validation,” respectively.

- *Internal validation* uses only the data that was used for clustering. Typically, internal validation consists of calculating an index that is supposed to measure how well the clustering fits the data (Halkidi et al., 2015). Such indices often exploit the proximity structure of the data, for example, by measuring the homogeneity and/or the separation of the clusters. Examples are the Average Silhouette Width index (Kaufman & Rousseeuw, 2009) and the Caliński–Harabasz index (Caliński & Harabasz, 1974). These indices combine measurements of the homogeneity and the separation of a clustering into a single value, in order to balance a small within-cluster heterogeneity and a large between-clusters heterogeneity. There are also indices that measure only isolated aspects of a clustering (e.g., only the homogeneity or only the separation of the clusters), see Akhanli and Hennig (2020).
- *External validation* makes use of additional (external) information that was *not* used for clustering. For example, when clustering a cancer gene expression dataset, one may use the survival time of patients to determine whether the clustering of patients based on gene expression can predict survival. The term “external validation” also encompasses the recovery of previously known “true labels” as presented in Section 2.1.

## 2.3 | Stability

Many authors consider *stability* to be a crucial aspect of cluster validity. The idea is that a good clustering method should yield similar partitions when applied to multiple datasets drawn from the same data distribution (Ben-David et al., 2006; Von Luxburg, 2010). In this spirit, a specific clustering of a single real dataset may be considered as validated if the clusterings obtained from datasets generated from the same data distribution are similar. There are several methods of generating multiple datasets to emulate the data distribution of the dataset to be analyzed, for example, by drawing subsamples from the original dataset (Hennig, 2007).

Stability analysis dates back to McIntyre and Blashfield (1980), Morey et al. (1983), and Breckenridge (1989). These authors considered the replicability of a clustering result on a validation dataset. To generate the validation dataset, the original data is split into two halves (by splitting along the objects to be clustered for inferential clustering, or by splitting across the variables of the dataset for descriptive clustering). This is followed by assessing whether the clustering obtained in the first half can be replicated in the second half. For descriptive clustering, because the objects in the two halves are the same, replicability can be assessed directly with a partition similarity index such as the Adjusted Rand Index (ARI; Hubert & Arabie, 1985; Rand, 1971), the Jaccard index (Jaccard, 1908), or the FM index (Fowlkes & Mallows, 1983). See Meila (2015) and Albatineh et al. (2006) for overviews of partition similarity indices. For inferential clustering, the objects to cluster are not the same in the two data halves, and thus the objects from the second half have to be classified into the clusters of the first half, before the clusterings can be compared with a partition similarity index (see Section 3.3 for details). Such stability analyses will indeed be a special case of the broader validation framework presented in Section 3.

In the decades that followed, however, the focus of stability analysis shifted away from this concept and more towards *method or model selection*. Like other validation techniques, stability analyses are used in Step 1 (see Figure 2) as a basis for the selection of a suitable clustering method and its parameters, such as the number of clusters (Ben-Hur et al., 2002; Bertrand & Mufti, 2006; Dolnicar & Leisch, 2010; Dudoit & Fridlyand, 2002; Fang & Wang, 2012; Fu & Perry, 2020; Lange et al., 2004; Levine & Domany, 2001; Monti et al., 2003; Tibshirani & Walther, 2005; Wang, 2010). In these approaches, stability analysis selects the clustering method that is most stable over multiple subsamples. The subsamples are drawn without replacement or in a cross-validation manner, or are bootstrap samples drawn with replacement from the data. For example, different numbers of clusters  $k$  can be considered in turn, and the  $k$  that leads to the most stable clustering, or the smallest  $k$  that exceeds a stability threshold, can be chosen. These studies typically consider inferential clustering, such that the term “subsamples” refers to subsets of objects to be clustered. Some schemes require the comparison of clusterings on subsets of objects that consist of disjunct subsamples of the original dataset and thus have no overlap (e.g., Dudoit & Fridlyand, 2002; Fang & Wang, 2012; Lange et al., 2004; Tibshirani & Walther, 2005; Wang, 2010). This requires the aforementioned supervised classification step for classifying observations of one sample to the clusters of the other sample. However, the approaches could in principle be modified to also apply to descriptive clustering.

When splitting the dataset multiple times to determine the stability of a clustering method or parameter, eventually information from the whole dataset enters the method selection process. Thus putting aside a validation dataset that is only used *after* the method selection is advised. Even if a clustering is chosen by stability analysis on a discovery dataset, it is *not* guaranteed that this clustering can be validated on a validation dataset.

Stability analysis can also be combined with classical internal validation indices by checking whether internal validation indices have similar values for multiple clusterings calculated on subsamples of the data (Jain & Moreau, 1987), see also Dangl and Leisch (2020) for a related approach. This idea will also be part of our framework in Section 3.

## 2.4 | Visual validation

Cluster analysis is often exploratory without fixed predefined expectations from the user. Patterns in the data that qualify to be interpreted as clusters can have very diverse appearances. Some key characteristics of clusters, such as being areas of high density separated by areas of lower density, are difficult to translate into easily computable statistics. Furthermore, many clustering methods rely on model assumptions and cluster concepts, the appropriateness of which is hard to diagnose by means other than visual. This explains why visual validation is important in cluster analysis.

Clusters can be declared valid based on visualization if they correspond to clearly visible patterns in the data, or in some cases if the assumptions required for the chosen clustering method look valid.

Useful plots for visual cluster validation can be distinguished into:

1. General purpose data plots in which found clusters can be indicated by colors or glyphs, such as scatterplots, matrix plots, principal components biplots, multidimensional scaling, or parallel coordinates plots (Cook & Swayne, 2007, chapter 5). There are also projection pursuit approaches that generate “interesting” data projections, potentially showing clustering structure, without requiring the clustering as input (e.g., Tyler et al., 2009).
2. Plots set up to visualize a specific clustering, which can be further classified as:
  - a. Plots that visualize the original data directly, such as cluster heatmaps (Hahsler & Hornik, 2011; Wilkinson & Friendly, 2009) or projections to optimally discriminate clusters (Hennig, 2004).
  - b. Plots that visualize the clustering solution without representing the original observations directly such as dendograms, silhouette plots, and neighborhood graphs (Leisch, 2008).

We refer to the Supporting Information for an illustration of some of these methods.

Plots that visualize the original data directly can be used to assess patterns in data space, although these plots come with either information loss by dimension reduction, or heavy reliance on aspects such as variable and observation ordering. The advantage of plots that optimize objective functions dependent on the clustering, such as discriminant projections or heatmaps with orderings determined by the clustering, is that they have better chances to bring out the data patterns corresponding to the clustering than general purpose plots. On the other hand, they may lead to an overoptimistic assessment of the validity of the clustering, or an interpretation of spurious patterns. Validation data that is kept separate from the beginning of the analysis may help to avoid overoptimism, see Section 3.4.

Some of the plots that do not represent the original observations directly can also be valuable for cluster validation. The silhouette plot accompanies the Average Silhouette Width index (Kaufman & Rousseeuw, 2009) and gives observation-wise information about the quality of assignment in the given clustering; dendograms visualize the hierarchical merging process and can sometimes reveal issues, such as potentially meaningful clusters disappearing at higher levels of the hierarchy.

### 3 | A SYSTEMATIC FRAMEWORK FOR VALIDATING A CLUSTERING ON A VALIDATION DATASET

In this section, we present a systematic framework for validating a clustering on a validation dataset that includes many existing approaches from the literature as special cases and revisits them more formally. We also show how the validation methods that we reviewed in the last section are incorporated into the framework.

We first discuss what is meant by a “validation dataset” in Section 3.1. In Section 3.2, we give an overview of properties of a clustering result that may be validated on the validation set (these properties are strongly related to the classical validation procedures discussed in Section 2). In Section 3.3, we outline the distinction between method-based and result-based validation on a validation dataset. In Section 3.4, we combine the concepts of Sections 3.2 and 3.3 into an overview of strategies for validation on validation data. In Section 3.5, we discuss how to judge whether “successful” validation has been achieved.

#### 3.1 | Validation datasets

The term “validation dataset” can refer to a dataset composed of independently collected data (e.g., collected by other researchers or in a different laboratory) which is similar enough to the original data for cluster evaluation to be possible. In practice, however, genuinely independent data is often not available. In this case, one might split a single dataset into a discovery and a validation set.

Apart from this consideration, the structure of the validation data depends on two further aspects: (a) The data for clustering can either be object by variable data or object by object proximity data (where the term “proximity” denotes

either similarities or dissimilarities), see Van Mechelen et al. (2018). Here, “objects” denote the entities which are to be clustered. (b) The aim of the clustering could either be inferential or descriptive, as defined in the introduction (Section 1).

For inferential clustering, the validation data consists of more objects to cluster. On the other hand, for descriptive clustering, validation data does *not* consist of more objects because the set of objects to be clustered is fixed. Consider the example of the  $n \times p$  gene expression dataset as described in the introduction. This dataset can be understood as an object by variable dataset in two ways. For inferential clustering of the patients, the patients are the “objects” and the genes the “variables.” A validation dataset consists of more patients. For descriptive clustering of the genes, now the genes constitute the “objects,” and the patients are the “variables.” A validation dataset consists of more variables, that is, again of more patients.

In Table 1, we give a general overview of the structure of the validation data, where we distinguish between inferential and descriptive clustering as well as between object by variable and object by object data.

If separately collected data is not available, and the dataset must be split into discovery and validation sets, a 50/50 split ratio is usually chosen. Indeed, we believe that this choice makes sense in most cases: validation strategies often require the number of data points in the validation set to not be too small when trying to validate certain properties obtained from the clustering on the discovery set. A similar argument has been made in the context of stability analysis (Lange et al., 2004).

### 3.2 | Clustering properties to be validated

In the literature, we identified four categories of properties of clusterings that researchers may want to validate.

(**Int**) Internal properties of the clusters (that turn up when clustering the discovery data), for example:

- descriptive measures of the clusters such as the values of the cluster centroids or the relative sizes of the clusters,
- the value of an internal validation index calculated for the clustering result, and
- subsets of variables that characterize the clusters.

(**Ext**) Associations of the clusters with external variables or agreement of the clustering with an externally known partition. Some examples:

- Clusters of cancer patients have different mean survival rates.
- A clustering of genes shows some agreement with known functional gene labels. For example, a clustering may be compatible with known partitions of the genes into functional categories. Less restrictively, some particular genes, of which this was previously expected, may be in the same cluster.

(**Vis**) Characteristics that can be assessed using visualization: do the clusters correspond to distinctive meaningful patterns in the data? Do the clusters look how they were supposed to look like? This could refer to model assumptions for the clustering method, or *a priori* hypotheses or requirements by the researcher.

TABLE 1 Structure of the validation data depending on inferential versus descriptive clustering and object by variable versus object by object data

	Inferential clustering	Descriptive clustering
<b>Object by variable data</b>	Validation data: further objects, same variables	Validation data: further variables, same objects
If a single dataset is split	Split performed along the objects	Split performed along the variables
<b>Object by object data</b>	Validation data: proximity matrix of further objects	Validation data: proximity matrix of same objects, but with proximities derived from another source (e.g., based on different underlying variables).
If a single dataset is split	Objects can be split into two disjoint sets, yielding two smaller proximity matrices (one representing the discovery data, the other the validation data).	Impossible to split proximity data directly into discovery and validation data, but may be possible to split underlying variables.

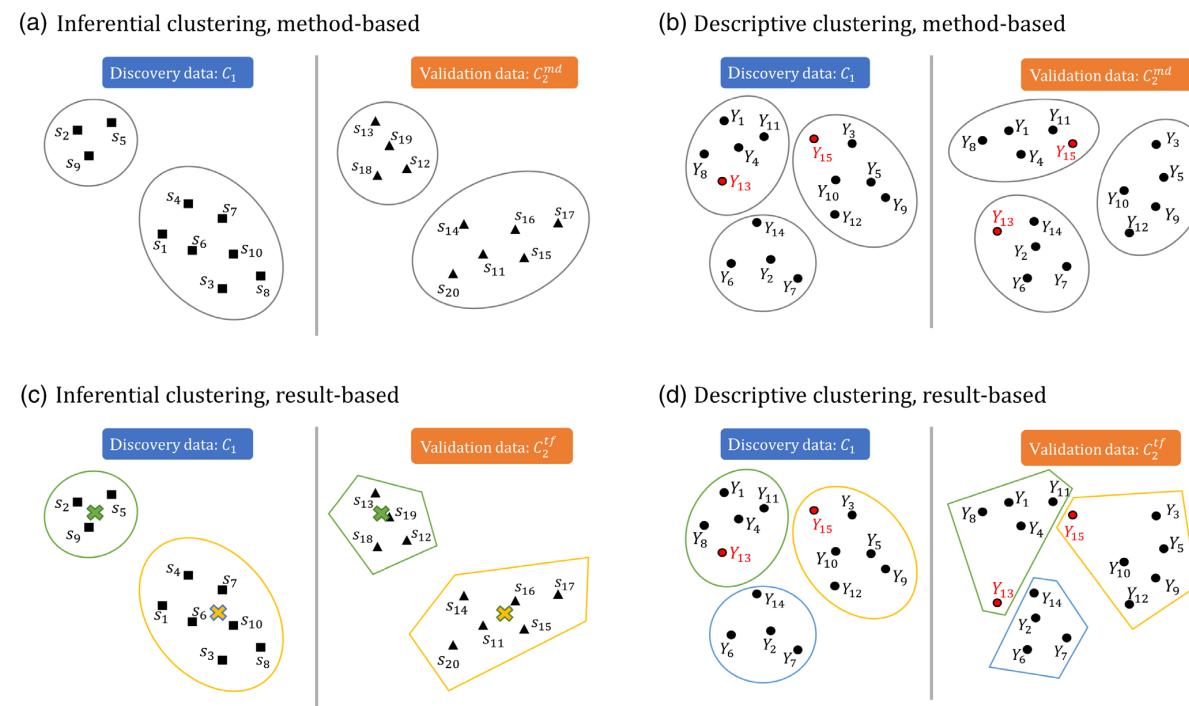
**(Stab)** Stability of cluster membership: Does cluster membership remain stable when the same method (algorithm, number of clusters, etc.) is applied to the validation data? Since the objects in the discovery and the validation set are disjunct in the case of inferential clustering, this involves supervised classification of objects of one dataset to clusters of the other dataset.

Most subsections of Section 2 correspond to a category in the above list, with the exception of Section 2.1 (recovery of “true” clusters). If the “true” cluster labels are indeed known, this can be considered as a part of (Ext).

### 3.3 | Method-based and result-based validation

The validation of a clustering on a specific dataset can refer either to the validity of the used clustering *method*, or the validity of the clustering *result* itself. While this distinction is often not made clear in the literature on classical validation procedures, it has important implications for how validation on a validation dataset is performed. We thus distinguish between *method-based* and *result-based* validation on validation data, as illustrated in Figure 3. In the following, we explain these terms in more detail.

We denote the discovery data by  $D_1$  and the validation data by  $D_2$ . The clustering chosen on  $D_1$  in Step 1 (method selection, see Figure 2) is called  $C_1$ . Given  $C_1$ , the validation dataset can be handled in two different ways:



**FIGURE 3** Method- and result-based validation for inferential and descriptive clustering. We use the same data example as in Figure 1. The top panel (a) and (b) (method-based validation) is from Figure 1. For inferential clustering (a), re-applying the clustering method to the validation data again detects a smaller cluster on the top left and a larger one on the bottom right. For descriptive clustering (b), the clustering  $C_2^{md}$  on the validation data groups the elements  $Y_{13}$  and  $Y_{15}$  (marked in red) differently than the clustering  $C_1$  on the discovery data. The bottom panel (c) and (d) (result-based validation) illustrates the classification procedures that yield  $C_2^{rf}$ . The clusterings  $C_2^{rf}$  are depicted as polygons. For inferential clustering, nearest-centroid classification is depicted: the green and yellow crosses represent the centroids of  $C_1$ . The samples in the validation data are then assigned to the nearest centroid. In this particular example, the resulting clustering  $C_2^{rf}$  in (c) is equal to  $C_2^{md}$  in (a); in our terminology, the criterion (Stab) is perfectly fulfilled. For descriptive clustering (d), the most obvious way of transferring  $C_1$  to the validation data is to set the cluster memberships in  $C_2^{rf}$  equal to those of  $C_1$ . In particular, the elements  $Y_{13}$  and  $Y_{15}$  are clustered as in  $C_1$ . Comparing  $C_2^{rf}$  with  $C_2^{md}$  in (b) shows that the cluster memberships are not perfectly stable according to criterion (Stab).

- a. The same clustering method that yielded  $C_1$  (i.e., same algorithm, same number of cluster  $k$ , etc.) can be applied to  $D_2$ , yielding a clustering  $C_2^{md}$  on  $D_2$  (“ $md$ ” for “method”).  $C_1$  and  $C_2^{md}$  can then be compared with respect to aspects (Int), (Ext), or (Vis). We call this approach *method-based validation*. It puts a focus on the structural similarity of the clustering results as generated by the method.
- b. Instead of applying the clustering method again,  $C_1$  can be “transferred” to the validation data by using a supervised classifier to predict the cluster labels of the validation set (explained in more detail below). This results in a clustering  $C_2^{tf}$  on  $D_2$  (“ $tf$ ” for “transferred”). The transferred clustering can be compared to the original clustering  $C_1$  with respect to aspects (Int), (Ext), or (Vis). We call this approach *result-based validation*. It puts a focus on whether the specific clustering result is also sensible for the validation data.

We now explain what we mean by “transferring” the clustering. For descriptive clustering,  $C_2^{tf}$  is simply  $C_1$  (recall that for descriptive clustering, the objects to be clustered are the same for  $D_1$  and  $D_2$ , and thus  $C_1$  can immediately be considered to be a clustering of  $D_2$ ). For inferential clustering, the objects to be clustered are different in the discovery and validation sets, so some proper “transfer” is required. This can be done using a supervised classifier (using the labeled discovery set  $(D_1, C_1)$  as “training set”) to assign the objects in  $D_2$  to the clusters in  $C_1$  (Akhanli & Hennig, 2020; Lange et al., 2004). For example, one can calculate the centroids of the clusters in  $C_1$ , and then assign each sample in  $D_2$  to its nearest centroid (“nearest-centroid classifier”). As  $C_2^{tf}$  is supposed to be an “extension” or “transfer” of the original clustering to the validation data, one should use a classifier that fits the assignment rule of the chosen clustering algorithm as closely as possible. The nearest-centroid classifier is suitable for  $k$ -means, which indeed clusters points by assigning them to the nearest centroid (Lloyd, 1982). For suitable classifiers for other clustering algorithms see Akhanli and Hennig (2020).

For (Stab) (stability of cluster membership), the clustering method needs to be applied again to  $D_2$ . We check whether the cluster memberships resulting from applying the method to the validation data are similar to the cluster memberships resulting from transferring the original clustering to the validation data. This combines (a) and (b).

### 3.4 | Overview of validation strategies

Table 2 combines the concepts of Sections 3.2 and 3.3 into an overview of strategies for validation on validation data. The precise choice of indices, plots, and so forth depends on the specific context of the analysis. We refer to Section 4 for illustrative examples from the applied literature.

Here are some considerations regarding the different strategies. The commented R code in the Supporting Information illustrates the following paragraphs with real-world datasets and concrete choices for indices and visualization tools.

#### 3.4.1 | Validating (Int): Internal properties of the clustering

When applying result-based validation, the clusters of  $C_2^{tf}$  correspond to those of  $C_1$ . This makes the comparison easier. For method-based validation, the clusters of  $C_2^{md}$  are not automatically associated one-to-one with the clusters of  $C_1$ . Such an association is not needed when calculating internal indices that refer to a whole clustering, and comparing the index values between the clusterings on discovery and validation data. However, one may also be interested in comparing characteristics of specific clusters such as cluster centroids. In this case, there needs to be a matching of the clusters of  $C_2^{md}$  to the clusters of  $C_1$ , usually assuming that their number is the same. There are various methods to do this. For

TABLE 2 Strategies for validation on validation data

	<b>Method-based validation</b>	<b>Result-based validation</b>
(Int)	Compare $C_1, C_2^{md}$ with respect to: Internal properties	Compare $C_1, C_2^{tf}$ with respect to: Internal properties
(Ext)	External associations	External associations
(Vis)	Visual properties	Visual properties
(Stab)	Compare $C_2^{md}, C_2^{tf}$ with respect to cluster membership	

example, in centroid-based clustering one could match the centroids so that the sum of distances between centroids of matched clusters is minimal (Mirkin, 2005). Breckenridge (2000) suggests associating each cluster of  $C_2^{md}$  to a cluster of  $C_2^f$  (e.g., by choosing the cluster association that maximizes the sum of the intersections of the clusters). The one-to-one cluster association of  $C_2^f$  to  $C_1$  can then be used to assign each cluster of  $C_2^{md}$  to one of  $C_1$ .

### 3.4.2 | Validating (Ext): Associations with external variables or agreement with externally known partitions

As for method-based validation of internal properties (Int), here too it may be necessary to match the clusters of  $C_2^{md}$  to those in  $C_1$  and the remarks made above apply again. Note that this is not necessarily required. For example, testing whether the clusters are associated with an external variable, such as survival time, without interpreting the association of specific clusters, does not require matching.

For result-based validation of descriptive clustering, the partition  $C_2^f$  is actually equal to  $C_1$ . This makes certain approaches such as testing an association between cluster membership and an external variable on both discovery and validation data meaningless.

### 3.4.3 | Validating (Vis): Visual patterns

Using the same variables for  $D_1$  and  $D_2$  as in inferential clustering, some plots such as scatterplots or parallel coordinates plots can visualize both  $C_2^{md}$  and  $C_2^f$  in a straightforward manner comparable to  $C_1$ . Some other plots such as principal components biplots, other linear projection plots such as those in Hennig (2004), and multidimensional scaling require a selection of an optimal projection space for the dataset to be plotted. Although this could be done on the validation data, for inferential clustering, plotting the validation dataset on the projection space defined by the discovery dataset (and its clustering, if the projection space depends on it) allows for a more direct comparison. For linear projection methods, this requires a standard linear projection given the coordinate axes determined from  $D_1$ . For multidimensional scaling, there are techniques to embed new observations into the projection space defined by the original observations, for example, Gower (1968). For descriptive clustering, on the other hand, embedding the observations of  $D_2$  in the space defined by  $D_1$  is not informative as the points would be identical, so here an optimized projection space for  $D_2$  must be found.

Some other plots, such as the silhouette plot and cluster heatmaps (as long as observations are ordered only by a partition rather than a full dendrogram), may benefit from matching clusters for determining their order, see the comments on internal validation (Int) in Section 3.4.1.

The results of visual validation are subjective, and although plots are reproducible given both discovery and validation datasets, the way the researcher arrives at a validity verdict will not be reproducible. Displaying the involved plots will give the reader the chance to form their own conclusions.

### 3.4.4 | Validating (Stab): Stability of cluster membership

Here one needs to compute both  $C_2^f$  and  $C_2^{md}$ . These are then compared with an index for comparing partitions. The rationale behind this is as follows: cluster memberships in  $C_1$  and  $C_2^{md}$  are compared to check whether repeated application of the clustering method leads to stable cluster memberships. For descriptive clustering,  $C_1$  can be compared to  $C_2^{md}$  directly (here  $C_1$  is equal to  $C_2^f$ ). For inferential clustering,  $C_1$  and  $C_2^{md}$  cannot be compared directly because they are partitions of different sets of objects. Thus  $C_2^f$  is used as a surrogate for  $C_1$  on  $D_2$ . Different choices of a partition similarity index are possible, for example, the ARI, the Jaccard index, or the FM index (for overviews, see Meila, 2015; Albatineh et al., 2006).

## 3.5 | When is a clustering successfully validated?

Due to random variation, researchers will hardly ever achieve the exact same results on discovery and validation data. So far, there seem to be no systematic approaches for judging “validation success” in the context of validating clustering

results on validation data. In this section, we review the current status and outline which aspects would be interesting to study in further research.

The problem of defining “successful” validation does not only arise in cluster analysis, but generally in validation or replication studies. Here we consider “validation” to be the broader term, and “replication” as more specific, for which strategies of the validation framework can be used. “Replication” refers to using new data to re-assess scientific claims made in a previous publication (Nosek & Errington, 2020). The discussion about judging replication success is ongoing in the field of methodological research on replication studies, mostly in the context of hypothesis tests and effect estimates. For example, Hedges (2019) and Held (2020) argue that, when trying to replicate a hypothesis test (that was significant on the original data), it is not enough to check whether the test on the replication data is significant again. Actually, the binary distinction between significance and insignificance may not be helpful, for example, when comparing  $p$  values of 0.04 and 0.06 (given a significance level of 0.05). Rather, we should also check whether the effect estimate in the replication study provides evidence for the claim about the effect in the original study. Some clustering validation aspects are connected to significance tests, particularly testing for external associations in (Ext). The same caveats apply here regarding general replication of test results.

The consideration of differences between (internal or external) validity measurements on discovery and validation data, or the consideration of an index value for stability between discovery and validation sets, could in principle also be framed as a testing problem of a null hypothesis formalizing some kind of equality of structure. To our knowledge, this has not been performed yet and is left as a potential direction of future research. It can be expected that validation data results will not be quite as good due to selection bias originating from basing selection of the final clustering on results of the discovery data: the more different clustering algorithms or parameters are tried during the analysis on the discovery data, the more likely it is that one of them yields a satisfying result. If only the best result is chosen, this might be “overoptimistic” to some extent. In other words, the multiplicity of possible analysis strategies may hinder replicability (Hoffmann et al., 2021), see also the discussion in Section 5. Observing slightly worse values on the validation data is thus to be expected and does not necessarily mean that the validation has failed. However, if the results are severely worse, then this suggests problematic overoptimism on the discovery data.

As it stands, it must be acknowledged that the question “is validation successful?” cannot simply be answered with “yes” or “no”. The validation dataset may deliver high or low agreement regarding various aspects (internal and external validity, stability, visual aspects) with what was found on the discovery data—where the clustering on the discovery data may already have been assessed as a weaker or stronger clustering in Step 1. For example, regarding an internal index, such as the Average Silhouette Width, it is of interest both whether the value is reasonably high on the discovery dataset alone, and whether the validation dataset supports whatever value was found on the discovery data. Guidelines or thresholds for interpreting index values are rarely given and in fact mostly arbitrary, so the researcher must rely on their understanding of the index, experience, and judgment.

#### 4 | EXAMPLES FROM THE APPLIED LITERATURE

In this section, we review application studies that conducted cluster analysis on a discovery set and then validated the results with a validation set. Our aim is to demonstrate how these studies fit into the framework outlined above. Given the vast amount of applied cluster analysis studies, it is impossible to list every cluster study that used a validation set. Rather, we start by giving a short historical overview and then present some exemplary studies in Table 3.

The appearance of clustering studies that used a discovery and a validation set dates back to at least the 1960s. One of the first clustering studies that used a validation set was Goldstein and Linden (1969) who clustered patients with alcohol use disorder. In our terms, they performed method-based validation with respect to internal properties. Rogers and Linden (1973) provided an early implementation of stability-based validation, (Stab). They clustered college freshwomen based on personality features and used discriminant analysis as the classifier to derive  $C_2^f$ . (Stab) was then presented more systematically by McIntyre and Blashfield (1980) and Breckenridge (1989).

In recent decades, many more clustering studies that use validation data have appeared. In Table 3, we list exemplary applied studies for the different validation types as outlined in Table 2. The studies are taken from our main field of expertise, that is, medicine and health science. Some of these studies used multiple aspects of the validation framework, but for the sake of illustration, we only list one validation type per study. We did not find an example for result-based validation of (Vis). In general, there appear to be few studies which performed validation of visual properties on

TABLE 3 Study examples for each validation type

Validation type and study	Clustering aim and validation motivation	Validation data	Cluster algorithm	Validation procedure
(Int) result-based: Kapp and Tibshirani (2007)	Inferential clustering of breast cancer patients based on microarray gene expression to validate breast cancer subtypes that were previously found by Sørlie et al. (2003).	Validation data consisted of independently collected samples from different countries.	hierarchical clustering	$C_2^f$ was derived via a variant of nearest-centroid classification (each sample in the validation data was assigned to the original cluster whose centroid had the maximum Pearson's correlation coefficient with the sample), $\tilde{C}_2^f$ was then evaluated with a newly introduced internal validation index (the “in-group proportion” IGP). This index was combined with a statistical test procedure that consists of generating centroids randomly placed in the data, classifying the samples of the validation data to these centroids to obtain clusterings $\tilde{C}_2^f$ , and comparing the values of the internal index for the $\tilde{C}_2^f$ 's with the index value for $C_2^f$ . The IGP was not applied to $C_1$ .
(Int) method-based: De Bourdeaudhuij and Van Oost (1998)	Inferential clustering of adolescents to find clusters of health behavior (with respect to smoking, alcohol use, sleeping, food choice, BMI, and physical activity). Validation was used by the authors to replicate their own findings.	Validation data was a separately collected dataset.	hierarchical clustering	Clusters of $C_1$ and $C_2^{md}$ were matched manually. Compared means of health behavior variables (centroids) between $C_1$ and $C_2^{md}$ (e.g., the mean amount of smoking for cluster 1 was compared between $C_1$ and $C_2^{md}$ and so on). Overall, the centroids were similar between both clusterings. In particular, both the most “healthy” and the most “unhealthy” cluster could be recovered from the validation data.
(Ext) result-based: Curtis et al. (2012)	Inferential clustering of breast cancer patients based on copy number and gene expression data to discover novel breast cancer subtypes. The authors chose result-based validation because in clinical practice, doctors would typically want to assign a new patient to a subtype, and the validity of such a procedure can be analyzed by classification of the validation samples to yield $C_2^f$ and then comparing $C_2^f$ to $C_1$ .	Validation data was a second cohort from the same tumor banks.	iCluster (Shen et al., 2009)	$C_2^f$ was derived via nearest shrunken centroid classification (Tibshirani et al., 2003), which is a modification of nearest-centroid classification where the cluster centroids are shrunk towards the overall centroid. Comparison of $C_1$ and $C_2^f$ w.r.t. their associations with survival: a Cox proportional hazards model was fitted to the discovery data (respectively validation data), with the cluster memberships of $C_1$ (resp. $C_2^f$ ) as covariates. The hazard ratios of the clusters were similar between the model for the discovery data and the model for the validation data.

(Continues)

TABLE 3 (Continued)

Validation type and study	Clustering aim and validation motivation	Validation data	Cluster algorithm	Validation procedure
(Ext) method-based: Freidenberg et al. (2009)	Descriptive clustering of cancer-related genes to find biologically meaningful clusters of co-expressed genes, which may help to elucidate biological pathways and generate hypotheses about transcriptional regulatory mechanisms. Authors performed validation to check the replicability of the clustering results.	Validation data was an independently collected breast cancer dataset.	hierarchical clustering combined with the CSIMM algorithm (Liu et al., 2006)	Each gene in the clustering was assigned a CLEAN score, a newly introduced external measure for agreement with previously known functional categories. The correlation between the gene CLEAN scores obtained with $C_1$ and $C_2^{md}$ was calculated.
(Vis) method-based: Sweatt et al. (2019)	Inferential clustering of pulmonary arterial hypertension (PAH) patients based on blood proteomic profiles to find distinct PAH immune phenotypes. The underlying idea was that patient subgroups might express distinct patterns of inflammation in blood, and the detection of these groups may in turn help to develop tailored treatments in future studies. Validation was performed to assess whether the results generalize to other patients.	Validation data consisted of independently collected samples from a different country.	Consensus Clustering (Monti et al., 2003)	Clusters of $C_1$ and $C_2^{md}$ were matched manually. Compared heatmaps and PCA plots for $C_1$ and $C_2^{md}$ which were generated separately for discovery and validation data, no common projection space was used. The heatmaps and PCA plots were deemed to be similar between discovery and validation data.
(Stab): Bergström et al. (2001)	Inferential clustering of spinal pain patients based on the Multidimensional Pain Inventory, which is a battery of questionnaires where patients self-report their pain severity, pain-related interference in everyday life, etc. Previous studies had detected distinct subgroups of spinal pain patients with respect to how well the patients coped with their disease, which has implications for tailored treatment. The authors sought to find similar clusters in their data, and performed validation to assess the replicability of their own findings.	Validation data was an independently collected dataset.	k-means	$C_2^f$ was derived via nearest-centroid classification. The kappa coefficient (a partition similarity index) was used to compare $C_2^{md}$ and $C_2^f$ . The resulting index value was 0.82, which was judged as indicating very good agreement.

a validation dataset in a thorough manner. We believe future studies would benefit from considering the procedures for (Vis), outlined above.

The studies cited in Table 3 mostly treat validation and discovery data *asymmetrically* (with the exception of Freudenberg et al., 2009, and Bergström et al., 2001). This is more obvious for result-based validation: the clustering  $C_1$  is transferred to the validation data (and not the other way around). Method-based validation may appear more symmetric because the same method is applied to both discovery and validation data and the results are typically compared descriptively in a symmetric fashion. However, method-based validation can be asymmetric to the extent of which the validation data is kept apart from the method selection on the discovery data, and is only used later without model selection to validate the results on the discovery data. Asymmetry could be made more explicit by using a suitable test procedure to judge validation success (inspired by the methodological research on judging replication success, for example, Held (2020) advocates for an asymmetric approach when comparing the replication study to the original study), but as discussed in Section 3.5, such approaches do not seem to exist yet for cluster analysis.

Many studies in the literature do not strictly set apart the validation data during Step 1 (method selection). That is, these studies use the result of the validation on the validation data for method selection (e.g., Brennan et al., 2012; Jamison et al., 1988; Sinclair et al., 2005). In contrast, we have argued in the introduction and in Section 2.3 that for the purpose of validating a clustering result on validation data in the sense of our framework, method selection should be finished after Step 1.

Another validation variant is also frequently found in the literature (e.g., Ailawadi et al., 2001; Gruber et al., 2010; Homburg et al., 2008; Kaluza, 2000; Phinney et al., 2005): method selection is performed on the whole dataset, after which the data is split into two sets. The chosen cluster method is applied to the first set, and then validation on the second set (the validation data) is assessed. Successful “validation” may indicate a certain robustness or stability of the result, but in order to avoid overoptimism on the validation data, method selection should be constrained to the first part of the split dataset, and not be performed on the whole data according to our framework.

Other studies (e.g., Alexe et al., 2006) perform a procedure that appears similar to method-based validation: they split a dataset into two halves, use the first half as the discovery set, but obtain  $C_2^{md}$  by clustering discovery and validation data *together* (instead of only clustering the validation data), which again will likely yield more optimistic validation results than if  $C_2^{md}$  had been obtained based on the validation data only.

## 5 | DISCUSSION

We have presented a systematic framework for validating clusterings on a validation dataset that encompasses procedures known from the literature. This framework might help researchers to identify a suitable approach to validate their clustering results in future studies. However, the procedure cannot be performed in an “automated” manner. Rather, it requires substantial input from the researchers who must decide which validation criteria are important for them depending on the substantive context. Furthermore, specific indices and plots need to be chosen, as well as whether the amount of agreement between results on the discovery and validation datasets is assessed as sufficient. We have given hints about when some aspects may be of interest, but as every application is different, there are no clear rules. This holds for the clustering process in general: while cluster analysis is often interpreted as being able to find meaningful structure in the data “on its own”, the choice of cluster concept and method requires thorough consideration by researchers (Akhanli & Hennig, 2020; Hennig, 2015b). The same is true for our validation framework.

Performing validation on the validation data adds some computational complexity to the cluster analysis. However, the overall complexity is often less than twice the complexity that would result from only analyzing the discovery data: frequently method selection is performed on the discovery data, and this possibly time-consuming process is not applied to the validation data.

Regarding the choice of validation data, a validation dataset could be obtained by splitting the original dataset, or it could be a separately collected dataset. On one hand, if the validation dataset and the discovery dataset are obtained by splitting an originally collected dataset, it is unclear whether a successful validation allows for generalization to data from other sources. Moreover, this reduces the size of the data and can make it more difficult to find meaningful cluster structure in the data. On the other hand, if the validation data have been independently collected (potentially coming from a different distribution) and the validation fails, it can be difficult to determine whether this is due to the clustering not being meaningful, or due to systematic differences between discovery and validation data. Conversely, if validation is successful, then this is all the more encouraging, because it suggests that the clustering result may be valid in a more general context.

Notably, the validation of clustering results on a validation dataset may also allow detection of “overoptimism” due to “overfitting” effects: when researchers try different clustering algorithms or parameters during the analysis, they can use classical internal and external validation methods to choose a single clustering out of these. However, the more clustering methods tried, the more likely it is that one of them yields a satisfying result by chance. Consequently, the reported results may be less reliable than they seem, similarly to the results of multiple tests if no adjustment is performed. While this is well-understood in the context of multiple testing, this is less so in the context of clustering. Repeating the same cluster analysis on another dataset is a sensible approach to ensure that seemingly satisfactory results are not (solely) the product of such overfitting effects.

In future work, it would be interesting to study further aspects of cluster validation in relation to validation data use. *Hypothesis testing* is an approach to cluster validation that we have not embedded in our framework. For example, one can test if a clustering result is significantly “better” than clusterings generated by the same method on homogeneous datasets (for an overview, see Huang et al., 2015). This can involve internal validation indices (Dubes, 1993; Gordon, 1998; Halkidi et al., 2002; Hennig & Lin, 2015) or stability analysis (Bertrand & Mufti, 2006; Dudoit & Fridlyand, 2002; John et al., 2020; Smith & Dubes, 1980). We do not know of work where hypothesis testing for cluster validation has involved validation data, but it could be of interest to derive distributions under suitable null hypotheses for statistics that are evaluated on validation data.

In conclusion, our hope for this framework is to improve the interpretation of clustering studies that use validation data, and to stimulate the use of validation sets in cluster analysis.

## ACKNOWLEDGMENTS

We thank Anna Jacob and Alethea Charlton for making valuable language corrections. This work has been partially supported by the German Federal Ministry of Education and Research (BMBF) [grant number 01IS18036A to Anne-Laure Boulesteix (Munich Center of Machine Learning)] and the German Research Foundation [grant number BO3139/7-1 to Anne-Laure Boulesteix]. The authors of this work take full responsibility for its content.

## CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

## AUTHOR CONTRIBUTIONS

**Theresa Ullmann:** Conceptualization (equal); methodology (lead); writing – original draft (lead); writing – review and editing (equal). **Christian Hennig:** Conceptualization (supporting); methodology (supporting); supervision (supporting); writing – original draft (supporting); writing – review and editing (equal). **Anne-Laure Boulesteix:** Conceptualization (equal); funding acquisition (lead); methodology (supporting); supervision (lead); writing – original draft (supporting); writing – review and editing (equal).

## DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## ORCID

Theresa Ullmann  <https://orcid.org/0000-0003-1215-8561>

Christian Hennig  <https://orcid.org/0000-0003-1550-5637>

Anne-Laure Boulesteix  <https://orcid.org/0000-0002-2729-0947>

## RELATED WIREs ARTICLE

[Method evaluation, parameterization, and result validation in unsupervised data mining: A critical survey](#)

## REFERENCES

- Ailawadi, K. L., Neslin, S. A., & Gedenk, K. (2001). Pursuing the value-conscious consumer: Store brands versus national brand promotions. *Journal of Marketing*, 65(1), 71–89.
- Akhanli, S. E., & Hennig, C. (2020). Comparing clusterings and numbers of clusters by aggregation of calibrated clustering validity indexes. *Statistics and Computing*, 30(5), 1523–1544.
- Albatineh, A. N., Niewiadomska-Bugaj, M., & Mihalko, D. (2006). On similarity indices and correction for chance agreement. *Journal of Classification*, 23(2), 301–313.

- Alexe, G., Dalgin, G. S., Ramaswamy, R., DeLisi, C., & Bhanot, G. (2006). Data perturbation independent diagnosis and validation of breast cancer subtypes using clustering and patterns. *Cancer Informatics*, 2, 243–227.
- Begley, C. G., & Ellis, L. M. (2012). Raise standards for preclinical cancer research. *Nature*, 483(7391), 531–533.
- Ben-David, S., Von Luxburg, U., & Pál, D. (2006). A sober look at clustering stability. In *International conference on computational learning theory* (pp. 5–19). Springer.
- Ben-Hur, A., Elisseeff, A., & Guyon, I. (2002). A stability based method for discovering structure in clustered data. *Pacific Symposium on Bio-computing*, 7, 6–17.
- Bergström, G., Bodin, L., Jensen, I. B., Linton, S. J., & Nygren, A. L. (2001). Long-term, non-specific spinal pain: Reliable and valid subgroups of patients. *Behaviour Research and Therapy*, 39(1), 75–87.
- Bertrand, P., & Mufti, G. B. (2006). Loevinger's measures of rule quality for assessing cluster stability. *Computational Statistics & Data Analysis*, 50(4), 992–1015.
- Boulesteix, A.-L., & Hatz, M. (2017). Benchmarking for clustering methods based on real data: A statistical view. In F. Palumbo, A. Montanari, & M. Vichi (Eds.), *Data science—Innovative developments in data analysis and clustering* (pp. 73–82). Springer.
- Boulesteix, A.-L., Strobl, C., Augustin, T., & Daumer, M. (2008). Evaluating microarray-based classifiers: An overview. *Cancer Informatics*, 6, 77–97.
- Breckenridge, J. N. (1989). Replicating cluster analysis: Method, consistency, and validity. *Multivariate Behavioral Research*, 24(2), 147–161.
- Breckenridge, J. N. (2000). Validating cluster analysis: Consistent replication and symmetry. *Multivariate Behavioral Research*, 35(2), 261–285.
- Brennan, T., Breitenbach, M., Dieterich, W., Salisbury, E. J., & Van Voorhis, P. (2012). Women's pathways to serious and habitual crime: A person-centered analysis incorporating gender responsive factors. *Criminal Justice and Behavior*, 39(11), 1481–1508.
- Burstein, M. D., Tsimelzon, A., Poage, G. M., Covington, K. R., Contreras, A., Fuqua, S. A., Savage, M. I., Osborne, C. K., Hilsenbeck, S. G., Chang, J. C., Mills, G. B., Lau, C. C., & Brown, P. H. (2015). Comprehensive genomic analysis identifies novel subtypes and targets of triple-negative breast cancer. *Clinical Cancer Research*, 21(7), 1688–1698.
- Calinski, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3(1), 1–27.
- Camerer, C. F., Dreber, A., Forsell, E., Ho, T.-H., Huber, J., Johannesson, M., Kirchler, M., Almenberg, J., Altmejd, A., Chan, T., Heiksten, E., Holzmeister, F., Imai, T., Isaksson, S., Nave, G., Pfeiffer, T., Razen, M., & Wu, H. (2016). Evaluating replicability of laboratory experiments in economics. *Science*, 351(6280), 1433–1436.
- Cook, D., & Swayne, D. F. (2007). *Interactive and dynamic graphics for data analysis with R and GGobi*. Springer.
- Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., Speed, D., Lynch, A. G., Samarajiwa, S., & Yuan, Y. (2012). The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403), 346–352.
- Dangl, R., & Leisch, F. (2020). Effects of resampling in determining the number of clusters in a data set. *Journal of Classification*, 37, 558–583.
- De Bourdeaudhuij, I., & Van Oost, P. (1998). Family characteristics and health behaviours of adolescents and families. *Psychology and Health*, 13(5), 785–803.
- Ding, Y., Chowdhury, G. G., & Foo, S. (2001). Bibliometric cartography of information retrieval research by using co-word analysis. *Information Processing & Management*, 37(6), 817–842.
- Dolnicar, S., & Leisch, F. (2010). Evaluation of structure and reproducibility of cluster solutions using the bootstrap. *Marketing Letters*, 21(1), 83–101.
- Dougherty, E. R., Hua, J., & Bittner, M. L. (2007). Validation of computational methods in genomics. *Current Genomics*, 8(1), 1–19.
- Dubes, R. C. (1993). Cluster analysis and related issues. In C. H. Chen, L. F. Pau, & P. S. P. Wang (Eds.), *Handbook of pattern recognition and computer vision* (pp. 3–32). World Scientific Publishing Company.
- Dudoit, S., & Fridlyand, J. (2002). A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7), research0036.1–0036.21.
- Fang, Y., & Wang, J. (2012). Selection of the number of clusters via the bootstrap method. *Computational Statistics & Data Analysis*, 56(3), 468–477.
- Färber, I., Günemann, S., Kriegel, H.-P., Kröger, P., Müller, E., Schubert, E., Seidl, T., & Zimek, A. (2010). On using class-labels in evaluation of clusterings. In *MultiClust: 1st international workshop on discovering, summarizing and using multiple clusterings held in conjunction with KDD*, Washington, DC.
- Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383), 553–569.
- Freudenberg, J. M., Joshi, V. K., Hu, Z., & Medvedovic, M. (2009). Clean: Clustering enrichment analysis. *BMC Bioinformatics*, 10(1), 234.
- Fu, W., & Perry, P. O. (2020). Estimating the number of clusters using cross-validation. *Journal of Computational and Graphical Statistics*, 29(1), 162–173.
- Garrido-Castro, A. C., Lin, N. U., & Polyak, K. (2019). Insights into molecular classifications of triple-negative breast cancer: Improving patient selection for treatment. *Cancer Discovery*, 9(2), 176–198.
- Goldstein, S. G., & Linden, J. D. (1969). Multivariate classification of alcoholics by means of the MMPI. *Journal of Abnormal Psychology*, 74(6), 661–669.
- Gordon, A. D. (1998). Cluster Validation. In C. Hayashi, K. Yajima, H. Bock, N. Ohsumi, Y. Tanaka, & Y. Baba (Eds.), *Data science, classification, and related methods. Proceedings of the fifth conference of the International Federation of Classification Societies (IFCS-96)* (pp. 22–39). Springer.
- Gower, J. C. (1968). Adding a point to vector diagrams in multivariate analysis. *Biometrika*, 55, 582–585.

- Gruber, M., Heinemann, F., Brettel, M., & Hungeling, S. (2010). Configurations of resources and capabilities and their performance implications: An exploratory study on technology ventures. *Strategic Management Journal*, 31(12), 1337–1356.
- Hahsler, M., & Hornik, K. (2011). Dissimilarity plots: A visual exploration tool for partitional clustering. *Journal of Computational and Graphical Statistics*, 20(2), 335–354.
- Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002). Cluster validity methods: Part I. *ACM SIGMOD Record*, 31(2), 40–45.
- Halkidi, M., Vazirgiannis, M., & Hennig, C. (2015). Method-independent indices for cluster validation and estimating the number of clusters. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (pp. 616–639). Chapman and Hall/CRC.
- Handl, J., Knowles, J., & Kell, D. B. (2005). Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15), 3201–3212.
- Hedges, L. V. (2019). The statistics of replication. *Methodology*, 15, 3–14.
- Held, L. (2020). A new standard for the analysis and design of replication studies. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 183(2), 431–448.
- Hennig, C. (2004). Asymmetric linear dimension reduction for classification. *Journal of Computational and Graphical Statistics*, 13, 930–945.
- Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis*, 52(1), 258–271.
- Hennig, C. (2015a). Clustering strategy and method selection. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (pp. 703–730). Chapman & Hall/CRC.
- Hennig, C. (2015b). What are the true clusters? *Pattern Recognition Letters*, 64, 53–62.
- Hennig, C., & Lin, C.-J. (2015). Flexible parametric bootstrap for testing homogeneity against clustering and assessing the number of clusters. *Statistics and Computing*, 25(4), 821–833.
- Hoffmann, S., Schönbrodt, F., Elsas, R., Wilson, R., Strasser, U., & Boulesteix, A.-L. (2021). The multiplicity of analysis strategies jeopardizes replicability: Lessons learned across disciplines. *Royal Society Open Science*, 8(4), 201925.
- Homburg, C., Jensen, O., & Krohmer, H. (2008). Configurations of marketing and sales: A taxonomy. *Journal of Marketing*, 72(2), 133–154.
- Huang, H., Liu, Y., Hayes, D. N., Nobel, A., Marron, J. S., & Hennig, C. (2015). Significance testing in clustering. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (pp. 336–357). Chapman and Hall/CRC.
- Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1), 193–218.
- Hutson, M. (2018). Artificial intelligence faces reproducibility crisis. *Science*, 359(6377), 725–726.
- Jaccard, P. (1908). Nouvelles recherches sur la distribution florale. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 44, 223–270.
- Jain, A. K., & Moreau, J. V. (1987). Bootstrap technique in cluster analysis. *Pattern Recognition*, 20(5), 547–568.
- Jamison, R. N., Rock, D. L., & Parris, W. C. (1988). Empirically derived symptom checklist 90 subgroups of chronic pain patients: A cluster analysis. *Journal of Behavioral Medicine*, 11(2), 147–158.
- John, C. R., Watson, D., Russ, D., Goldmann, K., Ehrenstein, M., Pitzalis, C., Lewis, M., & Barnes, M. (2020). M3C: Monte Carlo reference-based consensus clustering. *Scientific Reports*, 10(1), 1–14.
- Kaluza, G. (2000). Changing unbalanced coping profiles—a prospective controlled intervention trial in worksite health promotion. *Psychology and Health*, 15(3), 423–433.
- Kapp, A. V., Jeffrey, S. S., Langerød, A., Børresen-Dale, A.-L., Han, W., Noh, D.-Y., Bøkholm, I. R., Nicolau, M., Brown, P. O., & Tibshirani, R. (2006). Discovery and validation of breast cancer subtypes. *BMC Genomics*, 7(1), 231.
- Kapp, A. V., & Tibshirani, R. (2007). Are clusters found in one dataset present in another dataset? *Biostatistics*, 8(1), 9–31.
- Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: An introduction to cluster analysis*. John Wiley & Sons.
- Lange, T., Roth, V., Braun, M. L., & Buhmann, J. M. (2004). Stability-based validation of clustering solutions. *Neural Computation*, 16(6), 1299–1323.
- Lehmann, B. D., Bauer, J. A., Chen, X., Sanders, M. E., Chakravarthy, A. B., Shyr, Y., & Pietenpol, J. A. (2011). Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies. *The Journal of Clinical Investigation*, 121(7), 2750–2767.
- Leisch, F. (2008). Visualizing cluster analysis and finite mixture models. In C.-H. Chen, W. Härdle, & A. Unwin (Eds.), *Handbook of data visualization* (pp. 561–587). Springer.
- Levine, E., & Domany, E. (2001). Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, 13(11), 2573–2593.
- Liu, X., Sivaganesan, S., Yeung, K. Y., Guo, J., Bumgarner, R. E., & Medvedovic, M. (2006). Context-specific infinite mixtures for clustering gene expression profiles across diverse microarray dataset. *Bioinformatics*, 22(14), 1737–1744.
- Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 129–137.
- McIntyre, R. M., & Blashfield, R. K. (1980). A nearest-centroid technique for evaluating the minimum-variance clustering procedure. *Multivariate Behavioral Research*, 15(2), 225–238.
- Meila, M. (2015). Criteria for comparing Clusterings. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (pp. 640–657). Chapman and Hall/CRC.
- Milligan, G. W., & Cooper, M. C. (1987). Methodology review: Clustering methods. *Applied Psychological Measurement*, 11(4), 329–354.
- Mirkin, B. (2005). *Clustering for data mining: A data recovery approach*. CRC Press.
- Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1), 91–118.
- Morey, L. C., Blashfield, R. K., & Skinner, H. A. (1983). A comparison of cluster analysis techniques within a sequential validation framework. *Multivariate Behavioral Research*, 18(3), 309–329.

- Müller, H., & Hamm, U. (2014). Stability of market segmentation with cluster analysis—a methodological approach. *Food Quality and Preference*, 34, 70–78.
- Nosek, B. A., & Errington, T. M. (2020). What is replication? *PLoS Biology*, 18(3), e3000691.
- Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251), aac4716.
- Phinney, J. S., Dennis, J. M., & Gutierrez, D. M. (2005). College orientation profiles of Latino students from low socioeconomic backgrounds: A cluster analytic approach. *Hispanic Journal of Behavioral Sciences*, 27(4), 387–408.
- Prat, A., Pineda, E., Adamo, B., Galván, P., Fernández, A., Gaba, L., Dez, M., Viladot, M., Arance, A., & Muñoz, M. (2015). Clinical implications of the intrinsic molecular subtypes of breast cancer. *The Breast*, 24, S26–S35.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336), 846–850.
- Rogers, G., & Linden, J. D. (1973). Use of multiple discriminant function analysis in the evaluation of three multivariate grouping techniques. *Educational and Psychological Measurement*, 33(4), 787–802.
- Shen, R., Olshen, A. B., & Ladanyi, M. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, 25(22), 2906–2912.
- Simon, R., Radmacher, M. D., Dobbin, K., & McShane, L. M. (2003). Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *Journal of the National Cancer Institute*, 95(1), 14–18.
- Sinclair, R. R., Tucker, J. S., Cullen, J. C., & Wright, C. (2005). Performance differences among four organizational commitment profiles. *Journal of Applied Psychology*, 90(6), 1280.
- Smith, S. P., & Dubes, R. (1980). Stability of a hierarchical clustering. *Pattern Recognition*, 12(3), 177–187.
- Sørlie, T., Tibshirani, R., Parker, J., Hastie, T., Marron, J. S., Nobel, A., Deng, S., Johnsen, H., Pesich, R., & Geisler, S. (2003). Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proceedings of the National Academy of Sciences of the United States of America*, 100(14), 8418–8423.
- Sotiriou, C., Neo, S.-Y., McShane, L. M., Korn, E. L., Long, P. M., Jazaeri, A., Martiat, P., Fox, S. B., Harris, A. L., & Liu, E. T. (2003). Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proceedings of the National Academy of Sciences of the United States of America*, 100(18), 10393–10398.
- Sweatt, A. J., Hedlin, H. K., Balasubramanian, V., Hsi, A., Blum, L. K., Robinson, W. H., Haddad, F., Hickey, P. M., Condliffe, R., & Lawrie, A. (2019). Discovery of distinct immune phenotypes using machine learning in pulmonary arterial hypertension. *Circulation Research*, 124(6), 904–919.
- Tibshirani, R., Hastie, T., Narasimhan, B., & Chu, G. (2003). Class prediction by nearest shrunken centroids, with applications to DNA microarrays. *Statistical Science*, 18(1), 104–117.
- Tibshirani, R., & Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3), 511–528.
- Tyler, D. E., Critchley, F., Dümbgen, L., & Oja, H. (2009). Invariant co-ordinate selection (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71, 549–592.
- Van Mechelen, I., Boulesteix, A.-L., Dangl, R., Dean, N., Guyon, I., Hennig, C., Leisch, F. & Steinley, D. (2018). Benchmarking in cluster analysis: A white paper. arXiv preprint arXiv:1809.10496.
- Von Luxburg, U. (2010). *Clustering stability: An overview*. Now Publishers Inc.
- Von Luxburg, U., Williamson, R.C., & Guyon, I. (2012). Clustering: Science or art? In Guyon, I., Dror, G., Lemaire, V., and Taylor, G., editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 65–79. . PML Research Press.
- Wang, J. (2010). Consistent selection of the number of clusters via crossvalidation. *Biometrika*, 97(4), 893–904.
- Wilkinson, L., & Friendly, M. (2009). The history of the cluster heat map. *The American Statistician*, 63(2), 179–184.
- Yang, Y., Han, L., Yuan, Y., Li, J., Hei, N., & Liang, H. (2014). Gene co-expression network analysis reveals common system-level properties of prognostic genes across cancer types. *Nature Communications*, 5(1), 1–9.
- Zhang, Q., Burdette, J. E., & Wang, J.-P. (2014). Integrative network analysis of TCGA data for ovarian cancer. *BMC Systems Biology*, 8(1), 1–18.
- Zimmermann, A. (2020). Method evaluation, parameterization, and result validation in unsupervised data mining: A critical survey. *WIREs Data Mining and Knowledge Discovery*, 10(2), e1330.

## SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

**How to cite this article:** Ullmann, T., Hennig, C., & Boulesteix, A.-L. (2021). Validation of cluster analysis results on validation data: A systematic framework. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, e1444. <https://doi.org/10.1002/widm.1444>

## Charles Bouveyron

*Material list:*

Charles Bouveyron, Marco Corneli (2025) Scaling Optimal Transport to High-Dimensional Gaussian Distributions. hal-04930868v3



# Scaling Optimal Transport to High-Dimensional Gaussian Distributions

Charles Bouveyron, Marco Corneli

## ► To cite this version:

Charles Bouveyron, Marco Corneli. Scaling Optimal Transport to High-Dimensional Gaussian Distributions. 2025. hal-04930868v3

HAL Id: hal-04930868

<https://hal.science/hal-04930868v3>

Preprint submitted on 29 Apr 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Scaling Optimal Transport to High-Dimensional Gaussian Distributions

Charles BOUVEYRON<sup>1</sup> & Marco CORNELI<sup>1,2</sup>

<sup>1</sup> Université Côte d'Azur, INRIA, CNRS, LJAD, Maasai, Nice, France

<sup>2</sup> Université Côte d'Azur, CNRS, CEPAM, UMR 7264, Nice, France

April 29, 2025

## Abstract

Although optimal transport (OT) has recently become very popular in machine learning, it faces challenges when dealing with high-dimensional data, such as images or omics data. Current OT approaches for high-dimensional situations rely on projections of the data or measures onto low-dimensional spaces, which inevitably results in information loss. In this work, we consider the case of high-dimensional Gaussian distributions with parsimonious covariance structures and lower intrinsic dimension. We exhibit a simplified closed-form expression of the 2-Wasserstein distance with an efficient and robust calculation procedure based on a low-dimensional decomposition of empirical covariance matrices, without relying on data projections. Furthermore, we provide a closed-form expression for the Monge map, which involves the exact calculation of the square-root and inverse square-root of the source distribution covariance matrix. This approach offers analytical and computational advantages, as demonstrated by our numerical experiments, which quantitatively evaluate these benefits in comparison to existing methods. In addition to being able to compute both the  $W_2^2$ -distance and the transport map, our method outperforms model-free methods, in high dimension, even in the case of non-Gaussian distributions.

## 1 Introduction

Due to its proven versatility, optimal transport (OT) is becoming more and more popular within the machine learning community (Peyré et al., 2019). Basically, once the observed data is identified with a probability distribution (possibly the empirical mass function), optimal transport allows to consistently assess the similarity between complex instances such as point clouds, images or graphs. However, as the modern data are increasingly high-dimensional, OT is also now facing an old problem in optimization and statistical learning: the curse of dimensionality (Bellman, 1957). Among the OT problems that have to face the high dimensionality of the data, we can mention as a popular example the calculation of the Frechet inception distance (FID, Heusel et al., 2017) for comparing the distribution of generated images with the distribution of a set of ground-truth images, using the Wasserstein distance between two full Gaussian distributions.

### 1.1 Statistical learning in high-dimensional spaces

In many application domains of machine learning, such as image analysis, genomics, chemometrics or personalized medicine, the observed data are frequently high-dimensional and learning from such

data is a challenging problem. Indeed, statistical learning in such high-dimensional spaces is made difficult both because of estimation biases and numerical problems (Giraud, 2021; Wainwright, 2019). In particular, when considering the generative (model-based) framework, most learning methods show a disappointing behavior in high-dimensional spaces. They suffer from the well-known curse of dimensionality which is mainly due to the fact that generative methods turn to be dramatically over-parametrized in high-dimensional spaces (Bouveyron et al., 2019). Moreover, even though many variables are measured to describe the studied phenomenon, only a small subset of these original variables is in fact relevant for both modeling and learning. In recent years, several works tried to reduce the data dimensionality or select relevant variables while building a generative predictor, showing excellent results. In this context, there are two main approaches. On the one hand, some works assume that the data of each class live in different low-dimensional subspaces. On the other hand, some other works assume that the classes differ only with respect to some of the original features. Both approaches present two practical advantages: results are improved by the removing of non informative features and the result interpretation is eased by the visualization in the subspaces or the meaning of retained variables. We may recommend to refer to Bouveyron et al. (2019, Chap. 8) and Bouveyron and Brunet-Saumard (2014) for a full overview in the contexts of classification, clustering and dimension reduction. As we focus here on the question of an efficient modeling of high-dimensional distributions, a key work in this context is due to Tipping and Bishop (1999b) who have shown that the subspace of principal component analysis (PCA) could be retrieved from the maximum-likelihood estimator of a parameter, in a particular factor analysis model called probabilistic PCA (PPCA). This probabilistic framework led to diverse Bayesian analysis of PCA (Bishop, 1999; Minka, 2000) and extensions in various ML situations such as classification Bouveyron et al. (2007b) and clustering Tipping and Bishop (1999a); Bouveyron et al. (2007a); McNicholas and Murphy (2008). As it will be shown in this paper, this model will be once again a game-changer tool, here for the optimal transport between high-dimensional Gaussian distributions.

## 1.2 Optimal Transport with Wasserstein distance

Based on the modern formulation of Kantorovich (1942), standard optimal transport generally relies on the Wasserstein distance. Given two random variables  $X_1$  and  $X_2$  supported on  $\mathbb{R}^p$ , with finite second moments and whose marginal cumulative distribution functions are denoted by  $\mu_1$  and  $\mu_2$ , respectively, the squared 2-Wasserstein distance is defined as:

$$W_2^2(\mu_1, \mu_2) := \min_{\pi \in \Pi(\mu_1, \mu_2)} \mathbb{E}_{(X_1, X_2) \sim \pi} \|X_1 - X_2\|_2^2, \quad (1)$$

where  $\Pi(\mu_1, \mu_2)$  denotes the set of *joint* distributions with marginals  $\mu_1$  and  $\mu_2$ , respectively and  $\|\cdot\|_2$  denotes the standard Euclidean norm. The joint distribution  $\pi^*$  minimizing the expectation on the r.h.s. of Eq. (1) is known as optimal coupling or optimal transport plan. As it can be understood from the above equation, OT lifts a metric defined on some ground space (here  $\mathbb{R}^p$  with Euclidean metric) to a metric on the probability distributions supported on that space. The above definition extends to probability measures with support on more general separable metric spaces and higher order Wasserstein distances. However, in this paper we only focus on the 2-Wasserstein distance between measures supported on  $\mathbb{R}^p$ , for some integer  $p$ . For an in depth inspection of Wasserstein distances and their properties the reader is referred to Villani et al. (2009); Santambrogio (2015); Peyré et al. (2019).

In the particular where case the random variables  $X_1$  and  $X_2$  are Gaussian, it was shown that the Wasserstein distance can be computed in closed form (Dowson and Landau, 1982; Takatsu, 2011). Moreover, in force of the Brenier's theorem (see for instance Peyré et al., 2019, Theorem 2.1) there exists a unique transport or *Monge map*  $T^* : \mathbb{R}^p \rightarrow \mathbb{R}^p$  linked to the optimal transport plan  $\pi^*$  by the following relation<sup>1</sup>

$$\mathbb{E}_{(X_1, X_2) \sim \pi^*} [h(X_1, X_2)] = \mathbb{E}_{X_1 \sim \mu_1} [h(X_1, T^*(X_1))],$$

holding for any continuous function  $h : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ . Also  $T^*$  has closed form in the Gaussian case. If the Gaussian distributions of  $X_1$  and  $X_2$  must be inferred from the data, i.e. two point clouds in dimensions  $\mathbb{R}^p$ , the closed formulas for the Wasserstein distance and the Monge map  $T^*$  between  $\mu_1$  and  $\mu_2$  can always been computed. However, due to the difficulties in the estimation of the covariance matrices in high dimension, those formulas lead to poor estimates (shown in Section 3). The alternative approach, seeking to compute the Wasserstein distance between  $\mu_1$  and  $\mu_2$  via the empirical distributions, is also doomed to failure due to the known instability of OT in high dimension (Dudley, 1969; Fournier and Guillin, 2015). Among the existing solutions, we can cite Sliced Wasserstein distances (SWD, see Nguyen and Ho, 2024, and the references therein), which attack the high-dimensional problem by averaging the optimal OT costs between 1D measures, obtained by projecting the original measures onto several random directions. Another approach builds Subspace Robust Wasserstein distances (SRW, Paty and Cuturi, 2019), which are defined by modifying the 2-Wasserstein cost in such a way to find an optimal matching between projections of the original measures onto a  $k$ -dimensional subspace. However, both SWD and SRW do not allow to estimate the Monge map  $T^*$ . Still based on the empirical distributions, Muzellec and Cuturi (2019) introduced in the literature two methods to extend a Monge map which is optimal on a subspace to one that is *nearly* optimal on the entire space. However, all those approaches are non-parametric, meaning that the Gaussianity of the input data is never used (nor the closed formulas mentioned above).

### 1.3 Contributions of the paper

This work focuses on the use of the high-dimensional Gaussian (HD-Gaussian) distributions, induced by the probabilistic PCA (PPCA) model, for the optimal transport between high-dimensional data distributions. In particular, this paper features the three main contributions:

- (i) exhibition of a closed-form expression of the 2-Wasserstein distance between two HD-Gaussian distributions, with an efficient and robust calculation procedure based on a low-dimensional subspace decomposition.
- (ii) generalization to a less restrictive framework of previous state-of-the-art results, which considered Gaussian distributions with similar covariance orientations or structures.
- (iii) exhibition of a closed-form expression of the Monge map for the transport of a HD-Gaussian distribution on another one, involving an exact calculation of both the square-root and the inverse square-root of the covariance matrix of the source distribution, avoiding in turn many numerical drawbacks in high-dimensional practical situations.

---

<sup>1</sup>In a short-hand notation one writes  $\pi^* = (\text{Id}, T^*)_\# \mu_1$ , where  $\#$  is the push-forward operator.

Interestingly, these results remain valid in the case of HD-Gaussian distributions with different intrinsic dimensions. It is also worth underlying that the proposed approach, named hereafter OT-HDGauss, is able to compute both the  $W_2^2$ -distance and the transport map for HD-Gaussian distributions. Furthermore, the analytical and numerical advantages of our approach in high dimensions allow it to outperform model-free methods in the case of non-Gaussian distributions. These contributions are supported by numerical experiments that highlight the performance and robustness of the proposed OT-HDGauss method to both the dimensionality and the sample size, and this in comparison with the most recent OT approaches.

## 2 Optimal Transport between HD-Gaussian Distributions

### 2.1 The HD-Gaussian distribution

To overcome the well-known “curse of the dimensionality” in statistical learning, Tipping and Bishop (1999b) have proposed a parsimonious Gaussian distribution, induced by a probabilistic view of PCA, that splits the modelling between a low-dimensional subspace where the data actually live and a noise component. This HD-Gaussian distribution can be defined as follows.

**Definition 2.1.** A  $p$ -dimensional random vector  $X \in \mathbb{R}^p$  follows a HD-Gaussian distribution  $\mathcal{N}_{HD}(m, U, \Lambda, \sigma^2, d)$  if it exists a low-dimensional latent random vector  $Y \in \mathbb{R}^d$ , of intrinsic dimensionality  $d < p$ , and a  $p$ -dimensional noise random vector  $\varepsilon \in \mathbb{R}^p$  such that:

$$\begin{aligned} X &= UY + m + \varepsilon, \\ Y &\sim \mathcal{N}(0, \Lambda), \\ \varepsilon &\sim \mathcal{N}(0, \sigma^2 I_p), \end{aligned}$$

where  $U$  is a  $p \times d$  transformation matrix whose columns are orthonormal vectors,  $m \in \mathbb{R}^p$  is the mean vector,  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_d)$  and  $\sigma^2 > 0$ .

Under these assumptions, it can be shown that the HD-Gaussian distribution  $\mathcal{N}_{HD}(m, U, \Lambda, \sigma^2, d)$  is a specific Gaussian distribution with a structured covariance matrix.

**Proposition 2.2.** A  $p$ -dimensional random vector  $X \in \mathbb{R}^p$  following a HD-Gaussian distribution  $\mathcal{N}_{HD}(m, U, \Lambda, \sigma^2, d)$  is distributed as:

$$X \sim \mathcal{N}(m, Q\Delta Q^t),$$

where  $Q = [U, R]$ , the  $p \times p$  matrix made of  $U$  and an orthonormal complementary  $R$ , and  $\Delta$  is a block-diagonal matrix:

$$\Delta = \left( \begin{array}{cc|cc} \delta_1 & 0 & & 0 \\ & \ddots & & \\ 0 & \delta_d & & \\ \hline & & \sigma^2 & 0 \\ & 0 & & \ddots \\ & & 0 & \sigma^2 \end{array} \right) \quad \left. \begin{array}{l} d \\ (p-d) \end{array} \right\}$$

with  $\delta_j = \lambda_j + \sigma^2$  and  $\delta_j > \sigma^2$ , for  $j = 1, \dots, d$ .

*Proof.* Assuming that  $X = UY + m + \varepsilon$ , where  $Y \sim \mathcal{N}(0, \Lambda)$  and  $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_p)$ , the conditional distribution of  $Y$  is therefore Gaussian:

$$X | Y \sim \mathcal{N}(UY + m, \sigma^2 I_p),$$

and the marginal distribution of  $X$  is a Gaussian distribution with a specific structured covariance structure:

$$X \sim \mathcal{N}(m, \Sigma),$$

where  $\Sigma = U\Lambda U^t + \sigma^2 I_p$ . Introducing  $Q = [U, R]$ , the  $p \times p$  matrix made of  $U$  and an orthonormal complementary  $R$ , the covariance matrix  $\Sigma$  can be easily rewritten  $\Sigma = Q\Delta Q^t$  where  $\Delta = \text{diag}(\delta_1, \dots, \delta_d, \sigma^2, \dots, \sigma^2)$ . This allows to conclude.  $\square$

Therefore, the HD-Gaussian distribution is fully parametrized by the set of parameters  $\theta = \{m, U, \lambda_j, \sigma^2, d; \forall j = 1, \dots, d\}$ .

## 2.2 Calculation of the 2-Wasserstein distance

Let us now consider two HD-Gaussian probability distributions  $\mu_1 \sim \mathcal{N}_{HD}(m_1, U_1, \Lambda_1, \sigma_1^2, d_1)$  and  $\mu_2 \sim \mathcal{N}_{HD}(m_2, U_2, \Lambda_2, \sigma_2^2, d_2)$  on  $\mathbb{R}^p$  for which we would like to compute the 2-Wasserstein distance. The following proposition exhibits a closed-form expression of  $W_2(\mu_1, \mu_2)$ , which in turn yields to numerically efficient and stable calculations.

**Proposition 2.3.** *The 2-Wasserstein distance between two HD-Gaussian distributions  $\mu_1 \sim \mathcal{N}_{HD}(m_1, U_1, \Lambda_1, \sigma_1^2, d_1)$  and  $\mu_2 \sim \mathcal{N}_{HD}(m_2, U_2, \Lambda_2, \sigma_2^2, d_2)$  is*

$$\begin{aligned} W_2^2(\mu_1, \mu_2) &= \|m_1 - m_2\|_2^2 + \text{trace}(\Lambda_1) + \text{trace}(\Lambda_2) \\ &\quad + p(\sigma_1^2 + \sigma_2^2) - 2\text{trace}(A^{\frac{1}{2}}), \end{aligned}$$

where  $A$  can be expressed as:

$$A = U_1 \Lambda_1 U_1^t U_2 \Lambda_2 U_2^t + \sigma_1^2 U_2 \Lambda_2 U_2^t + \sigma_2^2 U_1 \Lambda_1 U_1^t + \sigma_1^2 \sigma_2^2 I_p.$$

*Proof.* In the case when  $c(x, y) = \|x - y\|_2^2$ , the 2-Wasserstein distance between two Gaussian distributions  $\mu_1 \sim \mathcal{N}(m_1, \Sigma_1)$  and  $\mu_2 \sim \mathcal{N}(m_2, \Sigma_2)$ , is known to have the following explicit form (Dowson and Landau, 1982; Takatsu, 2011)

$$\begin{aligned} W_2(\mu_1, \mu_2)^2 &= \|m_1 - m_2\|_2^2 + \text{trace}(\Sigma_1) + \text{trace}(\Sigma_2) \\ &\quad - 2\text{trace}\left[\left(\Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}}\right)^{\frac{1}{2}}\right]. \end{aligned} \tag{2}$$

Thanks to Proposition 2.2, this result can be extended to two HD-Gaussian distributions  $\mu_1 \sim \mathcal{N}_{HD}(m_1, U_1, \Lambda_1, \sigma_1^2, d_1)$  and  $\mu_2 \sim \mathcal{N}_{HD}(m_2, U_2, \Lambda_2, \sigma_2^2, d_2)$ , by considering that  $\Sigma_1$  and  $\Sigma_2$  have specific parsimonious structures, i.e.  $\Sigma_i = U_i \Lambda_i U_i^t + \sigma_i^2 I_p$ , with  $\Lambda_i = \text{diag}(\delta_{i1} - \sigma_i^2, \dots, \delta_{id} - \sigma_i^2)$ , for  $i = 1, 2$ . It is first straightforward to establish that  $\text{trace}(\Sigma_1) = \text{trace}(\Lambda_1) + p\sigma_1^2$ , and similarly for  $\Sigma_2$ .

Let's now consider the computation of  $\text{trace} \left[ \left( \Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}} \right)^{\frac{1}{2}} \right]$ . Reminding that trace of matrix  $M$  is equal to the sum of its eigenvalues  $\omega_1(M), \dots, \omega_p(M)$ , we can write:

$$\begin{aligned} \text{trace} \left[ \left( \Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}} \right)^{\frac{1}{2}} \right] &= \sum_{j=1}^p \omega_j \left[ \left( \Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}} \right)^{\frac{1}{2}} \right] \\ &= \sum_{j=1}^p \omega_j^{\frac{1}{2}} \left[ \Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}} \right] \\ &= \sum_{j=1}^p \omega_j^{\frac{1}{2}} [\Sigma_1 \Sigma_2]. \end{aligned}$$

This allows us to conclude that  $\text{trace} \left[ \left( \Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}} \right)^{\frac{1}{2}} \right] = \text{trace} \left[ (\Sigma_1 \Sigma_2)^{\frac{1}{2}} \right]$ . Then, exploiting the parsimonious structure of both  $\Sigma_1$  and  $\Sigma_2$ , one can get a form of  $\Sigma_1 \Sigma_2$  that depends only on low rank matrix calculations:

$$\begin{aligned} \Sigma_1 \Sigma_2 &= (U_1 \Lambda_1 U_1^t + \sigma_1^2 I_p)(U_2 \Lambda_2 U_2^t + \sigma_2^2 I_p) \\ &= U_1 \Lambda_1 U_1^t U_2 \Lambda_2 U_2^t + \sigma_1^2 U_2 \Lambda_2 U_2^t + \sigma_2^2 U_1 \Lambda_1 U_1^t \\ &\quad + \sigma_1^2 \sigma_2^2 I_p. \end{aligned}$$

Combining the different parts above allows us to conclude.  $\square$

*Remark 2.4.* It is first important to notice that Proposition 2.3 provides a numerically efficient ways to compute the 2-Wasserstein distance in high-dimensional spaces. Indeed, the formulae exhibited above involves the computing of the trace of the square root of a matrix which is expressed only with low-rank matrix calculations. This will even be more determinant when the different elements involved need to be estimated from the data, as discussed later in this paper.

*Remark 2.5.* Let us also notice that Proposition 2.3 is valid even when the intrinsic dimensions  $d_1$  and  $d_2$  of the two HD-Gaussian distributions are different.

In the specific case where the two distributions share the same subspace, i.e.  $U_1 = U_2$ , the previous result reduces to an even simpler form of the 2-Wasserstein distance, as stated in the next proposition.

**Proposition 2.6.** *The 2-Wasserstein distance between two HD-Gaussian distributions  $\mu_1 \sim \mathcal{N}_{HD}(m_1, U, \Lambda_1, \sigma_1^2, d)$  and  $\mu_2 \sim \mathcal{N}_{HD}(m_2, U, \Lambda_2, \sigma_2^2, d)$  is*

$$\begin{aligned} W_2^2(\mu_1, \mu_2) &= \|m_1 - m_2\|_2^2 + \sum_{j=1}^d \left( \sqrt{\delta_{1j}} - \sqrt{\delta_{2j}} \right)^2 \\ &\quad + p (\sigma_1 - \sigma_2)^2. \end{aligned}$$

*Proof.* Starting with the result of Proposition 2.3 and assuming now that  $U_1 = U_2 = U$ , we can first rewrite  $(\Sigma_1 \Sigma_2)^{\frac{1}{2}}$  as:

$$\begin{aligned} [\Sigma_1 \Sigma_2]^{\frac{1}{2}} &= [(U \Lambda_1 U^t + \sigma_1^2 I_p)(U \Lambda_2 U^t + \sigma_2^2 I_p)]^{\frac{1}{2}} \\ &= [(Q \Delta_1 Q^t)(Q \Delta_2 Q^t)]^{\frac{1}{2}} \\ &= [Q(\Delta_1 \Delta_2) Q^t]^{\frac{1}{2}} = Q [\Delta_1 \Delta_2]^{\frac{1}{2}} Q^t \\ &= Q \text{diag}(\sqrt{\delta_{11} \delta_{21}}, \dots, \sqrt{\delta_{1d} \delta_{2d}}, \sigma_1 \sigma_2, \dots, \sigma_1 \sigma_2) Q^t. \end{aligned}$$

Therefore, as  $Q$  is an orthonormal  $p \times p$  matrix,  $\text{trace}([\Sigma_1 \Sigma_2]^{\frac{1}{2}})$  becomes:

$$\text{trace}([\Sigma_1 \Sigma_2]^{\frac{1}{2}}) = \sum_{j=1}^d \sqrt{\delta_{1j}} \sqrt{\delta_{2j}} + (p-d)\sigma_1 \sigma_2.$$

Reporting this quantity in the final formulation of the 2-Wasserstein distance, we get

$$\begin{aligned} W_2^2(\mu_1, \mu_2) &= \|m_1 - m_2\|_2^2 + \text{trace}(\Lambda_1) + \text{trace}(\Lambda_2) \\ &\quad + p(\sigma_1^2 + \sigma_2^2) - 2\text{trace}([\Sigma_1 \Sigma_2]^{\frac{1}{2}}) \\ &= \|m_1 - m_2\|_2^2 + \sum_{j=1}^d (\lambda_{1j} + \lambda_{2j}) + p(\sigma_1^2 + \sigma_2^2) \\ &\quad - 2 \left( \sum_{j=1}^d \sqrt{\delta_{1j}} \sqrt{\delta_{2j}} + (p-d)\sigma_1 \sigma_2 \right). \end{aligned}$$

Finally, recalling that  $\delta_{ij} = \lambda_{ij} + \sigma_i^2$ , we get

$$\begin{aligned} W_2^2(\mu_1, \mu_2) &= \|m_1 - m_2\|_2^2 + \sum_{j=1}^d \left( \delta_{1j} + \delta_{2j} - 2\sqrt{\delta_{1j}} \sqrt{\delta_{2j}} \right) \\ &\quad + (p-d)(\sigma_1^2 + \sigma_2^2 - 2\sigma_1 \sigma_2) \\ &= \|m_1 - m_2\|_2^2 + \sum_{j=1}^d \left( \sqrt{\delta_{1j}} - \sqrt{\delta_{2j}} \right)^2 \\ &\quad + (p-d)(\sigma_1 - \sigma_2)^2 \end{aligned}$$

This concludes the proof.  $\square$

*Remark 2.7.* The above proposition recovers and generalizes results established by several previous works, including Dowson and Landau (1982), Takatsu (2011) and Peyré et al. (2019). Indeed, if we set  $d = d_1 = d_2 = p - 1$ , we recover exactly those well-known results. If  $d < p$ , and in particular if  $d$  is small compared to  $p$ , this new formula is proposing a sort of regularization of the general expression, which may have an interesting numerical behavior in practical high-dimensional situations.

*Remark 2.8.* Despite the elegant form of the 2-Wasserstein distance in the case  $U_1 = U_2$ , exploiting this formula is far from being trivial in practice and this is rarely highlighted in the literature. Indeed, the (statistical) estimation of a common subspace of dimension  $d$  of two sets of data distributed as two different HD-Gaussian distributions is a quite complex problem, that requires the use of iterative algorithms, such as the Flury-Gautschi algorithm Flury and Gautschi (1986), to solve this problem.

### 2.3 Calculation of the optimal transport plan

Let us now consider the calculation of the optimal transport plan between two HD-Gaussian distributions  $\mu_1$  and  $\mu_2$ . The following proposition exhibits a closed-form expression of the Monge map for the transport of  $\mu_1$  toward  $\mu_2$ , involving an exact calculation of the inverse square-root of the covariance matrix of the source distribution.

**Theorem 2.9.** *The optimal transport map  $T^*$  between two HD-Gaussian distributions  $\mu_1 \sim \mathcal{N}_{HD}(m_1, U_1, \Lambda_1, \sigma_1^2, d_1)$  and  $\mu_2 \sim \mathcal{N}_{HD}(m_2, U_2, \Lambda_2, \sigma_2^2, d_2)$  is*

$$\forall x \in \mathbb{R}^p, T^*(x) = m_2 + \Sigma_1^{-\frac{1}{2}} \left[ \Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}} \right]^{\frac{1}{2}} \Sigma_1^{-\frac{1}{2}}(x - m_1),$$

where both  $\Sigma_1^{\frac{1}{2}}$  and  $\Sigma_1^{-\frac{1}{2}}$  have the explicit closed-form formulations

$$\Sigma_1^{\frac{1}{2}} = \sigma_1 I_p + U_1 C_1 U_1^t,$$

with  $C_1 = \text{diag}(\sqrt{\delta_{11}} - \sigma_1, \dots, \sqrt{\delta_{1d}} - \sigma_1) > 0$  and

$$\Sigma_1^{-\frac{1}{2}} = \frac{1}{\sigma_1} (I_p - U_1 D_1 U_1^t)$$

with  $D_1 = \text{diag}\left(\frac{\sqrt{\delta_{11}} - \sigma_1}{\sqrt{\delta_{11}}}, \dots, \frac{\sqrt{\delta_{1d}} - \sigma_1}{\sqrt{\delta_{1d}}}\right)$ .

*Proof.* The optimal transport map  $T^*$  between two Gaussian distributions  $\mu_1 \sim \mathcal{N}(m_1, \Sigma_1)$  and  $\mu_2 \sim \mathcal{N}(m_2, \Sigma_2)$  is affine and is given by (Dowson and Landau, 1982; Takatsu, 2011):

$$\forall x \in \mathbb{R}^p, T^*(x) = m_2 + A^{-1}(x - m_1), \quad (3)$$

where  $A^{-1} = \Sigma_1^{-\frac{1}{2}} \left[ \Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}} \right]^{\frac{1}{2}} \Sigma_1^{-\frac{1}{2}}$  involves difficult computations in high-dimensional spaces. Assuming that  $\mu_1$  and  $\mu_2$  have structured covariance structures of the form of the HD-Gaussian distribution, i.e.  $\Sigma_i = U_i \Lambda_i U_i^t + \sigma_i^2 I_p$ , for  $i = 1, 2$ , let us first focus on the computation of  $\Sigma_1^{\frac{1}{2}}$ . To do so, we start with Theorem 1.35 of Higham (2008) which expresses the form of  $f(M)$  where  $M = AB + \alpha I_p$  and  $f$  is defined on the spectrum of  $AB + \alpha I_p$ :

$$f(M) = f(\alpha)I_p + A(BA)^{-1} (f(BA + \alpha I_d) - f(\alpha)I_d) B.$$

Applying this result to the function  $f(x) = x^{\frac{1}{2}}$ , we get:

$$M^{\frac{1}{2}} = \alpha^{\frac{1}{2}} I_p + A(BA)^{-1} \left( (BA + \alpha I_d)^{\frac{1}{2}} - \alpha^{\frac{1}{2}} I_d \right) B.$$

Working now with  $M = \Sigma_1 = U_1(\Lambda_1 U_1^t) + \sigma_1^2 I_p$ , we get:

$$\begin{aligned}\Sigma_1^{\frac{1}{2}} &= \sigma_1 I_p + U_1 ((\Lambda_1 U_1^t) U_1)^{-1} \\ &\quad \left( (\sigma_1^2 I_d + (\Lambda_1 U_1^t) U_1)^{\frac{1}{2}} - \sigma_1 I_d \right) (\Lambda_1 U_1^t).\end{aligned}$$

Since  $U_1^t U_1 = I_d$ , the equation reduces to:

$$\Sigma_1^{\frac{1}{2}} = \sigma_1 I_p + U_1 \Lambda_1^{-1} \left( (\sigma_1^2 I_d + \Lambda_1)^{\frac{1}{2}} - \sigma_1 I_d \right) \Lambda_1 U_1^t.$$

Furthermore, as  $\Lambda_1 = \text{diag}(\delta_{11} - \sigma_1^2, \dots, \delta_{1d} - \sigma_1^2)$ , we get  $(\sigma_1^2 I_d + \Lambda_1)^{\frac{1}{2}} = \text{diag}(\sqrt{\delta_{11}}, \dots, \sqrt{\delta_{1d}})$ , which can be reinjected in the above formula:

$$\begin{aligned}\Sigma_1^{\frac{1}{2}} &= \sigma_1 I_p + U_1 \Lambda_1^{-1} C_1 \Lambda_1 U_1^t \\ &= \sigma_1 I_p + U_1 C_1 U_1^t,\end{aligned}$$

where  $C_1 = \text{diag}(\sqrt{\delta_{11}} - \sigma_1, \dots, \sqrt{\delta_{1d}} - \sigma_1)$ . Let's now consider the computation of  $\Sigma_1^{-\frac{1}{2}}$ :

$$\Sigma_1^{-\frac{1}{2}} = (\sigma_1 I_p + U_1 \Lambda_1^{-1} C_1 \Lambda_1 U_1^t)^{-1}.$$

Using now the Woodbury formula, we get:

$$\begin{aligned}\Sigma_1^{-\frac{1}{2}} &= \frac{1}{\sigma_1} I_p - \frac{1}{\sigma_1} U_1 \left( C_1^{-1} + U_1^t \frac{1}{\sigma_1} I_d U_1 \right)^{-1} U_1^t \frac{1}{\sigma_1} I_p \\ &= \frac{1}{\sigma_1} \left( I_p - \frac{1}{\sigma_1} U_1 \left( C_1^{-1} + \frac{1}{\sigma_1} I_d \right)^{-1} U_1^t \right) \\ &= \frac{1}{\sigma_1} \left( I_p - \frac{1}{\sigma_1} U_1 \tilde{D}_1 U_1 \right),\end{aligned}$$

where  $\tilde{D}_1 = \left( C_1^{-1} + \frac{1}{\sigma_1} I_d \right)^{-1}$ . Taking into account the diagonal structure of  $C_1$ , we can write:

$$\begin{aligned}\tilde{D}_1 &= \left( C_1^{-1} + \frac{1}{\sigma_1} I_d \right)^{-1} \\ &= \left[ \text{diag}(\sqrt{\delta_{1j}} - \sigma_1)_{j=1,\dots,d}^{-1} + \frac{1}{\sigma_1} I_d \right]^{-1} \\ &= \left[ \text{diag}\left(\frac{1}{\sqrt{\delta_{1j}} - \sigma_1} + \frac{1}{\sigma_1}\right)_{j=1,\dots,d} \right]^{-1} \\ &= \sigma_1 \text{diag}\left(\frac{\sqrt{\delta_{1j}} - \sigma_1}{\sqrt{\delta_{1j}}}\right)_{j=1,\dots,d}.\end{aligned}$$

We finally get:

$$\Sigma_1^{-\frac{1}{2}} = \frac{1}{\sigma_1} (I_p - U_1 D_1 U_1),$$

with  $D_1 = \text{diag}\left(\frac{\sqrt{\delta_{1j}} - \sigma_1}{\sqrt{\delta_{1j}}}\right)_{j=1,\dots,d}$ .

□

*Remark 2.10.* The computation of the transport map  $T^*$  usually requires the inversion of a covariance matrix, which will be rarely of full rank in high-dimensional spaces. In our case, Proposition 2.9 provides an explicit and stable inverse of the square-root of the covariance matrix  $\Sigma_1$  and consequently an efficient and numerically stable way of computing the transport plan  $T$ , even in situations where  $\Sigma_1$  and  $\Sigma_2$  are not of full rank. In addition, Proposition 2.9 also provides an explicit form of the square-root of  $\Sigma_1$ .

*Remark 2.11.* Once again, the result of Proposition 2.9 is valid even when  $d_1$  and  $d_2$  are different. This is naturally a key point in practical situations where there is no reason to have distributions with identical intrinsic dimensions.

## 2.4 Inference and intrinsic dimension estimation

**Inference** Assuming that two point clouds  $X^{(1)}$  and  $X^{(2)}$  sampled from HD-Gaussian distributions are given and that their intrinsic dimensions  $d_1$  and  $d_2$  are known, the computation of both the 2-Wasserstein distance and the associated transport map requires the estimation of the parameters  $\mu_i, \lambda_{ij}, \sigma_i$  and  $U_i$ , for  $i = 1, 2$  and  $j = 1, \dots, d$ . Following Tipping and Bishop (1999b), the maximum likelihood estimates of those parameters are, for  $i = 1, 2$ :

$$\begin{aligned}\hat{\mu}_i &= \sum_{\ell=1}^{n_i} x_{\ell}^{(i)} / n_i, & \hat{\lambda}_{ij} &= \omega_j(S_i) - \hat{\sigma}_i \\ \hat{\sigma}_i &= \left( \text{trace}(S_i) - \sum_{j=1}^d \omega_j(S_i) \right) / (p - d),\end{aligned}$$

and  $\hat{U}_i$  is formed by the  $d_i$  leading eigenvectors (i.e. associated with the  $d_i$  largest eigenvalues  $\omega_j(S_i)$ ) of  $S_i$ . Finally,  $S_i = (X^{(i)} - \hat{\mu}_i)^t (X^{(i)} - \hat{\mu}_i)$  is the empirical covariance matrix.

**Estimation of the intrinsic dimensions** As in practical situations the intrinsic dimensionality of the data is not known, we also need to estimate  $d_1$  and  $d_2$  from the data. This question has been intensively studied in the last two decades and remains a difficult question in general. Among the possible solutions, we can cite the works of Cattell (1966), Bouveyron et al. (2011), Josse and Husson (2012) and Bouveyron et al. (2020). Even though intrinsic dimensionality estimation is a challenging task in general, the effect of some variation on the estimation of the actual dimensions of the source and target distributions will be limited in our case since we model the data in the whole high-dimensional space with a parsimonious approach, without effective dimension reduction. As illustrated in Appendix B, the cross-validation approach of Josse and Husson (2012) for PCA performs well in a variety of situations and we recommend to use it in practice. This technique will be used in the following for estimating the intrinsic dimensions of the source and target distributions.

## 3 Numerical experiments

### 3.1 Experimental setups and methods

**Simulated scenarios** For evaluating the proposed method performance in calculating both the Wasserstein distance and the Monge map, we designed 3 simulation scenarios:

- i) The first scenario, hereafter referred to as GaussHD, consists in drawing  $n$  observations in dimension  $p$  from two HD-Gaussian distributions, as defined by Definition 2.1. While  $n$  and  $p$  are allowed to vary in order to both test the effects “sample size” and “high dimension”, we assumed that both distributions share  $d_1 = d_2 = 5$  and that are centered (these requirements, i.e. same intrinsic dimension and centrality, are needed by some competitor approaches). Moreover, we set  $\sigma_1^2 = 0.4$  and  $\sigma_2^2 = 0.2$  whereas  $\text{diag}(\Lambda) = \{\lambda_i, \lambda_i, \lambda_i, \lambda_i, \lambda_i\}$ , with  $i = 1, 2$ ,  $\lambda_1 = 3.6$  and  $\lambda_2 = 1.8$ .
- ii) The second scenario, called FullGauss, assumes the data are sampled from two centered Gaussian distributions ( $d_1 = d_2 = p$ ). For the source distribution (respectively the destination distribution) we created a decreasing sequence of  $p$  eigenvalues ranging from 3.6 to 0.4 (1.8 to 0.2) representing the spectrum of the covariance matrix.
- iii) The last simulation scenario considers two non Gaussian distributions: the skew-Normal (Azzalini, 2013) and Student distributions. In this case, data are sampled from multivariate ( $p$ -dimensional) skew-Normal and Student distributions where the scaling / correlation matrices of the two distributions are simulated as for the FullGauss scenario.

**State-of-the-art methods** Once the source and target point clouds are sampled, in order to compute the Wasserstein distance (and possibly the Monge map) between the generating distributions, two global strategies exist, depending the considered approach. Either the parameters of the distributions are learned from the data and *then* the Wasserstein distance and Monge map are computed via the Gaussian closed formulas, otherwise each point is equipped with mass  $1/n$  and the Wasserstein distance (or another OT distance) is learned *directly*, numerically. We will compare hereafter the following 9 methods, adopting one or the other strategy: OT-Gauss, the classical  $W_2^2$ -distance and Monge map computations between 2 Gaussians, OT-GaussReg, its ridge-regularization, the Earth movers distance EMD, Sinkhorn (Cuturi, 2013), SRW (Paty and Cuturi, 2019), SWD (Bonneel et al., 2015), MK-dist and MI-dist Muzellec and Cuturi (2019), and OT-HDGauss, the approach proposed in this work. More details about these approaches are given in Appendix A. Let us notice that EMD, Sinkhorn, SWD and SRW are limited to the calculation of OT distances and they cannot compute the Monge map. Consequently, they won’t be used for comparisons about transport maps.

### 3.2 Computation of the $W_2^2$ -distance

In this first experiment, we focus on the numerical computation of the  $W_2^2$ -distance between two Gaussian distributions (HDGauss and FullGauss scenarios). In particular, we aim to study the robustness of the considered OT approaches against the data dimensionality  $p$  and the sample size  $n$ . To this end, we first simulated data from source and target distributions according to the HDGauss and FullGauss scenarios, with a fixed sample size  $n = 50$  and varying dimensions of the observations space  $p \in [10, 150]$ . We applied the OT methods listed above on these simulated data to calculate the  $W_2^2$ -distance between the source and target distributions. For methods working on subspaces, i.e. MK-dist, MI-dist and SRW, we always provided them with the actual intrinsic dimension  $d = 5$ . Same for OT-HDGauss. The performance of these approaches in computing the  $W_2^2$ -distance between  $\mu_s$  and  $\mu_t$  is assessed by the absolute-value difference with the exact Gaussian  $W_2^2$ -distance computed between the true distributions, whose parameters are known. All results are averaged over 25 replications. Figure 1 presents the performance in computing the  $W_2^2$ -distance for the different methods, according to the space dimensionality  $p$ , and this for both HD-Gaussian and full Gaussian distributions. In order to keep the exposition uncluttered, we did not report results for

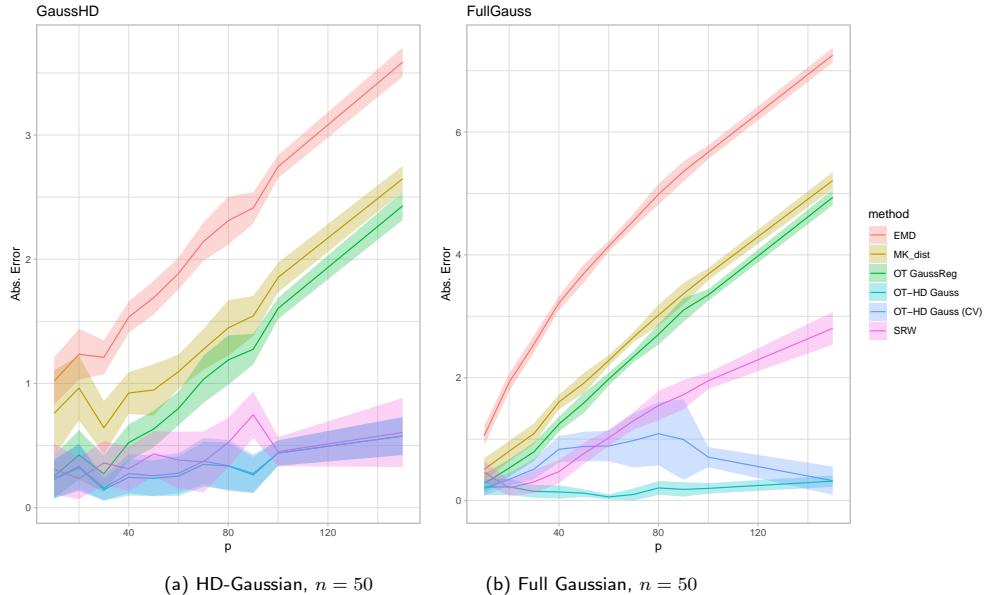


Figure 1: Absolute value difference between the actual  $W_2^2$ -distance (computed from the true distributions) and their estimations using the compared methods, with a fixed dimension  $n = 50$  of the observation space and where the dimension  $p$  of the observation space varies: (a) with simulated HD-Gaussian distributions, (b) with simulated full Gaussian distributions. Results are averaged over 25 replications.

all the methods listed in the previous section. In particular: the behaviour of OT-Gauss is almost indistinguishable from OT-GaussReg in low-dimension, and stops working for  $p > n$ . Same remark for Sinkhorn and EMD (not reported). Instead, SWD and MI-dist are systematically outperformed by (for instance) MK-dist and this is reported in Appendix C. From Figure 1, it clearly appears that EMD, MK-dist and OT-Gauss have a high sensibility to the data dimensionality in both scenarios and make important errors in the computation of the Wasserstein distance in high-dimensional spaces. In the case of the HDGauss scenario, SRW and the two OT-HDGauss approaches show a good robustness to the dimensionality and see their estimations of the Wasserstein distance are little impacted by the increase of the dimensionality. Not surprisingly, the two OT-HDGauss approaches only slightly outperform SRW here since the simulation scenario is favorable. This is however not the case for the FullGauss scenario (Figure 1-b) where the OT-HDGauss approaches outperform all approaches, including SRW, even though the data are not simulated according to their model. We also studied the robustness of the considered OT approaches against the sample size  $n$  in high dimensions. For this, we simulated data from source and target distributions according to the HDGauss and FullGauss scenarios, with a fixed dimension  $p = 100$  and varying sample sizes  $n \in [20, 250]$ . Figure 2 presents the performance in computing the  $W_2^2$ -distance for the different methods, according to the sample size  $n$  for both simulation scenarios. One can first notice that EMD performs badly whatever the sample size. Conversely, MK-dist and OT-Gauss benefit from the increase of the sample size and significantly improve their performance when the sample size is clearly larger than the space dimensionality. This experiment also reveals a surprising behavior of

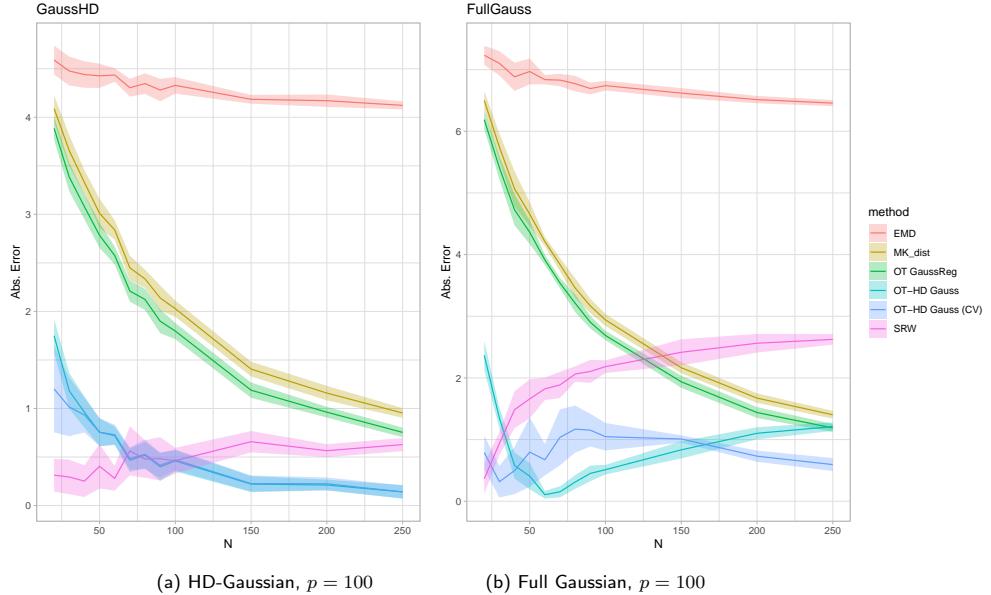


Figure 2: Absolute value difference between the actual  $W_2^2$ -distance (computed from the true distributions) and their estimations using the compared methods, with a fixed dimension  $p = 100$  of the observation space and where the sample size  $n$  varies: (a) with simulated HD-Gaussian distributions, (b) with simulated full Gaussian distributions. Results are averaged over 25 replications.

SRW, which was not possible to see in the previous experiment: the performance of SRW decreases with the increase of the sample size. In the FullGauss scenario OT-HDGauss exhibits the same behavior and this can be explained by the fact that generative model it is based on is no longer the true one, thing that emerges for large  $n$ . However in the HDGauss scenario the data *almost* live in a subspace of dimension  $d$ , thing that should favour SRW but apparently it does not. This is rather counter intuitive and probably linked to the fact that SRW does not compute the exact Wasserstein distance, but a lower-bound of it. Finally, OT-HDGauss demonstrates here again a clear robustness to the sample size in high dimensions, in both scenarios. In the FullGauss case, the OT-HDGauss with the CV procedure to select the intrinsic dimension has to be recommended since it better adapts to the data.

### 3.3 Computation of the Monge map

This experiment now focuses on the computation of the transport map, that our approach is also able to compute. Here again, we aim at studying the the robustness against the data dimensionality  $p$  and the sample size  $n$  of the OT approaches allowing the Monge map computation. To this end, we simulated data from source and target distributions according to the HDGauss and FullGauss scenarios, first with a fixed sample size  $n = 50$  and varying dimensions of the observations space  $p \in [10, 150]$ , and second with a fixed dimension  $p = 100$  and varying sample sizes  $n \in [20, 250]$ . On these simulated data set, we then applied the 3 methods able to compute the transport map. In order to measure the performance of the transport undertaken, we simulated two point clouds, one

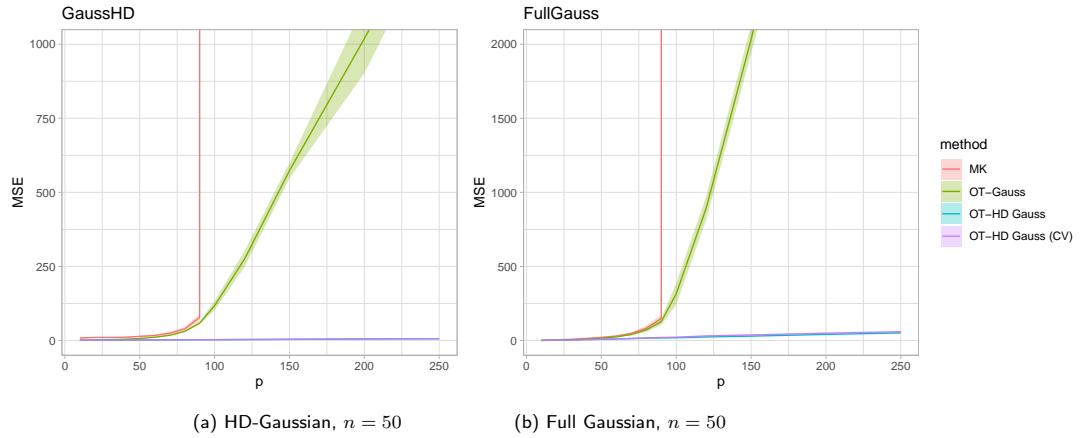


Figure 3: Mean squared error between the Monge map estimated by the compared methods and the actual transport on a test data set, with a fixed sample size  $n = 50$  and where the dimension  $p$  of the observation space varies: (a) with simulated HD-Gaussian distributions and (b) with simulated full Gaussian distributions. Results are averaged over 25 replications.

from a source measure  $\mu_1$  and the other from a destination measure  $\mu_2$ , both being either Gaussian or HD-Gaussian distributions. Then, each cloud was split in train and test. We used the source-train and the destination-train to estimate the Monge map between  $\mu_1$  and  $\mu_2$ , then we transported the source-test with the oracle Monge map as well as with the Monge maps estimated by all methods. Finally the mean squared error (MSE) between the test-transported points (oracle vs. estimated) was computed. The panels (a) and (b) of Figure 3 present the performance evolution of the best OT methods according to the space dimensionality (the sample size is fixed to  $n = 50$ ), for both the HD-Gaussian and full Gaussian scenarios. In both scenarios, one can first notice that MK-dist fails to compute the transport map in dimension higher than 80. After this dimension  $p = 80$ , one can also observe a rapid deterioration in performance of OT-Gauss, even with a numerical regularization. Conversely, the OT-HDGauss and OT-HDGAuss (CV) approaches show once again a good robustness in performance when the space dimensionality increases. Figure 4 presents the results of the same 4 OT methods when the sample size  $n$  varies and for a fixed dimensionality  $p = 100$  of the observation space. One can observe similar results here, and this for both simulated scenarios: MK-dist fails to compute the transport map for  $n$  smaller than 120 and this sample size is also a breakpoint for the performance of OT-Gauss. Here again, OT-HDGauss and OT-HDGAuss (CV) turn out to be robust in performance against the sample size, even when  $n \ll p$ .

### 3.4 Transport of non Gaussian distributions

This last experiment focuses on the transport of non Gaussian distributions. The aim here is to study the robustness of our approach to deviation from the Gaussian assumption and to compare

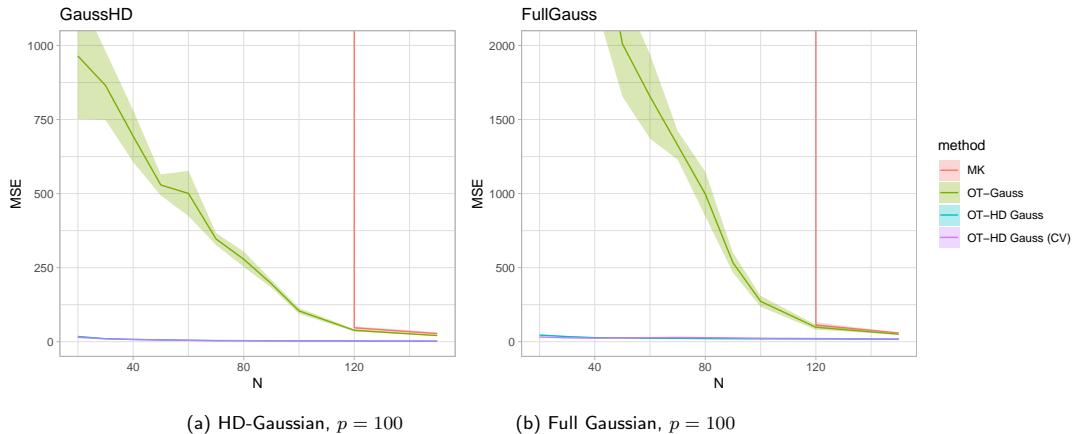


Figure 4: Mean squared error between the Monge map estimated by the compared methods and the actual transport on a test data set, with a fixed dimension  $p = 100$  of the observation space and where the sample size  $n$  varies: (a) with simulated HD-Gaussian distributions and (b) with simulated full Gaussian distributions. Results are averaged over 25 replications.

with other transport approaches (**MK-dist** and **MI-dist**) that are model-free, in the context of high-dimensional spaces. To this end, we simulated a point cloud from a source distribution either multivariate skew Normal or Student distribution for different sample sizes and varying dimensions of the observation space. Figures 10 and 11 of Appendix D present pairs plots of simulated data from these non Gaussian distributions. After splitting the source cloud into train and test, we transported the source-train with a fixed linear map in such a way to leave the transported points centered at the origin and used the source-train and transported source-train in order to estimate the Monge maps with the three methods. As before, the MSE between the transported source-test (oracle vs. estimated) was computed. Figures 5 and 6 present the performance evaluations of the same OT methods for multivariate Skew Normal and Student distributions respectively according to the space dimensionality  $p$  and the sample size  $n$ . The performances of **MK-dist** and **OT-Gauss** are similar to the Gaussian case (previous experiment). Even though its robustness is less impressive than in the Gaussian case, **OT-HDGauss** performs here also quite well in general even though the distributions clearly differ from the Gaussian one, and in any case clearly outperforms all tested OT methods.

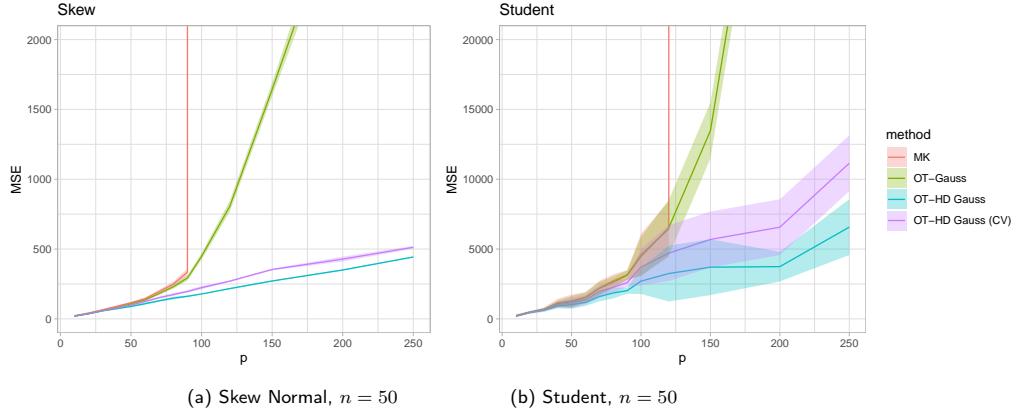


Figure 5: Mean squared error between the Monge map estimated by the compared methods and the actual transport on a test data set, with a fixed sample size  $n = 50$  and where the dimension  $p$  of the observation space varies: a) with simulated Skew Normal distributions and (b) with Student distributions. Results are averaged over 25 replications.

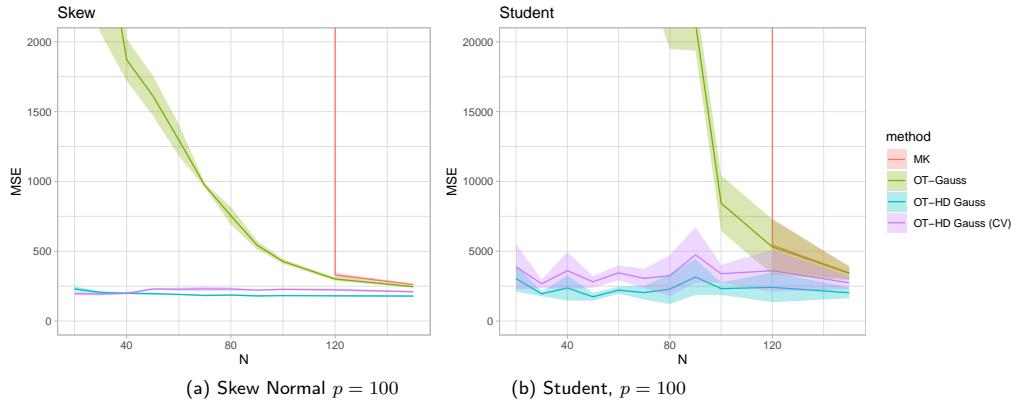


Figure 6: Mean squared error between the Monge map estimated by the compared methods and the actual transport on a test data set, with a fixed dimension  $p = 100$  of the observation space and where the sample size  $n$  varies: a) with simulated Skew Normal distributions and (b) with Student distributions. Results are averaged over 25 replications.

## 4 Conclusion and discussion

This work has focused on the optimal transport of high-dimensional Gaussian (HD-Gaussian) distributions, induced by the probabilistic PCA (PPCA) model. In particular, we exhibited of a closed-form expression of the Wasserstein distance between two HD-Gaussian distributions, with an efficient and robust calculation procedure based on a low-dimensional subspace decomposition, and this without relying on data projections. This result also generalizes previous state-of-the-art results which considered Gaussian distributions with similar covariance orientations or structures. Furthermore, we provided a closed-form expression of the Monge map for the transport of a HD-Gaussian distribution on another one, involving an exact calculation of both the square-root and the inverse square-root of the covariance matrix of the source distribution. This result avoids in turn many numerical drawbacks in high-dimensional practical situations and remain valid in the case of HD-Gaussian distributions with different intrinsic dimensions. These contributions are supported by numerical experiments that highlight the performance and robustness of the proposed OT-HDGauss procedure to both the dimensionality and the sample size, and this in comparison with the most recent OT approaches. The numerical experiments also showed that the analytical and numerical advantages of our approach in high dimensions allow it to also outperform model-free methods in the case of non-Gaussian distributions. Among the possible further work, it would be interesting to consider the extension of this approach to mixture models, in particular Gaussian mixture models.

## References

- Azzalini, A. (2013). *The skew-normal and related families*, volume 3. Cambridge University Press.
- Bellman, R. (1957). *Dynamic Programming*. Rand Corporation research study. Princeton University Press.
- Bishop, C. M. (1999). Variational principal components.
- Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. (2015). Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45.
- Bouveyron, C. and Brunet-Sauvad, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71:52–78.
- Bouveyron, C., Celeux, G., and Girard, S. (2011). Intrinsic dimension estimation by maximum likelihood in isotropic probabilistic pca. *Pattern Recognition Letters*, 32(14):1706–1713.
- Bouveyron, C., Celeux, G., Murphy, T. B., and Raftery, A. E. (2019). *Model-based clustering and classification for data science: with applications in R*, volume 50. Cambridge University Press.
- Bouveyron, C., Girard, S., and Schmid, C. (2007a). High-dimensional data clustering. *Computational statistics & data analysis*, 52(1):502–519.
- Bouveyron, C., Girard, S., and Schmid, C. (2007b). High-dimensional discriminant analysis. *Communications in StatisticsâTheory and Methods*, 36(14):2607–2623.
- Bouveyron, C., Latouche, P., and Mattei, P.-A. (2020). Exact dimensionality selection for bayesian pca. *Scandinavian Journal of Statistics*, 47(1):196–211.

- Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate behavioral research*, 1(2):245–276.
- Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.
- Dowson, D. and Landau, B. (1982). The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455.
- Dudley, R. M. (1969). The speed of mean glivenko-cantelli convergence. *The Annals of Mathematical Statistics*, 40(1):40–50.
- Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. (2021). Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8.
- Flury, B. N. and Gautschi, W. (1986). An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form. *SIAM Journal on Scientific and Statistical Computing*, 7(1):169–184.
- Fournier, N. and Guillin, A. (2015). On the rate of convergence in wasserstein distance of the empirical measure. *Probability theory and related fields*, 162(3):707–738.
- Giraud, C. (2021). *Introduction to high-dimensional statistics*. Chapman and Hall/CRC.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.
- Higham, N. J. (2008). *Functions of Matrices*. Society for Industrial and Applied Mathematics.
- Josse, J. and Husson, F. (2012). Selecting the number of components in principal component analysis using cross-validation approximations. *Computational Statistics & Data Analysis*, 56(6):1869–1879.
- Kantorovich, L. V. (1942). On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pages 199–201.
- McNicholas, P. D. and Murphy, T. B. (2008). Parsimonious gaussian mixture models. *Statistics and Computing*, 18:285–296.
- Minka, T. (2000). Automatic choice of dimensionality for pca. *Advances in neural information processing systems*, 13.
- Muzellec, B. and Cuturi, M. (2019). Subspace detours: Building transport plans that are optimal on subspace projections. *Advances in Neural Information Processing Systems*, 32.
- Nguyen, K. and Ho, N. (2024). Energy-based sliced wasserstein distance. *Advances in Neural Information Processing Systems*, 36.

- Paty, F.-P. and Cuturi, M. (2019). Subspace robust wasserstein distances. In *International conference on machine learning*, pages 5072–5081. PMLR.
- Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.
- Santambrogio, F. (2015). Optimal transport for applied mathematicians. *Birkhäuser, NY*, 55(58-63):94.
- Takatsu, A. (2011). Wasserstein geometry of gaussian measures.
- Tipping, M. E. and Bishop, C. M. (1999a). Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482.
- Tipping, M. E. and Bishop, C. M. (1999b). Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622.
- Villani, C. et al. (2009). *Optimal transport: old and new*, volume 338. Springer.
- Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press.

## Appendix

### A Details about the competitors used in numerical experiments

We provide below more details about the methods used as competitors in the numerical experiments:

- OT-Gauss: classical  $W_2^2$ -distance and Monge map computations between 2 full Gaussians, using respectively Eq. (2) and Eq. (3). The computation of the square-root matrices is based on a Schur decomposition (`sqrtm` function in R) and the inverse of covariance matrices is performed using the Moore-Penrose generalized inverse (`ginv` function).
- OT-GaussReg: classical  $W_2^2$ -distance and Monge map computations (as above), with an additional regularization of the rank of the covariance matrices ( $\tilde{\Sigma}_i = \Sigma_i + \gamma I_p$ , where  $\gamma = 1e^{-3}$  in the experiments,  $i = 1, 2$ ),
- EMD: Earth movers distance as implemented in the Python Optimal Transport (POT Flamary et al., 2021) library (`ot.emd2` function),
- Sinkhorn: Solution of the entropic regularized optimal transport problem, as described in Cuturi (2013) and implemented in the POT library (`ot.sinkhorn2` function),
- SRW: Subspace robust Wasserstein distance of Paty and Cuturi (2019), GitHub code<sup>2</sup>,
- SWD: Sliced Wasserstein distance as implemented in POT and following Bonneel et al. (2015),
- MK-dist: Monge-Knothe transport plan and relative distance as described in Muzellec and Cuturi (2019),
- MI-dist: Monge Independent distance as described in Muzellec and Cuturi (2019),
- OT-HDGauss: the approach proposed in this work that implements the computations of the  $W_2^2$ -distance with Theorem 2.3 and the Monge map with Theorem 2.9. Additionally we denote by OT-HDGauss (CV) the version of our approach where the intrinsic dimension is selected by cross-validation as in Josse and Husson (2012).

### B Intrinsic dimension estimation

This section aims to compare the performance of methods proposed repectively by Bouveyron et al. (2011) (hereafter PPCA-ds), Josse and Husson (2012) (PCA-CV) and Cattell (1966) (Cattell) for estimating the intrinsic dimension of HD-Gaussian distributions. In order to evaluate the effect of the estimation of the intrinsic dimensions using these techniques, we measured the error made in computing the Monge map between two HD-Gaussian distributions using our approach (based on the Theorem 2.9). For this comparison, we simulated two (isotropic) HD-Gaussian distributions with intrinsic dimensions  $d_1 = d_2 = 5$ , a signal-to-noise ratio of  $\delta/\sigma^2 = 5$ , in dimensions  $p = 100$  and with varying number of observations ( $n \in 50, 100, 250$ ). The OT-HD approach was used to compute the optimal transport plan between the two simulated distributions. Figure 7 presents the mean squared errors (MSE) measured between the two distributions with OT-HD on test data for different methods for the intrinsic dimension estimation and for different sample sizes of the data used for learning the transport map. The results are averaged over 25 simulated datasets. The results clearly show that the PCA-CV approach of Josse and Husson (2012) is the most efficient one for this task and should be recommended.

---

<sup>2</sup><https://github.com/francoispierrepaly/SubspaceRobustWasserstein>

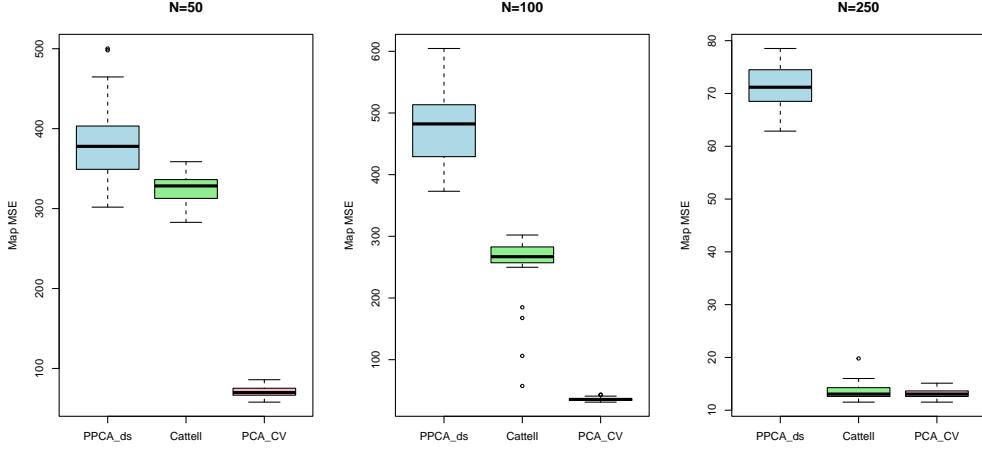


Figure 7: Effect of the choice of the intrinsic dimension estimation method on the mean squared errors of the transport of test data using OT-HD, for different sample sizes of the source distribution.

## C Computation of the $W_2^2$ -distance: additional results

We report in this section some additional results, visible in Figures 8 and 9, and comparing MI-dist and SWD with MK-dist. We are in the very same simulated scenarios described in Section 3.3 and, as it can be seen, MK-dist outperforms the two competitors.

## D Transport of Non Gaussian distributions

Figures 10 and 11 present pairs plots of simulated non Gaussian distributions used as source and target distributions in the experiment on non Gaussian data (Section 3.4).

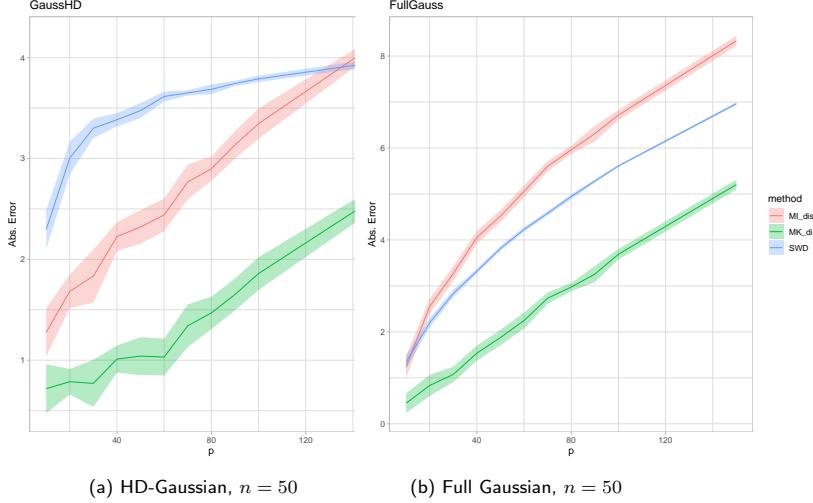
(a) HD-Gaussian,  $n = 50$ (b) Full Gaussian,  $n = 50$ 

Figure 8: Absolute value difference between the actual  $W_2^2$ -distance (computed from the true distributions) and their estimations using the compared methods, with a fixed sample size  $n = 50$  and where the dimension  $p$  of the observation space varies: (a) with simulated HD-Gaussian distributions, (b) with simulated full Gaussian distributions. Results are averaged over 25 replications.

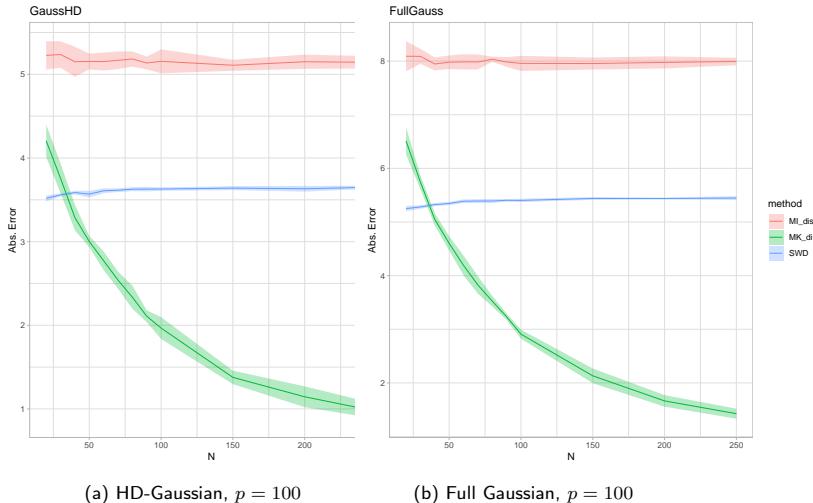
(a) HD-Gaussian,  $p = 100$ (b) Full Gaussian,  $p = 100$ 

Figure 9: Absolute value difference between the actual  $W_2^2$ -distance (computed from the true distributions) and their estimations using the compared methods, with a fixed dimension  $p = 100$  of the observation space and where the sample size  $n$  varies: (a) with simulated HD-Gaussian distributions, (b) with simulated full Gaussian distributions. Results are averaged over 25 replications.

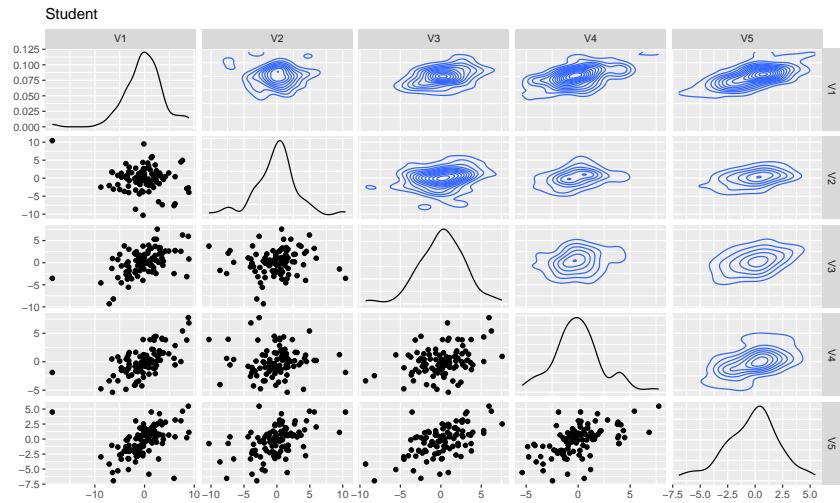


Figure 10: Pairs plot of the simulated Student distribution ( $p = 5$ ) used for the experiment on non Gaussian data.

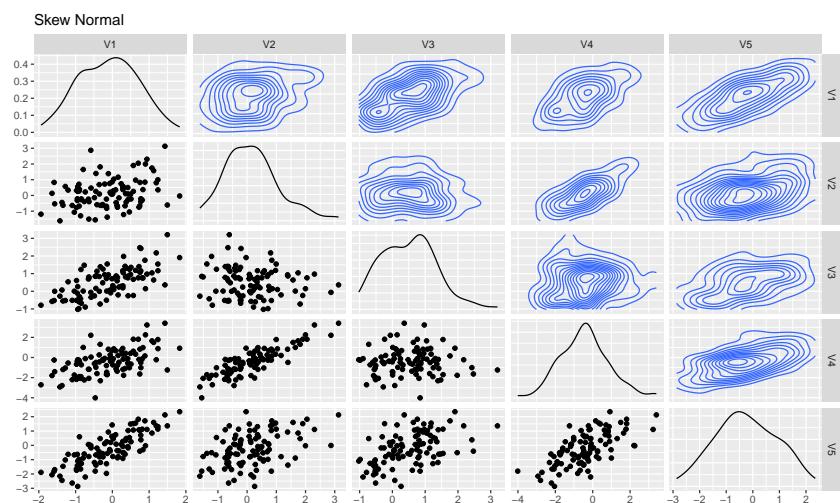


Figure 11: Pairs plot of the simulated Skew Normal distribution ( $p = 5$ ) used for the experiment on non Gaussian data.

## Luca Scrucca

### *Material list:*

Scrucca L. (2025) A model-based clustering approach for bounded data using transformation-based Gaussian mixture models. *Journal of Classification*.



## A Model-Based Clustering Approach for Bounded Data Using Transformation-Based Gaussian Mixture Models

Luca Scrucca<sup>1</sup>

Accepted: 9 June 2025

© The Author(s) under exclusive licence to The Classification Society 2025

### Abstract

The clustering of bounded data presents unique challenges in statistical analysis due to the constraints imposed on the data values. This paper introduces a novel method for model-based clustering specifically designed for bounded data. Building on the transformation-based approach to Gaussian mixture density estimation introduced by Scrucca (*Biometrical Journal*, 61(4), 873–888, 2019), we extend this framework to develop a probabilistic clustering algorithm for data with bounded support that allows for accurate clustering while respecting the natural bounds of the variables. In our proposal, a flexible range-power transformation is employed to map the data from its bounded domain to the unrestricted real space, hence enabling the estimation of Gaussian mixture models in the transformed space. Despite the close connection to density estimation, the behavior of this approach has not been previously investigated in the literature. Furthermore, we introduce a novel measure of clustering uncertainty, the normalized classification entropy (NCE), which provides a general and interpretable measure of classification uncertainty. The performance of the proposed method is evaluated through real-world data applications involving both fully and partially bounded data, in both univariate and multivariate settings, showing improved cluster recovery and interpretability. Overall, the empirical results demonstrate the effectiveness and advantages of our approach over traditional and advanced model-based clustering techniques that rely on distributions with bounded support.

**Keywords** Model-based clustering · Bounded data · Gaussian mixture models · Data transformation · Expectation-Maximization algorithm · Clustering uncertainty · Normalized classification entropy

### 1 Introduction

Clustering is a fundamental task in data analysis, aiming to identify natural groupings or patterns within a dataset and thus facilitate the discovery of underlying structures and relationships among data points. While numerous model-based clustering methods exist for unconstrained data, either having symmetric or asymmetric distributions, the clustering of

---

Luca Scrucca  
luca.scrucca@unibo.it

<sup>1</sup> Department of Statistical Sciences, University of Bologna, Via Belle Arti 41, Bologna 40126, Italy

bounded data presents unique challenges that require specialized approaches. Data with bounded support frequently arise in various fields, including economics, biology, environmental science, and social sciences. Examples include percentages, proportions, non-negative variables, e.g., arising from physical measurements, and any metric naturally limited to a fixed range. Traditional clustering methods often struggle when applied to bounded data due to their inability to adequately account for the constraints imposed by the data bounds, leading to inaccurate inference and suboptimal clustering results.

Model-based clustering, which assumes that data are generated from a mixture of probability distributions, provides a principled and flexible framework for clustering. Gaussian mixture models (GMMs) assume the data are generated from a mixture of Gaussian distributions, with each component representing a distinct cluster. They are widely used due to their flexibility to model complex data structures and their adaptability to a variety of clustering problems.

However, the direct application of GMMs to bounded data is problematic due to the inherent constraints on data values. In this context, alternative model-based approaches that account for natural bounds in the data can be adopted by assuming bounded distributions for the component densities.

For instance, in the case of positive, right-skewed data, a mixture of gamma distributions (John, 1970) provides a natural alternative. Bagnato and Punzo (2013) and Young et al. (2019) proposed a mixture of univariate unimodal gamma densities, while Wiper et al. (2001) discussed a similar model in a Bayesian framework, although also restricted to univariate data. Another potential family of distributions is introduced in Karlis and Santourian (2009), who employed a mixture of normal inverse-Gaussian distributions. This model extends naturally to the multivariate case, allowing for skewness, fat tails and, as by-product, can handle variables bounded at zero.

For data bounded at both the lower and upper limits, Bagnato and Punzo (2013) proposed the mixture of univariate unimodal beta densities. Building on this, Dean and Nugent (2013) developed a model-based clustering procedure for data confined within the unit hypercube using a mixture of univariate unimodal beta distributions. However, their approach in the multivariate case is constrained by the assumption of conditional independence, whereby variables are considered conditionally independent given component membership.

A different approach was introduced in speech processing, where a bounded Gaussian mixture model (BGMM; Hedelin & Skoglund, 2000) was employed and later extended to a bounded generalized Gaussian mixture model (BGGMM; Lindblom & Samuelsson, 2003). The BGGMM encompasses several models, including the standard GMM and BGMM, as special cases. These methods rely on truncated GMMs, where the unbounded Gaussian densities are multiplied by an indicator function that equals 1 if the component densities lie within the bounded data region and 0 otherwise; then, a subsequent normalization step ensures a proper marginal density distribution.

A common alternative to modeling bounded distributions directly is the use of data transformation techniques. Transforming bounded data onto an unbounded scale enables the application of standard statistical methods, then followed by an inverse transformation to express the results obtained in the original data space. For instance, transformations such as the Box-Cox (Lo & Gottardo, 2012) and Manly (Zhu & Melnykov, 2018) transformations have been employed to manage skewness in component distributions.

More recently, Gallaugher et al. (2020) compared the performance of Gaussian mixtures in handling skewed data or outliers with that of mixtures of skewed distributions, such as the variance-gamma distribution (VG; S.M. McNicholas et al., 2017) and the generalized hyperbolic distribution (GH; Browne & McNicholas, 2015). They also examined mixtures

incorporating transformations aimed at achieving near-normality, including the power transformation (Yeo & Johnson, 2000) and the Manly transformation (Manly, 1976). Their results indicate that no single method consistently outperforms the others, with the optimal choice depending on the specific characteristics and context of the data being analyzed.

Most of the research on data transformation focuses on skewed data, where the main goal is to reduce skewness, making the data more symmetric and closer to a Gaussian distribution. However, if skewness varies across different clusters or subsets of the data, applying a single global transformation might not be appropriate. In contrast, transformations for bounded data aim to convert data that is constrained within a specific range into an unbounded form, allowing for more flexible modeling. The underlying assumption is that a single transformation can be applied coordinate-wise to the entire dataset, as the bounding is consistent across all observations.

In this paper, we propose a novel model-based clustering method specifically designed for bounded data. Our approach builds upon the range-power transformation framework for Gaussian mixtures proposed by Scrucca (2019). The method involves mapping bounded data into an unbounded space, where standard Gaussian mixture models can be applied, followed by an inverse transformation to recover the clustering results in the original bounded space. While the original framework was developed for density estimation, we extend it to the clustering setting. To the best of our knowledge, no previous work has directly addressed clustering within this framework, despite the close connection to density estimation. We also introduce a novel measure of clustering uncertainty, called normalized classification entropy, which provides a general and interpretable measure of classification uncertainty that can be applied to model-based clustering approaches more broadly. The proposed approach has proven effective in real-data applications, offering a general and flexible framework for clustering bounded data.

In Section 2, we first review the model-based approach to clustering, with particular emphasis on the Gaussian mixture model (GMM). The proposed approach is then presented, introducing the range-power transformation and its integration within the finite mixture model framework. We discuss maximum likelihood estimation via the EM algorithm, as well as model selection and methods for assessing clustering and classification uncertainty in this specific context. Section 3 applies the method to real-data examples, covering both fully and partially bounded data in univariate and multivariate contexts. We compare our method against both the standard GMM and more advanced model-based clustering techniques that incorporate distributions with bounded support specific to the analyzed variables. Finally, Section 4 summarizes the main contributions of this work, discusses its strengths and limitations, and outlines potential directions for future research.

## 2 Methods

### 2.1 Finite Mixture Modeling for Clustering

Consider a multivariate dataset  $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$  of  $n$  observations, where each observation  $\mathbf{x}_i$  is drawn from a  $d$ -dimensional random vector  $\mathbf{x}$  with unbounded support  $\mathcal{S}_{\mathcal{X}} \equiv \mathbb{R}^d$ .

In model-based clustering, we aim to partition the observations into  $G$  distinct groups or clusters by typically employing a finite mixture model (FMM; McLachlan & Peel, 2000; McLachlan et al., 2019). Within this framework, each mixture component is directly

associated with a cluster, effectively representing a distinct grouping of the data. In its general form, a FMM can be expressed as:

$$f(\mathbf{x}; \boldsymbol{\Psi}) = \sum_{k=1}^G \pi_k f_k(\mathbf{x}; \boldsymbol{\theta}_k), \quad (1)$$

where  $\pi_k$  represents the mixing proportions or weights, subject to the constraints  $\pi_k > 0$  and  $\sum_{k=1}^G \pi_k = 1$ , and  $f_k(\mathbf{x}; \boldsymbol{\theta}_k)$  denotes the multivariate density function of the  $k$ th component with parameters vector  $\boldsymbol{\theta}_k$  ( $k = 1, \dots, G$ ). If the density function  $f_k()$  is usually assumed to be known, the parameters of the FMM,  $\boldsymbol{\Psi} = (\pi_1, \dots, \pi_{G-1}, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_G)$ , are unknown and need to be estimated from the data.

One of the earlier, and still the most popular, model for continuous data is the Gaussian mixture model (GMM), which is obtained from Eq. 1 assuming a multivariate Gaussian distribution for each component density (Fraley & Raftery, 2002; P.D. McNicholas, 2016b; Bouveyron et al., 2019; Gormley et al., 2023; Scrucca et al., 2023). A  $G$ -component GMM can be expressed as

$$f(\mathbf{x}; \boldsymbol{\Psi}) = \sum_{k=1}^G \pi_k \phi(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (2)$$

where  $\phi(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$  denotes the multivariate Gaussian density function with mean  $\boldsymbol{\mu}_k$  and covariance matrix  $\boldsymbol{\Sigma}_k$  for the  $k$ -th mixture component.

Parsimonious covariances decomposition can be obtained by imposing constraints on the geometric characteristics, such as volume, shape, and orientation, of corresponding ellipsoids. This is achieved using the following covariance matrices eigen-decomposition (Banfield & Raftery, 1993; Celeux & Govaert, 1995):

$$\boldsymbol{\Sigma}_k = \lambda_k \mathbf{U}_k \boldsymbol{\Delta}_k \mathbf{U}_k^\top, \quad (3)$$

where  $\lambda_k = |\boldsymbol{\Sigma}_k|^{1/d}$  is a scalar controlling the volume,  $\boldsymbol{\Delta}_k$  is a diagonal matrix controlling the shape, such that  $|\boldsymbol{\Delta}_k| = 1$  and with the normalized eigenvalues of  $\boldsymbol{\Sigma}_k$  in decreasing order, and  $\mathbf{U}_k$  is an orthogonal matrix of eigenvectors of  $\boldsymbol{\Sigma}_k$  that controls the orientation. Further details on the resulting 14 different models can be found in Scrucca et al. (2023, Sec. 2.2.1).

Maximum likelihood estimation of GMM parameters,  $\boldsymbol{\Psi} = (\pi_1, \dots, \pi_{G-1}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_G)$ , is commonly carried out using the expectation-maximization (EM) algorithm (Dempster et al., 1977). Let  $z_i$  denote the membership multinomial latent variable, with  $z_{ik} = 1$  if observation  $i$  belongs to cluster  $k$ , and  $z_{ik} = 0$  otherwise. The EM algorithm iteratively maximizes the complete-data log-likelihood:

$$\ell_C(\boldsymbol{\Psi}) = \sum_{i=1}^n \sum_{k=1}^G z_{ik} \left\{ \log \pi_k + \log \phi(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\},$$

by alternating between two steps, the expectation (E) step and the maximization (M) step. For a comprehensive treatment of estimation using the EM algorithm and its properties, refer to McLachlan and Krishnan (2008). Details on the M-step for the different covariance parameterizations can be found in Celeux and Govaert (1995) and Scrucca et al., (2023, Sec. 2.2.2).

## 2.2 Transformation-Based Approach for Gaussian Mixtures Clustering

A direct application of Gaussian mixtures, as formulated in Eq. 2, may lead to inaccurate density estimates and erroneous clustering assignments when some or all the variables of the dataset are bounded. To overcome these limitations, we propose employing the transformation-based approach introduced by Scrucca (2019). The core idea of this approach is to map the bounded variables to an unbounded space, where standard GMMs can be applied more effectively, and then transform the results back to the original bounded space, where clustering assignments can be straightforwardly obtained. An additional advantage of this approach is that it enables the fitting of parsimonious models by using component-covariance eigen-decompositions from Eq. 3, which are particularly beneficial in high-dimensional settings for reducing complexity and improving the interpretability of the clustering results.

Let  $\mathbf{x}$  represent a random vector from a distribution with bounded support  $\mathcal{S}_{\mathcal{X}} \subset \mathbb{R}^d$ . Consider the family of continuous monotonic transformations, denoted by  $\mathbf{y} = t(\mathbf{x}; \boldsymbol{\lambda})$ , where  $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_d]^T \in \Lambda$  represents the vector of transformation parameters, mapping the bounded support of data to an unbounded space,  $\mathcal{S}_{\mathcal{Y}} \equiv \mathbb{R}^d$ . Note that if one or more variables do not require transformation, the corresponding  $\lambda$  parameters can be set to 1 and kept fixed during the fitting process.

In the transformed space, the data can be modeled using a standard GMM:

$$h(\mathbf{y}; \boldsymbol{\Psi}) = \sum_{k=1}^G \pi_k \phi(\mathbf{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where  $\boldsymbol{\Psi}$  refers here to the parameters in the transformed scale, so  $\boldsymbol{\mu}_k$  and  $\boldsymbol{\Sigma}_k$  are, respectively, the mean vector and the covariance matrix of transformed variables in the  $k$ th component of the mixture. Then, by applying the change-of-variable theorem, the density on the original scale can be recovered as:

$$f(\mathbf{x}; \boldsymbol{\Psi}, \boldsymbol{\lambda}) = h(t(\mathbf{x}; \boldsymbol{\lambda})) \times |\mathbf{J}(t(\mathbf{x}; \boldsymbol{\lambda}))|, \quad (4)$$

where  $\mathbf{J}(t(\mathbf{x}; \boldsymbol{\lambda}))$  is the jacobian of the transformation. Since we consider coordinate-independent transformations

$$t(\mathbf{x}; \boldsymbol{\lambda}) = [t(x_1; \lambda_1), t(x_2; \lambda_2), \dots, t(x_d; \lambda_d)]^T,$$

the jacobian simplifies to a product of first derivatives:

$$\mathbf{J}(t(\mathbf{x}; \boldsymbol{\lambda})) = \prod_{j=1}^d t'(x_j; \lambda_j). \quad (5)$$

The transformation  $\mathbf{y} = t(\mathbf{x}; \boldsymbol{\lambda})$  adopted is based on the *range-power transformation* introduced in Scrucca (2019), which is suitable for both partially and completely bounded data. For a variable  $x_j \in (l_j, +\infty)$ , where  $l_j > -\infty$  represents its lower bound, the range-power transformation is defined as:

$$t(x_j; \lambda_j) = \begin{cases} \frac{(x_j - l_j)^{\lambda_j} - 1}{\lambda_j} & \text{if } \lambda_j \neq 0 \\ \log(x_j - l_j) & \text{if } \lambda_j = 0, \end{cases} \quad (6)$$

with continuous first derivative equal to

$$t'(x_j; \lambda_j) = \frac{\partial t(x_j; \lambda_j)}{\partial x_j} = (x_j - l_j)^{\lambda_j - 1} \quad (7)$$

for any  $\lambda_j \in \Lambda$  ( $j = 1, \dots, d$ ).

If a variable  $x_j \in (l_j, u_j)$ , where  $l_j > -\infty$  and  $u_j < +\infty$  represent the lower and upper bounds, respectively, the range-power transformation is defined as:

$$t(x_j; \lambda_j) = \begin{cases} \frac{\left(\frac{x_j - l_j}{u_j - x_j}\right)^{\lambda_j} - 1}{\lambda_j} & \text{if } \lambda_j \neq 0 \\ \log\left(\frac{x_j - l_j}{u_j - x_j}\right) & \text{if } \lambda_j = 0, \end{cases} \quad (8)$$

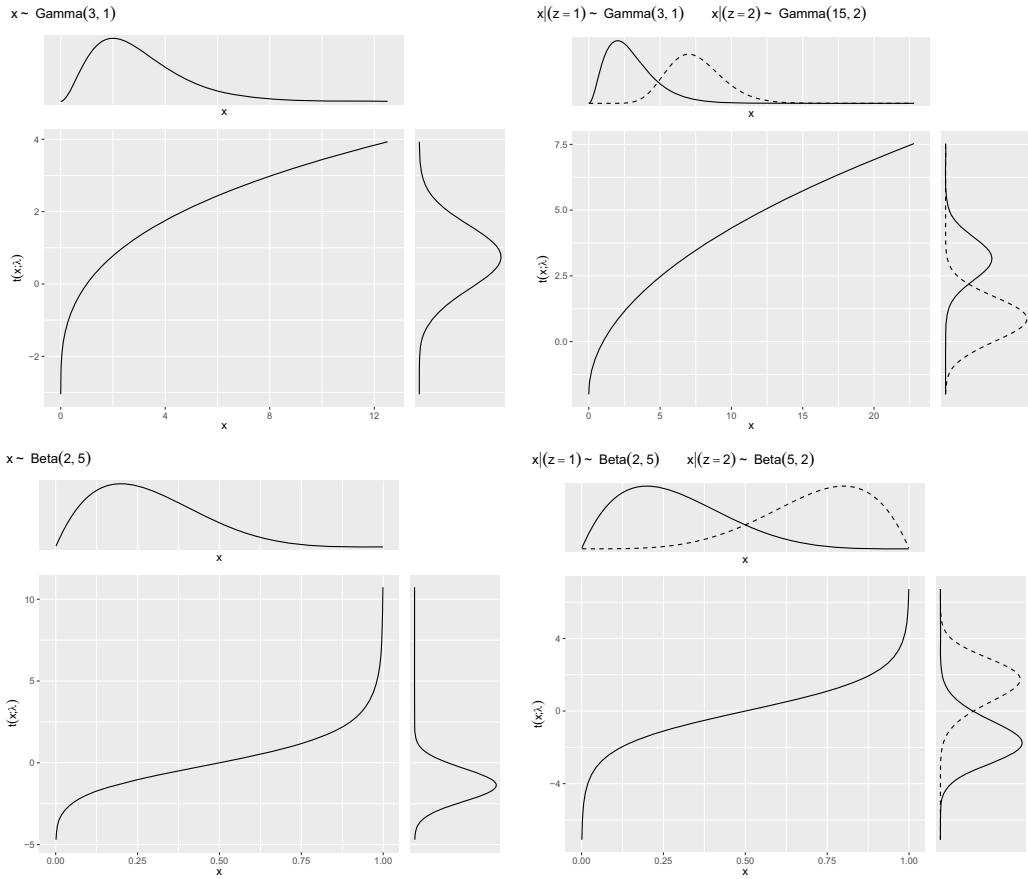
with continuous first derivative given by

$$t'(x_j; \lambda_j) = \frac{\partial t(x_j; \lambda_j)}{\partial x_j} = \begin{cases} \left(\frac{x_j - l_j}{u_j - x_j}\right)^{\lambda_j-1} \frac{u_j - l_j}{(u_j - x_j)^2} & \text{if } \lambda_j \neq 0 \\ \frac{1}{x_j - l_j} + \frac{1}{u_j - x_j} & \text{if } \lambda_j = 0. \end{cases} \quad (9)$$

Equations 7 and 9 can be used to compute the jacobian in Eq. 5. Together with the range-power transformations in Eqs. 6 and 8, these allow for the estimation of the mixture density in Eq. 4. However, both the unknown mixture parameters  $\Psi$  and the transformation parameters  $\lambda$  need to be estimated, as discussed in Section 2.3.

The range-power transformation was selected for its flexibility in accommodating both lower- and upper-bounded data while preserving interpretability. Alternative families of transformations could also be considered. For instance, the Manly (1976) transformation can handle both positive and negative values, although it does not naturally account for bounded support. The Yeo and Johnson (2000) transformation extends the Box-Cox approach to allow for both positive and negative values, but it similarly does not accommodate data with explicit bounds. In contrast, the range-power transformation is specifically designed for bounded data, making it particularly well-suited for the settings considered in this work.

Figure 1 illustrates the behavior of the range-power transformation on selected single- and two-component mixtures for univariate bounded data. Each panel shows the curve of the optimal transformation along with the corresponding marginal densities on both the original and transformed scales. The left panels contain the transformations for single-component distributions. In particular, the top-left panel presents the transformation for a lower-bounded Gamma(3, 1) distribution, which results in a smooth, monotonic transformation. The bottom-left panel shows the transformation for a Beta(2, 5) distribution, which is bounded on both sides. The transformation is nonlinear, with stronger effects near the boundaries due to the bounded support. The right panels display the transformations for two-component mixtures of bounded distributions. The top-right panel shows the transformation for a mixture of two Gamma distributions, namely Gamma(3, 1) and Gamma(15, 2). The bimodal structure of the mixture is evident from the density plot, and the transformation effectively reflects the underlying mixture components. The bottom-right panel illustrates the transformation for a mixture of Beta(2, 5) and Beta(5, 2) distributions. The bimodal nature is clearly visible, and the transformation adapts to the complexity of the mixture structure. These examples demonstrate the flexibility of the range-power transformation approach in handling different types of bounded data, including single and mixture distributions, while preserving the main clustering structure, if any, of the data.



**Fig. 1** Plots of range-power transformations for single-component distributions (left panels) and two-component mixtures (right panels) for univariate bounded data. The top panels show data with only a lower bound, while the bottom panels show data with both lower and upper bounds. Each panel displays the curve of the optimal transformation and the corresponding marginal densities on both the original and transformed scales

### 2.3 Maximum Likelihood Estimation via the EM Algorithm

Recalling the density in Eq. 4, the log-likelihood of the observed data can be expressed as:

$$\ell(\Psi, \lambda) = \sum_{i=1}^n \log \left( \sum_{k=1}^G \pi_k \phi(t(\mathbf{x}_i; \lambda); \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \times |\mathbf{J}(t(\mathbf{x}_i; \lambda))| \right).$$

Maximum likelihood estimation can be pursued via the EM algorithm by maximizing the complete-data log-likelihood:

$$\ell_C(\Psi, \lambda) = \sum_{i=1}^n \sum_{k=1}^G z_{ik} \{ \log \pi_k + \log \phi(t(\mathbf{x}_i; \lambda); \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \log |\mathbf{J}(t(\mathbf{x}_i; \lambda))| \},$$

where  $z_i$  is the latent variable for cluster membership described in Section 2.1.

At iteration  $m$  of the EM algorithm, the conditional expectation of the complete-data log-likelihood given the observed data and the current parameter values can be expressed as:

$$Q(\Psi, \lambda; \Psi^{(m)}, \lambda^{(m)}) = \sum_{i=1}^n \sum_{k=1}^G \hat{z}_{ik}^{(m+1)} \{ \log \pi_k + \log \phi(t(\mathbf{x}_i; \lambda); \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \log |\mathbf{J}(t(\mathbf{x}_i; \lambda))| \},$$

where  $\hat{z}_{ik}^{(m+1)} = \mathbb{E}(I(z_i = k)|t(\mathbf{x}_i; \hat{\lambda}^{(m)}), \Psi^{(m)})$ . Therefore, in the E-step the posterior probabilities are updated using

$$\hat{z}_{ik}^{(m+1)} = \frac{\hat{\pi}_k^{(m)} \phi\left(t(\mathbf{x}_i; \hat{\lambda}^{(m)}); \hat{\mu}_k^{(m)}, \hat{\Sigma}_k^{(m)}\right)}{\sum_{g=1}^G \hat{\pi}_g^{(m)} \phi\left(t(\mathbf{x}_i; \hat{\lambda}^{(m)}); \hat{\mu}_g^{(m)}, \hat{\Sigma}_g^{(m)}\right)}.$$

In the M-step, the parameters  $(\Psi, \lambda)$  are updated by maximizing the  $Q$ -function, given the previous values of the parameters and the updated posterior probabilities. Following the expectation-conditional-maximization (ECM) algorithm introduced by Meng and Rubin (1993), this maximization is carried out in two steps. In the first step, the updated value  $\hat{\lambda}^{(m+1)}$  is computed by maximizing the  $Q$ -function with respect to  $\lambda$ . Since no closed-form solution is available, this requires numerical optimization, such as a Newton-type algorithm or a related method. Our implementation uses the L-BFGS-B method (Byrd et al., 1995) available in the `optim()` function for the R statistical software (R Core Team, 2025). The remaining parameters are then obtained as in standard EM algorithm but accounting for the updated transformation parameters  $\hat{\lambda}^{(m+1)}$ , namely

$$\hat{\pi}_k^{(m+1)} = \frac{\sum_{i=1}^n \hat{z}_{ik}^{(m+1)}}{n} \quad \text{and} \quad \hat{\mu}_k^{(m+1)} = \frac{\sum_{i=1}^n \hat{z}_{ik}^{(m+1)} t(\mathbf{x}_i; \hat{\lambda}^{(m+1)})}{\sum_{i=1}^n \hat{z}_{ik}^{(m+1)}}.$$

The update formula for the covariance matrices depends on the assumed eigen-decomposition model. In the most general case of unconstrained covariance matrices, i.e., the VVV model in `mclust` nomenclature (see Scrucca et al. (2023), Table 2.1), we have

$$\hat{\Sigma}_k^{(m+1)} = \frac{\sum_{i=1}^n \hat{z}_{ik}^{(m+1)} \left( t(\mathbf{x}_i; \hat{\lambda}^{(m+1)}) - \hat{\mu}_k^{(m+1)} \right) \left( t(\mathbf{x}_i; \hat{\lambda}^{(m+1)}) - \hat{\mu}_k^{(m+1)} \right)^\top}{\sum_{i=1}^n \hat{z}_{ik}^{(m+1)}}.$$

The initialization of the EM algorithm introduced above requires initial estimates of optimal marginal transformation parameters and an initial partition of the data. The former can be obtained by numerically maximizing the marginal log-likelihood (using the L-BFGS-B method as mentioned above), i.e., by estimating  $\lambda_j^{(0)} = \arg \max_{\lambda_j} \ell(\lambda_j)$ , where

$$\ell(\lambda_j) = \sum_{i=1}^n \log \phi(t(x_{ij}; \lambda_j); m_j, s_j^2) + \log t'(x_{ij}; \lambda_j),$$

with  $x_{ij}$  being the  $i$ th observation on variable  $j$ , so  $t(x_{ij}; \lambda_j)$  is the corresponding range-power transformation for parameter  $\lambda_j$ , with mean  $m_j$  and variance  $s_j^2$  ( $j = 1, \dots, d$ ). The initial partition can then be obtained using the final classification from a  $k$ -means algorithm on the range-power transformed variables. Alternatively, partitions obtained from model-based agglomerative hierarchical clustering (MBAHC; Scrucca & Raftery, 2015) can be used. This initial partition of data points is used to start the algorithm from the M-step. Finally, the EM algorithm is stopped when the log-likelihood improvement falls below a specified tolerance value or a maximum number of iterations is reached.

Model selection in finite mixture modeling is typically carried out using information criteria, such as the Bayesian information criterion (BIC; Schwarz, 1978), or the integrated

complete-data likelihood criterion (ICL; Biernacki et al., 2000). Both criteria penalize model complexity, favoring more parsimonious models unless the additional parameters are justified by a significant improvement in the likelihood. Additionally, ICL introduces an extra penalization for clustering overlap, promoting models that produce well-separated clusters.

Lastly, we note that estimating the  $\lambda$  parameters introduces additional computational overhead to the fitting procedure, primarily due to the increased complexity of the likelihood surface rather than the number of parameters alone. The number of EM iterations required for convergence — and its impact on runtime — depends on problem-specific factors, as EM-based optimization is often more sensitive to the curvature of the parameter space than to its dimensionality. Empirical results suggest that while estimating  $\lambda$  parameters increases runtime, the additional cost remains reasonable in practical scenarios. For instance, on the wholesale dataset discussed in Section 3.2, the median runtime increased by a factor of approximately four (89 ms vs. 22 ms when  $\lambda$  parameters were estimated rather than fixed).

## 2.4 Clustering and Classification Uncertainty

In model-based clustering, assigning a data point to one of the identified clusters is straightforward using the maximum a posteriori (MAP) principle. According to MAP, each observation is assigned to the cluster that has the highest posterior probability.

Consider a partition of the data  $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$  into  $G$  clusters, denoted as  $\mathcal{C} = \{C_1, C_2, \dots, C_G\}$ , where  $C_k \cap C_g = \emptyset$  (for  $k \neq g$ ) and  $\bigcup_{k=1}^G C_k = \mathcal{D}$ . The MAP procedure assigns an observation  $\mathbf{x}_i$  to a cluster  $C_{\hat{k}}$  according to the rule:

$$\mathbf{x}_i \in C_{\hat{k}} \quad \text{with} \quad \hat{k} = \arg \max_{k \in \{1, \dots, G\}} \hat{z}_{ik},$$

where  $\hat{z}_{ik}$  represents the posterior probability of observation  $\mathbf{x}_i$  belonging to cluster  $k$ :

$$\hat{z}_{ik} = \frac{\hat{\pi}_k \phi\left(t(\mathbf{x}_i; \hat{\lambda}); \hat{\mu}_k, \hat{\Sigma}_k\right)}{\sum_{g=1}^G \hat{\pi}_g \phi\left(t(\mathbf{x}_i; \hat{\lambda}); \hat{\mu}_g, \hat{\Sigma}_g\right)}.$$

Hard classification based on the MAP principle assumes the most likely cluster is the correct one. However, once each data point is assigned to its most probable cluster, it becomes essential to evaluate the uncertainty of these assignments. Assessing clustering uncertainty is thus crucial, as some data points may not clearly belong to a single cluster, especially when they lie near cluster boundaries. This also allows to distinguish between well-separated clusters and those that exhibit substantial overlap.

To measure this uncertainty, we can examine the distribution of posterior probabilities across all clusters rather than focusing solely on the maximum probability. The uncertainty for each data point is captured by the following index:

$$u_i = 1 - \max_k \hat{z}_{ik}.$$

This score ranges from 0 to  $(G - 1)/G$ , with values close to 0 indicating low classification uncertainty, while values near the upper bound suggest greater uncertainty.

An alternative measure of classification uncertainty can be derived from the entropy. For each data point, we can compute

$$e_i = - \sum_{k=1}^G \hat{z}_{ik} \log(\hat{z}_{ik}),$$

which yields values in the range  $[0, \log(G)]$ , with higher entropy values indicating greater uncertainty in the assignment. For easier interpretation, a normalized version of this classification entropy, ranging between 0 and 1, can be calculated by taking the ratio of each value to its maximum:

$$e_i^* = \frac{e_i}{\log(G)} \in [0, 1]. \quad (10)$$

An overall measure of classification uncertainty can thus be obtained by averaging the normalized entropy values in Eq. 10 across all data points, yielding the normalized classification entropy (NCE):

$$\text{NCE} = \frac{1}{n} \sum_{i=1}^n e_i^* = \frac{\sum_{i=1}^n e_i}{n \log(G)}, \quad (11)$$

for  $G > 1$ , and with the implicit assumption that NCE = 0 when  $G = 1$ . The index NCE quantifies the clustering uncertainty, taking a value of zero when each data point is assigned to its respective cluster with probability 1. As uncertainty increases, the index rises accordingly, reaching its maximum value of one when  $z_{ik} = 1/G$  for all observations  $i = 1, \dots, n$ , and clusters  $k = 1, \dots, G$ .

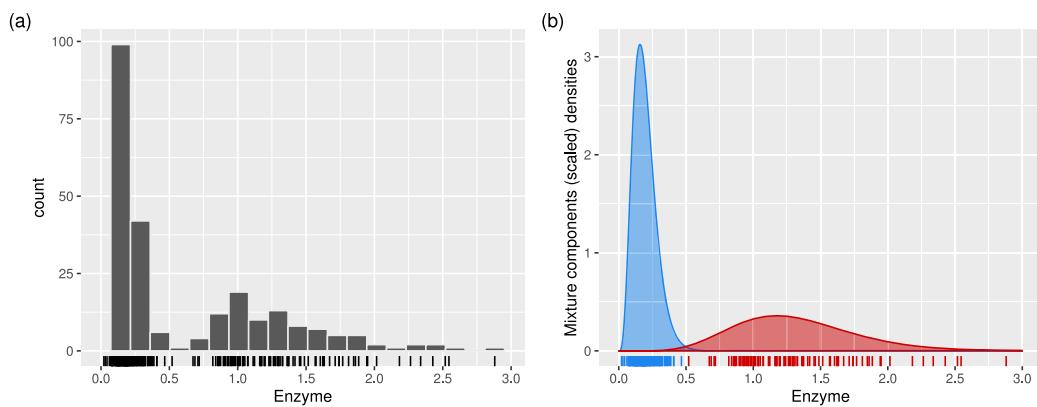
The above proposal is reminiscent of the normalized entropy criterion (NEC) introduced by Celeux and Soromenho (1996), and later improved by Biernacki et al. (1999), which also uses the classification entropy  $E = \sum_{i=1}^n e_i$  as in the numerator of Eq. 11, but applies a different denominator for scaling. Moreover, the goals of NEC and NCE are fundamentally different. NEC was proposed as a criterion for model selection, whereas our proposal aims to measure the uncertainty in the clustering partition obtained. For this reason, we caution against interpreting a very small NCE as necessarily indicative of a better clustering result. A low NCE suggests well-separated clusters with low uncertainty in the MAP assignments. On the contrary, an increase in NCE may reflect substantial overlap among clusters. While this could indicate poor clustering, it often reflects the inherent nature of the data, where the clusters are not well separated. Therefore, while NCE provides a useful measure of overall clustering uncertainty, it should not be used as the sole criterion for evaluating the quality of a clustering solution.

### 3 Applications

#### 3.1 Enzyme Data

The enzyme data set, originally described by Bechtel et al. (1993), contains enzymatic activity measurements in the blood for an enzyme involved in the metabolism of carcinogenic substances. Enzymatic activity is usually greater than zero, as some baseline activity may be observed even in healthy individuals. However, such activity can be recorded as zero when the enzyme is completely absent, fully inhibited, or even if some minimal activity is present but falls below the detection limit of a laboratory test. The analyzed data were collected from a sample of  $n = 245$  unrelated individuals. Here, the primary interest lies in identifying subgroups of slow or fast metabolizers, which serve as markers of genetic polymorphism in the general population.

This benchmark data set for mixture models is notable for containing at least two components, one of which exhibits clear skewness. Richardson and Green (1997) employed a Bayesian GMM estimated using reversible jump MCMC to analyze the distribution of



**Fig. 2** **a** Histogram of the enzyme data and **b** plot of component densities, scaled by the corresponding mixing probabilities, estimated using the GMMB model

enzymatic activity. Their analysis suggests that a model with 3 to 5 mixture components is plausible, with the three-component solution being preferred, particularly in line with a simple underlying genetic model. In contrast, Karlis and Santourian (2009) applied a normal inverse-Gaussian mixture (NIGM) and identified two distinct, non-overlapping clusters.

Figure 2a presents the histogram of the enzyme data, while Fig. 2b shows the estimated densities from the selected two-component GMMB model with unconstrained variances and a range-power transformation parameter in Eq. 6 equals to  $\hat{\lambda} = 0.3666$ .

This model, identified as the best fit according to both the BIC and ICL criteria, distinguishes between two clusters: one consisting of 152 observations characterized by a skewed distribution and enzymatic activity levels below 0.5, and a second, comprising 93 observations, forming a nearly symmetric group with enzymatic activity levels above 0.5. These results align with those reported by Karlis and Santourian (2009) and have been reproduced here using the function `MGHM()` from the **MixtureMissing** R package (Tong & Tortora, 2024).

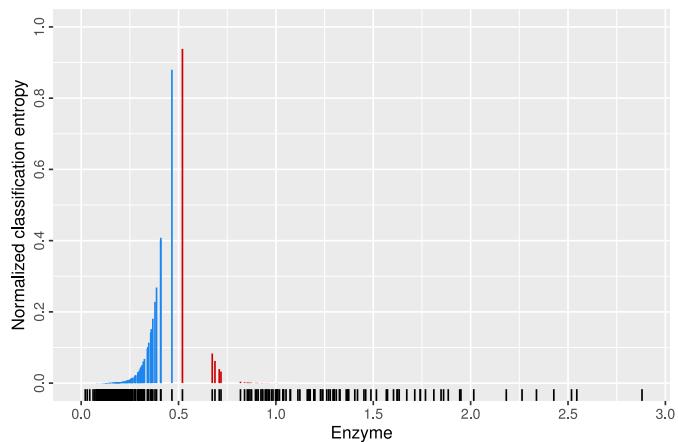
Although the GMMB model provides a better fit, as indicated by higher BIC and ICL values, and is more parsimonious with fewer parameters to estimate, it reflects a slightly larger classification uncertainty than the NIGM model, as measured by NCE. Figure 3 shows the normalized classification entropy values from Eq. 10, which represent the contribution of each data point to the overall classification uncertainty provided by NCE. As can be clearly seen, most of the contribution comes from those observations near the classification boundary at an enzymatic level of 0.5. In any case, both models represent a clear improvement over the standard GMM, as shown in Table 1.

**Table 1** Model comparison for the clustering of the enzyme data using Gaussian mixture model (GMM), mixture of normal inverse-Gaussian (NIGM) distributions, and Gaussian mixture model for bounded data (GMMB), with the number of components selected according to the BIC criterion

Model	Log-likelihood	df	BIC	ICL	NCE
GMM(V,2)	-54.6401	5	-136.7865	-148.9526	0.1109
NIGM(2)	-41.4723	9	-132.4558	-132.6530	0.0052
GMMB(V,2)	-46.1870	6	-125.3815	-127.9385	0.0208

The nomenclature (V,2) for both the GMM and GMMB models denotes two-component heteroscedastic mixture models (i.e., models with different variances)

**Fig. 3** Normalized classification entropy for each data point in the enzyme dataset showing clustering uncertainty



### 3.2 Wholesale Customer Segmentation

Customer segmentation aims to divide customers into distinct groups based on common characteristics, such as purchasing behavior. This partition helps businesses to tailor their marketing efforts, products, and services to specific customer segments, potentially maximizing customer satisfaction and profitability.

The wholesale customers dataset, freely available on the UCI Machine Learning Repository (Cardoso, 2013), is a widely used dataset to explore clustering and segmentation techniques. The dataset includes 440 observations, each representing a customer of a wholesale distributor. For each client, annual spending, in monetary units, is recorded for six product categories: fresh, milk, grocery, frozen, detergents paper, and delicatessen. Two additional categorical variables are available, corresponding to the region and customer channel. Note that, unlike other authors who have analyzed this dataset, such as Punzo and Tortora (2021), we conduct our analysis using the variables in their original scale.

Since annual expenditure cannot take negative values, variables used in the segmentation process are naturally bounded at zero. Standard clustering techniques often assume unbounded data, which is clearly not appropriate in the current context and, by failing to capture the true underlying structure, can lead to suboptimal segmentation.

Table 2 reports the results obtained by fitting the standard GMM for unbounded variables, the GMMB model proposed in this paper, and some models discussed in Punzo and Tortora (2021), namely the mixture of contaminated normal distributions (MCNM) and the mixture

**Table 2** Model comparison for the clustering of the wholesale dataset, using Gaussian mixture model (GMM), mixture of contaminated normals (MCNM), mixture of multiple scaled contaminated normals (MSCNM), and Gaussian mixture model for bounded data (GMMB), all fitted using  $G = 2$  mixture components

Model	log-likelihood	df	BIC	ICL	NCE	ER	ARI
GMM(VVV,2)	-25,069.70	55	-50,474.18	-50,495.52	0.0796	0.3091	0.1028
MCNM(VVV,2)	-24,614.60	59	-49,588.32	-49,626.70	0.1439	0.1909	0.3808
MSCNM(2)	-24,498.57	79	-49,477.99	-49,491.15	0.1290	0.1977	0.3642
GMMB(VVE,2)	-23,909.79	46	-48,099.57	-48,141.26	0.1539	0.0932	0.6585

Specification of models follow the `mclust` nomenclature (see Scrucca et al. (2023), Table 2.1), so (VVV,2) denotes a model with two mixture components and unconstrained component-covariance matrices, while model VVE indicates heteroscedastic component-covariance matrices with equal orientation

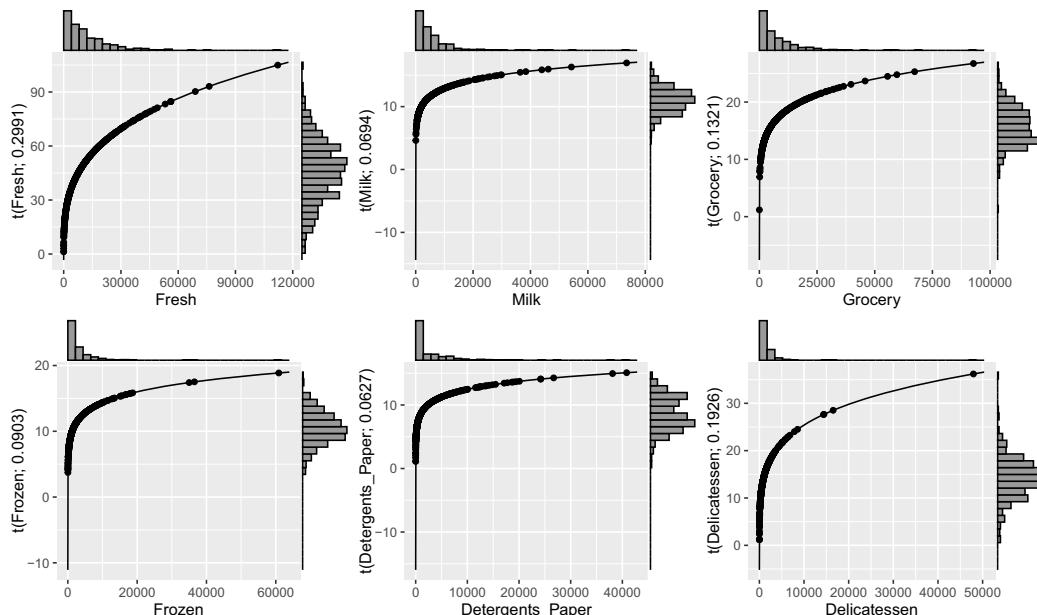
of multiple scaled contaminated normal distributions (MSCNM). For all models, the number of mixing components is fixed at  $G = 2$ . The MCNM model is fitted using the function `CNmixt()` in the **ContaminatedMixt** R package (Punzo et al., 2018, 2023), while the MSCNM is fitted using the function `mscn()` in the **MSclust** R package (Tortora et al., 2024). Models are compared using the BIC, ICL, NCE, and the adjusted Rand index (ARI; Hubert & Arabie, 1985).

Both the MCNM and MSCNM models provide significant improvements over the standard GMM, offering superior fit as evidenced by higher BIC values, and enhanced classification performance, with lower error rates (ER) and higher adjusted Rand index (ARI). However, both models are outperformed by the GMMB model, which achieves higher BIC and ICL values, indicating a better overall fit, while also approximately halving the ER and doubling the ARI, further emphasizing its superior classification capabilities. As for the MCNM and MSCNM models, the classification uncertainty of GMMB, measured by NCE, is higher than that of GMM, reflecting a larger overlap among mixture components in GMMB model.

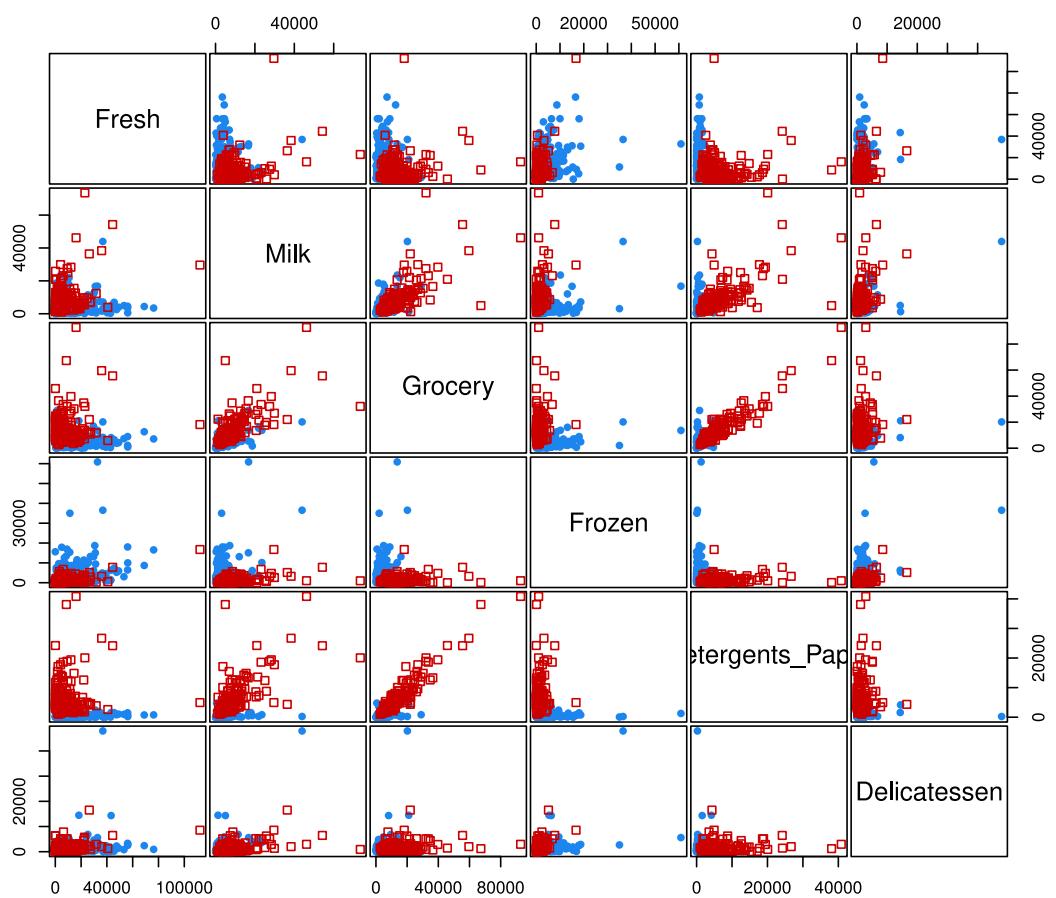
Figure 4 shows the range-power transformations in Eq. 6 for each variable in the wholesale dataset, as estimated by the GMMB model. All transformations exhibit a log-type shape, with  $\lambda$  values ranging from 0.05 to 0.3. These transformations effectively reduce the pronounced positive skewness in the marginal distributions, as illustrated in the corresponding marginal histograms.

Figure 5 presents the scatterplot matrix of the variables from the wholesale dataset, with points representing wholesale customers marked by their assigned cluster. The distributions within each segment exhibit noticeable skewness due to the zero-lower bound constraint of the variables, a characteristic effectively captured by the GMMB model used for the segmentation.

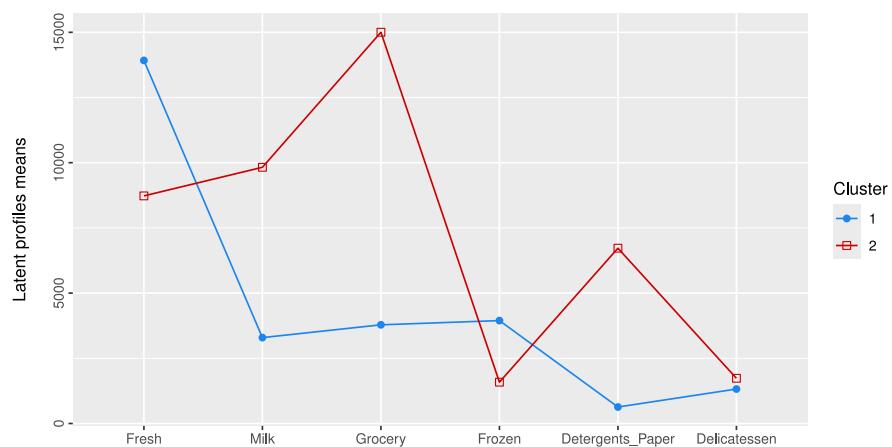
Finally, Fig. 6 shows the average latent profiles for the two identified clusters. This graph clearly highlights the key variables that distinguish the two segments of wholesale distributor



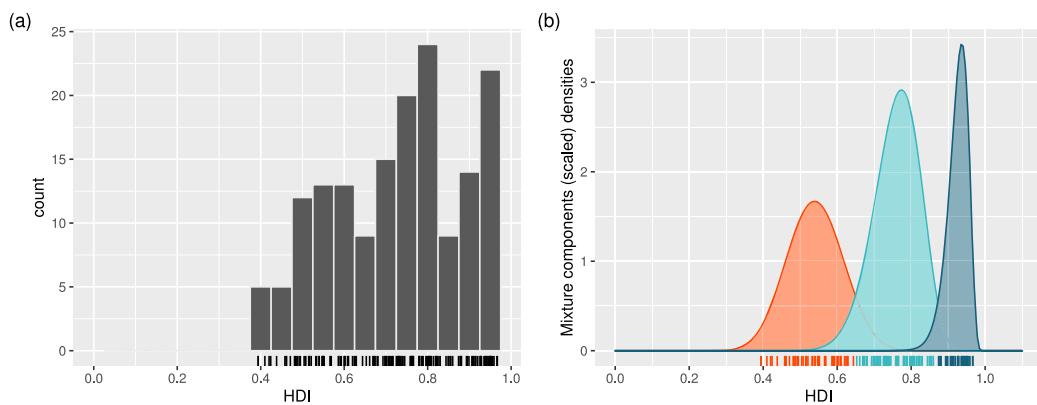
**Fig. 4** Plots displaying the selected range-power transformations  $t(X_j; \hat{\lambda})$  as a function of the original variables  $X_j$  in the wholesale dataset, with marginal histograms showing the distribution before and after the applied transformation



**Fig. 5** Scatterplot matrix of variables in the wholesale dataset, with wholesale customers marked by their cluster membership



**Fig. 6** Latent profiles plot showing the estimated cluster means from the fitted GMMB model for each variable in the wholesale dataset



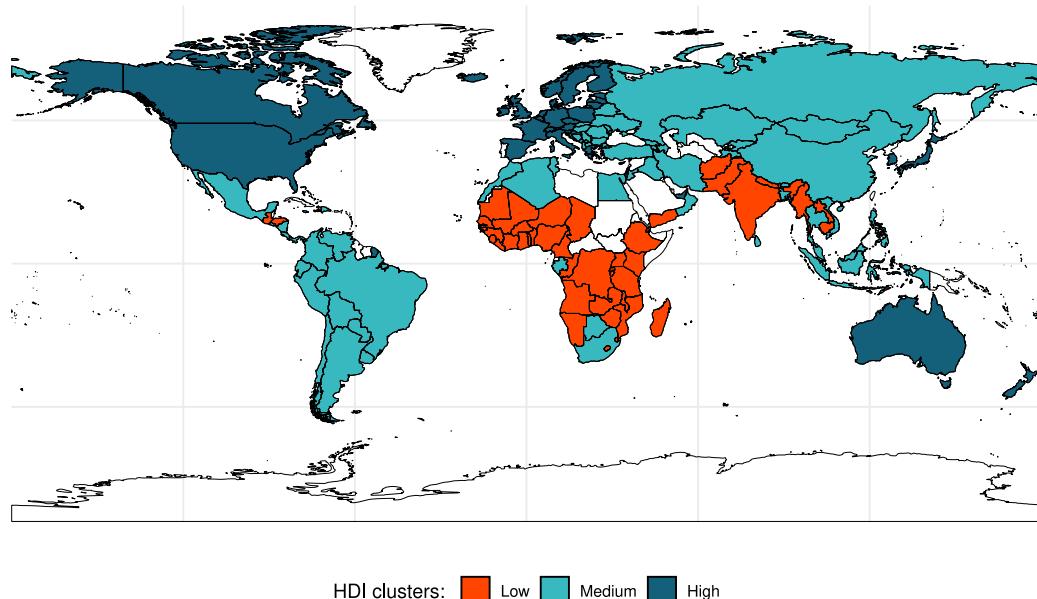
**Fig. 7** **a** Histogram of the HDI distribution for the year 2022. **b** Plot of component densities, scaled by the corresponding mixing probabilities, estimated using the GMMB model

customers, with the “delicatessen” and “frozen” product categories not playing a significant role in the segmentation.

### 3.3 Human Development Index

The Human Development Index (HDI) is a composite index provided by the United Nations Development Programme (UNDP) and designed to measure the level of human development in world countries (Herre & Arriagada, 2023).

The HDI combines three key dimensions of human development: *health*, measured by life expectancy at birth, *education*, measured by expected years of schooling (for children of school entering age) and average years of schooling (for adults aged 25 and older), and *standard of living*, measured by gross national income (GNI) per capita. These three indicators



**Fig. 8** World map showing countries clustered by HDI using the GMMB model

**Table 3** Model comparison for the clustering of the Human Development Index (HDI), using Gaussian mixture model (GMM), Beta mixture model (BMM), and Gaussian mixture model for bounded data (GMMB), with the number of components selected according to the BIC criterion

Model	log-likelihood	df	BIC	ICL	NCE
GMM(V,3)	96.6144	8	152.5775	127.8651	0.1660
BMM(3)	97.8045	8	154.9578	128.7258	0.1609
GMMB(E,3)	97.8727	7	160.1756	135.8538	0.1575

Following `mclust` nomenclature (see Scrucca et al. (2023), Table 2.1), model (V,3) indicates a model with three mixture components and varying variances, while (E,3) denotes a model with three mixture components and equal variance across components

are first normalized to have a minimum of 0 and a maximum of 1, then are combined by calculating the geometric mean. The resulting HDI value for each country represents an overall score in the range [0, 1], with higher values indicating a better level of human development.

Figure 7a shows the distribution of HDI for the year 2022, revealing a clearly multimodal pattern. HDI values span a range from approximately 0.4 to just under 1, reflecting significant variation across countries.

According to the BIC criterion, the best-fitting GMMB model comprises three components with equal variance and a range-power transformation parameter in Eq. 8 equals to  $\hat{\lambda} = -0.12$ . The corresponding component densities (scaled by their mixing probabilities) are shown in Fig. 7b. This model indicates the presence of three distinct clusters of countries, which can be broadly categorized as low, medium, and high human development clusters.

Figure 8 presents a world map with countries colored according to their respective cluster memberships. The high-HDI cluster includes mainly developed countries, such as those in North America and Europe, along with Japan, Australia, New Zealand, and the United Arab Emirates. The low-HDI cluster consists mostly of underdeveloped countries, primarily in Africa, with additional members in Asia (Afghanistan, Pakistan, Yemen, Nepal, Myanmar, Laos, Cambodia) and Central America (Guatemala, Haiti, Honduras). The remaining countries, mainly from Central and South America, Eastern Europe, and parts of Asia, fall into the medium-HDI cluster.

The selected GMMB model can be compared to the standard GMM model, and to the mixture of Beta distributions (BMM; Bagnato & Punzo, 2013; Dean & Nugent, 2013), with the number of components again selected using the BIC criterion. The results in Table 3 indicate that the GMMB model achieves a better fit, as evidenced by a higher BIC, along with a slightly reduced classification uncertainty, as shown by a lower NCE. Overall, the GMMB approach offers a reasonable and interpretable clustering solution with lowered uncertainty, further highlighting its effectiveness in this context.

## 4 Conclusion

This study aimed to develop a model-based clustering approach for bounded data, extending earlier work of Scrucca (2019) on Gaussian mixture density estimation. The primary research question focused on overcoming the unique challenges posed by bounded data, particularly by transforming bounded variables into an unbounded space to facilitate the application of GMMs. Through the proposed range-power transformation, our method provides a flexible

and interpretable clustering framework that preserves data bounds, enhancing the accuracy and applicability of GMMs for real-world bounded data scenarios.

The analyzed datasets demonstrated that this transformation-based approach offers a unified and flexible framework capable of handling diverse bounded data structures while ensuring model parsimony and interpretability. Compared to traditional GMMs and other model-based approaches using distributions with limited support, our method showed improved clustering partitions and interpretability, highlighting its relevance for practical applications involving the analysis of bounded data.

Despite the advantages mentioned above, some limitations remain. The proposed transformation-based approach requires selecting a set of appropriate global coordinate-wise transformation parameters. However, this approach may be suboptimal if different transformations are needed for different clusters or subsets of the data, a topic warranting further exploration in future research.

Additionally, the suitability of the proposal for handling heavy-tailed clusters should be investigated. This would allow to extend the transformation-based approach also in case of unbounded but non-spherical cluster distributions, hence offering an alternative approach to the several existing methods for non-Gaussian clusters (P.D. McNicholas, 2016a).

**Code and Data Availability** All the analyses have been conducted in R (R Core Team, 2025) using the `mclust` and `mclustAddons` packages (Fraley et al., 2024; Scrucca, 2025). Source code and datasets to reproduce the analyses are available in a GitHub repository at <https://github.com/luca-scr/MclustBounded>.

## Declarations

**Ethical Approval** The research study did not involve any human participants or animals.

**Conflict of Interest** The author declares no competing interests.

## References

- Bagnato, L., & Punzo, A. (2013). Finite mixtures of unimodal beta and gamma densities and the  $k$ -bumps algorithm. *Computational Statistics*, 28(4), 1571–1597. <https://doi.org/10.1007/s00180-012-0367-4>
- Banfield, J., & Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49, 803–821. <https://doi.org/10.2307/2532201>
- Bechtel, Y. C., Bonaiti-Pellie, C., Poisson, N., Magnette, J., & Bechtel, P. R. (1993). A population and family study N-acetyltransferase using caffeine urinary metabolites. *Clinical Pharmacology & Therapeutics*, 54(2), 134–141. <https://doi.org/10.1038/clpt.1993.124>
- Biernacki, C., Celeux, G., & Govaert, G. (1999). An improvement of the NEC criterion for assessing the number of clusters in a mixture model. *Pattern Recognition Letters*, 20(3), 267–272. [https://doi.org/10.1016/S0167-8655\(98\)00144-5](https://doi.org/10.1016/S0167-8655(98)00144-5)
- Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7), 719–725. <https://doi.org/10.1109/34.865189>
- Bouveyron, C., Celeux, G., Murphy, T.B., Raftery, A.E. (2019). *Model-based clustering and classification for data science: With applications in R*. Cambridge University Press.
- Browne, R. P., & McNicholas, P. D. (2015). A mixture of generalized hyperbolic distributions. *Canadian Journal of Statistics*, 43(2), 176–198. <https://doi.org/10.1002/cjs.11246>
- Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5), 1190–1208.
- Cardoso, M. (2013). *Wholesale customers*. UCI machine learning repository. <https://archive.ics.uci.edu/ml/datasets/wholesale+customers>

- Celeux, G., & Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28, 781–793. [https://doi.org/10.1016/0031-3203\(94\)00125-6](https://doi.org/10.1016/0031-3203(94)00125-6)
- Celeux, G., & Soromenho, G. (1996). An entropy criterion for assessing the number of clusters in a mixture model. *Journal of Classification*, 13(2), 195–212.
- Dean, N., & Nugent, R. (2013). Clustering student skill set profiles in a unit hypercube using mixtures of multivariate betas. *Advances in Data Analysis and Classification*, 7(3), 339–357. <https://doi.org/10.1007/s11634-013-0149-z>
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39, 1–38. <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>
- Fraley, C., Raftery, A.E., & Scrucca, L. (2024). <https://CRAN.R-project.org/package=mclust> (R package version 6.1.1)
- Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458), 611–631. <https://doi.org/10.1198/016214502760047131>
- Gallaugh, M.P.B., McNicholas, P.D., Melnykov, V., & Zhu, X. (2020). *Skewed distributions or transformations? Modelling skewness for a cluster analysis*. arXiv:2011.09152
- Gormley, I. C., Murphy, T. B., & Raftery, A. E. (2023). Model-based clustering. *Annual Review of Statistics and Its Application*, 10, 573–595. <https://doi.org/10.1146/annurev-statistics-033121-115326>
- Hedelin, P., & Skoglund, J. (2000). Vector quantization based on Gaussian mixture models. *IEEE Transactions on Speech and Audio Processing*, 8(4), 385–401. <https://doi.org/10.1109/89.848220>
- Herre, B., & Arriagada, P. (2023). The human development index and related indices: What they are and what we can learn from them. *Our World in Data*, <https://ourworldindata.org/human-development-index>
- Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2, 193–218.
- John, S. (1970). On identifying the population of origin of each observation in a mixture of observations from two gamma populations. *Technometrics*, 12(3), 565–568. <https://doi.org/10.1080/00401706.1970.10488697>
- Karlis, D., & Santourian, A. (2009). Model-based clustering with non-elliptically contoured distributions. *Statistics and Computing*, 19(1), 73–83. <https://doi.org/10.1007/s11222-008-9072-0>
- Lindblom, J., & Samuelsson, J. (2003). Bounded support Gaussian mixture modeling of speech spectra. *IEEE Transactions on Speech and Audio Processing*, 11(1), 88–99. <https://doi.org/10.1109/TSA.2002.805639>
- Lo, K., & Gottardo, R. (2012). Flexible mixture modeling via the multivariate t distribution with the box-cox transformation: An alternative to the skew-t distribution. *Statistics and Computing*, 22(1), 33–52.
- Manly, B. (1976). Exponential data transformations. *The Statistician*, 25(1), 37–42. <https://doi.org/10.2307/2988129>
- McLachlan, G. J., & Krishnan, T. (2008). *The EM algorithm and extensions* (2nd ed.). Hoboken, New Jersey: Wiley-Interscience.
- McLachlan, G. J., Lee, S. X., & Rathnayake, S. I. (2019). Finite mixture models. *Annual Review of Statistics and Its Application*, 6(1), 355–378. <https://doi.org/10.1146/annurev-statistics-031017-100325>
- McLachlan, G. J., & Peel, D. (2000). *Finite mixture models*. New York: Wiley.
- McNicholas, S.M., McNicholas, P.D., & Browne, R.P. (2017). A mixture of variance-gamma factor analyzers. S.E. Ahmed (Ed.), *Big and complex data analysis: Methodologies and applications* (pp. 369–385). Cham: Springer International Publishing.
- McNicholas, P. D. (2016). *Mixture model-based classification*. Boca Raton: Chapman & Hall/CRC.
- McNicholas, P. D. (2016). Model-based clustering. *Journal of Classification*, 33(3), 331–373. <https://doi.org/10.1007/s00357-016-9211-9>
- Meng, X.-L., & Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2), 267–278. <https://doi.org/10.1093/biomet/80.2.267>
- Punzo, A., Mazza, A., & McNicholas, P.D. (2023). <https://CRAN.R-project.org/package=ContaminatedMixt> (R package version 1.3.8)
- Punzo, A., Mazza, A., & McNicholas, P. D. (2018). ContaminatedMixt: An R package for fitting parsimonious mixtures of multivariate contaminated normal distributions. *Journal of Statistical Software*, 85(10), 1–25. <https://doi.org/10.18637/jss.v085.i10>
- Punzo, A., & Tortora, C. (2021). Multiple scaled contaminated normal distribution and its application in clustering. *Statistical Modelling*, 21(4), 332–358. <https://doi.org/10.1177/1471082X19890935>
- R Core Team (2025). Vienna, Austria. <https://www.R-project.org/>
- Richardson, S., & Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4), 731–792. <https://doi.org/10.1111/1467-9868.00095>

- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2), 461–464. <https://doi.org/10.1214/aos/1176344136>
- Scrucca, L. (2025). <https://CRAN.R-project.org/package=mclustAddons> (R package version 0.9.2)
- Scrucca, L. (2019). A transformation-based approach to Gaussian mixture density estimation for bounded data. *Biometrical Journal*, 61(4), 873–888. <https://doi.org/10.1002/bimj.201800174>
- Scrucca, L., Fraley, C., Murphy, T. B., & Raftery, A. E. (2023). *Model-based clustering, classification, and density estimation using mclust in R*. Boca Raton, FL: Chapman & Hall/CRC.
- Scrucca, L., & Raftery, A. E. (2015). Improved initialisation of model-based clustering using Gaussian hierarchical partitions. *Advances in Data Analysis and Classification*, 4(9), 447–460. <https://doi.org/10.1007/s11634-015-0220-z>
- Tong, H., & Tortora, C. (2024). <https://CRAN.R-project.org/package=MixtureMissing> (R package version 3.0.2)
- Tortora, C., Punzo, A., & Tran, L. (2024). <https://CRAN.R-project.org/package=MSclust> (R package version 1.0.4)
- Wiper, M., Insua, D. R., & Ruggeri, F. (2001). Mixtures of gamma distributions with applications. *Journal of Computational and Graphical Statistics*, 10(3), 440–454. <https://doi.org/10.1198/106186001317115054>
- Yeo, I.-K., & Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4), 954–959. <https://doi.org/10.1093/biomet/87.4.954>
- Young, D. S., Chen, X., Hewage, D. C., & Nilo-Poyanco, R. (2019). Finite mixture-of-gamma distributions: Estimation, inference, and model-based clustering. *Advances in Data Analysis and Classification*, 13(4), 1053–1082. <https://doi.org/10.1007/s11634-019-00361-y>
- Zhu, X., & Melnykov, V. (2018). Manly transformation in finite mixture modeling. *Computational Statistics & Data Analysis*, 121, 190–208. <https://doi.org/10.1016/j.csda.2016.01.015>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Pierre Mattei

*Material list:*

## Naisyin Wang

*Material list:*

## Daniel Sewell

### *Material list:*

Li H. & Sewell D.K. (2025) Model-based edge clustering for weighted networks with a noise component. Computational Statistics & Data Analysis, 209, 108172.

Arakkal A.T. & Sewell D.K. (2025) JANE: Just Another latent space NEtwork clustering algorithm. Computational Statistics & Data Analysis, 108228.



## Model-based edge clustering for weighted networks with a noise component

Haomin Li, Daniel K. Sewell \*,

Department of Biostatistics, University of Iowa, Iowa City, 52242, IA, USA



### ARTICLE INFO

**Keywords:**

Community detection  
Edge thresholding  
Latent space models  
Network analysis  
Weighted edges

### ABSTRACT

Clustering is a fundamental task in network analysis, essential for uncovering hidden structures within complex systems. Edge clustering, which focuses on relationships between nodes rather than the nodes themselves, has gained increased attention in recent years. However, existing edge clustering algorithms often overlook the significance of edge weights, which can represent the strength or capacity of connections, and fail to account for noisy edges—connections that obscure the true structure of the network. To address these challenges, the Weighted Edge Clustering Adjusting for Noise (WECAN) model is introduced. This novel algorithm integrates edge weights into the clustering process and includes a noise component that filters out spurious edges. WECAN offers a data-driven approach to distinguishing between meaningful and noisy edges, avoiding the arbitrary thresholding commonly used in network analysis. Its effectiveness is demonstrated through simulation studies and applications to real-world datasets, showing significant improvements over traditional clustering methods. Additionally, the R package “WECAN”<sup>1</sup> has been developed to facilitate its practical implementation.

### 1. Introduction

Networks arise as dominant structures in various fields, including epidemiology, sociology, biology, neuroscience, and computer science. In network analysis, entities of interest are depicted as nodes, while the connections between them are typically denoted by edges. These edges can take the form of unordered pairs, indicating undirected relationships, or ordered pairs, signifying directed relationships (Butts, 2009). One of the pivotal research areas within network analysis is clustering, of which community detection is a special case. Communities in a network are typically characterized by dense groups of nodes, exhibiting dense internal connections while maintaining sparser connections with nodes outside the group, although more broadly clusters of nodes tend to exhibit structural similarities. Network clustering serves as a cornerstone in unraveling the functionality of complex networks (Khan and Niazi, 2017).

As one of the most broadly-studied topics in network science, various algorithms and statistical models have been developed for community detection. Among them, there are algorithmic methods such as the Kernighan-Lin algorithm for graph partitioning (Kernighan and Lin, 1970), the Girvan Newman (GN) algorithm (Newman and Girvan, 2004), the spectral clustering algorithm (Ng et al., 2001); in addition, many model-based methods exist such as the latent position cluster model (Handcock et al., 2007), and the stochastic block model (Nowicki and Snijders, 2001) and their many variants. These algorithms have been widely accepted and used

\* Corresponding author.

E-mail address: [daniel-sewell@uiowa.edu](mailto:daniel-sewell@uiowa.edu) (D.K. Sewell).

<sup>1</sup> <https://github.com/HaominLi7/WECAN>.

<https://doi.org/10.1016/j.csda.2025.108172>

Received 3 April 2024; Received in revised form 20 January 2025; Accepted 12 March 2025

in various fields. However, despite advantages in interpretation and intuition in many settings, all of these models focus on grouping nodes into clusters based on similarity, but ignore the potential that lay in edge clustering.

The idea of edge-centric clustering is attractive through the following aspects. First, it is common for communities to be built based on the identities of the edges in realistic networks. For example, in a scientific collaboration network, nodes represent researchers and edges represent joint research efforts or co-authorship of papers. In such a network, communities emerge based on research topics, academic disciplines, or shared research interests. Second, membership in multiple organizations is extremely common. Through notions of mixed-membership, node-centric network clustering methods have been developed to address this; yet difficulties remain in interpreting the results, as a node can be divided into multiple proportions, each of which belongs to a distinct community (Amelio and Pizzuti, 2014). For instance, a person may be a member of multiple interest groups, yet it is counter-intuitive to say that 50% of that person belongs to a dance club. Clustering on edges overcomes this issue, as edges represent relationships, and in most cases a relationship forms within a single context/cluster.

Despite the advantages above, very little work has been devoted to the area of edge clustering when compared with the numerous node-centric clustering algorithms. A few algorithms tried to solve this question by performing standard community detection on the line graph (Evans and Lambiotte, 2009); (Evans and Lambiotte, 2010); (Yoshida, 2013); (Tian et al., 2023). In recent years, the latent space edge clustering (LSEC) model (Sewell, 2021) was introduced as the first model-based edge community detection algorithm for directed networks. Building on this, the automated latent space edge clustering (aLSEC) model (Pham and Sewell, 2024) was developed, using an overfitted mixture prior to automatically determine the number of clusters. This addresses the limitation of the LSEC model, which requires the number of clusters to be specified in advance. Even though the LSEC model and aLSEC model have been shown to be accurate and computationally efficient, there are two fundamental limitations, described below.

The first limitation is that none of these edges clustering algorithms takes edge weights into consideration. Incorporating edge weights into the clustering model is crucial as they quantify the strength of relationships between connected nodes. Strongly weighted edges often signify more substantial interactions, indicative of specific communities or functional modules (Opsahl and Panzarasa, 2009). While node-centric clustering algorithms, such as weighted stochastic blockmodels (Aicher et al., 2015), have integrated edge weights to enhance accuracy, edge-centric clustering methods have largely overlooked this factor. In this study, we aim to fill this gap by developing the first model-based edge clustering algorithm tailored for weighted networks.

Second, while noise contamination has long been recognized as an important consideration in clustering of traditional data (Banfield and Raftery, 1993), this has not been addressed in network analysis. Through clustering edges and analyzing their weights, we gain a distinct advantage in identifying “noisy edges”. The notion of a “noisy edge” refers to edges that do not conform to the patterns or structures present in the majority of the edges in the network. For instance, in the email network, while a high-level manager may send regular emails to various teams under her supervision (which may share the same nodes between them), she may send a spurious email to everyone regarding a phishing attempt, or she may simply relay a message on behalf of someone else. The first example email highlights meaningful structure in a communication network, whereas the latter two may only serve to obscure the team structure. In a contact tracing network, this could be contacts caused by pedestrians passing by each other. The presence of noisy edges can sometimes pose challenges in clustering analysis, as they may obscure the underlying structure of the network, lead to over- or underestimation of the number of clusters, or lead to misinterpretations of the clusters. Therefore, it is essential to account for noisy edges and handle them appropriately in clustering algorithms to ensure accurate and meaningful results. Though such noisy edges widely exist in real-world networks, to the authors’ knowledge, currently no algorithms have been proposed to identify them in networks.

In this paper, we propose the Weighted Edge Clustering Adjusting for Noise (WECAN) model to address these two methodological research gaps by clustering weighted directed edges and identifying the noisy edges. The WECAN model automatically selects the number of clusters, makes use of edge weights to help inform clustering patterns, and incorporates a noise component to account for edges which act to obscure the underlying clustering structure of the network. WECAN is flexible in the types of edge weights that can be modeled by using a general exponential family modeling approach. Estimation is achieved by a variational Bayes generalized EM (VB-GEM) algorithm.

The remainder of the paper is as follows. Section 2 describes the proposed edge clustering model and the VB-GEM estimation algorithm. Section 3 describes a simulation study in which we compare our proposed approach to the existing state-of-the-art. Section 4 illustrates our proposed approach by applying WECAN to patient transfer networks. Finally, Section 5 provides a brief discussion.

## 2. Models and Methodology

### 2.1. Setup and notation

Consider the case where we observe a directed weighted network with  $n$  nodes represented by the set  $\mathcal{A}$  and  $M$  edges denoted by  $\mathcal{E} \subset \mathcal{A} \times \mathcal{A}$ . Without loss of generality, we will assume  $\mathcal{A} = \{1, 2, \dots, n\}$ . The set of edges is represented as  $\mathcal{E} = \{e_m\}_{m=1}^M$ , where each edge is denoted by  $e_m$ . Each edge  $e_m$  is a 3-dimensional vector comprised of the following: the sending node, denoted by  $e_{m1}$ ; the receiving node, denoted by  $e_{m2}$ ; and the weight of the edge, denoted by  $w_m$ . Thus,  $e_m = (e_{m1}, e_{m2}, w_m)$ . It is important to note that self-loops are not considered in our model. For clarity, we illustrate this with a running example from the domain of hospital epidemiology, namely patient transfer networks where nodes signify hospitals, and edges signify patient transfers between hospitals. Here,  $e_{m1}$  represents the sending hospital,  $e_{m2}$  represents the receiving hospital and  $w_m$  represents the number of patients transferred between these hospitals.

We presuppose the existence of a maximum of  $(K + 1)$  latent clusters for edges in this network, comprising up to  $K$  genuine clusters and 1 noise cluster. Edges within the “noise cluster” do not adhere to any distinct cluster structure, while edges within a “genuine cluster” exhibit meaningful patterns or structures within the network. Each edge is assigned to a specific latent cluster, represented by the binary variable  $Z_{mk}$ , where  $Z_{mk} = 1$  if edge  $e_m$  belongs to cluster  $k$ , and  $Z_{mk} = 0$  otherwise. In the patient transfer network context, these edge clusters can be understood as groups of patient transfers exhibiting distinct patterns. For instance, transfers within a cluster may involve patients who are moving “in network” based on insurance coverage, moving based on specialty clinics, or moving within a certain geographical region. Meanwhile, the noisy cluster could encompass rare or unexpected transfers, such as patients having health emergencies while traveling.

In the context of clustering nodes, it is well known that various topological features may obfuscate the clustering structure of the network (hence, for example, the development of the degree-corrected stochastic blockmodel proposed by Karrer and Newman, 2011). To avoid this from occurring in edge clustering, we utilize the following node-specific attributes:

- $\mathbf{S}_1 = (S_{11}, S_{12}, \dots, S_{1n})$ : representing overall propensities of nodes to send a large or small number of edges.
- $\mathbf{R}_1 = (R_{11}, R_{12}, \dots, R_{1n})$ : representing overall propensities of nodes to receive a large or small number of edges.
- $\mathbf{S}_2 = (S_{21}, S_{22}, \dots, S_{2n})$ : representing overall propensities of nodes to send edges with large or small weights.
- $\mathbf{R}_2 = (R_{21}, R_{22}, \dots, R_{2n})$ : representing overall propensities of nodes to receive edges with large or small weights.

These attributes address degree heterogeneity as well as heterogeneity in expected edge weights. In the patient transfer example, a large academic university will typically send and receive a large number of edges, and such a university that transfers patients from and to large urban hospitals will typically have edges with larger weights than one transferring patients from and to smaller rural hospitals.

Beyond degree heterogeneity and heterogeneity in expected edge weights, a core feature found ubiquitously in networks is that of homophily, the phenomenon where nodes that are close in some feature space have a higher probability of having an edge between them. We estimate homophilic effects by representing each node in a latent  $p$ -dimensional space. Stacking these  $n$  latent feature vectors forms an  $n \times p$  matrix  $\mathbf{U}$  for latent sending features and an  $n \times p$  matrix  $\mathbf{V}$  for latent receiving features. Originally motivated by the Aldous-Hoover theorem, this builds off of the rich class of latent space network models first put forth by Hoff et al. (2002). Furthermore, we posit that the contexts in which edges form, that is, the edge clusters, can also be described using this latent feature space, and that each edge context dictates how node features interact to build stronger or weaker edges. Notationally, we have the following:

- $\mathbf{Y}$ : a  $K \times p$  matrix where the  $k^{th}$  row,  $\mathbf{Y}_k$ , represents the latent features of the  $k^{th}$  edge cluster, elucidating how node features interact with edge clusters to form edges.
- $\Lambda$ : a  $K \times p \times p$  array where the  $k^{th}$   $p \times p$  diagonal matrix describes how node features interact with the  $k^{th}$  edge cluster to yield large or small edge weights.

In the patient transfer network, these variables unveil the underlying characteristics shared by hospitals and patients influencing transfers, such as geographical proximity or hospital specialization in treating specific diseases.

## 2.2. Model

The Weighted Edge Clustering Adjusting for Noise (WECAN) model is developed from the latent space edge cluster (LSEC) model (Sewell, 2021) and automated latent space edge cluster (aLSEC) model (Pham and Sewell, 2024). Suppose there are  $(K + 1)$  latent edge clusters in the observed network;  $Z_{m0} = 1$  represents that the edge is in the noise cluster, while  $Z_{mk} = 1$ ,  $k > 0$ , represents that the edge belongs to the cluster  $k$ . The WECAN model is given by:

$$\begin{aligned}\pi(\mathcal{E}|Z) &= \prod_{m=1}^M \prod_{k=0}^K (\pi(e_m|Z_{mk} = 1))^{Z_{mk}} \\ &= \prod_{m=1}^M \prod_{k=0}^K \left( \pi(e_{m1}|Z_{mk} = 1) \times \pi(e_{m2}|e_{m1}, Z_{mk} = 1) \times \pi(w_m|e_{m2}, e_{m1}, Z_{mk} = 1) \right)^{Z_{mk}}\end{aligned}\quad (1)$$

For the first two components in (1), we assume that for a noisy edge, the sending and receiving nodes should be completely random and, therefore, follow a uniform distribution. For a non-noisy, or structural, edge, we mirror the LSEC and aLSEC models, letting the sending and receiving nodes involved in the  $m^{th}$  edge depend on the nodes' overall propensities ( $\mathbf{S}_1, \mathbf{R}_1$ ), the nodes' latent features ( $\mathbf{U}, \mathbf{V}$ ), and the interactions between nodes' features and the edge's cluster ( $\mathbf{UY}, \mathbf{VY}$ ). This leads to the following:

$$\pi(e_{m1} = i|Z_{mk} = 1) = \begin{cases} \frac{1}{N} & k = 0 \\ \frac{e^{S_{1i} + U_i Y_k^T}}{f_{uk}} & o.w., \end{cases}$$

$$\pi(e_{m2} = j | e_{m1} = i, Z_{mk} = 1) = \begin{cases} \frac{1}{N-1} & k = 0, i \neq j \\ \frac{e^{R_{1j} + \mathbf{V}_j \mathbf{Y}_k^T}}{f_{vk} - e^{R_{1i} + \mathbf{V}_i \mathbf{Y}_k^T}} & k \neq 0, i \neq j \\ 0 & i = j, \end{cases}$$

where:  $f_{uk} = \sum_{i=1}^n e^{S_{1i} + \mathbf{U}_i \mathbf{Y}_k^T}$ ,  $f_{vk} = \sum_{i=1}^n e^{R_{1i} + \mathbf{V}_i \mathbf{Y}_k^T}$ . The quantities  $f_{uk}$  and  $f_{vk}$  are introduced for normalization.

We assume that the weight of a noisy edge follows an exponential distribution with user-defined rate  $\lambda_a$ ; this value should typically be large to induce a small expected weight for those edges in the noise component. The weight of a non-noisy edge follows an exponential family distribution with canonical parameter as  $\eta_{ijk}$  and dispersion parameter as  $\phi_k$ . Therefore we are able to model both continuous and discrete edge weights.

$$\begin{aligned} \pi(w_m = w | e_{m1} = i, e_{m2} = j, Z_{mk} = 1) \\ = \begin{cases} \lambda_a \exp(\lambda_a w) & k = 0 \\ h(w, \phi_k) \exp\left(\frac{\eta_{ijk} w - A(\eta_{ijk})}{a(\phi_k)}\right) / \Pr(w \neq 0 | \eta_{ijk}) & o.w., \end{cases} \end{aligned} \quad (2)$$

The canonical parameter  $\eta_{ijk}$  is decided by weights-related parameters consisting of the sending and receiving nodes' overall propensities ( $S_{2i}, R_{2j}$ ) to be involved in large or small edge weights, the interaction between the nodes' and edge cluster's features  $\mathbf{U}_i \Lambda \mathbf{V}_j'$ , and the cluster-specific intercept  $\beta_k$ . Thus we set

$$\eta_{ijk} := \beta_k + S_{2i} + R_{2j} + \mathbf{U}_i \Lambda_k \mathbf{V}_j^T.$$

The remaining elements in (2) are determined by the specific exponential family distribution chosen by the user. Taking the normal distribution as an example, we have

$$\begin{aligned} h(w_m, \phi_k) &\propto \frac{1}{\phi_k} \exp\left(-\frac{w_m^2}{2\phi_k^2}\right), \\ A(\eta_{ijk}) &= \frac{\eta_{ijk}^2}{2}, \\ a(\phi_k) &= \phi_k^2. \end{aligned}$$

Note that  $\phi_k$  is the standard deviation in normal distribution.

As in the aLSEC model, we propose a sparse finite mixture model for  $Z_m$ , this allows for automatic detection of the number of clusters. By deliberately overfitting the model with more components than necessary, the Dirichlet prior with small expected concentration parameter encourages zero weights for irrelevant components, effectively leaving some clusters empty. This approach ensures that the model adapts to the true number of clusters without requiring prior specification.

Each cluster assignment vector  $Z_m$  is drawn from a multinomial distribution with probabilities  $\boldsymbol{\pi} := (\pi_0, \pi_1, \dots, \pi_K)$ . To make sure  $\sum_{k=0}^K \pi_k = 1$ , we introduce  $\mathbf{t} = c(t_0, t_1, \dots, t_K)$ , where for the noise cluster,  $\pi_0 = t_0$ ; for non-noise cluster,  $\pi_k = t_k(1 - t_0)$ . So we have:

$$Z_m | \boldsymbol{\pi} \stackrel{iid}{\sim} \text{Multinomial}(1, \boldsymbol{\pi})$$

$$t_0 \sim \text{Beta}(c_0, d_0)$$

$$(t_1, \dots, t_K) \sim \text{Dir}(\alpha \mathbf{1}_K)$$

$$\alpha \sim \Gamma(a_\alpha, b_\alpha)$$

The complete likelihood is:

$$\begin{aligned} f(\mathcal{E}, Z | \boldsymbol{\theta}) \\ = \prod_{m=1}^M \left[ t_0 \times \lambda_a \times \frac{1}{N(N-1)} \times \exp(-\lambda_a w_m) \right]^{Z_{m0}} \\ \times \prod_{k=1}^K \left[ t_k(1 - t_0) \times \frac{h(w, \phi_k) \exp\left(\frac{\eta_{em} w - A(\eta_{em})}{a(\phi_k)}\right) \times \exp\left(S_{1e_{m1}} + R_{1e_{m2}} + \mathbf{U}_{e_{m1}} \mathbf{Y}'_k + \mathbf{V}_{e_{m2}} \mathbf{Y}'_k\right)}{f_{uk} (f_{vk} - \exp(R_{1e_{m1}} + \mathbf{V}_{e_{m1}} \mathbf{Y}'_k)) \Pr(w_m \neq 0 | \eta_{em})} \right]^{Z_{mk}} \end{aligned} \quad (3)$$

Fig. 1 shows all the parameters for the WECAN model. Note that if we ignore the weights of the network, WECAN model will be equivalent to LSEC model.

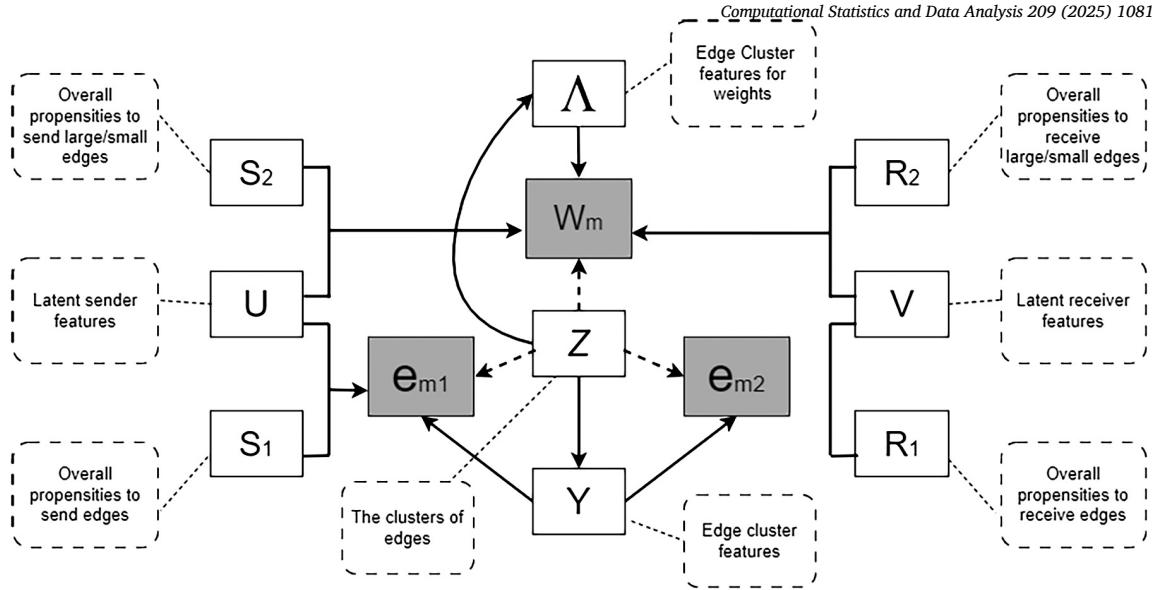


Fig. 1. Schematic of the WECAN model.

### 2.3. Parameter Estimation

We propose performing Bayesian estimation in the same manner as in Pham and Sewell (2024). For unknown quantities  $(S_1, S_2, R_1, R_2, U, V, \Lambda, Y, \beta, \phi)$ , following priors are assumed:

$$\begin{aligned} (U_i, V_i)^T &\stackrel{iid}{\sim} MVN(\mathbf{0}, \Sigma_{UV} \otimes I_p), & \Sigma_{UV} &\sim IW(\Psi_{0UV}, v_{0UV}), \\ (S_{1i}, R_{1i})^T &\sim MVN(\mathbf{0}, \Sigma_{SR1}), & \Sigma_{SR1} &\sim IW(\Psi_{0SR1}, v_{0SR1}), \\ (S_{2i}, R_{2i})^T &\sim MVN(\mathbf{0}, \Sigma_{SR2}), & \Sigma_{SR2} &\sim IW(\Psi_{0SR2}, v_{0SR2}), \\ Y_k &\sim MVN(\mathbf{0}, I_p), & \text{diag}(\Lambda_k) &\sim MVN(\mathbf{0}, \lambda I_p), \\ Z_m &\sim \text{Multinomial}(1, \alpha), & \alpha &\sim Dir(\alpha_0 \mathbb{1}_K), \\ \beta_k &\propto 1, & \lambda &\sim \Gamma^{-1}(a_0, b_0), \end{aligned}$$

where  $MVN(\mathbf{a}, \Sigma)$  represents the multivariate normal distribution with  $\mathbf{a}$  as the mean vector and  $\Sigma$  as the covariance matrix,  $IW(\Psi, v)$  represents the Inverse Wishart distribution with  $\Psi$  as scale matrix,  $v$  as degrees of freedom,  $Dir(\mathbf{a})$  denotes the Dirichlet distribution with  $\mathbf{a}$  as concentration parameter,  $\Gamma^{-1}(a, b)$  denotes inverse Gamma distribution with  $a$  as shape parameter and  $b$  as scale parameter.  $I_p$  is  $p \times p$  identity matrix,  $\mathbb{1}_K$  is the K-dimensional vector of ones. The prior on the dispersion parameter will be dependent on the specific exponential family used for the edge weights. For example, in Section 4 we assume a normal distribution where we use a half- $t$  prior with  $v_0$  denoting the degrees of freedom and  $\eta_0$  denoting the scale parameter on the standard deviation:

$$\phi_k \sim \text{half-}t(v_0, \eta_0).$$

Then we propose a variational-Bayes generalized expectation–maximization (VB-GEM) algorithm (Dempster et al., 1977; Bernardo et al., 2003). In the E-step, both  $Z$  and  $t = t_0, t_1, \dots, t_k$  are the latent variables. In the M-step, we update the unknown parameters  $\theta := \{S_1, S_2, R_1, R_2, U, V, \Lambda, Y, \Sigma_{SR1}, \Sigma_{SR2}, \Sigma_{UV}, \lambda, \alpha\}$  by performing a conjugate gradient approach that maximizes the expected log posterior  $Q(\theta|\theta^t)$ , where  $\theta^t$  denotes the current estimates of the parameters.

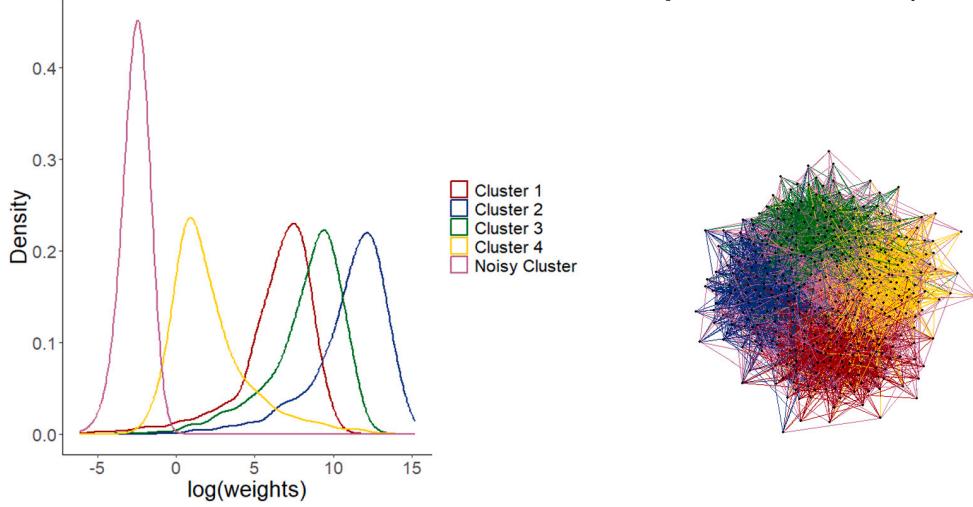
For the expectation step (E step), the variational distributions are selected by maximizing evidence lower bound (ELBO) of the log marginal likelihood, which can be derived as below:

$$ELBO := E_{q(Z,t)} \log(f(\theta, Z, t | \mathcal{E})) - E_{q(Z,t)} \log(q(Z, t)).$$

The full log posterior and the details of ELBO can be found in the Appendix A. In short, the full conditional of  $Z$  is the multinomial distribution; the full conditional of  $t$  is a Dirichlet distribution; and the full conditional of  $t_0$  is a Beta distribution. This semi-conjugacy lends itself to a mean-field variational Bayes estimation of the means for the E step of the algorithm, providing us closed form solutions when iteratively updating the parameters of the variational approximation distribution for  $Z$ ,  $t_0$ , and  $t$  until convergence.

For the maximization step (M step), we propose a coordinate ascent approach similar to Sewell (2021) to find:

$$\theta^{t+1} = \text{argmax}_{\theta} (Q(\theta|\theta^t))$$



**Fig. 2.** Example of simulated network. The yellow, red, green, and blue clusters represent meaningful network structures with varying but overlapping edge weights, whereas the mauve cluster is comprised of pairs selected uniformly at random with smaller edge weights. (Figure colors are available in the web version of this article.)

The updates of the  $\{S_1, S_2, R_1, R_2, U, V, \Lambda, Y\}$  are by conjugate gradient; the updates of the  $\{\Sigma_{SR_1}, \Sigma_{SR_2}, \Sigma_{UV}, \lambda, \alpha\}$  are from analytical solutions. The details of gradients can be found in Appendix B, and the analytical solutions can be found in Appendix C.

To obtain a more reasonable initialization, we first run the aLSEC model once to get initialization for the variables  $(S_1, R_1, U, V, Y)$ ; for the initial values of  $S_2$  and  $R_2$ , the sum of in-degree and out-degree edge weights of the adjacent edges for each node was calculated; and when the weights were fit using a normal distribution, the standard deviation of the weights was used to set the initial value of  $\phi$ . The convergence in the E-step was evaluated by monitoring the changes in the ELBO. Convergence of the whole algorithm was based on a lack of change in the cluster assignments between iterations.

In the analyses of Sections 3 and 4, we repeated the VB-GEM estimation procedure 15 times with different random starting points and computed the integrated completed likelihood (ICL) to select the optimal (highest) result (Biernacki et al., 2000).

### 3. Simulation Studies

#### 3.1. Study Design

We simulated networks with edges in 5 clusters: 1 noise cluster and 4 non-noise clusters. We varied the proportion of noisy edges in the set  $\{0\%, 5\%, 10\%, 15\%, 20\%\}$ . Under each noisy edge proportion setting, we simulated 200 networks, each with 400 nodes and an average of 7065 edges. Each pair of  $(S_1, R_1)$  and  $(S_2, R_2)$  are sampled from two independent bivariate normal distributions with mean 0, variances 2, and correlation 0.75. We set  $p = 2$ , and the row vectors of the latent features  $Y$  were equally spaced on a circle. The direction of the  $U_i$ 's and  $V_i$ 's were drawn from a mixture of von Mises–Fisher distributions with concentration parameter of 50, and their magnitudes were drawn from a gamma distribution with shape equal to 150 and rate equal to 40. The  $\beta_k$  were simulated by a random sample without replacement from  $\{-1, -2, 1, 2\}$ , and  $\Lambda_k$  was set to be a  $2 \times 2$  diagonal matrix with the diagonal as  $\{0.4, 0.4\}$ ,  $\{0.4, -0.4\}$ ,  $\{-0.4, 0.4\}$ ,  $\{-0.4, -0.4\}$  for each non-noisy cluster respectively. The weights of edges in the noise cluster follow a gamma distribution with shape equal to 2 and rate equal to 20. The edge weights within the genuine clusters are modeled as follows a normal distribution with a mean of  $\beta_k + S_{2i} + R_{2j} + U_i \Lambda_k V_j^T$ , while the standard deviations are sampled from a uniform distribution,  $Unif(0.05, 0.5)$ .

Fig. 2 presents an example of a simulated network, with different colors indicating the clusters to which the edges belong. The pink edges represent noisy edges, uniformly distributed across the network. In the left plot, the densities of log-transformed edge weights are shown, helping to highlight the distinction between noisy and non-noisy edges. While the noisy edges generally have smaller weights, the weights of non-noisy edges are heavily overlapping. The right plot displays the same network structure without showing the weights, using the Fruchterman-Reingold layout. This visualization illustrates how noisy edges can obscure the underlying clustering structure.

For each simulated network, we clustered the edges using WECAN and two other competitors and compared their performance. First, we used spectral clustering on the line graph (SCLG). Spectral clustering detects communities by leveraging the eigenvalues of the graph Laplacian matrix. The process involves constructing a similarity graph, computing the Laplacian matrix, and embedding the graph into a lower-dimensional space based on the smallest eigenvalues. Traditional clustering algorithms like k-means are then applied to identify clusters (Von Luxburg, 2007). In the SCLG method, we first convert the original network into a line graph, and then apply spectral clustering to the resulting line graph. Second, we used the aLSEC model.

Following (Pham and Sewell, 2024), we set the parameter  $p$  to 4 for both the aLSEC and WECAN models to effectively capture latent space information. For the WECAN model, the non-noisy edges' weights are assumed to follow a normal distribution. To

**Table 1**

Performance of the WECAN model in correctly identifying the true number of clusters.

Proportion of Noisy Edges	0%	5%	10%	15%	20%
Proportion of Correctly Estimated # of clusters	97.9%	97.0%	98.5%	100%	100%

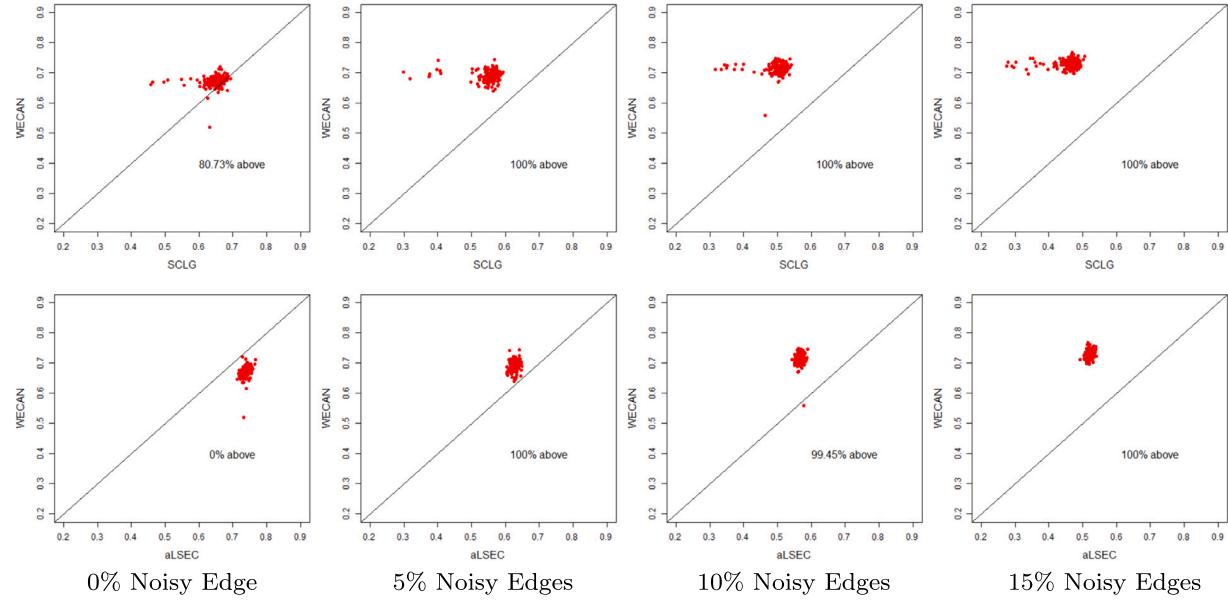


Fig. 3. NMI comparison of SCLG, aLSEC model and WECAN model in our simulation study.

evaluate estimation accuracy, we compared true cluster assignments with estimated cluster assignments using the normalized mutual information (NMI) metric (Danon et al., 2005). NMI is a criteria that measures the similarity between clusters of the same data set; it ranges from 0 to 1, with higher values indicating better agreement between clusters. Notably, we also attempted to compare our results with those from the linkcomm R package (Kalinka and Tomancak, 2011), but due to consistently poor performance, as evidenced by NMI values approaching zero in all cases, we have omitted these results from our analysis.

We also conducted a study to evaluate the computational time of the WECAN model. Previous research has demonstrated that the aLSEC model is highly efficient, reducing run time by 10 to over 100 times compared to the original LSEC model. In our study, we compared the running times of the WECAN model and the aLSEC model by simulating 100 networks of varying sizes, ranging from 100 to 1000 nodes. Both models were applied to each of these networks for comparison. Computation for the simulation study was done on University of Iowa High-performance Computing (HPC) system. The tests of model computation time were performed on a server with an Intel(R) Xeon(R) Gold 6230 CPU @ 2.10 GHz. All code was executed by R in version 4.0.5 (R Core Team, 2023). Package RcppArmadillo (version 0.12.6.4.0) (Eddelbuettel and Sanderson, 2014) was used to increase the processing speed.

### 3.2. Results

Table 1 presents the results of applying the WECAN model to 200 simulated networks under each scenario, showing the proportion of cases in which the correct number of non-noisy clusters ( $K = 4$ ) was detected. Overall, the WECAN model performed exceptionally well in identifying the correct number of clusters. In all scenarios, it successfully detected the correct number of clusters in over 97% of the simulations. Notably, when the proportion of noisy clusters exceeded 10%, the model accurately identified the correct number of clusters in 100% of the simulations.

Fig. 3 presents a visual comparison of the Normalized Mutual Information (NMI) for the estimation results obtained from spectral clustering on the line graph, aLSEC model, and WECAN model. The top row depicts comparisons between spectral clustering on the line graph and the WECAN model, while the bottom row illustrates comparisons between the aLSEC model and the WECAN model. Each red point represents the NMI obtained from spectral clustering on the line graph or the aLSEC model plotted against the NMI from the WECAN model for a simulated network. The black line indicates the line  $y = x$ , where points above it indicate instances where the WECAN model outperforms the other model, while points below it denote the opposite.

The WECAN model consistently outperforms spectral clustering on the line graph across all noisy proportion settings. Remarkably, even in scenarios where no noisy edges are present and an incorrect model is assigned to the WECAN model, it still achieves better prediction accuracy in over 80% of the networks.

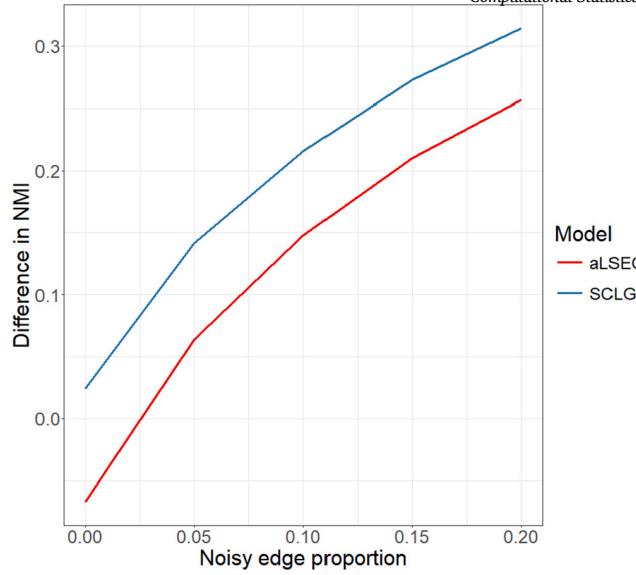


Fig. 4. Mean difference in NMI between WECAN and either aLSEC or SCLG across varying noisy proportion settings.

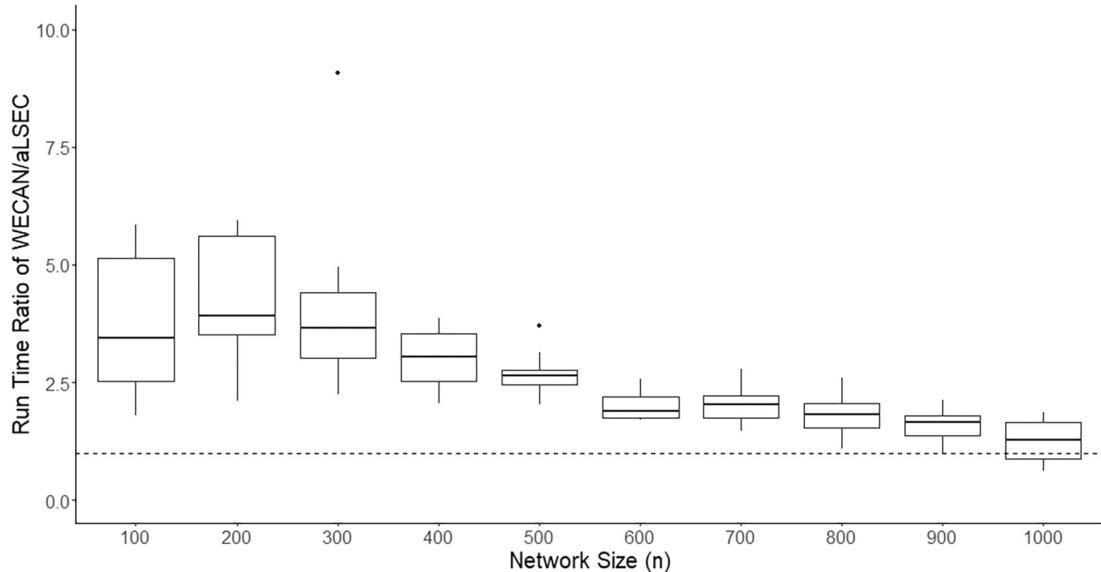
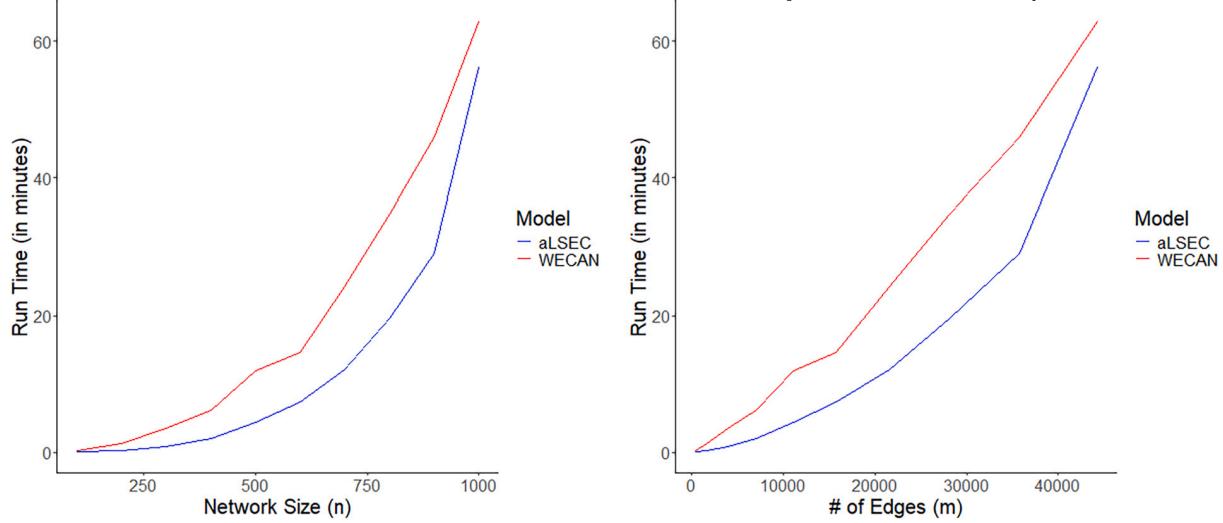


Fig. 5. The ratios of the run time of the WECAN model over the aLSEC model.

When there are no noisy edges present in the simulated network, the aLSEC model demonstrates superior prediction compared to the WECAN model. However, even a small proportion of noisy edges (as low as 5%) leads to an improvement in prediction accuracy for the WECAN model for all simulations. While not shown, these results hold true when the user-input parameter  $p$  is varied from 4 to 2, indicating the robustness of the WECAN model across different settings.

Fig. 4 shows the mean difference in NMI across various noisy proportion settings. The blue curve represents the difference between the mean NMI of the WECAN model and SCLG, while the red curve represents the difference between the mean NMI of the WECAN model and the aLSEC model. Both differences increase as the proportion of noisy edges rises. At a noisy edge proportion of 20%, the WECAN model outperforms SCLG by 0.31 in NMI and the aLSEC model by 0.26.

Fig. 5 compares the running times of the aLSEC model and the WECAN model. The results show that while the aLSEC model generally runs faster than the WECAN model, the difference in computational time decreases as the network size increases. This is because the WECAN model uses the aLSEC model as an initial step, providing a strong starting point that reduces the additional computational time required. For smaller networks, the WECAN model takes approximately three times longer than the aLSEC model, but since the overall time is still short, this difference is less impactful. For larger networks, the running times of the two models converge, ensuring that the WECAN model maintains a reasonable computational time overall. Fig. 6 presents the actual runtime



**Fig. 6.** Analysis of computational time on synthetic networks.

(in minutes) as a function of the network's number of nodes ( $n$ ) and edges ( $m$ ), respectively. The WECAN model shows only a slight increase in runtime compared to the aLSEC model. With a computational complexity of  $\mathcal{O}(N + M)$ , the right plot highlights that for large and dense networks where  $m \gg n$ , the runtime grows approximately linearly with the number of edges in the network.

#### 4. Real Data Example

To illustrate the effectiveness of the WECAN model on real-world data, we conducted an analysis using patient transfer network from New York state. These datasets were obtained from the Healthcare Cost and Utilization Project (HCUP) State Inpatient Databases (SID).

This patient transfer network is constructed from the New York SID from 2005 to 2016. In this network, the nodes represent New York hospitals, and the edges indicate patients who are transferred between them. Throughout the study period, we interpreted a directed edge from hospital A to hospital B as indicating patient transfers from hospital A to hospital B. The weight of each edge corresponds to the number of patients transferred between the two hospitals during the study period. The network encompasses 167 nodes and 6187 edges, with an average weight of 67.16.

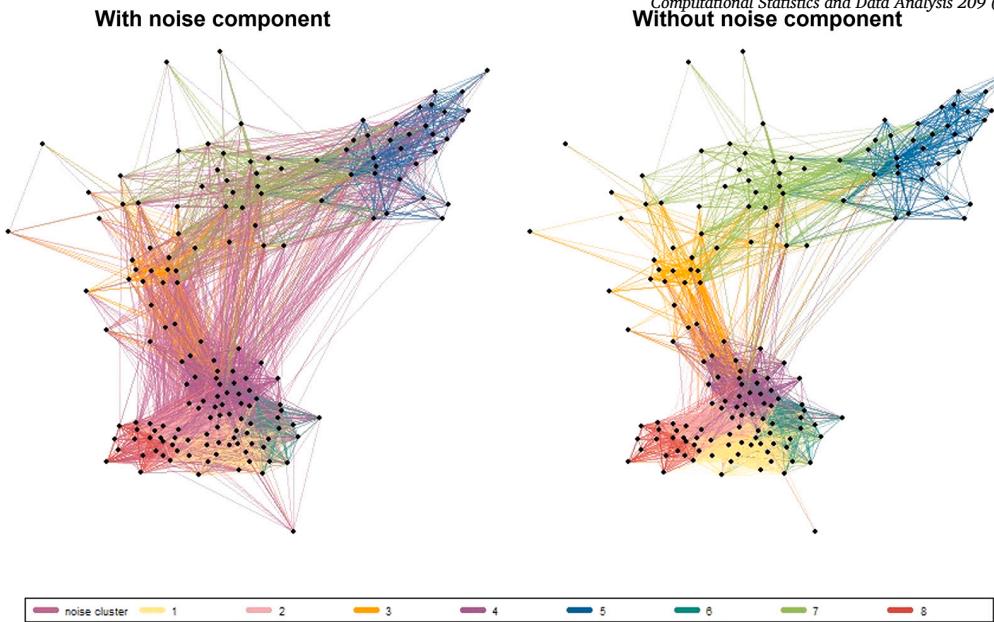
Analyzing the patient transfer network is crucial because transfer patients have been found to play a significant role in infectious disease outbreaks. Studies have demonstrated that patient transfer patterns profoundly impact the spread of healthcare-associated infections (HAI), such as hospital-acquired MRSA (Donker et al., 2010). As patients are represented by edges in the patient transfer network, utilizing edge clustering during HAI outbreaks can provide valuable insights into transmission patterns and flow.

We applied the WECAN model to the New York State patient transfer network, assuming a log-normal distribution for the non-noisy edges. The WECAN approach relying on sparse finite mixtures automatically selected eight non-noise clusters. We then plotted the estimated clusters both with and without the noisy edges included. Finally, we visualized the clusters based on their geographic locations to explore potential real-world implications for these clusters.

Fig. 7 visually illustrates the edge clusters within the New York patient transfer network. Our analysis revealed 8 distinct non-noise clusters, while 1823 edges were identified as noisy edges, with an average weight of 1.1. In the left plot of Fig. 7, the noisy edges are colored in pink, visually confirming our assumption that they are uniformly distributed throughout the network. Upon removing the noisy edges, as shown in the right plot, the patterns of the remaining 8 non-noise clusters become clearer. These non-noise clusters exhibit high concentration, which aligns with our intuition.

Fig. 8 shows the geographic distribution of the eight non-noisy clusters on the map, reflecting the locations of the hospitals involved in patient transfers. This sheds light on the real-world significance of the edge clusters. For instance, Cluster 1 represents patient transfers from areas surrounding New York City into the city itself. Cluster 2 captures transfers from rural areas to New York City and Long Island. Cluster 3 corresponds to transfers directed toward the Albany Med Health System, the largest healthcare provider in the Capital District of New York. Cluster 4 represents transfers from smaller cities to New York City. Cluster 5 reflects patient transfers within the northwestern regions of New York state. Cluster 6 involves transfers within New York City itself. Cluster 7 is centered around transfers between hospitals in the Syracuse and Rochester areas. Finally, Cluster 8 represents transfers between Springfield, New York City, and Long Island. Overall, the clusters identified by the WECAN model align with meaningful real-world healthcare dynamics, indicating that the model's results are interpretable and realistic.

We conducted a Pearson correlation test between the estimated parameters from the WECAN model and key hospital features. The results show a significant positive correlation between the estimated  $S_2$  and total admissions at the hospital (correlation = 0.26, p-value < 0.001). This indicates that larger hospitals, in terms of admissions, tend to have a higher propensity to send more



**Fig. 7.** Clustering of the New York state patient transfer network by WECAN model.

patients. Additionally, we found a positive correlation between the hospital's estimated  $R_1$  and the total number of beds (correlation = 0.19, p-value < 0.05), suggesting that hospitals with more beds are more likely to receive patients. These findings are consistent with real-world expectations and demonstrate that the WECAN model produces interpretable and meaningful output parameters.

In network analysis, a common approach to mitigate the impact of such noisy edges is to remove edges with weights below a certain cutoff value. For instance, in patient transfer networks, researchers often disregard edges with a weight of 1, indicating that only one patient was transferred between two hospitals during the study period. In the New York patient transfer network, out of the 1823 identified "noisy edges," only 914 have a weight of 1. This finding suggests that setting a cutoff value may overlook edges with weights slightly above the threshold that still mask the underlying connectivity patterns.

We also applied the WECAN model to the Wisconsin patient transfer network, generated from SID database from 2013 to 2017. This network exhibits a much sparser structure (density = 0.10 for WI versus 0.22 for NY). In the Wisconsin patient transfer network example, there are 678 edges associated with a weight of 1. However, only 153 of them have been identified as "noisy edges" by the WECAN model. In this case, the WECAN model has the potential to help preserve information for network analysis by accurately identifying and excluding noisy edges. Compared with results from the NY network, these suggest that WECAN can differentiate small and unimportant edges from small and important edges.

This observation highlights the limitations of using a cutoff value alone to identify noisy edges. Therefore, a more sophisticated approach, such as the WECAN model, which takes into account the entire distribution of edge weights and other network properties, is essential for accurately identifying and removing noisy edges in network analysis.

## 5. Discussion

In this paper we have introduced the WECAN model, an innovative model-based clustering algorithm developed from the aLSEC model, incorporating the information in the edge weights and augmented with a noisy component. It has been demonstrated that dichotomizing network edges with thresholds may distort the latent structure of the network, potentially resulting in biased outcomes (Berardo et al., 2020). Our model fills a significant gap in current network edge clustering methods by utilizing edge weights, thus offering a more nuanced understanding of network structures.

Furthermore, the incorporation of a noise component in the WECAN model strengthens the resilience of edge clustering in the presence of extraneous edges. Noise components are known to disrupt clustering algorithms, making it challenging to detect the cluster structure of the remaining domain points (Ben-David and Haghtalab, 2014). Our real data analysis demonstrates that the WECAN model effectively distinguishes between small yet significant edges and small, less significant ones, thereby aiding in the preservation of valuable information for network analysis.

Through the simulation study, we demonstrated the effectiveness of the WECAN model in discerning meaningful patterns from complex network structures, particularly in scenarios involving noisy edges. The WECAN model can make a substantial improvement in prediction accuracy achieved by the WECAN model, underscoring its utility in practical applications.

Our analysis of patient transfer network data underscores the prevalence of unstructured "noisy edges" in real networks, further emphasizing the importance of robust clustering algorithms like WECAN. The ability to accurately identify and delineate edge clusters is applicable to a wide range of applications. For instance, in the context of infectious disease transmission, understanding patient

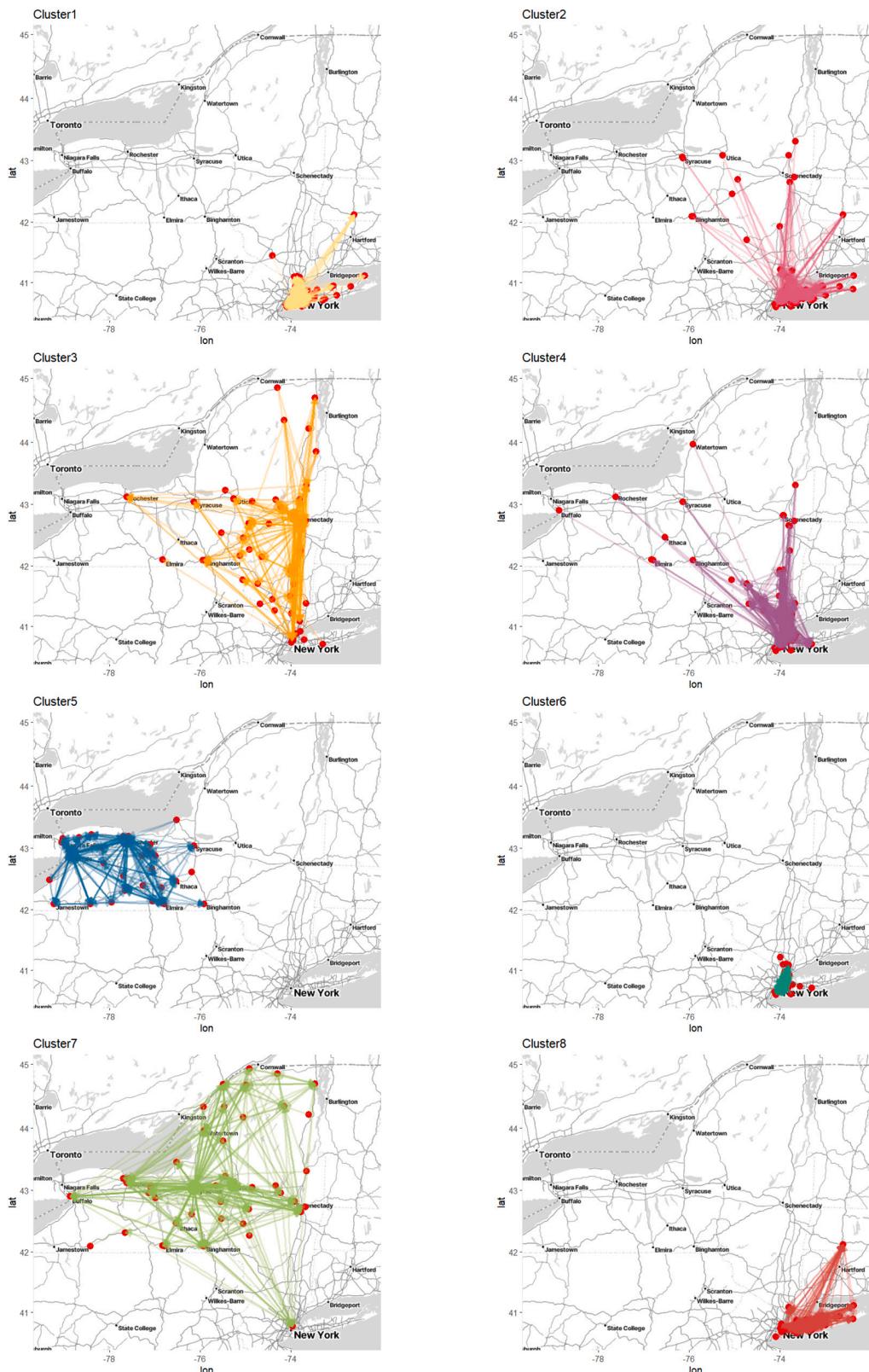


Fig. 8. Geographic Distribution of Non-Noisy Clusters in New York State's Patient Transfer Network.

clusters can facilitate proactive measures to mitigate risks and control outbreaks, while in social networks we may better understand the contexts in which relationships form.

Limitations of our proposed approach include the following. First, we currently do not have a method for determining whether the noise component ought to be included or excluded. Future work in selecting either a model with or without a noise component would be important for practitioners. Second, while our method is computationally efficient for sparse networks, growing linearly with respect to the number of nodes and edges in the network, this scalability does not hold for dense networks. Third, we assumed an exponential distribution on the weights of the noise component. This was to induce small edge weights overall while allowing some edge weights from the noise component to be large. This may not be appropriate in all cases, e.g., when the edge weights may be negative. Despite these limitations, the WECAN model provides a novel way to uncover hidden structures and dynamics within networks, helping to improve insights and decision-making in different areas.

While the current application of the WECAN model focuses on static networks, it has the potential to be extended to longitudinal network data by incorporating time as an additional dimension. Moreover, the model could be adapted to account for node and edge attributes, providing richer insights by incorporating additional information such as node characteristics or edge-specific properties. Future work may explore these extensions to enhance the model's applicability in more complex network structures.

### Acknowledgements

This work was supported by the US Centers for Disease Control and Prevention (5 U01CK000594-04-00) as part of the MInd-Healthcare Program.

### Appendix A. Posterior and ELBO

The full log posterior is given by:

$$\begin{aligned}
& E_{q(\mathbf{Z}, \mathbf{t})} \log(f(\theta, \mathbf{Z}, \mathbf{t} | \mathcal{E})) \\
& \propto \sum_{k=1}^K \sum_{m=1}^M E(Z_{mk}) \times \left[ \log(h(w_m, \phi_k)) + \frac{\eta_{e_{m1} e_{m2} k} w_m - A(\eta_{e_{m1} e_{m2} k})}{a(\phi_k)} \right. \\
& \quad - \log(\Pr(w_m \neq 0 | \eta_{e_{m1} e_{m2} k})) + S_{1e_{m1}} + R_{1e_{m2}} + U_{e_{m1}} \mathbf{Y}'_k + V_{e_{m2}} \mathbf{Y}'_k \\
& \quad - \log(f_{uk}) - \log(f_{vk} - e^{R_{1e_{m1}} + V_{e_{m2}} \mathbf{Y}'_k}) + \log(t_k) + \log(1 - t_0) \Big] \\
& \quad + \sum_{m=1}^M E(Z_{m0}) \times (\log(t_0) - \log(\frac{N(N-1)}{\lambda_a}) - \lambda_a w_m) \\
& \quad + \log \Gamma(K\alpha) - K \log \Gamma(\alpha) + (\alpha_0 - 1) \times \log(E(t_0)) + (\beta_0 - 1) \times \log(1 - E(t_0)) + \sum_{k=1}^K (\alpha - 1) E(\log(t_k)) \\
& \quad - \frac{1}{2} \text{tr}((S_1 \quad R_1) \Sigma_{SR1}^{-1} (S_1 \quad R_1)' - \frac{n}{2} \log |\Sigma_{SR1}|) \\
& \quad - \frac{1}{2} \text{tr}((S_2 \quad R_2) \Sigma_{SR2}^{-1} (S_2 \quad R_2)' - \frac{n}{2} \log |\Sigma_{SR2}|) \\
& \quad - \frac{1}{2} \text{tr}((U \quad V) (\Sigma_{UV}^{-1} \otimes I_p) (U \quad V)' - \frac{np}{2} \log |\Sigma_{UV}|) \\
& \quad + a_\alpha \log \alpha - b_\alpha \alpha - \sum_k \frac{\|\mathbf{Y}_k\|^2}{2} \\
& \quad - \frac{v_{SR1} + p + 1}{2} \log(|\Sigma_{SR1}|) - \frac{1}{2} \text{tr}(\Phi_{SR1} \Sigma_{SR1}^{-1}) \\
& \quad - \frac{v_{SR2} + p + 1}{2} \log(|\Sigma_{SR2}|) - \frac{1}{2} \text{tr}(\Phi_{SR2} \Sigma_{SR2}^{-1}) \\
& \quad - \frac{v_{UV} + p + 1}{2} \log(|\Sigma_{UV}|) - \frac{1}{2} \text{tr}(\Phi_{UV} \Sigma_{UV}^{-1}) \\
& \quad - \frac{v_0 + 1}{2} \sum_k \log(1 + \frac{\phi_k^2}{v_0 \eta_0^2}) - \frac{\sum_k \|\text{diag}(\Lambda_k)\|^2 + 2b_0}{2\lambda} - (a_0 + 1 + \frac{Kp}{2}) \log \lambda.
\end{aligned}$$

Besides:

$$\begin{aligned}
E_{q(\mathbf{Z}, \mathbf{t})} \log(q(\mathbf{Z}, \mathbf{t})) &= [\sum_{m=1}^M E_{q(\mathbf{Z})}(\log q(\mathbf{Z}_m))] + E_{q(\mathbf{t})}(\log q(\mathbf{t})) \\
\sum_{m=1}^M E_{q(\mathbf{Z})}(\log q(\mathbf{Z}_m)) &= E(\mathbf{Z}_0) \times \log \left( t_0 \times \lambda_a \times \frac{1}{N(N-1)} \times \exp(-\lambda_a w_m) \right)
\end{aligned}$$

$$\begin{aligned}
& \sum_{k=1}^K E(Z_{mk}) \times \log \left( t_k (1 - t_0) \times \right. \\
& \left. \frac{h(w, \phi_k) \exp(\frac{\eta_{em} w_m - A(\eta_{em})}{a(\phi_k)}) \times \exp(S_{1e_{m1}} + R_{1e_{m2}} + U_{e_{m1}} Y'_k + V_{e_{m2}} Y'_k)}{f_{uk}(f_{vk} - \exp(R_{1e_{m1}} + V_{e_{m1}} Y'_k)) \Pr(w_m \neq 0 | \eta_{em})} \right) \\
E_{q(t)}(\log q(t)) = & (E(\Sigma_{m=1}^M Z_{m0}) + \alpha_0 - 1) \log t_0 + (M - E(\Sigma_{m=1}^M Z_{m0}) + \beta_0 - 1) \log(1 - t_0) \\
& + \sum_{k=1}^K (E(\Sigma_{m=1}^M Z_{mk}) + \alpha - 1) \times \log(t_k)
\end{aligned}$$

## Appendix B. The Gradients

$$\begin{aligned}
\frac{\partial Q}{\partial S_{1i}} &= \sum_k \left( p_{(1)k} - p_{-k} \frac{e^{S_{1i} + U_i Y'_k}}{f_{uk}} \right) - \Sigma_{SR1,11} S_{1i} - \Sigma_{SR1,12} R_{1i} \\
\frac{\partial Q}{\partial S_{2i}} &= \sum_{k=1}^K \sum_{m:e_{m1}=i}^M p_{mk} \left( \frac{1}{a(\phi_k)} \left( w_m - \frac{\partial A(\eta_{ie_{m2}k})}{\partial \eta_{ie_{m2}k}} \right) - \frac{\frac{\partial}{\partial \eta_{ie_{m2}k}} \Pr(w_m \neq 0 | \eta_{ie_{m2}k})}{\Pr(w_m \neq 0 | \eta_{ie_{m2}k})} \right) \\
&\quad - \Sigma_{SR2,11} S_{2i} - \Sigma_{SR2,12} R_{2i} \\
\frac{\partial Q}{\partial U_i} &= \sum_{k=1}^K \left\{ \sum_{m:e_{m1}=i}^M p_{mk} \left( \frac{1}{a(\phi_k)} \left( w_m - \frac{\partial A(\eta)}{\partial \eta} \right) V_{e_{m2}} \Lambda_k - \frac{\frac{\partial}{\partial \eta_{ie_{m2}k}} \Pr(w_m \neq 0 | \eta_{ie_{m2}k})}{\Pr(w_m \neq 0 | \eta_{ie_{m2}k})} V_{e_{m2}} \Lambda_k \right) \right. \\
&\quad \left. + \left( p_{(1)k} - \frac{p_{-k}}{f_{uk}} e^{S_{1i} + U_i Y'_k} \right) Y_k \right\} - \Sigma_{UV,11} U_i - \Sigma_{UV,12} V_i \\
\frac{\partial Q}{\partial R_{1i}} &= \sum_k \left( p_{(2)k} - e^{R_{1i} + V_i Y'_k} \left( H_k - \frac{p_{(1)k}}{f_{vk} - e^{R_{1i} + V_i Y'_k}} \right) \right) - \Sigma_{SR1,22} R_{1i} - \Sigma_{SR1,12} S_{1i} \\
\frac{\partial Q}{\partial R_{2i}} &= \sum_{k=1}^K \sum_{m:e_{m2}=i}^M p_{mk} \left( \frac{1}{a(\phi_k)} \left( w_m - \frac{\partial A(\eta)}{\partial \eta} \right) - \frac{\frac{\partial}{\partial \eta_{e_{m1}ik}} \Pr(w_m \neq 0 | \eta_{e_{m1}ik})}{\Pr(w_m \neq 0 | \eta_{e_{m1}ik})} \right) \\
&\quad - \Sigma_{SR2,22} R_{2i} - \Sigma_{SR2,12} S_{2i} \\
\frac{\partial Q}{\partial V_i} &= \sum_{k=1}^K \left\{ \sum_{m:e_{m2}=i}^M p_{mk} \left( \frac{1}{a(\phi_k)} \left( w_m - \frac{\partial A(\eta)}{\partial \eta} \right) U_{e_{m1}} \Lambda_k - \frac{\frac{\partial}{\partial \eta_{e_{m1}ik}} \Pr(w_m \neq 0 | \eta_{e_{m1}ik})}{\Pr(w_m \neq 0 | \eta_{e_{m1}ik})} U_{e_{m1}} \Lambda_k \right) \right. \\
&\quad \left. + \left( p_{(2)k} - e^{R_{1i} + V_i Y'_k} \left( H_k - \frac{p_{(1)k}}{f_{vk} - e^{R_{1i} + V_i Y'_k}} \right) \right) Y_k \right\} - \Sigma_{UV,22} V_i - \Sigma_{UV,12} U_i \\
\frac{\partial Q}{\partial \text{diag}(\Lambda_k)} &= \sum_{m=1}^M p_{mk} \left[ \frac{1}{a(\phi_k)} \left( w_m - \frac{\partial A(\eta)}{\partial \eta} \right) (\mathbf{U}_{e_{m1}} \circ \mathbf{V}_{e_{m2}}) \right. \\
&\quad \left. - \frac{\frac{\partial}{\partial \eta_{e_{m1}e_{m2}k}} \Pr(w_m \neq 0 | \eta_{e_{m1}e_{m2}k})}{\Pr(w_m \neq 0 | \eta_{e_{m1}e_{m2}k})} (\mathbf{U}_{e_{m1}} \circ \mathbf{V}_{e_{m2}}) \right] - \frac{\text{diag}(\Lambda_k)}{\lambda} \\
\frac{\partial Q}{\partial Y_k} &= \sum_{m=1}^M p_{mk} (\mathbf{U}_{e_{m1}} + \mathbf{V}_{e_{m2}} - \frac{s_{uk}}{f_{uk}} - \frac{s_{vk} - e^{R_{1e_{m1}} + V_{e_{m1}} Y'_k} \mathbf{V}_{e_{m1}}}{f_{vk} - e^{R_{1e_{m1}} + V_{e_{m1}} Y'_k}}) - \mathbf{Y}_k \\
\frac{\partial Q}{\partial \beta_k} &= \sum_{m=1}^M p_{mk} \left[ \frac{1}{a(\phi_k)} \left( w_m - \frac{\partial A(\eta)}{\partial \eta} \right) - \frac{\frac{\partial}{\partial \eta_{e_{m1}e_{m2}k}} \Pr(w_m \neq 0 | \eta_{e_{m1}e_{m2}k})}{\Pr(w_m \neq 0 | \eta_{e_{m1}e_{m2}k})} \right] \\
\frac{\partial Q}{\partial \phi_k} &= \sum_{m=1}^M p_{mk} \left[ \frac{\frac{\partial}{\partial \phi_k} h(w_m, \phi_k)}{h(w_m, \phi_k)} - \frac{\eta_{e_{m1}e_{m2}k} w_m - A(\eta)}{a^2(\phi_k)} \frac{d a(\phi_k)}{d \phi_k} - \frac{\frac{\partial}{\partial \phi_k} \Pr(w_m \neq 0 | \eta_{e_{m1}e_{m2}k})}{\Pr(w_m \neq 0 | \eta_{e_{m1}e_{m2}k})} \right] - \frac{(v_0 + 1) \phi_k}{v_0 \eta_0^2 + \phi_k^2}
\end{aligned} \tag{B.1}$$

Where in the example of normal distribution, we have:

$$\begin{aligned} \Pr(w \neq 0 | \eta_{ijk}) &= 1, \\ (\partial A(\eta_{ijk})) / (\partial \eta_{ijk}) &= \eta_{ijk}, \\ \frac{\frac{\partial}{\partial \eta_{ijk}} \Pr(w_m \neq 0 | \eta_{ijk})}{\Pr(w_m \neq 0 | \eta_{ijk})} &= 0, \\ \frac{\partial a(\phi_k)}{\partial \phi_k} &= 2\phi_k, \\ \frac{\partial \log(h(w_m, \phi_k))}{\partial \phi_k} &= \frac{w_m^2}{\phi_k^3} - \frac{1}{\phi_k} \end{aligned}$$

Due to the potentially high computation cost, we introduce the following quantities to increase the estimation efficiency. It has been proven that, with these precomputed quantities, the time of computing these derivations can be reduced from  $O_{(nM)}$  to  $O_{(M+n)}$  (Sewell, 2021):

$$\begin{aligned} H_k &:= \sum_{m=1}^M \frac{p_{mk}}{f_{vk} - e^{R_{1e_m} + V_{e_m} Y'_k}}, \\ s_{uk} &:= \sum_{i=1}^n e^{S_{1i} + U_i Y'_k} \mathbf{U}_i, \\ \text{and } s_{vk} &:= \sum_{i=1}^n e^{R_{1i} + V_i Y'_k} \mathbf{V}_i. \end{aligned}$$

### Appendix C. The analytical solutions

The full conditionals of covariance-related variables  $\{\hat{\Sigma}_{SR1}, \hat{\Sigma}_{SR2}, \hat{\Sigma}_{UV}, \alpha_k, \lambda\}$  can be found as follows:

$$\begin{aligned} \Sigma_{SR1} | S_1, R_1 &\sim IW(\Psi_{nSR1}, v_{nSR1}), \\ \Sigma_{SR2} | S_2, R_2 &\sim IW(\Psi_{nSR2}, v_{nSR2}), \\ \Sigma_{UV} | U, V &\sim IW(\Psi_{nUV}, v_{nUV}), \\ \alpha | \cdot &\sim Dir(\alpha_0 + \sum_m p_{m1}, \alpha_0 + \sum_m p_{m2}, \dots, \alpha_0 + \sum_m p_{mk}), \\ \lambda | \cdot &\sim \Gamma^{-1} \left( a_0 + \frac{Kp}{2}, b_0 + \frac{1}{2} \sum_k \|\text{diag}(\Lambda_k)\|^2 \right), \end{aligned}$$

with current estimations,  $\{\Sigma_{SR1}, \Sigma_{SR2}, \Sigma_{UV}, \lambda, \alpha\}$  are from analytical solutions:

$$\begin{aligned} \hat{\alpha}_k &= \frac{\alpha_0 + \sum_m p_{mk} - 1}{K\alpha_0 + M - K}, \quad \hat{\lambda} = \frac{b_0 + \frac{1}{2} \sum_k \|\text{diag}(\Lambda_k)\|^2}{a_0 + \frac{Kp}{2} + 1}. \\ \hat{\Sigma}_{SR1} &= \frac{\Psi_{nSR1}}{v_{nSR1} + 3}, \quad \hat{\Sigma}_{SR2} = \frac{\Psi_{nSR2}}{v_{nSR2} + 3}, \quad \hat{\Sigma}_{UV} = \frac{\Psi_{nUV}}{v_{nUV} + 3}, \\ \text{where: } \Psi_{nSR1} &= \Psi_{0SR1} + \sum_{i=1}^n \begin{pmatrix} S_{1i}^2 & S_{1i} R_{1i} \\ S_{1i} R_{1i} & R_{1i}^2 \end{pmatrix}, \text{ and } v_{nSR1} = v_{0SR1} + n; \\ \Psi_{nSR2} &= \Psi_{0SR2} + \sum_{i=1}^n \begin{pmatrix} S_{2i}^2 & S_{2i} R_{2i} \\ S_{2i} R_{2i} & R_{2i}^2 \end{pmatrix}, \text{ and } v_{nSR2} = v_{0SR2} + n; \\ \Psi_{nUV} &= \Psi_{0UV} + \sum_{i=1}^n (\mathbf{U}'_i \quad \mathbf{V}'_i)' (\mathbf{U}_i \quad \mathbf{V}_i), \quad v_{nUV} = v_{0UV} + np. \end{aligned}$$

### References

- Aicher, C., Jacobs, A.Z., Clauset, A., 2015. Learning latent block structure in weighted networks. *J. Complex Netw.* 3, 221–248.
- Amelio, A., Pizzati, C., 2014. Overlapping community discovery methods: a survey. In: *Social Networks: Analysis and Case Studies*, pp. 105–125.
- Banfield, J.D., Raftery, A.E., 1993. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 803–821.
- Ben-David, S., Haghtalab, N., 2014. Clustering in the presence of background noise. In: *International Conference on Machine Learning*. PMLR, pp. 280–288.

- Berardo, R., Fischer, M., Hamilton, M., 2020. Collaborative governance and the challenges of network-based research. *Am. Rev. Public Adm.* **50**, 898–913.
- Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., West, M., et al., 2003. The variational Bayesian em algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian Stat.* **7**, 210.
- Biernacki, C., Celeux, G., Govaert, G., 2000. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 719–725.
- Butts, C.T., 2009. Revisiting the foundations of network analysis. *Science* **325**, 414–416.
- Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A., 2005. Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **2005**, P09008.
- Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc., Ser. B, Methodol.* **39**, 1–22.
- Donker, T., Wallinga, J., Grundmann, H., 2010. Patient referral patterns and the spread of hospital-acquired infections through national health care networks. *PLoS Comput. Biol.* **6**, e1000715.
- Eddelbuettel, D., Sanderson, C., 2014. Rcpparmadillo: accelerating r with high-performance C++ linear algebra. *Comput. Stat. Data Anal.* **71**, 1054–1063. <https://doi.org/10.1016/j.csda.2013.02.005>.
- Evans, T.S., Lambiotte, R., 2009. Line graphs, link partitions, and overlapping communities. *Phys. Rev. E* **80**, 016105.
- Evans, T.S., Lambiotte, R., 2010. Line graphs of weighted networks for overlapping communities. *Eur. Phys. J. B* **77**, 265–272.
- Handcock, M.S., Raftery, A.E., Tantrum, J.M., 2007. Model-based clustering for social networks. *J. R. Stat. Soc., Ser. A, Stat. Soc.* **170**, 301–354.
- Hoff, P.D., Raftery, A.E., Handcock, M.S., 2002. Latent space approaches to social network analysis. *J. Am. Stat. Assoc.* **97**, 1090–1098.
- Kalinka, A.T., Tomancak, P., 2011. Linkcomm: an r package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type. *Bioinformatics* **27**, 2011–2012. <https://doi.org/10.1093/bioinformatics/btr311>.
- Karrer, B., Newman, M.E.J., 2011. Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **83**, 016107. <https://doi.org/10.1103/PhysRevE.83.016107>. <https://link.aps.org/doi/10.1103/PhysRevE.83.016107>.
- Kernighan, B.W., Lin, S., 1970. An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. J.* **49**, 291–307.
- Khan, B.S., Niazi, M.A., 2017. Network community detection: a review and visual survey. *arXiv preprint arXiv:1708.00977*.
- Newman, M.E., Girvan, M., 2004. Finding and evaluating community structure in networks. *Phys. Rev. E* **69**, 026113.
- Ng, A., Jordan, M., Weiss, Y., 2001. On spectral clustering: analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* **14**.
- Nowicki, K., Snijders, T.A.B., 2001. Estimation and prediction for stochastic blockstructures. *J. Am. Stat. Assoc.* **96**, 1077–1087.
- Opsahl, T., Panzarasa, P., 2009. Clustering in weighted networks. *Soc. Netw.* **31**, 155–163.
- Pham, H.T., Sewell, D.K., 2024. Automated detection of edge clusters via an overfitted mixture prior. *Netw. Sci.*, 1–19.
- R Core Team, 2023. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- Sewell, D.K., 2021. Model-based edge clustering. *J. Comput. Graph. Stat.* **30**, 390–405.
- Tian, Y., Lubberts, Z., Weber, M., 2023. Mixed-membership community detection via line graph curvature. In: NeurIPS Workshop on Symmetry and Geometry in Neural Representations. PMLR, pp. 219–233.
- Von Luxburg, U., 2007. A tutorial on spectral clustering. *Stat. Comput.* **17**, 395–416.
- Yoshida, T., 2013. Weighted line graphs for overlapping community discovery. *Soc. Netw. Anal. Min.* **3**, 1001–1013.



Contents lists available at ScienceDirect

## Computational Statistics and Data Analysis

journal homepage: [www.elsevier.com/locate/csda](http://www.elsevier.com/locate/csda)



# JANE: Just Another latent space NEtwork clustering algorithm

Alan T. Arakkal \*, Daniel K. Sewell

Department of Biostatistics, University of Iowa, College of Public Health Building, 145 N. Riverside Drive, Iowa City, IA 52242, USA

## ARTICLE INFO

**Keywords:**

Clustering  
Community detection  
Network analysis  
Latent space cluster model  
EM algorithm

## ABSTRACT

While latent space network models have been a popular approach for community detection for over 15 years, major computational challenges remain, limiting the ability to scale beyond small networks. The R statistical software package, JANE, introduces a new estimation algorithm with massive speedups derived from: (1) a low dimensional approximation approach to adjust for degree heterogeneity parameters; (2) an approximation of intractable likelihood terms; (3) a fast initialization algorithm; and (4) a novel set of convergence criteria focused on clustering performance. Additionally, the proposed method addresses limitations of current implementations, which rely on a restrictive spherical-shape assumption for the prior distribution on the latent positions; relaxing this constraint allows for greater flexibility across diverse network structures. A simulation study evaluating clustering performance of the proposed approach against state-of-the-art methods shows dramatically improved clustering performance in most scenarios and significant reductions in computational time — up to 45 times faster compared to existing approaches.

## 1. Introduction

Network analysis involves the study of information arising from relations and interconnections (i.e., edges) from a set of interacting units (i.e., actors). Social networks, a class of networks where the actors represent a social unit, typically share a common set of key features, including homophily (i.e., an edge between actors is more likely to occur among actors sharing similar characteristics), reciprocity (i.e., if there is a directed edge from actor *A* to actor *B*, then there is an increased likelihood of an edge from *B* to *A*), transitivity (i.e., if edges exist among pairs of actors *AB* and *BC*, then there is an increased likelihood of an edge to occur among actors *A* and *C*), and clustering (i.e., groups or clusters where actors are more densely connected to one another than to actors in other clusters). Analyzing social network data can provide valuable insights into community structures, influential actors, and the flow of information through the network.

Numerous methods exist to model network data within a statistical framework. One such model is the latent space model by Hoff et al. (2002). The latent space model, specifically the distance model, assumes that each actor in the network has a latent position in an unobserved *D*-dimensional social space that captures unobserved attributes of the actor. The probability of an edge forming between two actors is modeled such that it is inversely related to the distance between the actors' latent positions. The latent space model inherently accounts for reciprocity, transitivity, and homophily on observed and unobserved attributes. Handcock et al. (2007) integrated clustering into the latent space model through a clever specification of the prior on the actors' latent positions, specifically

\* Corresponding author.

E-mail addresses: [alan.arakkal@uiowa.edu](mailto:alan.arakkal@uiowa.edu) (A.T. Arakkal), [daniel-sewell@uiowa.edu](mailto:daniel-sewell@uiowa.edu) (D.K. Sewell).

<https://doi.org/10.1016/j.csda.2025.108228>

Received 3 December 2024; Received in revised form 1 May 2025; Accepted 28 May 2025

Available online 2 June 2025

0167-9473/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

a finite mixture of  $K$  spherical multivariate normal distributions. Krivitsky et al. (2009) extended the latent space cluster model to account for degree heterogeneity (i.e., the tendency of some actors to send and/or receive more connections than others) through the inclusion of random effects.

Estimation of the latent space cluster models of Handcock et al. (2007) and Krivitsky et al. (2009) both rely on a Bayesian approach via a Markov chain Monte Carlo (MCMC) algorithm. The MCMC approach requires computing the likelihood during each iteration, which is of the scale  $\mathcal{O}(N^2)$  with respect to computational time, where  $N$  is the number of actors in the network. Thus, the computational time is prohibitively slow for even medium-sized networks. A few approaches have been developed to improve on the computational burden of fitting latent space cluster models. Raftery et al. (2012) proposed using a case-control likelihood approximation to make the cost of evaluating the likelihood linear, rather than quadratic, in  $N$ . That is, the likelihood includes all edges (where most of the information lies) but uses a Monte Carlo approximation of terms involving the unconnected dyads of the network (i.e., non-edges). An alternative approach proposed by Salter-Townshend and Murphy (2013) utilizes a variational Bayes (VB) inference method to perform estimation on the latent space cluster model, thereby avoiding MCMC altogether. Through the use of three first order Taylor expansions, they were able to derive a tractable approximation to the expectation of the log-likelihood with respect to the variational posterior. Using the tractable approximation, parameter estimation proceeds through an iterative algorithm, which is still of the scale  $\mathcal{O}(N^2)$  with respect to computation time, but involves far less computational overhead compared to a sampling-based MCMC algorithm.

In this paper, we develop an expectation–maximization (EM) algorithm for the estimation of latent space cluster models. Our primary objective is to obtain accurate clustering results using latent space cluster models, while attenuating the computational burden associated with fitting these models. Unlike Handcock et al. (2007), Krivitsky et al. (2009), and Salter-Townshend and Murphy (2013), we do not make a restrictive spherical-shape assumption on the finite mixture of  $K$  multivariate normal distributions for the prior on the actors' latent positions. The importance of addressing this issue has been highlighted in the published discussion of Handcock et al. (2007) (see comments by Robinson, Forster, Hennig, and Lawson). Moreover, we incorporate several additional techniques to further improve computational efficiency: (1) we propose a low dimensional approximation approach to adjust for degree heterogeneity parameters; (2) we implement an approximation to the intractable likelihood; (3) we utilize a simple graphical neural network (Liu and Zhou, 2020) approach to initialize starting values for the EM algorithm; and (4) we develop a set of convergence criteria for the EM algorithm that focuses on the clustering aspect of the latent space cluster model.

The remainder of this paper is as follows: Section 2 gives the model, Section 3 provides the details of the proposed estimation approach, Section 4 presents a simulation study, Section 5 demonstrates an application to Twitter network data, and Section 6 gives a discussion. The algorithms described in this paper are implemented in an R (R Core Team, 2021) package, JANE, available on the CRAN repository (<https://CRAN.R-project.org/package=JANE>).

## 2. Model

We represent a network of  $N$  actors using an  $N \times N$  adjacency matrix,  $\mathbf{A}$ , with entries  $A_{ij}$  depicting the edge from actor  $i$  to actor  $j$ . The diagonal entries of  $\mathbf{A}$  are defined to be 0 to prevent self-loops. While edges in the network can be weighted or unweighted through the judicious use of link functions (see, e.g., Sewell and Chen (2016)), we primarily focus on unweighted networks. Furthermore, networks can be directed or undirected, as in the edges between actors can be directional or bidirectional. In what follows, we focus on directed networks, as it is trivial to reduce the model to the undirected case (and is accommodated in the R package JANE).

We use logistic regression to model the probability of an edge between two actors, i.e.,  $\text{logit}(P(A_{ij} = 1 | \beta_0, \mathbf{u}_i, \mathbf{u}_j)) = \beta_0 - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2$ , where  $\mathbf{u}_i$  denotes the  $i^{\text{th}}$  actor's latent position in a  $D$ -dimensional latent space and  $\beta_0$  is an intercept parameter controlling network density. We elect to use the squared Euclidean distance between the latent positions of actors  $i$  and  $j$  as it simplifies the derivations for the EM algorithm and has nice theoretical properties (Rastelli et al., 2016; Gollini and Murphy, 2016). To account for degree heterogeneity, Krivitsky et al. (2009) includes actor-specific random sender and receiver effects, which we will denote by  $s_i$  and  $r_i$ , respectively. The likelihood for the random sender and receiver (RSR) model is as follows:

$$\pi(\mathbf{A} | \mathbf{U}, \beta_0, \mathbf{s}, \mathbf{r}) = \prod_{i=1}^N \prod_{j \neq i} \left[ \left( \exp \left\{ \beta_0 + s_i + r_j - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 \right\} \right)^{A_{ij}} \left( \frac{1}{1 + \exp \left\{ \beta_0 + s_i + r_j - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 \right\}} \right) \right],$$

where  $\mathbf{U}$  is the  $N \times D$  matrix stacking the  $N$   $\mathbf{u}_i$ 's. While accounting for important network features that could otherwise distort the cluster structure, these sender and receiver effects require the estimation of an additional  $2N$  parameters, further exacerbating the computational challenges of estimating the latent space cluster model. However, if we allow that the observed degree of the network can be well approximated by a bijective smooth function of the degree heterogeneity effects, we can significantly reduce the dimensionality of the problem by approximating that function's inverse using splines. Hence we propose approximating the actor-specific sender and receiver effects through a set of natural cubic splines with  $\zeta$  interior knots on an actor's out- and in-degree, letting  $s_i = \mathbf{h}(\deg_i^{(\text{out})})^\top \boldsymbol{\gamma}_s$  and  $r_i = \mathbf{h}(\deg_i^{(\text{in})})^\top \boldsymbol{\gamma}_r$ , where  $\mathbf{h}(\cdot)$  are the  $\zeta + 1$  basis functions for the natural cubic spline with  $\zeta$  interior knots, and  $\deg_i^{(\text{out})}$  and  $\deg_i^{(\text{in})}$  denote the  $i^{\text{th}}$  actor's out- and in-degree, respectively. Here the intercept is not included in the basis expansion as it is unidentifiable in the likelihood. Using this representation, we have

$$\pi(\mathbf{A} | \mathbf{U}, \mathbf{X}, \boldsymbol{\beta}) = \prod_{i=1}^N \prod_{j \neq i} \left[ \left( \exp \left\{ \mathbf{x}_{ij}^\top \boldsymbol{\beta} - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 \right\} \right)^{A_{ij}} \left( \frac{1}{1 + \exp \left\{ \mathbf{x}_{ij}^\top \boldsymbol{\beta} - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 \right\}} \right) \right], \quad (1)$$

where  $\mathbf{x}_{ij} = (\mathbf{1}, \mathbf{h}(\deg_i^{(\text{out})})^\top, \mathbf{h}(\deg_j^{(\text{in})})^\top)^\top$  and  $\boldsymbol{\beta} = (\beta_0, \gamma_s^\top, \gamma_r^\top)^\top$ . Notice that the natural cubic spline approximation approach suffers from an endogeneity problem, as  $P(A_{ij} = 1)$  is a function of the degree of actors  $i$  and  $j$ , which in itself is a function of  $\mathbf{A}$ . This will lead to biased estimates of  $\boldsymbol{\beta}$ . However, since our main goal is clustering, we are willing to sacrifice accuracy in estimating the intercept and actor-specific degree heterogeneity effects in favor of faster clustering results.

To fit a latent space cluster model to an undirected network with no degree heterogeneity (NDH), we simply specify  $s_i = 0$  and  $r_i = 0$  in (1) for  $i = 1, \dots, N$ . Furthermore, for undirected networks with degree heterogeneity, we fit the random sociality (RS) model by setting  $r_i = s_i$  in (1) for  $i = 1, \dots, N$ .

## 2.1. Prior distributions

To integrate clustering into the model, we follow Handcock et al. (2007) and specify a finite mixture of  $K$  multivariate normal distributions as the prior on the actors' latent positions. However, unlike existing approaches that make a spherical-shape assumption on the multivariate normal distributions, which may not be appropriate for all network structures, here we make no such constraints, i.e.,  $\mathbf{u}_i \stackrel{i.i.d.}{\sim} \sum_{k=1}^K p_k \text{MVN}_D(\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k^{-1})$  for  $i = 1, \dots, N$ , where  $\mathbf{p} = (p_1, \dots, p_K)$  are the mixture weights, and  $\boldsymbol{\mu}_k$  and  $\boldsymbol{\Omega}_k$  are the  $D \times 1$  mean and  $D \times D$  positive definite precision matrix of the  $k^{\text{th}}$  multivariate normal mixture component, respectively. As commonly done with the estimation of mixture models, we introduce latent variables  $\mathbf{z}_i = (z_{i1}, \dots, z_{iK})^\top$ , where  $z_{ik} = 1$  if  $\mathbf{u}_i$  belongs to cluster  $k$  and 0 otherwise, letting the  $\mathbf{z}_i$ 's be iid multinomial random variables with probability vector  $\mathbf{p}$ .

The remaining prior distributions are specified as follows:

$$\begin{aligned} \boldsymbol{\beta} &\sim \text{MVN}_{\dim(\boldsymbol{\beta})}(\mathbf{e}, \mathbf{F}^{-1}), \\ \mathbf{p} &\sim \text{Dirichlet}(\mathbf{v}), \\ \boldsymbol{\Omega}_k &\stackrel{i.i.d.}{\sim} \text{Wishart}_D(c, \mathbf{G}^{-1}) \text{ for } k = 1, \dots, K, \text{ and} \\ \boldsymbol{\mu}_k | \boldsymbol{\Omega}_k &\stackrel{i.i.d.}{\sim} \text{MVN}_D(\mathbf{a}, (\mathbf{b}\boldsymbol{\Omega}_k)^{-1}) \text{ for } k = 1, \dots, K, \end{aligned}$$

where  $\mathbf{a}_{D \times 1}$ ,  $\mathbf{b}$ ,  $c$ ,  $\mathbf{e}_{\dim(\boldsymbol{\beta}) \times 1}$ ,  $\mathbf{F}_{\dim(\boldsymbol{\beta}) \times \dim(\boldsymbol{\beta})}$ ,  $\mathbf{G}_{D \times D}$ , and  $\mathbf{v}_{K \times 1}$  are user-specified hyperparameters.

## 3. Estimation

### 3.1. Expectation–maximization algorithm

To estimate the cluster assignments, we employ a generalized EM algorithm. The objective function for the expectation (E)-step is defined as

$$\begin{aligned} Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(t)}) &= \log(\pi(\mathbf{A} | \mathbf{U}, \mathbf{X}, \boldsymbol{\beta})) + \left[ \sum_{i=1}^N \sum_{k=1}^K \left[ \hat{z}_{ik}^{(t)} \left[ \log(p_k) - \frac{D}{2} \log(2\pi) + \frac{1}{2} \log(|\boldsymbol{\Omega}_k|) - \frac{1}{2} (\mathbf{u}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Omega}_k (\mathbf{u}_i - \boldsymbol{\mu}_k) \right] \right] \right] \\ &\quad + \log(\pi(\boldsymbol{\theta} \setminus \{\mathbf{U}\})), \end{aligned} \tag{2}$$

where  $t$  denotes the  $t^{\text{th}}$  iteration of the EM algorithm,  $\boldsymbol{\theta} = \{\mathbf{U}, \boldsymbol{\beta}, \mathbf{p}, \{\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k\}_{k=1}^K\}$ ,  $\hat{z}_{ik}^{(t)} = \frac{p_k^{(t)} \phi(\mathbf{u}_i^{(t)}, \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Omega}_k^{(t)-1})}{\sum_{m=1}^K p_m^{(t)} \phi(\mathbf{u}_i^{(t)}, \boldsymbol{\mu}_m^{(t)}, \boldsymbol{\Omega}_m^{(t)-1})}$ , and  $\phi(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$  is the multivariate normal density.

For the maximization (M)-step of the EM algorithm, we find the parameters that maximize (2), i.e.,  $\boldsymbol{\theta}^{(t+1)} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(t)})$ . Closed-form updates can be derived in a straightforward manner for  $\mathbf{p}$  and  $\{\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k\}_{k=1}^K$ . Closed-form updates for  $\mathbf{U}$  and  $\boldsymbol{\beta}$  can also be derived by utilizing a first-order and second-order Taylor expansion of the log-likelihood, respectively. In particular, we take the corresponding Taylor expansions with respect to the following term in the log-likelihood,  $-\log(1 + \exp\{\eta_{ij}(\boldsymbol{\beta}, \mathbf{u}_i, \mathbf{u}_j)\})$ , where  $\eta_{ij}(\boldsymbol{\beta}, \mathbf{u}_i, \mathbf{u}_j)$  varies by model (e.g., for the RSR model,  $\eta_{ij}(\boldsymbol{\beta}, \mathbf{u}_i, \mathbf{u}_j) = \mathbf{x}_{ij}^\top \boldsymbol{\beta} - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2$ ). See Section A.1 for the closed-form update equations and details on the log-likelihood approximation.

#### 3.1.1. Initialization

While in general EM and VB algorithms tend to be sensitive to initial values due to getting trapped in local optima, this is particularly problematic for the high-dimensional latent space model, and is highly unlikely to produce reasonable clustering estimates if the initialization is not chosen thoughtfully. Additionally, selecting initial values close to the global optimum can significantly reduce the number of iterations required for convergence, thereby decreasing computation time. Therefore, effective initialization is essential to avoid convergence to poor local optima and to allow for faster convergence of the algorithm. The initialization method of Salter-Townshend and Murphy (2013), implemented in the VBLPCM R package, combines the Fruchterman–Reingold (FR) method (Fruchterman and Reingold, 1991) for initializing latent positions and model-based clustering to initialize clustering parameters. Although this approach provides effective initializations for the variational parameters, its computational cost escalates with increasing

network size. The main bottleneck of VBLPCM's initialization approach is attributed to the FR method, which has an  $\mathcal{O}(N^2)$  time complexity and is prohibitively slow in generating initial latent positions for medium to large networks.

We propose an alternative method for generating initial values. To initialize the EM algorithm, we utilize a graphical neural network (GNN) approach in conjunction with K-means clustering to generate initial values. We begin by randomly generating a latent position matrix, i.e.,  $\mathbf{U}^{(0)}$ . Next, we perform feature propagation for a fixed set of  $T$  iterations, where at each step, each actor updates its latent position by averaging its neighboring actors' latent positions. Hence with each iteration, the values of the latent positions among linked actors become more similar. Define  $\mathbf{U}^{(T)}$  as the latent position matrix after the  $T^{th}$  iteration. Then, using maximum-likelihood, we fit a constrained logistic regression model on  $\mathbf{U}^{(T)}$  using downsampling to maintain scalability to obtain the initialization for  $\beta$  and a scaling factor for  $\mathbf{U}^{(T)}$ . For example, with the RSR latent space model, we utilize  $\text{logit}(P(A_{ij} = 1 | \beta, \beta_U, \mathbf{x}_{ij}, \mathbf{u}_i^{(T)}, \mathbf{u}_j^{(T)})) = \mathbf{x}_{ij}^\top \beta + \beta_U \|\mathbf{u}_i^{(T)} - \mathbf{u}_j^{(T)}\|_2^2$  and impose a constraint such that  $\beta_U < 0$  to preserve the inverse relationship with the distance of the actors' latent positions. The rescaled  $\mathbf{U}^{(T)}$ , i.e.,  $\sqrt{-\beta_U} \times \mathbf{U}^{(T)}$ , is set as the initialization for the latent positions. This is then followed by using K-means with  $K$  clusters on the rescaled  $\mathbf{U}^{(T)}$  to obtain the initializations for  $\{\mu_k, \Omega_k\}_{k=1}^K$  and  $p$ . The choice of  $T$  is a very important component; additional details for determining  $T$  involving the Brier score, evaluating  $\beta_U$ , and integrating downsampling can be found in Section A.2.

We attempted alternative initialization approaches, including a maximum likelihood embedding method proposed by O'Connor et al. (2020). However, the alternative initialization approaches considered failed to provide adequate starting values, resulting in poor clustering and/or computational performance.

### 3.1.2. Convergence

The EM algorithm proceeds by iterating between the E-step and the M-step until convergence is achieved. Determining convergence of the EM algorithm in the setting of latent space cluster models is a non-trivial task. This is illustrated by the unique convergence criteria utilized by VBLPCM: for a given iteration of the VB algorithm, if the change in a particular variational parameter between the previous and current iteration is less than some user-specified threshold, then the optimal solution for that variational parameter is assumed to be attained and is retained for all proceeding iterations until convergence for the remaining variational parameters is achieved. This parameter-wise convergence criteria may be problematic as it ignores the circular dependencies found in the updates of the variational parameters.

Convergence criteria not only have an impact on clustering results, but also on computational performance. Overly strict convergence criteria may lead to an excessive number of iterations, while too lenient criteria risk premature termination, potentially compromising the quality of the solutions. Thus, a balance is needed to ensure accurate parameter estimation while minimizing computation time. We developed a set of criteria to assess convergence that focuses on the clustering aspect of the model, since that is our primary goal. To that end, we determine convergence of the EM algorithm by evaluating changes in the  $N \times K$  matrix  $\hat{\mathbf{Z}} := (\hat{z}_1, \dots, \hat{z}_N)^\top$ , where the  $K \times 1$  vector  $\hat{z}_i$  contains the estimated conditional probabilities that the  $i^{th}$  actor belongs to the  $k^{th}$  cluster (i.e., the  $i^{th}$  actor's cluster membership probabilities). It can be reasonable to assume that if the  $\hat{\mathbf{Z}}$ 's between subsequent iterations of the EM algorithm are similar, then an optimal solution with respect to clustering is attained. While this is a straightforward criteria to implement, in practice determining convergence by evaluating changes in  $\hat{\mathbf{Z}}$  between subsequent iterations is not sufficient. In some cases, oscillations in cluster membership probabilities and latent positions can obfuscate convergence. To combat this issue, we evaluate the stability in the cumulative average of the absolute change in  $\hat{\mathbf{Z}}$  and  $\mathbf{U}$  across iterations. By assessing the cumulative average, we are able to smooth out fluctuations and focus on overall stability with respect to the changes in  $\hat{\mathbf{Z}}$  and  $\mathbf{U}$ . Specific details, along with a decision tree, can be found in Section A.3.1. Section A.3 also contains alternative convergence criteria available in JANE.

### 3.2. Case-control likelihood approximation

We implement the case-control likelihood approximation by Raftery et al. (2012) in our R package, JANE. However, instead of implementing the stratified sampling approach, where the random sampling of non-edges (i.e., controls) is based on geodesic distance, we elect to use a simple random sample. In experiments, utilizing a simple random sample of non-edges seems to have comparable clustering performance, while significantly cutting down on the computational overhead associated with implementing a stratified sampling procedure based on geodesic distances. Moreover, in our implementation of the case-control likelihood approximation, we specify a fixed number of non-edges to sample for each actor, unlike VBLPCM's implementation that uses a fixed proportion. By using a fixed number of non-edges, instead of a fixed proportion, the number of non-edges sampled does not grow with  $N$ .

### 3.3. Choosing the number of clusters

Handcock et al. (2007) proposed a double BIC approach for choosing the number of clusters. Specifically, from all potential number of clusters considered, select the  $K$  that results in the smallest BIC, defined as  $\text{BIC} = \text{BIC}_{\text{logistic regression}} + \text{BIC}_{\text{model-based clustering (MBC)}}$ . However, instead of using BIC for the model-based clustering component, we choose to use the integrated complete likelihood (ICL) (Biernacki et al., 2000). Define  $\tilde{z}_{ik} = 1$  if  $\hat{z}_{ik} = \max_l \hat{z}_{il}$  and 0 otherwise. Biernacki et al. (2000) defined  $\text{ICL}_{\text{MBC}}$  as the  $\text{BIC}_{\text{MBC}}$  plus an additional penalty, i.e.,  $\text{ICL}_{\text{MBC}} = \text{BIC}_{\text{MBC}} + 2E$ , where  $E = -\sum_{i=1}^N \sum_{k=1}^K \tilde{z}_{ik} \log(\hat{z}_{ik}) \geq 0$ . The penalty term,  $E$ , is defined as the entropy of classification, which measures the overlap of clusters. The entropy is largest when all  $\hat{z}_{ik} = 1/K$  for  $i = 1, \dots, N$  and  $k = 1, \dots, K$ , which results in  $E = N \log(K)$ . In contrast, the entropy is smallest when all the  $\hat{z}_{ik}$  are equal to either 0 or 1, which results in  $E = 0$ ,

where  $0 \log(0)$  is defined as 0. As a result,  $\text{ICL}_{\text{MBC}}$  tends to favor models with more distinctly separated clusters than  $\text{BIC}_{\text{MBC}}$ , and so  $\text{ICL}_{\text{MBC}}$  tends to choose the same or smaller number of clusters than  $\text{BIC}_{\text{MBC}}$ . Based on simulation studies, Biernacki et al. (2000) recommends that when the interest in mixture modeling is cluster analysis, instead of density estimation, the  $\text{ICL}_{\text{MBC}}$  criterion should be favored, as the  $\text{BIC}_{\text{MBC}}$  criterion tends to overestimate the number of clusters. Since our main goal is clustering, we elect to use  $\text{ICL}_{\text{MBC}}$ . Thus, to select the number of clusters, we define the  $\text{BICL}$  criterion, denoted as,

$$\text{BICL} = \text{BIC}_{\text{logistic regression}} + \text{ICL}_{\text{MBC}}, \quad (3)$$

where smaller  $\text{BICL}$  values are favored. However, JANE allows users to easily implement model selection based on either  $\text{BIC}$  or  $\text{BICL}$ .

#### 4. Simulation study

We conducted a simulation study to evaluate the clustering performance of our proposed approach against state-of-the-art methods. We examined scenarios where the spherical-shape assumption for the prior distribution on the latent positions is met and when it is violated. Additionally, we assessed performance across networks of varying sizes and densities. Furthermore, we compared the computational speed of our approach with that of competing methods to provide a more comprehensive performance analysis.

##### 4.1. Simulation design

We specified the true dimension of the latent space  $D = 2$  and the number of clusters  $K = 3$ . For each simulation, we drew  $\boldsymbol{p} \sim \text{Dirichlet}(31_3)$  to allow for unbalanced cluster sizes. We then independently simulated  $\mathbf{z}_i$  from a multinomial distribution with probabilities defined by  $\boldsymbol{p}$ . Given  $\mathbf{z}_i$ ,  $\mathbf{u}_i$  was simulated independently from a finite mixture of three multivariate normal distributions with parameters  $\{\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k\}_{k=1}^3$ . To induce a spherical shape with respect to the latent positions, we specified  $\boldsymbol{\mu} = \{(-5/6, -5/6)^T, (5/6, -5/6)^T, (5/6, 5/6)^T\}$  and  $\boldsymbol{\Omega} = \{7\mathbf{I}_2, 7\mathbf{I}_2, 7\mathbf{I}_2\}$ . For an ellipsoidal shape, we specified  $\boldsymbol{\mu} = \{(1, 0)^T, (-1, 1.15)^T, (-1, -1.15)^T\}$  and  $\boldsymbol{\Omega} = \{\text{Diag}(8, 1), \text{Diag}(1, 4), \text{Diag}(1, 4)\}$ . Contour plots of the spherical and ellipsoidal multivariate normal distributions can be found in Section A.4. Random sender and receiver effects were generated from  $(s_i, r_i) \stackrel{i.i.d.}{\sim} \text{MVN}_2(\mathbf{0}_2, 0.4\mathbf{I}_2 + 0.1\mathbf{J}_2)$ . Finally, we simulated a network with  $N$  actors according to an RSR latent space model, i.e.,  $A_{ij} \stackrel{\text{indep.}}{\sim} \text{Bernoulli}\left(\left(1 + \exp\left\{-1 \times (\beta_0 + s_i + r_j - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2)\right\}\right)^{-1}\right)$  for  $i, j = 1, \dots, N$ ,  $j \neq i$ , and  $A_{ii} = 0$  if  $i = j$ .

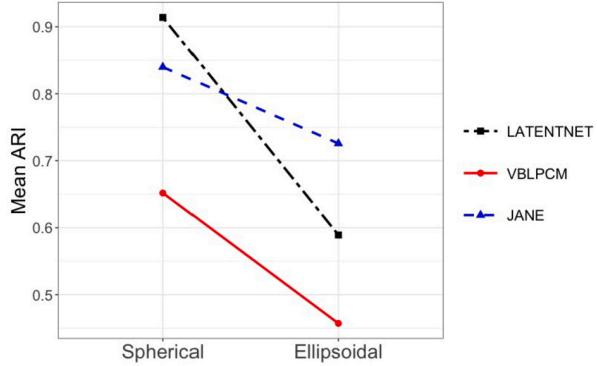
##### 4.2. Competing methods

We compared our proposed approach using the R package JANE against the MCMC-based implementation using the R package latentnet (Krivitsky and Handcock, 2008) and the VB implementation using the R package VBLPCM. We focused specifically on comparing clustering performance using the adjusted Rand index (ARI), with higher values closer to one suggesting stronger agreement between the estimated hard cluster assignments and the true cluster assignments of actors in the network. When fitting the respective approaches, we considered  $K = 2, \dots, 10$  and used  $\text{BIC}$  to select the number of clusters, with JANE using the alternative  $\text{BICL}$  specification given in (3) (with the exception of a  $\text{BIC}$  versus  $\text{BICL}$  comparison presented in Section 4.3). We utilized the defaults for VBLPCM and latentnet, with the exception of increasing the maximum number of iterations to 1000 and requiring analysis of the whole network for VBLPCM to match JANE (the default for VBLPCM is to only analyze the largest connected component of the network). For JANE, we specified  $\zeta = 5$ ,  $\boldsymbol{a} = \mathbf{0}_2$ ,  $b = 1$ ,  $c = 3$ ,  $\boldsymbol{e} = \mathbf{0}_{1+2(5+1)}$ ,  $\boldsymbol{F} = \text{Diag}(1/100, (1/2.5^2)\mathbf{I}_{2(5+1)})$ ,  $\boldsymbol{G} = \mathbf{I}_2$ , and  $\nu = 31_K$ . Finally, with VBLPCM and JANE, we allowed for five separate initializations of the algorithms in an attempt to prevent convergence to local optima. Selection of the best initialization was based on minimum Kullback–Leibler divergence for VBLPCM (as recommended by Salter-Townshend and Murphy (2013)) and minimum  $\text{BICL}$  given by (3) for JANE (with the exception of a  $\text{BIC}$  versus  $\text{BICL}$  comparison presented in Section 4.3). JANE was initialized by the GNN approach described in Section 3.1.1 and VBLPCM was initialized using the default FR method. The VBLPCM package has an alternative random initialization scheme and although the package documentation describes this as a random initialization of the variational parameters, the computational time and clustering performance was similar to using the FR approach. Therefore, we only considered results using the default FR method. All simulations were run on the University of Iowa's high performance computing system, with computing resources constrained to emulate an 8-core machine.

##### 4.3. Impact of violating spherical-shape assumption

For both the spherical- and ellipsoidal-shape specifications, we simulated 50 networks with  $N = 300$  and a density of 0.1. Fig. 1 shows the comparisons of average ARI across simulations by method and shape specification, assuming the true  $K$  was known. For all methods, clustering performance seems to decrease when the true shape specification is ellipsoidal. This is in part due to the greater degree of overlap in the multivariate normals specified for the latent positions (see Section A.4). As expected, latentnet seems to perform the best when the true shape specification is spherical. However, when the true shape specification is ellipsoidal, JANE appears to have the best performance. This highlights the benefits of relaxing the spherical-shape assumption made on the prior covariance structure of the latent positions. In both scenarios, JANE performs substantially better than VBLPCM.

Table 1 shows the performance comparisons when model selection was used to select the number of clusters from  $K \in \{2, \dots, 10\}$ . VBLPCM failed to select the best  $K$  for all simulations due to infinite  $\text{BIC}$  values. This was true for all simulation scenarios considered. JANE and latentnet experienced no such issues. Comparing the results of JANE using  $\text{BICL}$  versus  $\text{BIC}$  for choosing the



**Fig. 1.** Simulation study results of clustering performance by method and shape specification, assuming the true  $K$  was known. Note: ARI = Adjusted Rand Index.

**Table 1**  
Simulation study results of clustering performance by method and shape specification, where the true  $K$  was unknown.

Method	Spherical				Ellipsoidal			
	Mean ARI	% Selected $K = 3$	% Selected $K > 3$	Mean Time (minutes)	Mean ARI	% Selected $K = 3$	% Selected $K > 3$	Mean Time (minutes)
JANE (BICL) <sup>1</sup>	0.77	72.0	10.0	4.5	0.70	66.0	10.0	6.0
JANE (BIC) <sup>2</sup>	0.77	60.0	24.0	4.5	0.68	78.0	18.0	6.0
latentnet <sup>2</sup>	0.91	38.0	62.0	167.6	0.49	2.0	98.0	172.7
VBLPCM <sup>2,3</sup>	-	-	-	46.8	-	-	-	51.8

Note: ARI = Adjusted Rand Index; True  $K = 3$ .

<sup>1</sup> Optimal  $K$  selected using BICL.

<sup>2</sup> Optimal  $K$  selected using BIC.

<sup>3</sup> Missing values as VBLPCM failed to select the optimal  $K$  for all simulations due to infinite BIC values.

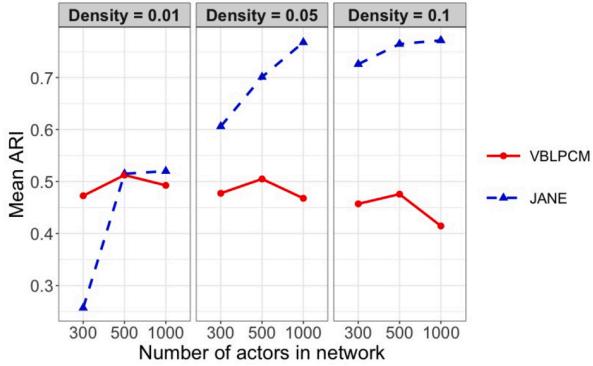
number of clusters, we observed that for both the spherical- and ellipsoidal-shape specifications, BIC tended to favor a larger  $K$  in a greater proportion of the simulations than BICL, consistent with the observations of Biernacki et al. (2000). Additionally, under the ellipsoidal-shape specification, we observed that BIC selected the correct  $K$  in a greater proportion of the simulations than BICL. The ellipsoidal-shape specification considered involves a greater degree of overlap of the multivariate normals (see Section A.4), consequently, as expected BICL tended to favor models with fewer, more well-separated clusters (i.e.,  $K < 3$ ) than BIC, penalizing against overlapping clusters with elevated clustering uncertainty, and favoring clustering configurations with less overlap. However, in general the mean ARI using BIC and BICL to select the optimal  $K$  were nearly identical. When the true shape specification was spherical, in general latentnet had better clustering performance than JANE. However, latentnet selected a  $K > 3$  for 62% of the simulations compared to 10% for JANE. This is expected, as latentnet uses BIC instead of ICL for the model-based clustering component, which has a tendency to overestimate the number of clusters. When the true shape specification was ellipsoidal, JANE produced a higher mean ARI and selected the true  $K$  in a greater proportion of simulations than latentnet. We observed that latentnet seems to especially overestimate the number of clusters when the true shape specification is ellipsoidal, which is expected since more spherical-shape clusters will be needed to approximate an ellipsoidal shape.

#### 4.4. Impact of network size and density

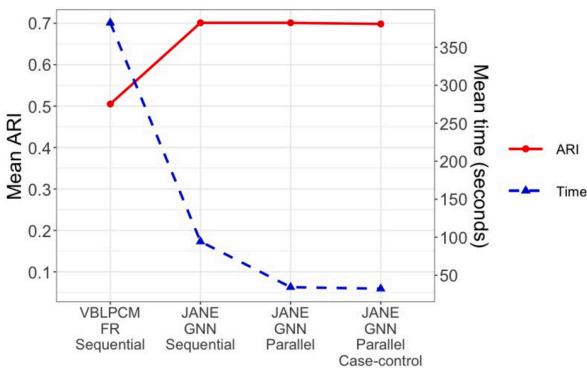
We simulated 50 networks under the ellipsoidal-shape specification for all combinations of network density  $\in \{0.01, 0.05, 0.1\}$  and  $N \in \{300, 500, 1000\}$ . Due to the extremely long computation times associated with latentnet, we only compared VBLPCM and JANE. Fig. 2 compares average ARI across the simulations by method, network density, and number of actors, assuming the true  $K$  was known. VBLPCM had issues fitting the model when density was 0.01. The worst was observed for density = 0.01 and  $N = 300$ , where VBLPCM failed to fit 30 of the 50 simulated networks. JANE encountered no such issues. For a fairer comparison, Fig. 2 only presents the average ARI of simulations where both VBLPCM and JANE were successfully fit. Poor clustering performance was observed with JANE for density = 0.01 and  $N = 300$ . However, in all other scenarios, JANE outperformed VBLPCM. In general, JANE's clustering performance improved as density or as  $N$  increased; troublingly, no such trends were observed for VBLPCM.

#### 4.5. Comparisons of computational performance

We simulated 50 networks under the ellipsoidal-shape specification with  $N = 500$  and a density of 0.05 to compare computational performance between VBLPCM and JANE. Once again, latentnet was excluded due to excessively long computational times. For a more comprehensive understanding of computational performance, we also considered the parallel and case-control approximation implementations of JANE that are readily available in the package. The case-control approximation implementation integrates the



**Fig. 2.** Simulation study results of clustering performance by method with varying network sizes and densities, assuming the true  $K$  was known. Note: ARI = Adjusted Rand Index.



**Fig. 3.** Simulation study results of clustering and computational performance by method, assuming the true  $K$  was known. Note: ARI = Adjusted Rand Index, GNN = Graphical Neural Network, FR = Fruchterman-Reingold.

methodology proposed by Raftery et al. (2012) alongside the novel strategies introduced in this paper. Fig. 3 shows the clustering and computational performance by method, assuming the true  $K$  was known. On average, compared to VBLPCM, all implementations of JANE exhibited superior clustering performance and required a substantially shorter amount of time to fit the models. In fact, the sequential implementation of JANE was on average 75.3% faster than VBLPCM. Parallelizing JANE resulted in additional speedups. The parallelized case-control approximation implementation achieved the fastest average computation time, while maintaining similar clustering performance. However, computation times for the parallel case-control approximation and parallel implementations of JANE were similar. Although the case-control approach required less time per iteration on average (0.03 vs. 0.10 average seconds per iteration), it generally needed more total iterations to converge (944.6 vs. 332.6 average iterations), typically offsetting computational gains.

When model selection was used to select the number of clusters from  $K \in \{2, \dots, 10\}$ , the sequential application of JANE took an average of 13.7 minutes to fit the models (2.1 minutes when parallelized), resulting in an average ARI of 0.65. The parallel case-control approximation implementation required an average of 1.7 minutes to fit the models and achieved an average ARI of 0.66. VBLPCM took an average of 1.6 hours to fit the models, yet failed to select the best  $K$  for all simulations due to infinite BIC values.

Additional simulation studies, including an ablation study and evaluations of the impact of increasing the number of clusters and varying the degree of cluster overlap, can be found in Section A.5.

## 5. Application to Irish politician Twitter network data

To evaluate clustering and computational performance using real-world network data, we utilized a Twitter network of 348 Irish politicians from 2012. The Irish politician Twitter network is a directed and unweighted network, with out-going edges representing who a particular politician (i.e., actor) “follows” (Greene and Cunningham, 2013). The network comprises of 16,856 edges, yielding a density of 0.1396. Each politician is affiliated with one of seven political parties, serving as the ground truth for clustering evaluation: Fianna Fáil (49), Fine Gael (143), Green (7), Independent (31), Labour (79), Sinn Féin (31), and United Left Alliance (8). The assortativity coefficient on political affiliation is 0.52, indicating that the network is assortative. As in the simulation study, we compare the clustering and computational performance of JANE with `latentnet` and VBLPCM, using a setup consistent with Section 4.2. The simulation results in Section 4.5 demonstrate significant computational speedups when using the parallel implementation of JANE, so we employed the parallel approach for the real-data application.

**Table 2**  
Clustering performance results of Irish politician Twitter network data by method.

Method	True $K$ Known		True $K$ Unknown			
	ARI	Time (minutes)	ARI	Selected $K$ <sup>3</sup>	Unique Hard Cluster Assignments <sup>4</sup>	Time (minutes)
JANE <sup>1</sup>	0.88	0.48	0.87	5	4	1.32
latentnet	0.88	26.59	0.87	10	8	237.03
VBLPCM <sup>2</sup>	0.71	4.67	-	-	-	38.19

Note: ARI = Adjusted Rand Index; True  $K$  = 7.

<sup>1</sup> The parallel implementation of JANE was utilized.

<sup>2</sup> Missing values as VBLPCM failed to select the optimal  $K$  due to infinite BIC values.

<sup>3</sup> Optimal number of clusters,  $K$ , selected using BICL for JANE and BIC for VBLPCM and latentnet.

<sup>4</sup> Unique number of clusters based on hard cluster assignments.

**Table 3**  
Confusion matrix of Irish politician Twitter network data using JANE hard cluster assignments, assuming the true  $K$  was unknown.

JANE Clusters	Political Party						
	FF	FG	GREEN	IND	LABOUR	SF	ULA
1	47	0	1	1	0	0	0
2	0	140	0	1	0	0	0
3	2	3	6	26	0	31	8
4	0	0	0	3	79	0	0

Note: FF = Fianna Fáil, FG = Fine Gael, IND = Independent, SF = Sinn Féin, ULA = United Left Alliance.

Table 2 presents the performance comparisons assuming the true  $K$  (= 7) was known, as well as when model selection was used to determine the number of clusters from  $K \in \{2, \dots, 10\}$ . In both scenarios, JANE and latentnet produced nearly identical ARI values. However, latentnet required just over 55 times the amount of time needed for JANE to fit the model when the true  $K$  was known, and nearly 180 times the time required by JANE when the true  $K$  was unknown. Consistent with behavior observed in the simulation study, latentnet selected a greater number of clusters than JANE. However, both JANE and latentnet resulted in fewer number of unique hard cluster assignments than the selected  $K$ . The ARI between the cluster assignments from the optimal JANE and LATENTNET fits was 0.82, indicating a high degree of agreement between the clustering solutions obtained by the two methods. VBLPCM produced a lower ARI than JANE and latentnet when the true  $K$  was known and failed to select the best  $K$  due to infinite BIC values. In both scenarios, VBLPCM required more time to fit the models than JANE — nearly 10 and 30 times the time required by JANE when the true  $K$  was known and unknown, respectively.

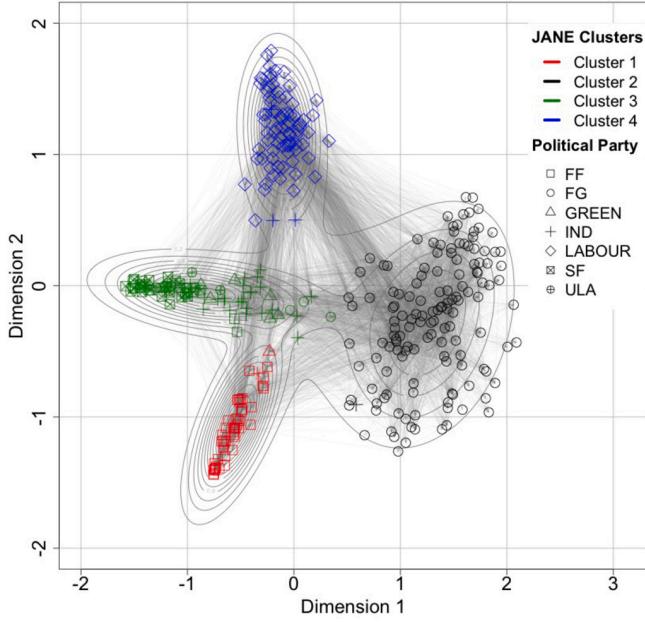
Table 3 shows the confusion matrix between the Irish political parties and the hard cluster assignments produced by JANE, assuming the true  $K$  was unknown. The first, second, and forth JANE clusters mainly comprised of politicians from the Fianna Fáil, Fine Gael, and Labour parties, respectively. In contrast, the third JANE cluster was observed to have a mix of politicians from all parties, with the exception of the Labour party.

Fig. 4 depicts a plot of the estimated latent positions produced by JANE, assuming the true  $K$  was unknown, with actors color-coded by their hard cluster assignments and represented by different shapes according to their party affiliations. Also shown in the plot are the contours of the estimated multivariate normal distributions. Fig. 5 presents the actor-specific hard clustering uncertainty, computed as  $1 - \max_k \hat{z}_{ik}$ ; this quantity was bounded between 0 and  $1 - (1/K)$ , where  $K$  was determined to be 5 by BICL. Approximately 4.0% of all actors were estimated to have a clustering uncertainty  $> 0.1$ , and they were affiliated with the Fianna Fáil (1), Fine Gael (8), Green (2), and Independent (3) parties.

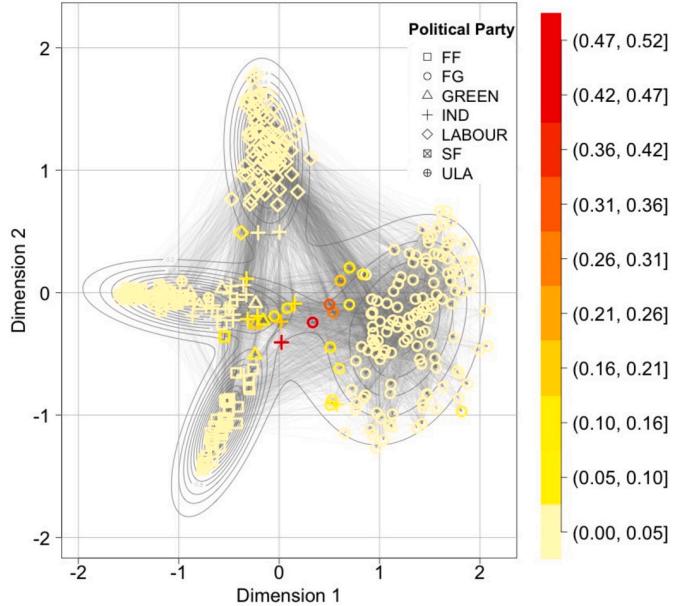
As a sensitivity analysis, we re-ran the application of JANE to the Irish politician Twitter network data using BIC instead of BICL to select the optimal  $K$ . We observed identical final clustering results, which is logical, as based on Fig. 5 the clusters are well separated and the actor-specific hard clustering uncertainty is relatively small for the majority of actors in the network. Subsequently, the penalty term (i.e., entropy of classification) added to the BIC was relatively small, as reflected in the observed metric values, BIC = 50745.36 versus BICL = 50761.15, leading to the selection of the same optimal fit.

## 6. Discussion

We have proposed a fast generalized EM algorithm-based estimation approach to fit latent space cluster models. Our method integrates a low-dimensional approximation approach to adjust for degree heterogeneity parameters, a likelihood approximation method, a fast initialization algorithm, and a novel set of convergence criteria focused on the clustering aspect of the model. Together, as demonstrated in our simulation study, these methods reduce the computational burden of fitting latent space cluster models by up to 45-fold. Our approach also relaxes the restrictive spherical-shape assumption on the prior of the latent positions, which in simulations have shown substantial improvements in clustering performance, especially in situations where the spherical-shape assumption is violated. By relaxing this constraint, our method offers greater flexibility across diverse network structures.



**Fig. 4.** Estimated latent positions and hard cluster assignments of Irish politician Twitter network data using JANE, assuming the true  $K$  was unknown. Note: FF = Fianna Fáil, FG = Fine Gael, IND = Independent, SF = Sinn Féin, ULA = United Left Alliance.



**Fig. 5.** Actor-specific hard clustering uncertainty for the Irish politician Twitter network data using JANE, assuming the true  $K$  was unknown. Note: FF = Fianna Fáil, FG = Fine Gael, IND = Independent, SF = Sinn Féin, ULA = United Left Alliance.

Our estimation approach has some limitations. First, our approach does not perform well on small networks with densities  $\leq 0.01$ . However, in real-world data, smaller networks often have higher densities, a scenario in which our approach performs well. Regardless, our proposed approach is designed to make estimation of latent space cluster models feasible for large networks; for small networks, we recommend using the MCMC-based approach by latentnet.

Second, with respect to the low dimensional approximation approach to adjust for degree heterogeneity parameters, by explicitly linking the sociability parameters with the degree of the actor, there could be potential confounding of the centrality of the latent positions and the magnitude of the sender and receiver effects, which in turn could affect the clustering structure. For example, an actor's observed degree may be induced through different mechanisms: (1) relatively small sociability parameters, but numerous connections due to the centrality of their latent position to other actors in the latent space; or (2) large sociability parameters

inducing many connections, but with a latent position that is distant from other actors in the latent space. Since our low dimensional approximation approach relies on the observed degree, it is challenging to disentangle these characteristics. However, the simulation study presented in the paper includes these confounding scenarios, as we simulated data based on the original degree heterogeneity model by Krivitsky et al. (2009), and our low dimensional approximation approach performed well in spite of these confounding issues. To further highlight the value of the low dimensional approximation approach to adjust for degree heterogeneity parameters, we re-ran our application to the Irish politician Twitter network data using a model that excluded degree heterogeneity parameters. As expected, we observed a complete degradation of clustering performance when ignoring degree heterogeneity. Specifically, using the approach that ignores degree heterogeneity, we obtained an ARI of 0.0 in both scenarios — assuming the true  $K$  ( $= 7$ ) was known and unknown. In contrast, utilizing our proposed low dimensional approximation approach to adjust for degree heterogeneity parameters produced an ARI of 0.88 and 0.87 assuming the true  $K$  was known and unknown, respectively.

Third, our approach focuses primarily on the clustering aspect of the latent space cluster model, and while further work is needed to thoroughly evaluate the accuracy of the other model parameters, the strong clustering performance observed suggests that these parameters are likely to be reasonably accurate. To illustrate this, we fit JANE to 50 simulated networks with 300 actors and a density of 0.1, under the spherical- and ellipsoidal-shape specifications. We assumed the true  $K$  was known and compared the mean Spearman correlation of the true and estimated latent position euclidean distances of all pairs of actors across all simulated networks. We observed a mean correlation coefficient of 0.92 and 0.95 for the networks under the spherical- and ellipsoidal-shape specifications, respectively. The strong correlation suggests that our approach was able to recover the latent distances of the actors reasonably well.

Fourth, with the additional simulation studies presented in Section A.5, for both latentnet and JANE we observed that the number of clusters selected was sensitive to the specification of the hyperparameters of the prior distributions, especially with respect to the Dirichlet prior on the mixture weights of the mixture model. Future work should explore principled approaches for hyperparameter selection.

Finally, the EM algorithm is sensitive to initial parameter values. As typically done with EM algorithms, we try to combat this issue by using multiple starting values, which is computationally inefficient if one lacks the resources to perform this embarrassingly parallelizable task. To overcome this issue, in work not included here, we also considered a deterministic annealing EM (DAEM) algorithm (i.e., deterministic variant of simulated annealing) (Ueda and Nakano, 1998). Currently, JANE can leverage the DAEM algorithm and while some initial experiments have shown promising results, further work is needed to fine-tune the implementation of the DAEM algorithm.

## Appendix A

### A.1. Closed-form update equations

$$\underline{\mathbf{u}_i^{(t+1)}}$$

#### NDH:

Take the first-order Taylor expansion of  $-\log(1 + \exp\{\eta_{ij}(\beta_0, \mathbf{u}_i, \mathbf{u}_j)\})$  about  $\mathbf{u}_i^{(t)}$  evaluated at  $\mathbf{u}_i$ , where  $\eta_{ij}(\beta_0, \mathbf{u}_i, \mathbf{u}_j) = \beta_0 - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2$ . Use this approximation in (2) and maximize with respect to  $\mathbf{u}_i$ :

$$\mathbf{u}_i^{(t+1)} = \left( \sum_{k=1}^K [\hat{z}_{ik}^{(t)} \Omega_k^{(t)}] + \sum_{j \neq i} [2A_{ij}] I_D \right)^{-1} \left( \sum_{k=1}^K [\hat{z}_{ik}^{(t)} \Omega_k^{(t)} \mu_k^{(t)}] + \sum_{j \neq i} [2A_{ij} \mathbf{u}_j] + \sum_{j \neq i} \left[ \frac{2 \exp\{\eta_{ij}(\beta_0^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j)\}}{1 + \exp\{\eta_{ij}(\beta_0^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j)\}} \right] (\mathbf{u}_i^{(t)} - \mathbf{u}_j) \right)$$

for  $i = 1, \dots, N$  where,

$$\mathbf{u}_j = \begin{cases} \mathbf{u}_j^{(t)}, & \text{if } j > i \\ \mathbf{u}_j^{(t+1)}, & \text{if } j < i \end{cases}$$

#### RS:

Take the first-order Taylor expansion of  $-\log(1 + \exp\{\eta_{ij}(\beta, \mathbf{u}_i, \mathbf{u}_j)\})$  about  $\mathbf{u}_i^{(t)}$  evaluated at  $\mathbf{u}_i$ , where  $\eta_{ij}(\beta, \mathbf{u}_i, \mathbf{u}_j) = \mathbf{x}_{ij}^\top \beta - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2$ ,  $\mathbf{x}_{ij(1+(\zeta+1)) \times 1} = (1, \mathbf{h}(\deg_i)^\top + \mathbf{h}(\deg_j)^\top)^\top$ ,  $\beta_{(1+(\zeta+1)) \times 1} = (\beta_0, \gamma^\top)^\top$ , and  $\mathbf{h}(\deg_i)^\top \gamma$  is the linear basis expansion for a natural cubic spline with  $\zeta$  interior knots, excluding an intercept, on the degree of actor  $i$ . Use this approximation in (2) and maximize with respect to  $\mathbf{u}_i$ :

$$\mathbf{u}_i^{(t+1)} = \left( \sum_{k=1}^K [\hat{z}_{ik}^{(t)} \Omega_k^{(t)}] + \sum_{j \neq i} [2A_{ij}] I_D \right)^{-1} \left( \sum_{k=1}^K [\hat{z}_{ik}^{(t)} \Omega_k^{(t)} \mu_k^{(t)}] + \sum_{j \neq i} [2A_{ij} \mathbf{u}_j] + \sum_{j \neq i} \left[ \frac{2 \exp\{\eta_{ij}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j)\}}{1 + \exp\{\eta_{ij}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j)\}} \right] (\mathbf{u}_i^{(t)} - \mathbf{u}_j) \right)$$

for  $i = 1, \dots, N$  where,

$$\mathbf{u}_j = \begin{cases} \mathbf{u}_j^{(t)}, & \text{if } j > i \\ \mathbf{u}_j^{(t+1)}, & \text{if } j < i \end{cases}$$

**RSR:**

Take the first-order Taylor expansion of  $-\log(1 + \exp\{\eta_{ij}(\beta, \mathbf{u}_i, \mathbf{u}_j)\})$  about  $\mathbf{u}_i^{(t)}$  evaluated at  $\mathbf{u}_i$ , where  $\eta_{ij}(\beta, \mathbf{u}_i, \mathbf{u}_j) = \mathbf{x}_{ij}^\top \beta - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2$ ,  $\mathbf{x}_{ij(1+2(\zeta+1)) \times 1} = (1, \mathbf{h}(\deg_i^{(out)})^\top, \mathbf{h}(\deg_j^{(in)})^\top)^\top$ ,  $\beta_{(1+2(\zeta+1)) \times 1} = (\beta_0, \gamma_s^\top, \gamma_r^\top)^\top$ , and  $\mathbf{h}(\deg_i^{(out)})^\top \gamma_s$  and  $\mathbf{h}(\deg_i^{(in)})^\top \gamma_r$  are the linear basis expansions for a natural cubic spline with  $\zeta$  interior knots, excluding an intercept, on the out- and in-degree of actor  $i$ , respectively. Use this approximation in (2) and maximize with respect to  $\mathbf{u}_i$ :

$$\begin{aligned}\mathbf{u}_i^{(t+1)} = & \left( \sum_{k=1}^K \left[ \hat{z}_{ik}^{(t)} \Omega_k^{(t)} \right] + \sum_{j \neq i} [2(A_{ij} + A_{ji})] I_D \right)^{-1} \times \\ & \left( \sum_{k=1}^K \left[ \hat{z}_{ik}^{(t)} \Omega_k^{(t)} \mu_k^{(t)} \right] + \sum_{j \neq i} [2(A_{ij} + A_{ji}) \mathbf{u}_j] + \sum_{j \neq i} \left[ 2(p_{ij}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j) + p_{ji}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j)) (\mathbf{u}_i^{(t)} - \mathbf{u}_j) \right] \right)\end{aligned}$$

for  $i = 1, \dots, N$  where,

$$\begin{aligned}& p_{ij}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j) = \frac{\exp\{\mathbf{x}_{ij}^\top \beta^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j\|_2^2\}}{1 + \exp\{\mathbf{x}_{ij}^\top \beta^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j\|_2^2\}}, \quad \mathbf{x}_{ij} = \begin{bmatrix} 1 \\ \mathbf{h}(\deg_i^{(out)}) \\ \mathbf{h}(\deg_j^{(in)}) \end{bmatrix} \\ & p_{ji}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j) = \frac{\exp\{\mathbf{x}_{ji}^\top \beta^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j\|_2^2\}}{1 + \exp\{\mathbf{x}_{ji}^\top \beta^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j\|_2^2\}}, \quad \mathbf{x}_{ji} = \begin{bmatrix} 1 \\ \mathbf{h}(\deg_j^{(out)}) \\ \mathbf{h}(\deg_i^{(in)}) \end{bmatrix} \\ & \mathbf{u}_j = \begin{cases} \mathbf{u}_j^{(t)}, & \text{if } j > i \\ \mathbf{u}_j^{(t+1)}, & \text{if } j < i \end{cases}\end{aligned}$$

$$\underline{\beta^{(t+1)}}$$

**NDH:**

Take the second-order Taylor expansion of  $-\log(1 + \exp\{\eta_{ij}(\beta_0, \mathbf{u}_i, \mathbf{u}_j)\})$  about  $\beta_0^{(t)}$  evaluated at  $\beta_0$ , where  $\eta_{ij}(\beta_0, \mathbf{u}_i, \mathbf{u}_j) = \beta_0 - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2$ . Use this approximation in (2) and maximize with respect to  $\beta_0$ :

$$\beta_0^{(t+1)} = \frac{\left( f e + \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i} [A_{ij}] + \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i} \left[ p_{ij}(\beta_0^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) (1 - p_{ij}(\beta_0^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \beta_0^{(t)} - p_{ij}(\beta_0^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \right] \right)}{\left( f + \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i} \left[ p_{ij}(\beta_0^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) (1 - p_{ij}(\beta_0^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \right] \right)}$$

where,

$$p_{ij}(\beta_0^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) = \frac{\exp\{\beta_0^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}\|_2^2\}}{1 + \exp\{\beta_0^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}\|_2^2\}}$$

**RS:**

Take the second-order Taylor expansion of  $-\log(1 + \exp\{\eta_{ij}(\beta, \mathbf{u}_i, \mathbf{u}_j)\})$  about  $\beta^{(t)}$  evaluated at  $\beta$ , where  $\eta_{ij}(\beta, \mathbf{u}_i, \mathbf{u}_j) = \mathbf{x}_{ij}^\top \beta - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2$ ,  $\mathbf{x}_{ij(1+(\zeta+1)) \times 1} = (1, \mathbf{h}(\deg_i)^\top + \mathbf{h}(\deg_j)^\top)^\top$ ,  $\beta_{(1+(\zeta+1)) \times 1} = (\beta_0, \gamma^\top)^\top$ , and  $\mathbf{h}(\deg_i)^\top \gamma$  is the linear basis expansion for a natural cubic spline with  $\zeta$  interior knots, excluding an intercept, on the degree of actor  $i$ . Use this approximation in (2) and maximize with respect to  $\beta$ :

$$\begin{aligned}\beta^{(t+1)} = & \left( F + \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i} \left[ p_{ij}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) (1 - p_{ij}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \mathbf{x}_{ij} \mathbf{x}_{ij}^\top) \right] \right)^{-1} \times \\ & \left( F e + \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i} [A_{ij} \mathbf{x}_{ij}] + \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i} \left[ p_{ij}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) (1 - p_{ij}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \mathbf{x}_{ij} \mathbf{x}_{ij}^\top \beta^{(t)} - p_{ij}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \mathbf{x}_{ij}) \right] \right)\end{aligned}$$

where,

$$p_{ij}(\beta^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) = \frac{\exp\{\mathbf{x}_{ij}^\top \beta^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}\|_2^2\}}{1 + \exp\{\mathbf{x}_{ij}^\top \beta^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}\|_2^2\}}$$

**RSR:**

Take the second-order Taylor expansion of  $-\log(1 + \exp\{\eta_{ij}(\beta, \mathbf{u}_i, \mathbf{u}_j)\})$  about  $\beta^{(t)}$  evaluated at  $\beta$ , where  $\eta_{ij}(\beta, \mathbf{u}_i, \mathbf{u}_j) = \mathbf{x}_{ij}^\top \beta - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2$ ,  $\mathbf{x}_{ij(1+2(\zeta+1)) \times 1} = (1, \mathbf{h}(\deg_i^{(out)})^\top, \mathbf{h}(\deg_j^{(in)})^\top)^\top$ ,  $\beta_{(1+2(\zeta+1)) \times 1} = (\beta_0, \gamma_s^\top, \gamma_r^\top)^\top$ , and  $\mathbf{h}(\deg_i^{(out)})^\top \gamma_s$  and  $\mathbf{h}(\deg_i^{(in)})^\top \gamma_r$  are the

linear basis expansions for a natural cubic spline with  $\zeta$  interior knots, excluding an intercept, on the out- and in-degree of actor  $i$ , respectively. Use this approximation in (2) and maximize with respect to  $\beta$ :

$$\begin{aligned} \boldsymbol{\beta}^{(t+1)} = & \left( \mathbf{F} + \sum_{i=1}^N \sum_{j \neq i} \left[ p_{ij}(\boldsymbol{\beta}^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \left( 1 - p_{ij}(\boldsymbol{\beta}^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \right) \mathbf{x}_{ij} \mathbf{x}_{ij}^\top \right] \right)^{-1} \times \\ & \left( \mathbf{F}\mathbf{e} + \sum_{i=1}^N \sum_{j \neq i} [A_{ij} \mathbf{x}_{ij}] + \sum_{i=1}^N \sum_{j \neq i} \left[ p_{ij}(\boldsymbol{\beta}^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \left( 1 - p_{ij}(\boldsymbol{\beta}^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \right) \mathbf{x}_{ij} \mathbf{x}_{ij}^\top \beta^{(t)} - p_{ij}(\boldsymbol{\beta}^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) \mathbf{x}_{ij} \right] \right) \end{aligned}$$

where,

$$p_{ij}(\boldsymbol{\beta}^{(t)}, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)}) = \frac{\exp \left\{ \mathbf{x}_{ij}^\top \boldsymbol{\beta}^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}\|_2^2 \right\}}{1 + \exp \left\{ \mathbf{x}_{ij}^\top \boldsymbol{\beta}^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}\|_2^2 \right\}}$$

$\underline{\mathbf{p}}^{(t+1)}$   
Maximize (2) with respect to  $\mathbf{p}$  utilizing a Lagrange multiplier:

$$p_k^{(t+1)} = \frac{\sum_{i=1}^N [\hat{z}_{ik}^{(t)}] + v_k - 1}{\sum_{m=1}^K \left[ \sum_{i=1}^N [\hat{z}_{im}^{(t)}] + v_m - 1 \right]} = \frac{\sum_{i=1}^N [\hat{z}_{ik}^{(t)}] + v_k - 1}{N - K + \sum_{m=1}^K [v_m]} \text{ for } k = 1, \dots, K$$

$\underline{\boldsymbol{\mu}}_k^{(t+1)}$   
Maximize (2) with respect to  $\boldsymbol{\mu}_k$ :

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{i=1}^N [\hat{z}_{ik}^{(t)} \mathbf{u}_i^{(t)}] + ba}{\sum_{i=1}^N [\hat{z}_{ik}^{(t)}]} \text{ for } k = 1, \dots, K$$

$\underline{\boldsymbol{\Omega}}_k^{(t+1)}$   
Maximize (2) with respect to  $\boldsymbol{\Omega}_k$ :

$$\boldsymbol{\Omega}_k^{(t+1)} = \left( \sum_{i=1}^N [\hat{z}_{ik}^{(t)}] + c - d \right) \left( \mathbf{G} + \sum_{i=1}^N [\hat{z}_{ik}^{(t)} \mathbf{u}_i^{(t)} \mathbf{u}_i^{(t)\top}] + baa^\top - \left( \sum_{i=1}^N [\hat{z}_{ik}^{(t)}] + b \right) \boldsymbol{\mu}_k^{(t)} \boldsymbol{\mu}_k^{(t)\top} \right)^{-1} \text{ for } k = 1, \dots, K$$

## A.2. Initialization method

Let  $T$  denote the total number of iterations specified for the initialization approach and let  $t$  denote the current iteration of the initialization algorithm. For some adjacency matrix  $\mathbf{A}$ , construct a row and column normalized matrix denoted as  $\mathbf{A}_{RN}$  and  $\mathbf{A}_{CN}$ , respectively, i.e.,

$$\begin{aligned} \mathbf{A}_{RN} &:= \text{diag} \left( 1/\text{Deg}_1^{(out)}, \dots, 1/\text{Deg}_N^{(out)} \right) \times \mathbf{A}, \\ \mathbf{A}_{CN} &:= \mathbf{A} \times \text{diag} \left( 1/\text{Deg}_1^{(in)}, \dots, 1/\text{Deg}_N^{(in)} \right), \end{aligned}$$

where  $\text{Deg}_i^{(out)}$  and  $\text{Deg}_i^{(in)}$  are the out- and in-degree of actor  $i$ , respectively. Additionally, define  $1/\text{Deg}_i^{(out)} = 1$  if  $\text{Deg}_i^{(out)} = 0$  and  $1/\text{Deg}_i^{(in)} = 1$  if  $\text{Deg}_i^{(in)} = 0$ . Let  $\mathbf{U}^{(t)}$  denote the averaged values of the latent positions across linked actors at iteration  $t$ , i.e.,

$$\mathbf{U}^{(t)} := \frac{1}{2} (\mathbf{A}_{RN} \mathbf{U}^{(t-1)} + \mathbf{A}_{CN}^\top \mathbf{U}^{(t-1)}).$$

The pseudocode for the basic initialization algorithm is shown below. Let  $\theta^{start}$  denote the resulting starting value for the parameter  $\theta$  generated by the initialization algorithm.

---

Input:  $\mathbf{A}$ ,  $N$ ,  $K$ ,  $D$ ,  $T$ , model (NDH, RS, or RSR);

- 1: Generate random  $\mathbf{U}_{N \times D}^{(0)}$  from  $\text{MVN}_D(\mathbf{0}, \mathbf{I})$
- 2: Construct  $\mathbf{A}_{RN}$  and  $\mathbf{A}_{CN}$
- 3:    for  $t = 1, \dots, T$
- 4:        $\mathbf{U}^{(t)} = \frac{1}{2} (\mathbf{A}_{RN} \mathbf{U}^{(t-1)} + \mathbf{A}_{CN}^\top \mathbf{U}^{(t-1)})$
- 5:    end for
- 6: Fit constrained logistic regression on  $\mathbf{U}^{(T)}$ , with logit based on model. Obtain  $\boldsymbol{\beta}^{start}$  and  $\mathbf{U}^{start} = \sqrt{-\beta_U} \times \mathbf{U}^{(T)}$
- 7: Fit K-means with  $K$  clusters on  $\mathbf{U}^{start}$  and obtain  $\{\boldsymbol{\mu}_k^{start}, \boldsymbol{\Omega}_k^{start}\}_{k=1}^K$  and  $\mathbf{p}^{start}$

---

The logit used for the logistic regression fit in step 6 of the pseudocode above is model dependent. For example, with the RSR latent space model, we utilize  $\text{logit}(P(A_{ij} = 1 | \beta, \beta_U, \mathbf{x}_{ij}, \mathbf{u}_i^{(T)}, \mathbf{u}_j^{(T)})) = \mathbf{x}_{ij}^\top \boldsymbol{\beta} + \beta_U \|\mathbf{u}_i^{(T)} - \mathbf{u}_j^{(T)}\|_2^2$ . The constraint imposed in the MLE fit relates to the  $\beta_U$  coefficient. Specifically, we constrain  $\beta_U < 0$  to preserve the inverse relationship with the distance of the actors' latent positions. Furthermore, define  $\{\mu_{k_{DX1}}^{start}\}_{k=1}^K$  and  $\mathbf{z}_{N \times 1}^{start}$  as the cluster centers and cluster assignments resulting from the K-means fit with  $K$  clusters using  $\mathbf{U}^{start}$ , respectively. We define  $\Omega_{k_{DXD}}^{start}$  as the sample precision matrix using observations  $\mathbf{u}_i^{start} \forall i \in \{h | z_h^{start} = k\}$  and  $p_k^{start} = \frac{|\{h | z_h^{start} = k\}|}{N}$ , for  $k = 1, \dots, K$ .

Determining the adequate number of iterations to repeatedly average the latent positions across linked actors is important. Too many iterations gives rise to large marginal precisions with respect to  $\mathbf{U}^{start}$  and results in poor performance with the EM algorithm. Here we describe an approach to determine the optimal number of iterations. Define  $T^* \leq T$  as the optimal number of iterations. To determine  $T^*$ , within each iteration of the initialization algorithm, fit a logistic regression model on  $\mathbf{U}^{(t)}$  with the logit specified as  $\text{logit}(P(A_{ij} = 1 | \beta_0, \beta_U, \mathbf{u}_i^{(t)}, \mathbf{u}_j^{(t)})) = \beta_0 + \beta_U \|\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}\|_2^2$ . Then, compute the Brier score of the fit at each iteration, denoted as  $BS^{(t)}$ . Next, define  $T^* = \{v | BS^{(v)} = \min_t BS^{(t)}\}$  for  $t, v = 1, \dots, T$  and replace  $\mathbf{U}^{(T)}$  with  $\mathbf{U}^{(T^*)}$  in step 6 of the initialization algorithm pseudocode.

An additional check is put in place to evaluate the adequacy of  $\mathbf{U}^{start}$ . For the model-dependent logistic regression fit to determine the scaling factor of  $\mathbf{U}^{(T^*)}$  (i.e., step 6 of the initialization algorithm pseudocode), a one-sided Wald-type test is performed to evaluate  $H_0: \beta_U = 0$  vs.  $H_1: \beta_U < 0$ . If we fail to reject the null at  $\alpha = 0.05$ , then we restart the initialization algorithm with a new randomly generated  $\mathbf{U}^{(0)}$  and reduce the total number of iterations,  $T$ , by 50%.

Finally, to improve on computational performance, we perform downsampling on the respective logistic regression fits to allow for a balanced design, such that all links are considered and only a random sample of non-links are utilized. This results in an  $\mathcal{O}(N + |E|)$  time complexity with respect to the proposed initialization algorithm, where  $|E|$  is the number of edges in the network.

### A.3. Convergence criteria

#### A.3.1. Stability in $\hat{\mathbf{Z}}$ and $\mathbf{U}$ (default in JANE)

Let  $\epsilon > 0$  be a tolerance threshold. For some matrix  $\mathbf{M}$  with elements  $M_{ij}$ , let  $\mathbf{M}^{(t)}$  denote the value of  $\mathbf{M}$  at the  $t^{th}$  iteration of the EM algorithm after both the E and M steps. Let  $|M^{(t)}|_q$  be a scalar denoting the  $q^{th}$  quantile of the absolute difference of the elements of  $\mathbf{M}^{(t)}$  and  $\mathbf{M}^{(t-1)}$ , i.e.,  $|M_{ij}^{(t)} - M_{ij}^{(t-1)}|$ . Let  $CA_M^{(t)}$  represent the cumulative average of  $|M^{(t)}|_q$  starting at some predefined iteration  $t^*$  that acts as a burn-in period, i.e.,

$$CA_M^{(t)} := \frac{\sum_{s=t^*}^t |M^{(s)}|_q}{t - t^* + 1}.$$

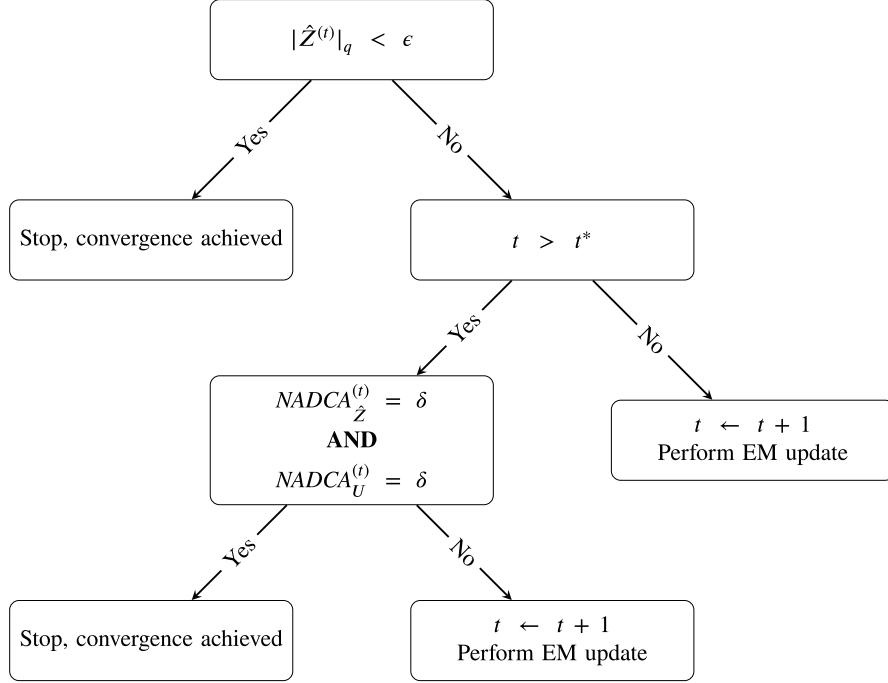
Let  $ADCA_M^{(t)}$  be the absolute difference in the cumulative average of  $|M^{(t)}|_q$  between the current and previous iteration, i.e.,

$$ADCA_M^{(t)} := |CA_M^{(t)} - CA_M^{(t-1)}|.$$

Finally, for some user-specified look-back iterations  $\delta \geq 1$ , let  $NADCA_M^{(t)}$  be the number of the most recent  $\delta$  iterations that display stability, defined by

$$NADCA_M^{(t)} := \left| \left\{ w \mid \max(t - \delta + 1, t^*) \leq w \leq t, ADCA_M^{(w)} < \epsilon \right\} \right|.$$

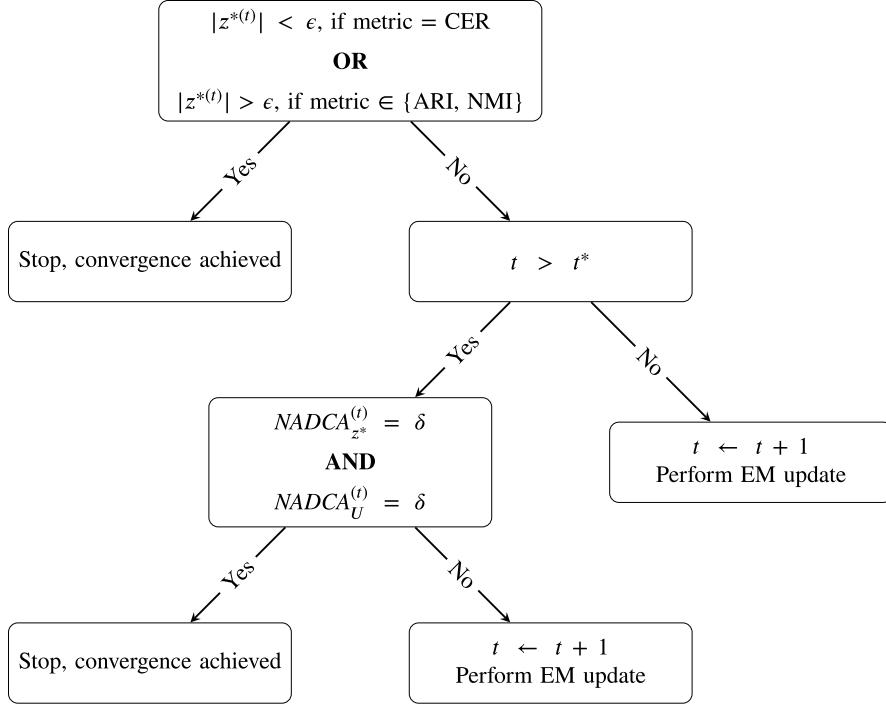
The flowchart below depicts the convergence decision rule based on user inputs,  $\epsilon$ ,  $q$ ,  $t^*$ , and  $\delta$ .



We recommend  $q$  be set to 1, i.e., maximum. Smaller  $q$  results in a less stringent threshold for convergence (i.e., at least  $100(q)\%$  of the element-wise absolute differences of  $\mathbf{M}^{(t)}$  and  $\mathbf{M}^{(t-1)}$  are less than  $\epsilon$ ). As mentioned previously, there are situations where oscillations in the cluster membership probabilities and latent positions can hinder convergence when using the simple criteria of  $|\hat{Z}^{(t)}|_q < \epsilon$ . To combat this issue, we look at how the cumulative average of  $|\hat{Z}^{(t)}|_q$  and  $|U^{(t)}|_q$  changes across iterations. By looking at the cumulative average, we are able to smooth out fluctuations and focus on overall stability with respect to the changes in  $|\hat{Z}^{(t)}|_q$  and  $|U^{(t)}|_q$  from  $t^*$  up to  $t$ . We allow for a burn-in period,  $t^*$ , (e.g., 20 iterations) as it is expected that  $|\hat{Z}^{(t)}|_q$  and  $|U^{(t)}|_q$  will be unstable during the first few iterations of the EM algorithm. We assess stability with respect to the moving average by looking at the absolute difference between iterations. Specifically, we determine stability by evaluating if the absolute difference in the respective moving averages between iterations is less than some threshold (i.e.,  $\epsilon$ ) for  $\delta$  (e.g., 5) consecutive iterations. The  $\delta$  consecutive iterations requirement is for added confidence that stability is achieved, which is useful if there are dramatic changes in  $|\hat{Z}^{(t)}|_q$  and  $|U^{(t)}|_q$ . Smaller  $\delta$  results in a less stringent assessment of stability.

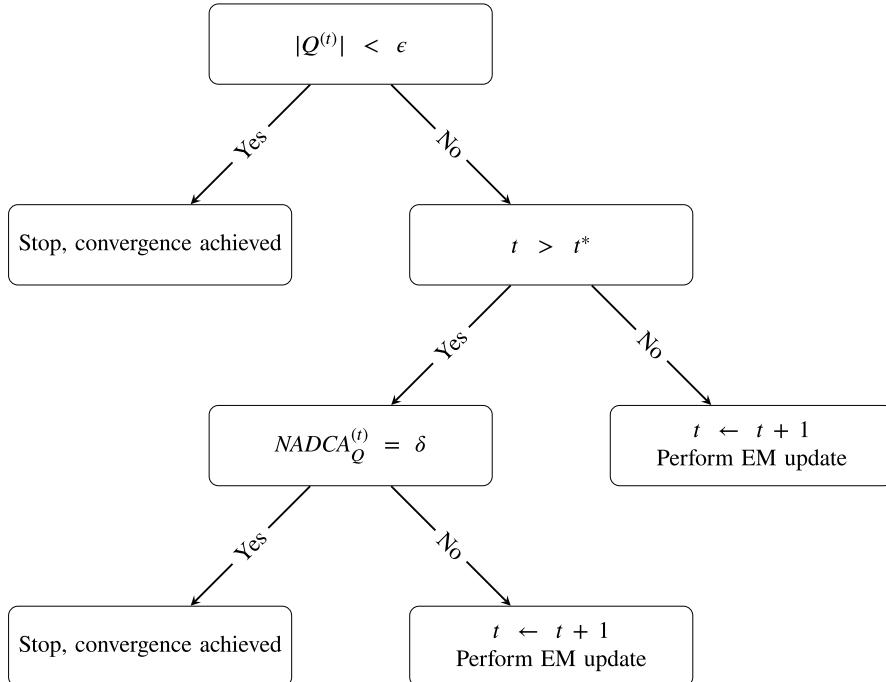
### A.3.2. Stability in $\mathbf{z}^*$ and $\mathbf{U}$

Instead of assessing changes in  $\hat{\mathbf{Z}}$  between iterations of the EM algorithm, here the convergence criteria focuses on  $\mathbf{z}_{N \times 1}^* := (z_1^*, \dots, z_N^*)^\top$ , where  $z_i^* \in \{1, \dots, K\}$  is defined as  $z_i^* = \{l | \hat{z}_{il} = \max_k \hat{z}_{ik}\}$  (i.e., the hard cluster assignment for actor  $i$ ). We use the same definitions as in Section A.3.1 and introduce two more definitions. Let  $|z^{*(t)}|$  be a scalar metric that measures similarity (i.e., adjusted Rand index [ARI], normalized mutual information [NMI], or classification error rate [CER]) between  $\mathbf{z}^{*(t)}$  vs.  $\mathbf{z}^{*(t-1)}$  and let  $CA_{z^*}^{(t)} := \frac{\sum_{s=t^*}^t |z^{*(s)}|}{t-t^*+1}$ . The flowchart below depicts the convergence decision rule based on user inputs,  $\epsilon$ ,  $q$ ,  $t^*$ , and  $\delta$ .



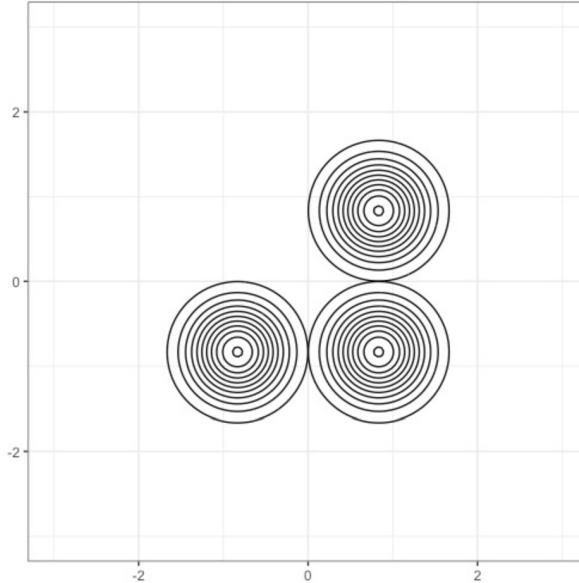
### A.3.3. Stability in $Q(\theta|\theta^{(t)})$

Here the convergence criteria focuses on the changes in the objective function,  $Q(\theta|\theta^{(t)})$ , evaluated using parameters from subsequent iterations. With this convergence criteria, we do not track stability with respect to  $U$ , as we assume that information is captured in  $Q(\theta|\theta^{(t)})$ . We use the same definitions as in Section A.3.1 and introduce two more definitions. Let  $|Q^{(t)}|$  be the absolute difference between  $Q(\theta^{(t)}|\theta^{(t)})$  and  $Q(\theta^{(t-1)}|\theta^{(t-1)})$ , i.e.,  $|Q(\theta^{(t)}|\theta^{(t)}) - Q(\theta^{(t-1)}|\theta^{(t-1)})|$  and let  $CA_Q^{(t)} := \frac{\sum_{s=t^*}^t |Q^{(s)}|}{t-t^*+1}$ . The flowchart below depicts the convergence decision rule based on user inputs,  $\epsilon$ ,  $t^*$ , and  $\delta$ .

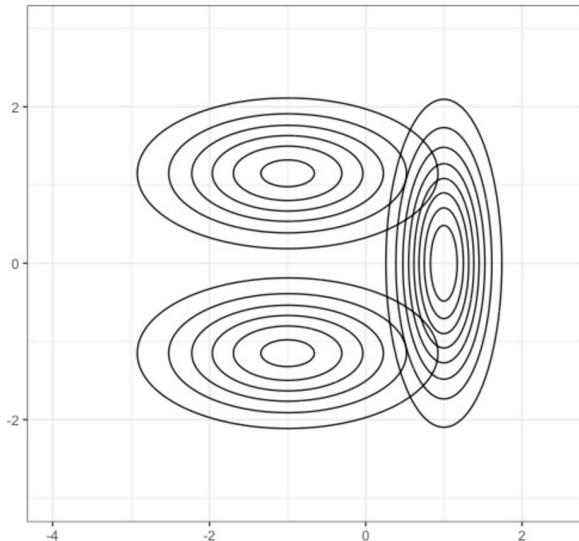


*A.4. Multivariate normal contour plots for spherical- and ellipsoidal-shape specifications used in the simulation study*

**Spherical**



**Ellipsoidal**



*A.5. Additional simulation studies*

*A.5.1. Ablation study*

An ablation study was conducted where we systematically evaluated the impact on clustering and computational performance of three key strategies implemented in JANE: (1) a low dimensional approximation approach to adjust for degree heterogeneity parameters; (2) a graphical neural network approach to initialize starting values for the EM algorithm; and (3) a set of convergence criteria for the EM algorithm that focuses on the clustering aspect of the latent space cluster model. Specifically, we simulated 50 networks under the ellipsoidal-shape specification with a network density of 0.1 and considered  $N \in \{300, 500, 1000, 1500\}$ , assuming the true  $K$  was known. For each combination, we evaluated the mean ARI and mean time required to fit the models across all simulations using the following implementations of JANE:

1. The unmodified JANE algorithm, which we label as “Original”.
2. The JANE algorithm initialized using the Fruchterman–Reingold (FR) method utilized by VBLPCM, instead of the proposed GNN initialization approach. This is labeled as “FR start”.
3. The JANE algorithm with convergence evaluated using a similar approach as implemented in VBLPCM. Specifically, we determined convergence based on parameter-wise changes between iterations. For  $\beta$ ,  $p$ , and  $\{\mu_k, \Omega_k\}_{k=1}^K$  we evaluated changes in the maximum element-wise absolute differences between iterations, whereas for  $\mathbf{U}$  we evaluated changes in the maximum absolute differences of the L2 norm squared distance metric,  $\|\mathbf{u}_i - \mathbf{u}_j\|_2^2$ , between iterations for all actor pairs in the network. Similar to the approach utilized by VBLPCM, if a particular parameter achieves convergence, then the optimal solution for that parameter is assumed to be attained and is retained for all proceeding iterations until convergence for the remaining parameters is achieved. This is labeled as “VBLPCM like convergence”.
4. The JANE algorithm without utilizing the low dimensional approximation approach to adjust for degree heterogeneity parameters, which we label as “Non-spline approach”. For this implementation we return to an approach we initially considered to fit the degree heterogeneity models, which is based on the following likelihood:

$$\pi(\mathbf{A}|\mathbf{U}, \beta_0, \mathbf{s}, \mathbf{r}) = \prod_{i=1}^N \prod_{j \neq i} \left[ \left( \exp \left\{ \beta_0 + s_i + r_j - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 \right\} \right)^{A_{ij}} \left( \frac{1}{1 + \exp \left\{ \beta_0 + s_i + r_j - \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 \right\}} \right) \right],$$

where we assume  $(s_i, r_i) \stackrel{i.i.d.}{\sim} N(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \boldsymbol{\omega} = \begin{bmatrix} \omega_s^2 & \omega_{sr} \\ \omega_{sr} & \omega_r^2 \end{bmatrix})$  and  $\boldsymbol{\omega}$  is the precision matrix. Utilizing a flat Wishart prior on  $\boldsymbol{\omega}$  (i.e.,  $\boldsymbol{\omega} \sim \text{Wishart}(\text{df} = 2 + 1, \text{scale} = \mathbf{I}_2)$ ), a close-form update for  $\boldsymbol{\omega}$  was derived in a straightforward manner, whereas closed-form updates for  $\{\mathbf{s}_i, \mathbf{r}_i\}_{i=1}^N$  were derived using respective second-order Taylor expansions of the log-likelihood. The specific update equations are as follows:

$$\begin{aligned} \boldsymbol{\omega}^{(t+1)} &= N \left( \sum_{i=1}^N [\mathbf{x}_i^{(t)} \mathbf{x}_i^{(t)\top}] + \mathbf{I}_2 \right)^{-1}, \\ s_i^{(t+1)} &= \frac{\left( \sum_{j \neq i} [p_{ij}(s_i^{(t)}) (1 - p_{ij}(s_i^{(t)})) s_i^{(t)} - p_{ij}(s_i^{(t)}) + A_{ij}] - r_i^{(t)} \omega_{sr}^{(t)} \right)}{\left( \sum_{j \neq i} [p_{ij}(s_i^{(t)}) (1 - p_{ij}(s_i^{(t)}))] + \omega_s^{2(t)} \right)}, \\ r_i^{(t+1)} &= \frac{\left( \sum_{j \neq i} [p_{ji}(r_i^{(t)}) (1 - p_{ji}(r_i^{(t)})) r_i^{(t)} - p_{ji}(r_i^{(t)}) + A_{ji}] - s_i^{(t)} \omega_{sr}^{(t)} \right)}{\left( \sum_{j \neq i} [p_{ji}(r_i^{(t)}) (1 - p_{ji}(r_i^{(t)}))] + \omega_r^{2(t)} \right)}, \end{aligned}$$

where  $\mathbf{x}_i^{(t)} = (s_i^{(t)}, r_i^{(t)})^\top$ ,  $p_{ij}(s_i^{(t)}) = \frac{\exp\{\beta_0^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}\|_2^2 + s_i^{(t)} + r_j^{(t)}\}}{1 + \exp\{\beta_0^{(t)} - \|\mathbf{u}_i^{(t)} - \mathbf{u}_j^{(t)}\|_2^2 + s_i^{(t)} + r_j^{(t)}\}}$ , and  $p_{ji}(r_i^{(t)}) = \frac{\exp\{\beta_0^{(t)} - \|\mathbf{u}_j^{(t)} - \mathbf{u}_i^{(t)}\|_2^2 + s_j^{(t)} + r_i^{(t)}\}}{1 + \exp\{\beta_0^{(t)} - \|\mathbf{u}_j^{(t)} - \mathbf{u}_i^{(t)}\|_2^2 + s_j^{(t)} + r_i^{(t)}\}}$ .

Fig. A.1 presents the clustering and computational performance results of the ablation study by network size. Compared to utilizing the proposed GNN approach to initialize parameters, the FR method seems to result in slightly worse clustering performance and increased computational time, especially as the network size grows. Our novel convergence criteria resulted in a similar clustering performance as the VBLPCM like convergence criteria, however required significantly less time to fit the models, especially with larger networks. Finally, for smaller networks the non-spline degree-heterogeneity approach outperformed the proposed low-dimensional spline approach in terms of both computational efficiency and clustering performance. However, as network size increased, the two methods yielded comparable clustering performance, while the non-spline approach exhibited a notable decline in computational efficiency relative to the proposed low-dimensional spline approach. For example, compared to the low-dimensional spline approach, the average computation time for the non-spline approach was approximately 23% and 40% greater for network sizes of 1000 and 1500 actors, respectively.

Fig. A.2 presents a plot depicting the ratio of average iterations per second for the non-spline versus original approach by number of actors in the network. Fig. A.2 allows for a better understanding of the computational gains with utilizing the low dimensional approximation approach instead of the “unconstrained” approach. The non-spline approach involves updating  $2N + 3$  degree-heterogeneity parameters during each iteration of the EM-algorithm, however the updates involve simple arithmetic operations. In contrast, with respect to degree-heterogeneity parameters, the low-dimensional spline approach involves a one time cost of forming the basis matrix for the natural cubic spline and iterative updates to the  $2\zeta + 3$  parameters of  $\beta$ , which involves computationally intensive matrix inversions; here  $\zeta$  is a fixed number of interior knots for the natural cubic spline that does not depend on  $N$ . Thus, for small networks, it makes sense that the non-spline approach is faster per iteration than the proposed low-dimensional spline approach. However, as network sizes increase we see a decrease in the relative computational superiority with the non-spline approach – updating  $2N + 3$  parameters during each iteration of the EM-algorithm becomes a bigger problem as  $N$  increases. At a network size of 1500 actors we observe that our proposed low-dimensional spline approach surpasses the non-spline approach with respect to computational efficiency as measured by iterations per second (i.e. ratio  $< 1$ ). Extrapolating this observed trend, we would expect the ratio of iterations per second to continue to decrease as  $N$  increases. Furthermore, in experiments, we observed instability when fitting models using the “unconstrained” non-spline approach to account for degree heterogeneity. Specifically, as the variance of the random degree heterogeneity effects increased, the estimates of  $s_i$  and  $r_i$  often exhibited divergent behavior, ultimately preventing successful model

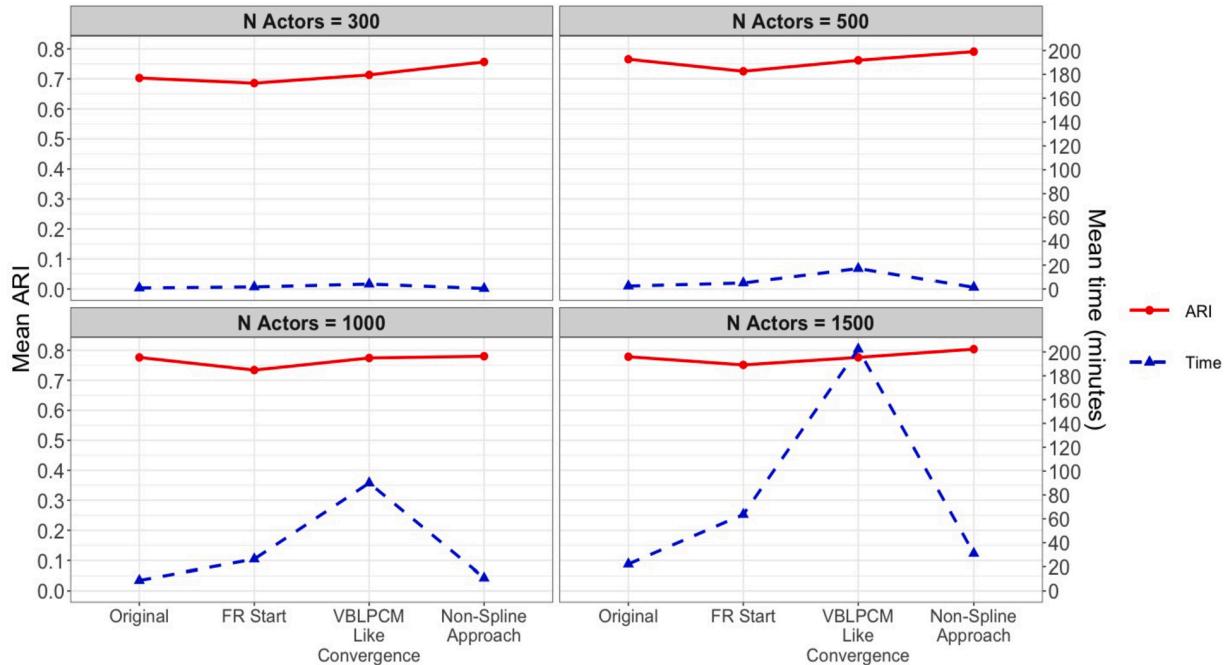


Fig. A.1. Ablation study results of clustering and computational performance.

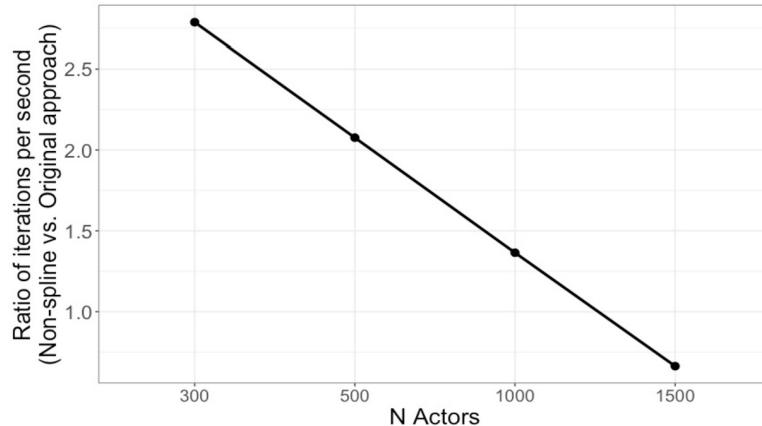


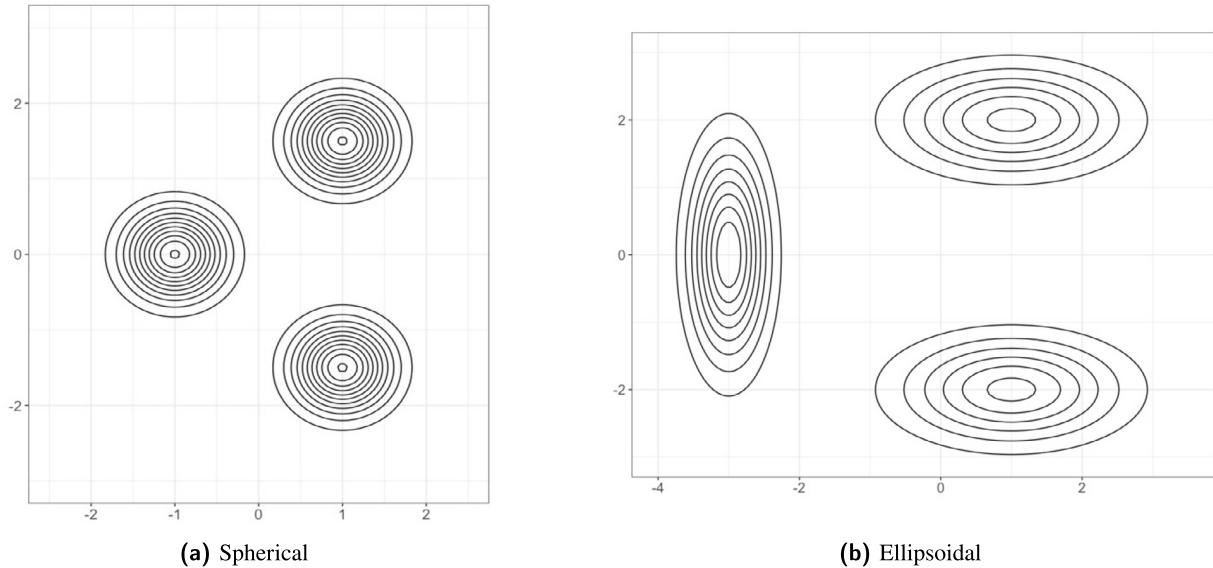
Fig. A.2. Ratio of average iterations per second for the non-spline versus original approach by number of actors in the network.

fitting. Additionally, the non-spline approach was observed to be highly sensitive to starting values. For instance, while the proposed low-dimensional spline approach reliably converged even with random starting values, the non-spline method consistently failed to do so under similar conditions. The combined advantages of computational efficiency on large networks, comparable clustering performance, and improved stability provide justification for utilizing the low-dimensional spline approach.

In summary, the ablation study provides evidence to suggest that each novel implementation incorporated into JANE individually offers improvements in either clustering or computational performance. Combined, the novel strategies allow for substantial improvements in both computational efficiency and clustering accuracy.

#### A.5.2. Increasing degree of separation and number of clusters

To better understand the impact of increasing the number of clusters and varying degree of overlap of the clusters, we conducted additional simulation studies. First, we considered a new configuration of the multivariate normal distributions where we increased the degree of separation, which we label as “More Separation” (Fig. A.3). Second, we considered a new configuration of the multivariate normal distributions where we doubled the number of clusters by essentially reflecting the more separated cluster configuration, which we label as “More Clusters” (Fig. A.4).



**Fig. A.3.** Configurations for more separated clusters.

For each new simulation configuration considered, we simulated 50 networks with 300 actors and a density of 0.1. Fig. A.5 presents the results under the new configurations, where we replicate the comparisons shown in Fig. 1, assuming the true  $K$  is known. The first panel of Appendix Fig. A.5 presents the results from Fig. 1 (i.e., “Original Configuration”). The next two panels present the results using configurations specified in Fig. A.3 (i.e., “More Separation”) and Fig. A.4 (i.e., “More Clusters”), respectively. Increasing the degree of separation dramatically improves clustering performance across all methods considered. This observation is expected as increasing the degree of separation between the clusters will subsequently result in a lower probability of edges between clusters, making it a much simpler clustering problem to fit. Similarly, we observed improved clustering performance when doubling the number of well-separated clusters. The superior performance observed with JANE under the ellipsoidal-shape specification seems to no longer hold as the degree of separation is increased. This finding may be explained by the lack of overlap with the elongated ellipsoidal-shaped clusters, which, under the scenario of assuming the true  $K$  is known, appears to be sufficiently represented by spherical-shaped clusters. However, these findings do not hold under the assumption where the true  $K$  is not known, which is explored and presented below. Another interesting observation was that VBLPCM seems to marginally outperform JANE under the ellipsoidal-shape setting with more clusters. Caution needs to be used when drawing conclusions from this observation as under this configuration VBLPCM failed to converge in 22 of the 50 simulations, while JANE and latentnet experienced no such issues. The fact that VBLPCM produced such impressive clustering results even when failing to converge for a considerable number of the simulations suggests that the improved performance may be an artifact of exceptional starting values produced by the FR approach when the degree of separation is increased.

Tables A.1 and A.2 present the clustering performance comparisons when model selection was used to select the number of clusters from  $K \in \{2, \dots, 10\}$  for the new configurations considered. As expected, increasing the degree of separation seems to substantially improve clustering performance across all methods considered, regardless of the shape specification (Table A.1). Furthermore, we also observed patterns consistent with the original configurations utilized in the manuscript. Specifically, (1) with JANE when using BIC rather than BICL for choosing the number of clusters, BIC tended to favor a larger  $K$  in a greater proportion of the simulations than BICL; (2) when the true shape specification was spherical, in general latentnet selected a  $K > 3$  for a greater proportion of simulations than JANE; and (3) when the true shape specification was ellipsoidal, JANE produced a higher mean ARI and selected the true  $K$  in a greater proportion of simulations than latentnet.

Table A.2 presents the clustering performance results when doubling the number of well-separated clusters. When the number of clusters was increased, we observed a substantial improvement in `latentnet`'s ability to accurately select the true  $K$ , particularly under the spherical-shape specification. In contrast, `JANE`, while producing a comparable mean ARI, seemed to underestimate the number of clusters, regardless if BIC or BICL was used for model selection. However, under the ellipsoidal-shape specification, `JANE` produced the best clustering performance and `latentnet` tended to overestimate the number of clusters, consistent with prior findings. The improvements with `latentnet`'s accuracy in selecting the true  $K$  as the true number of clusters increases may be a result of the hyperparameters utilized. As described in Krivitsky et al. (2009), `latentnet` employs strong priors for the hyperparameters associated with the mixture model prior variances, prior means, and prior on the mixture probabilities (see Section 3.1 Bayesian Estimation and Prior Distributions). The hyperparameters utilized are informed by the data through the number of actors and clusters considered in the network. We suspect that these strong priors, especially in the setting of relatively small network sizes, has a significant impact on the posterior. In contrast, with `JANE` the hyperparameters were selected in such a manner to ensure flat priors, so as to not have a large impact on the posterior. However, with respect to  $v$  (i.e., the hyperparameter for the Dirichlet prior on the

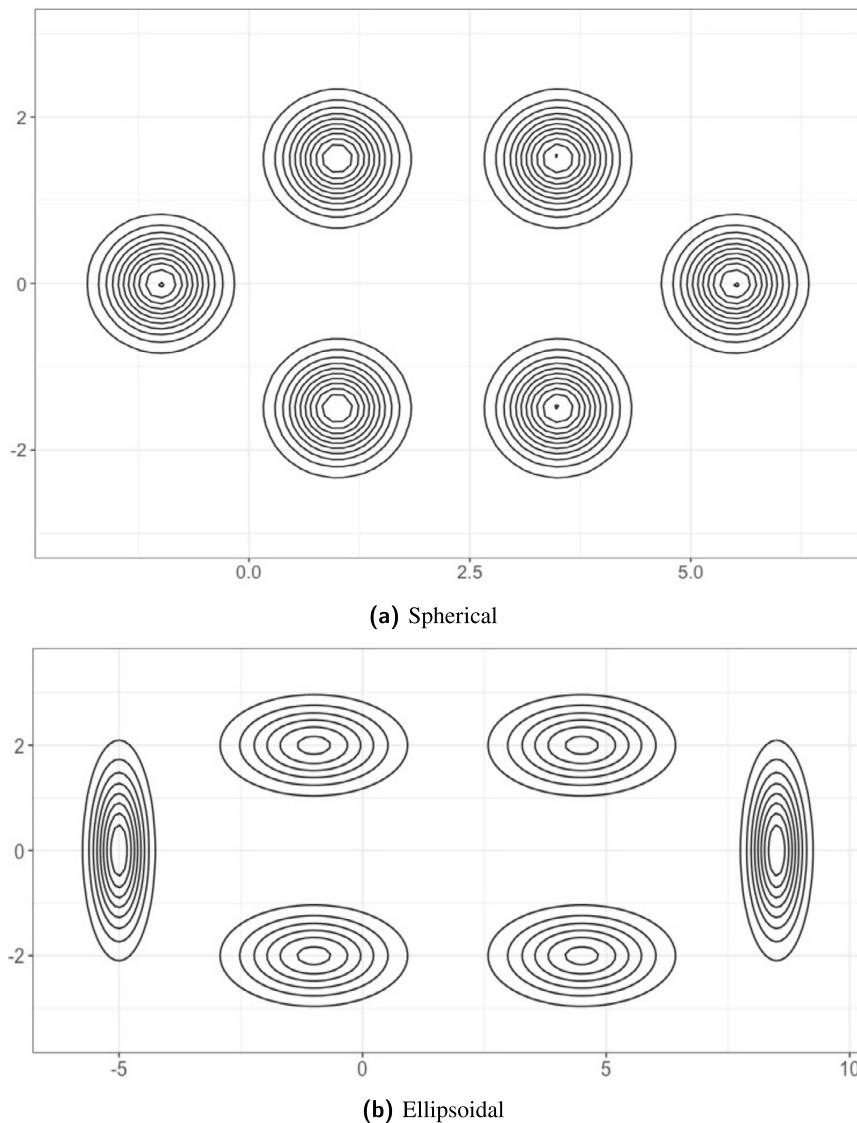


Fig. A.4. Configurations for doubling number of clusters.

**Table A.1**

Simulation study results of clustering performance by method with more separated clusters, where the true  $K$  was unknown.

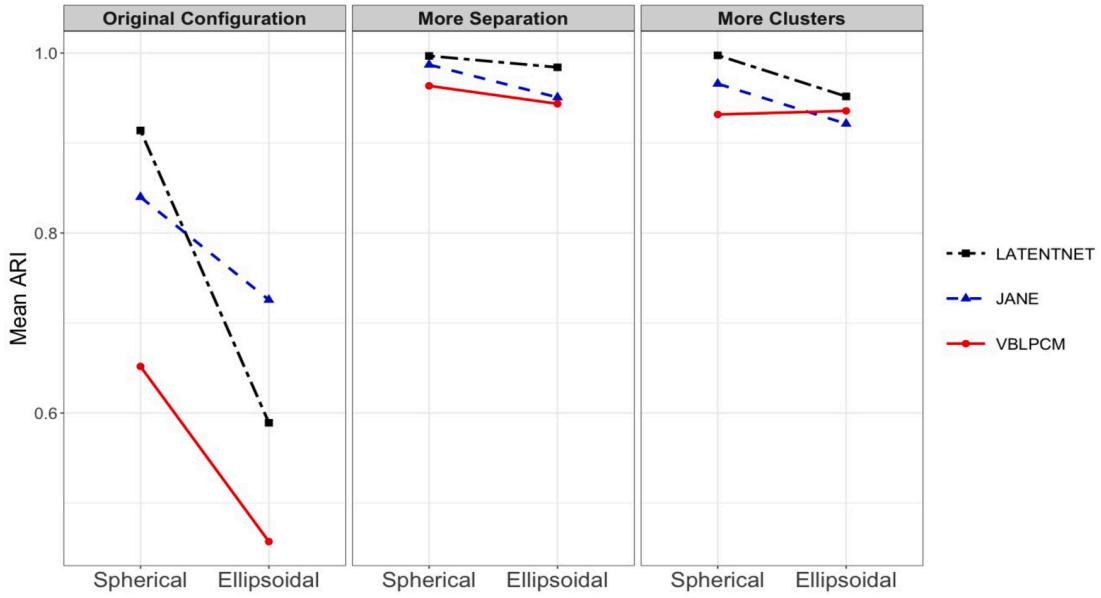
Method	Spherical			Ellipsoidal		
	Mean	% Selected $K = 3$	% Selected $K > 3$	Mean	% Selected $K = 3$	% Selected $K > 3$
	ARI			ARI		
JANE (BICL) <sup>1</sup>	0.99	78.0	22.0	0.86	50.0	46.0
JANE (BIC) <sup>2</sup>	0.98	74.0	26.0	0.80	34.0	62.0
latentnet <sup>2</sup>	0.99	62.0	38.0	0.62	0.0	100.0
VBLPCM <sup>2,3</sup>	-	-	-	-	-	-

Note: ARI = Adjusted Rand Index; True  $K = 3$ .

<sup>1</sup> Optimal  $K$  selected using BICL.

<sup>2</sup> Optimal  $K$  selected using BIC.

<sup>3</sup> Missing values as VBLPCM failed to select the optimal  $K$  for all simulations due to infinite BIC values.

Fig. A.5. Simulation study results for additional configurations, assuming the true  $K$  was known.**Table A.2**

Simulation study results of clustering performance by method with larger number of clusters, where the true  $K$  was unknown.

Method	Spherical			Ellipsoidal		
	Mean ARI	% Selected $K = 6$	% Selected $K > 6$	Mean ARI	% Selected $K = 6$	% Selected $K > 6$
JANE (BICL) <sup>1</sup>	0.92	36.0	6.0	0.88	46.0	22.0
JANE (BIC) <sup>2</sup>	0.92	36.0	6.0	0.87	40.0	28.0
latentnet <sup>2</sup>	0.97	82.0	4.0	0.77	12.0	88.0
VBLPCM <sup>2,3</sup>	-	-	-	-	-	-

Note: ARI = Adjusted Rand Index; True  $K = 6$ .

<sup>1</sup> Optimal  $K$  selected using BICL.

<sup>2</sup> Optimal  $K$  selected using BIC.

<sup>3</sup> Missing values as VBLPCM failed to select the optimal  $K$  for all simulations due to infinite BIC values.

**Table A.3**

Simulation study results of clustering performance by method with larger number of clusters and weaker prior on  $v$ , where the true  $K$  was unknown.

Method	Spherical			Ellipsoidal		
	Mean ARI	% Selected $K = 6$	% Selected $K > 6$	Mean ARI	% Selected $K = 6$	% Selected $K > 6$
JANE (BICL) <sup>1</sup>	0.90	64.0	6.0	0.94	70.0	16.0
JANE (BIC) <sup>2</sup>	0.90	64.0	6.0	0.94	70.0	16.0

Note: ARI = Adjusted Rand Index; True  $K = 6$ .

<sup>1</sup> Optimal  $K$  selected using BICL.

<sup>2</sup> Optimal  $K$  selected using BIC.

mixture weights), the default for JANE follows Handcock et al. (2007) and utilizes  $v_k = 3$  for  $k = 1, \dots, K$ , which puts low probability on small cluster sizes. This hyperparameter specification is potentially contributing towards the tendency for JANE to underestimate the number of clusters when the true number of clusters is increased. By keeping the number of actors in the network constant at 300 and doubling the number of clusters, we are essentially reducing cluster sizes. In the original setting with a true  $K = 3$  and 300 actors, under the ideal scenario of balanced cluster sizes we would expect 100 actors per cluster (i.e.,  $\frac{300}{3}$ ). In contrast, the new configuration considers a true  $K = 6$ , so with 300 actors and under the ideal scenario of balanced cluster sizes we would expect 50 actors per cluster (i.e.,  $\frac{300}{6}$ ). The potential of small cluster sizes is further exacerbated in our simulation study as we allow for unbalanced cluster sizes. Thus, given the presence of small clusters in the simulated networks, the default prior utilized by JANE, which puts low probability on small cluster sizes, will tend to favor a smaller  $K$  with larger cluster sizes, as observed in Table A.2. This is also supported by

the high mean ARI produced by JANE despite selecting a smaller  $K$  for the majority of simulations, suggesting that the discrepancy between the true cluster partitions and the hard cluster partitions produced by JANE are attributable to a small proportion of actors that were assigned to relatively small groups. In fact, for the simulation study with more clusters, by using a much weaker prior on  $\nu$  (i.e.,  $\nu_k = 1.5$  for  $k = 1, \dots, K$ ) we observed that JANE's accuracy in selecting the true  $K$  improved substantially, regardless of shape-specification (see Table A.3). As network size increases, the influence of hyperparameter specification on clustering results is expected to diminish.

In summary, relative to latentnet and VBLPCM, JANE exhibits comparable or even superior clustering performance for a variety of simulation configurations considered, all while significantly cutting down on computation time. Interestingly, for both latentnet and JANE we observed that the number of clusters selected was sensitive to the specification of the Dirichlet prior on the mixture weights of the mixture model. Future work should explore principled approaches for hyperparameter selection.

## References

- Biernacki, C., Celeux, G., Govaert, G., 2000. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 719–725.
- Fruchterman, T.M., Reingold, E.M., 1991. Graph drawing by force-directed placement. *Softw. Pract. Exp.* 21, 1129–1164.
- Gollini, I., Murphy, T.B., 2016. Joint modeling of multiple network views. *J. Comput. Graph. Stat.* 25, 246–265.
- Greene, D., Cunningham, P., 2013. Producing a unified graph representation from multiple social network views. In: Proceedings of the 3rd Annual ACM Web Science Conference. WebSci 2013 2.
- Handcock, M.S., Raftery, A.E., Tantrum, J.M., 2007. Model-based clustering for social networks. *J. R. Stat. Soc., Ser. A, Stat. Soc.* 170, 301–354.
- Hoff, P.D., Raftery, A.E., Handcock, M.S., 2002. Latent space approaches to social network analysis. *J. Am. Stat. Assoc.* 97, 1090–1098.
- Krivitsky, P.N., Handcock, M.S., 2008. Fitting position latent cluster models for social networks with latentnet. *J. Stat. Softw.* 24.
- Krivitsky, P.N., Handcock, M.S., Raftery, A.E., Hoff, P.D., 2009. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Soc. Netw.* 31, 204–213.
- Liu, Z., Zhou, J., 2020. Introduction to Graph Neural Networks. Springer, Cham.
- O'Connor, L.J., Medard, M., Feizi, S., 2020. Maximum likelihood embedding of logistic random dot product graphs. *Proc. AAAI Conf. Artif. Intell.* 34, 5289–5297.
- R Core Team, 2021. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- Raftery, A.E., Niu, X., Hoff, P.D., Yeung, K.Y., 2012. Fast inference for the latent space network model using a case-control approximate likelihood. *J. Comput. Graph. Stat.* 21, 901–919.
- Rastelli, R., Friel, N., Raftery, A.E., 2016. Properties of latent variable network models. *Netw. Sci.* 4, 407–432.
- Salter-Townshend, M., Murphy, T.B., 2013. Variational Bayesian inference for the latent position cluster model for network data. *Comput. Stat. Data Anal.* 57, 661–671.
- Sewell, D.K., Chen, Y., 2016. Latent space models for dynamic networks with weighted edges. *Soc. Netw.* 44, 105–116.
- Ueda, N., Nakano, R., 1998. Deterministic annealing em algorithm. *Neural Netw.* 11, 271–282.

## Riccardo Rastelli

### *Material list:*

Chaoyi Lu C., Rastelli R. and Friel N. (2025) A Zero-Inflated Poisson Latent Position Cluster Model. arXiv 2502.13790.

# A Zero-Inflated Poisson Latent Position Cluster Model

Chaoyi Lu<sup>\*†‡</sup>, Riccardo Rastelli<sup>†</sup>, Nial Friel<sup>†‡</sup>

<sup>†</sup>School of Mathematics and Statistics, University College Dublin, Ireland

<sup>‡</sup>Insight Centre for Data Analytics, University College Dublin, Ireland

## Abstract

The latent position network model (LPM) is a popular approach for the statistical analysis of network data. A central aspect of this model is that it assigns nodes to random positions in a latent space, such that the probability of an interaction between each pair of individuals or nodes is determined by their distance in this latent space. A key feature of this model is that it allows one to visualize nuanced structures via the latent space representation. The LPM can be further extended to the Latent Position Cluster Model (LPCM), to accommodate the clustering of nodes by assuming that the latent positions are distributed following a finite mixture distribution. In this paper, we extend the LPCM to accommodate missing network data and apply this to non-negative discrete weighted social networks. By treating missing data as “unusual” zero interactions, we propose a combination of the LPCM with the zero-inflated Poisson distribution. Statistical inference is based on a novel partially collapsed Markov chain Monte Carlo algorithm, where a Mixture-of-Finite-Mixtures (MFM) model is adopted to automatically determine the number of clusters and optimal group partitioning. Our algorithm features a truncated absorb-eject move, which is a novel adaptation of an idea commonly used in collapsed samplers, within the context of MFMs. Another aspect of our work is that we illustrate our results on 3-dimensional latent spaces, maintaining clear visualizations while achieving more flexibility than 2-dimensional models. The performance of this approach is illustrated via two carefully designed simulation studies, as well as four different publicly available real networks, where some interesting new perspectives are uncovered.

**Keywords:** social networks; latent position cluster model; weighted networks; zero-inflated Poisson; clustering; mixture of finite mixtures.

## 1 Introduction

The Latent Position Model (LPM) was introduced for social network analysis by Hoff et al. (2002). An important feature of the LPM is that it facilitates network visualization via latent space modeling. The LPM assumes that the data distribution depends on the stochastic latent positions of the nodes of the network. Usually these latent positions are assumed to lie in a Euclidean space. We refer the reader to Kaur et al. (2023), Rastelli, Friel, and Raftery (2016), and Salter-Townshend, White, et al. (2012) for detailed reviews of this model. Clustering of nodes plays a central role in statistical network analysis:

---

<sup>\*</sup>Address for correspondence: `chaoyi.lu@ucd.ie`

the LPM can be extended to this context following the pioneering work of Handcock et al. (2007). This leads us to the well-known Latent Position Cluster Model (LPCM) whereby the latent positions of the nodes are assumed to follow a mixture of multivariate normal distributions. With these assumptions, the network model can better support the presence of communities and assortative mixing.

A limitation of the original LPM and LPCM models is that they are only defined for the networks with binary interactions between the individuals. An interesting extension of these models to the weighted setting is provided by Sewell and Chen (2015) and Sewell and Chen (2016), where the authors extend the LPM to dynamic unipartite networks with weighted edges and propose a general link function for the expectation of the edge weights. In this paper, we focus on non-dynamic unipartite networks with non-negative discrete weighted edges, corresponding to networks where the edge values are typically integer counts.

Missing data is a common issue in statistical data analysis which generally leads to an overabundance of zeros. In this paper, we classify zero entries in a dataset either as “true” zeros or as “unusual” zeros. A zero-inflated model proposed by Lambert (1992) assumes a probability model for the occurrence of unusual zeros, which can be paired with the Poisson distribution as a canonical choice for count weighted data. In this situation, an unusual zero may correspond to an edge that although present, the corresponding edge weight is not recorded.

Zero-inflation is well-explored in the area of regression models with many extensions, for example, Hall (2000), Ridout et al. (2001), Ghosh et al. (2006), Lemonte et al. (2019) to cite a few. By contrast, zero-inflated models are not widely applied in the statistical analysis of network data. Zero-inflation is mentioned in Sewell and Chen (2016): while their dynamic latent space model may be extended to the zero-inflated case for sparse networks, their applications do not cover this extension. An advancement in this line of literature is provided by Lu et al. (2024) where the authors incorporate the zero-inflated model within a Stochastic Block Model (SBM) (Nowicki and Snijders, 2001). The SBM is widely used clustering model for network analysis that shares some similarities with the LPM, since both models are based on a latent variable framework. Differently from the LPM, the SBM is capable of representing disassortative patterns, but, on the other hand, the LPM can provide clearer and more interpretable network visualizations.

In this paper, we propose to incorporate a zero-inflated Poisson distribution within the LPCM leading to the Zero-Inflated Poisson Latent Position Cluster Model (ZIP-LPCM), and, in doing so, we create a simultaneous framework that can be used to characterize zero inflation, clustering and the latent space visualization. Similar model assumptions can be found in C. Ma (2024) where a Latent Space Zero-Inflated Poisson (LS-ZIP) model is proposed. However, differently from our model structure, their LS-ZIP embeds an inner-product latent space model (Hoff, 2003; Z. Ma et al., 2020) and proposes two different sets of latent positions for the Poisson rate and for the probability of unusual zeros, respectively. Furthermore, their model does not account for the clustering of nodes, which is a central feature of our proposed model.

We employ a Bayesian framework to infer the model parameters and the latent variables of our ZIP-LPCM. Our inferential framework takes inspiration from the literature on the LPCM, the Mixture of Finite Mixtures (MFM) model (Miller and Harrison, 2018; Geng et al., 2019), and collapsed Gibbs sampling, and we combine some key ideas from

the available literature to create our own original procedure. As regards the LPCM, commonly used inference methods include a variational expectation maximization algorithm (Salter-Townshend and Murphy, 2013), and Markov chain Monte Carlo methods (Handcock et al., 2007). A contribution close to our own is that of Ryan et al. (2017) where the authors exploit conjugate priors to calculate a marginalized posterior in analytic form, and then target this distribution using a so-called “collapsed” sampler. Similar parameter-collapsing ideas are also employed in McDaid et al. (2012) and Wyse and Friel (2012). In this paper, we propose to follow a similar strategy as that presented in Lu et al. (2024) for the inference task leveraging the partially collapsed Gibbs sampler introduced by Van Dyk and Park (2008) and Park and Van Dyk (2009).

As regards the choice of the number of clusters, we also make an original contribution by combining some approaches and ideas available from the literature. We highlight Nobile and Fearnside (2007), where the authors introduced an Absorb-Eject (AE) move to automatically choose the number of clusters. Here we propose a variant of this move to better match our framework. Indeed, as a prior distribution for the clustering variables and number of clusters, we adopt a MFM model, along with the supervision idea introduced by Legramanti et al. (2022). In this case, the AE move can further facilitate the estimation of clusters, but such a step can only be defined if the framework allows for the existence of empty clusters. Unfortunately, this is not the case for MFMs, and so the move and the model are incompatible. To address this impasse, we propose a new Truncated Absorb-Eject (TAE) move which allows us to efficiently explore the sample space thus obtaining good estimates of the clustering variables and of the number of groups.

This paper is organized as follows. Section 2 provides a detailed introduction of our proposed zero-inflated Poisson latent position cluster model. Section 3 explains how we design the Bayesian inference process of our model where the incorporation of a mixture of finite mixtures model as well as its supervised version is included in Section 3.1. The idea of partially collapsing the model parameters in the posterior distribution, and the newly proposed truncated absorb-eject move, are introduced in Section 3.2 and Section 3.3, respectively. The detailed steps and designs of the partially collapsed Metropolis-within-Gibbs algorithm for the inference are illustrated in Section 3.4. In Section 4, we show the performance of our strategy via two carefully designed simulation studies within each of which there are two different scenarios tackling different real world situations. Applications on four different real social networks with different network sizes are included in Section 5. Finally, Section 6 concludes this paper and provides a few possible future directions.

## 2 Model

In this paper we focus on weighted networks with non-negative discrete edges between each pair of individuals, and denote  $N$  as the total number of individuals in the network. The model introduced in this section is designed for directed networks, however it is straightforward to apply it to the undirected case. A network is usually observed by an  $N \times N$  adjacency matrix denoted as  $\mathbf{Y}$ , where each element  $y_{ij}$  is a non-negative integer indicating the interaction from node  $i$  to node  $j$ , and reflecting the corresponding interaction strength. An element  $y_{ij} = 0$  corresponds to a non-interaction or a zero

interaction. Self-loops are not allowed.

## 2.1 Zero-inflated Poisson model

In real networks which are observed in practice, there is often an overabundance of zeros, indicating the possibility that some of these are missing data. We consider the Zero-Inflated Poisson (ZIP) model (Lambert, 1992), which is a commonly used framework to deal with an excessive number of zeros in  $\mathbf{Y}$ . The ZIP model assumes that each observed interaction  $y_{ij}$  follows

$$P(y_{ij}|\lambda_{ij}, p_{ij}) = \begin{cases} p_{ij} + (1 - p_{ij})f_{\text{Pois}}(0|\lambda_{ij}), & \text{if } y_{ij} = 0; \\ (1 - p_{ij})f_{\text{Pois}}(y_{ij}|\lambda_{ij}), & \text{if } y_{ij} = 1, 2, \dots, \end{cases} \quad (1)$$

for  $i, j = 1, 2, \dots, N$ ;  $i \neq j$ , where  $f_{\text{Pois}}(\cdot|\lambda_{ij})$  is the probability mass function of the Poisson distribution with parameter  $\lambda_{ij}$ . Here the zeros in the network can be classified into two types. A “structural” zero is observed with probability  $p_{ij}$ , whereas a Poisson zero that naturally arises from the Poisson distribution is observed with probability  $(1 - p_{ij})f_{\text{Pois}}(0|\lambda_{ij})$ . We formalize the two possibilities by augmenting the model in Eq. (1) with a new indicator variable,  $\nu_{ij}$ , indicating whether the corresponding  $y_{ij}$  is a structural zero or not, and thus the data distribution is determined separately for the two cases below:

$$y_{ij}|\lambda_{ij}, \nu_{ij} \sim \begin{cases} \mathbb{1}(y_{ij} = 0), & \text{if } \nu_{ij} = 1; \\ \text{Pois}(\lambda_{ij}), & \text{if } \nu_{ij} = 0, \end{cases} \quad (2)$$

where, for every  $i$  and  $j$ , the collection of  $\nu_{ij} \sim \text{Bernoulli}(p_{ij})$  constitutes the  $N \times N$  structural zero indicator matrix  $\boldsymbol{\nu}$ . The function  $\mathbb{1}(y_{ij} = 0)$  is an indicator function returning 1 if  $y_{ij} = 0$  and returning 0 otherwise.

As far as  $\nu_{ij} = 1$ , the observed  $y_{ij}$  is a structural zero with probability 1. Here, we interpret such a “structural” zero as an “unusual” zero or missing data that replaces a true interaction weight which follows the corresponding  $\text{Pois}(\lambda_{ij})$  distribution. A zero arising from  $f_{\text{Pois}}(\cdot|\lambda_{ij})$  is thus treated as a “true” zero. We denote the covert true interaction as  $x_{ij}$ : we assume that this value is not observed when  $\nu_{ij} = 1$ , and that it follows the same Poisson distribution of  $y_{ij}|\lambda_{ij}, \nu_{ij} = 0$ . Thus, based on the augmented zero-inflated model in Eq. (2) as well as the observed  $\mathbf{Y}$ , the augmented data  $\mathbf{X}$ , which is a  $N \times N$  matrix with entries  $\{x_{ij}\}$ , is in the form

$$\begin{cases} x_{ij} \sim \text{Pois}(\lambda_{ij}), & \text{if } \nu_{ij} = 1; \\ x_{ij} = y_{ij}, & \text{if } \nu_{ij} = 0. \end{cases} \quad (3)$$

Here, the case  $x_{ij} = y_{ij}$  is equivalent to  $x_{ij} \sim \mathbb{1}(x_{ij} = y_{ij})$  when  $\nu_{ij} = 0$ . A similar data augmentation framework has appeared in other works, for example, Tanner and Wong (1987) and Ghosh et al. (2006). The augmented  $\mathbf{X}$  is known as the missing data imputed adjacency matrix.

## 2.2 Zero-inflated Poisson latent position cluster model

To characterize the  $\text{Pois}(\lambda_{ij})$  distribution under the  $\nu_{ij} = 0$  case of the augmented ZIP model in Eq. (2), we employ the Latent Position Cluster Model (LPCM) (Handcock

et al., 2007), which is an extended version of the Latent Position Model (LPM) (Hoff et al., 2002). Each node  $i : i = 1, 2, \dots, N$  in the network is assumed to have a latent position  $\mathbf{u}_i \in \mathbb{R}^d$ , and we denote the collection of all latent positions as  $\mathbf{U} := \{\mathbf{u}_i\}$ . A generalization of the latent position model without considering covariates can be expressed in the form  $g[\mathbb{E}(y_{ij})] = h(\mathbf{u}_i, \mathbf{u}_j)$ , where  $g(\cdot)$  is some link function, and  $h : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  is a function of two nodes' latent positions. Here, we make standard assumptions on the functions  $g(\cdot)$  and  $h(\cdot, \cdot)$ , which link the Poisson rate  $\lambda_{ij}$  in Eq. (2) with the latent positions  $\mathbf{U}$  as follows:

$$\log(\lambda_{ij}) = \beta - \|\mathbf{u}_i - \mathbf{u}_j\|. \quad (4)$$

In the equation above,  $\|\cdot\|$  is the Euclidean distance, while  $\beta \in \mathbb{R}$  can be interpreted as an intercept term where higher values bring larger interaction weights as well as lower chance of a true zero.

The LPCM further assumes that each latent position  $\mathbf{u}_i$  is drawn from a finite mixture of  $\bar{K}$  multivariate normal distributions, each corresponding to a different group:

$$\mathbf{u}_i \sim \sum_{k=1}^{\bar{K}} \pi_k \text{MVN}_d(\boldsymbol{\mu}_k, 1/\tau_k \mathbb{I}_d). \quad (5)$$

Here, each  $\pi_k$  is the probability that a node is clustered into group  $k$ , and  $\sum_{k=1}^{\bar{K}} \pi_k = 1$ . The combination of Eqs. (2), (4) and (5) defines our Zero-Inflated Poisson Latent Position Cluster Model (ZIP-LPCM).

Letting  $z_i \in \{1, 2, \dots, \bar{K}\}$  denote the group membership of node  $i$ , the mixture in Eq. (5) can be augmented as

$$\mathbf{u}_i | z_i = k \sim \text{MVN}_d(\boldsymbol{\mu}_k, 1/\tau_k \mathbb{I}_d),$$

where a multinomial( $1, \boldsymbol{\Pi}$ ) distribution is assumed for each  $\mathbf{z}_i$ , and  $\boldsymbol{\Pi} := (\pi_1, \pi_2, \dots, \pi_{\bar{K}})$ . The notation  $\mathbf{z}_i := (z_{i1}, z_{i2}, \dots, z_{i\bar{K}})$  is equivalent to  $z_i = k$  if we let  $z_{ig} = \mathbb{1}(g = k)$  for  $g = 1, 2, \dots, \bar{K}$ , and thus  $f_{\text{multinomial}}(\mathbf{z}_i | 1, \boldsymbol{\Pi}) = \pi_{z_i}$ . Note that  $\bar{K}$  here denotes the total number of possible groups that the network is assumed to have, even though some groups may be empty. Further, we denote  $\mathbf{z} := (z_1, z_2, \dots, z_N)$  as a vector of group membership indicators for all  $\{z_i\}$ , while  $\boldsymbol{\mu} := \{\boldsymbol{\mu}_k : k = 1, 2, \dots, \bar{K}\}$ , and  $\boldsymbol{\tau} := (\tau_1, \tau_2, \dots, \tau_{\bar{K}})$ .

The indicator of unusual zero  $\nu_{ij}$  in Eq. (2) is proposed to instead follow a Bernoulli( $p_{z_i z_j}$ ), where we replace the probability of unusual zero  $p_{ij}$  with a cluster-dependent counterpart  $p_{z_i z_j}$ , as in a network block-structure. The intuition behind this is that the probability of missing data observed between two nodes is expected to vary depending on their respective groups. Thus, we denote  $\mathbf{P}$  as a  $\bar{K} \times \bar{K}$  matrix with each entry  $p_{gh}$  denoting the probability of unusual zeros for the interactions from group  $g$  to group  $h$ , where  $g, h = 1, 2, \dots, \bar{K}$ .

This leads to the complete likelihood of the ZIP-LPCM being written as:

$$\begin{aligned}
f(\mathbf{Y}, \boldsymbol{\nu}, \mathbf{U}, \mathbf{z} | \beta, \mathbf{P}, \boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{\Pi}) &\propto f(\mathbf{Y} | \beta, \mathbf{U}, \boldsymbol{\nu}) f(\boldsymbol{\nu} | \mathbf{P}, \mathbf{z}) f(\mathbf{U} | \boldsymbol{\mu}, \boldsymbol{\tau}, \mathbf{z}) f(\mathbf{z} | \boldsymbol{\Pi}) \\
&= \prod_{\substack{i,j:i \neq j, \\ \nu_{ij}=0}}^N f_{\text{Pois}}(y_{ij} | \exp(\beta - \|\mathbf{u}_i - \mathbf{u}_j\|)) \prod_{i,j:i \neq j}^N f_{\text{Bern}}(\nu_{ij} | p_{z_i z_j}) \times \\
&\quad \times \prod_{i=1}^N f_{\text{MVN}_d}(\mathbf{u}_i | \boldsymbol{\mu}_{z_i}, 1/\tau_{z_i} \mathbb{I}_d) \prod_{i=1}^N f_{\text{multinomial}}(\mathbf{z}_i | 1, \boldsymbol{\Pi}),
\end{aligned} \tag{6}$$

which can be calculated analytically and efficiently for all choices of parameter values.

### 3 Partially collapsed Bayesian inference

In this section, we illustrate our original inference processes which aim to jointly infer the intercept  $\beta$ , the indicator of unusual zeros  $\boldsymbol{\nu}$ , the latent positions  $\mathbf{U}$ , the latent clustering indicator  $\mathbf{z}$  and the number of occupied groups  $K$ . The model includes other parameters, such as  $\boldsymbol{\mu}$  and  $\boldsymbol{\tau}$ , which are dealt with via marginalization, as shown in Section 3.2. We emphasize that  $K$  here indicates the number of occupied groups, not to be confused with the total number of groups  $\bar{K}$ . This distinction is crucial since we leverage mixture-of-finite-mixtures (Miller and Harrison, 2018; Geng et al., 2019) to marginalize  $\bar{K}$  in Section 3.1. As a result, we only need to focus on non-empty groups during the inference procedures.

#### 3.1 Mixture of finite mixtures and supervision

A natural conjugate prior for the multinomial parameters  $\boldsymbol{\Pi}$  is a Dirichlet( $\alpha, \dots, \alpha$ ) distribution (Nowicki and Snijders, 2001). By further proposing a prior  $\bar{K} \sim \pi_{\bar{K}}(\cdot)$ , the Mixture-of-Finite-Mixtures (MFM) can marginalize both  $\boldsymbol{\Pi}$  and  $\bar{K}$  leading to the probability mass function of a MFM, which is defined for the unlabeled clustering and reads as follows:

$$f(\mathbf{C}(\mathbf{z})) = \sum_{k=1}^{\infty} \frac{k_{(K)}}{(k\alpha)^{(N)}} \pi_{\bar{K}}(k) \prod_{G \in \mathbf{C}(\mathbf{z})} \alpha^{(|G|)}, \tag{7}$$

where  $\alpha^{(n)} := \alpha(\alpha + 1) \cdots (\alpha + n - 1)$  is the ascending factorial notation, and  $k_{(K)} := k(k - 1) \cdots (k - (K - 1))$  is the descending factorial notation. Here,  $\mathbf{C}(\mathbf{z}) := \{G_k : k = 1, 2, \dots, \bar{K}; |G_k| > 0\}$  is a set of non-empty unordered collections of nodes where each collection  $G_k := \{i : i = 1, 2, \dots, N; z_i = k\}$  contains all the nodes from group  $k$ , while  $|G_k|$  denotes the number of nodes inside the collection. In the case that  $G_k$  is the empty set, we have  $|G_k| = 0$ . The parameter  $\bar{K}$  is collapsed by summing over all the possible  $\bar{K} = k$  values from  $k = 1$  to  $k = \infty$ . Note that  $\mathbf{C}(\mathbf{z})$  is invariant under any relabeling of  $\mathbf{z}$ . The  $k_{(K)} \equiv \binom{k}{K} K!$  term in Eq. (7) is the number of ways to relabel a specific  $\mathbf{z}$  or to label the unlabeled partition  $\mathbf{C}(\mathbf{z})$  provided with  $\bar{K} = k$ . The natural choice of  $\bar{K}$  prior is a zero-truncated Poisson(1) distribution (Geng et al., 2019; Nobile, 2005; McDaid et al., 2012) that is assumed throughout this paper.

To ensure that the clustering  $\mathbf{z}$  is invariant under any relabeling of it, we adopt a particular labeling method following the procedure used in Rastelli and Friel (2018). We assign node 1 to group 1 by default, and then iteratively assign the next node either to a new empty group or to an existing group. In this way, the defined  $\mathbf{z}$  only contains  $K$  occupied groups and is one-to-one correspondence to  $\mathbf{C}(\mathbf{z})$  regardless of whether the clustering before label-switching has empty groups or not. The clustering dependent parameters,  $\boldsymbol{\mu}$ ,  $\boldsymbol{\tau}$  and  $\mathbf{P}$  are relabeled accordingly and the entries relevant to empty groups are treated as redundant. The probability mass function of the MFM in Eq. (7) may be rewritten as

$$f(\mathbf{z}) = \mathcal{W}_{N,K} \prod_{g=1}^K \alpha^{(n_g)}, \quad (8)$$

where  $n_g$  is the number of nodes in group  $g$ . The non-negative weight

$$\mathcal{W}_{N,K} := \sum_{k=1}^{\infty} \frac{k_{(K)}}{(k\alpha)^{(N)}} \pi_K(k)$$

satisfies the recursion  $\mathcal{W}_{N,K} := (N + K\alpha)\mathcal{W}_{N+1,K} + \alpha\mathcal{W}_{N+1,K+1}$  with  $\mathcal{W}_{1,1} = 1/\alpha$ , and we refer to Miller and Harrison (2018) for the details of the computation of  $\mathcal{W}_{N,K}$ .

One could proceed to determine a generative/predictive urn scheme by checking the formulae differences between the  $f(\mathbf{z})$  and each  $f(\{\mathbf{z}, z_{N+1}\})$  for  $z_{N+1} = 1, 2, \dots, K, K+1$ . Then, sampling from Eq. (8) can be performed using the following procedure: assuming that the first node labeled as node 1 is assigned to group 1 by default, then the probability of the subsequent node being assigned to existing non-empty groups or to a new empty group is defined as:

$$P(z_{N+1} = k | \mathbf{z}) \propto \begin{cases} n_k + \alpha, & \text{for } k = 1, 2, \dots, K; \\ \frac{\mathcal{W}_{N+1,K+1}}{\mathcal{W}_{N+1,K}} \alpha, & \text{for } k = K+1, \end{cases} \quad (9)$$

that is conditional on the current clustering  $\mathbf{z}$  with parameters  $N$  and  $K$ . This generative scheme also belongs to the Ewens-Pitman two-parameter family of Exchangeable Partitions (Gnedin, Haulk, et al., 2009; Pitman, 2006).

Some exogenous node attributes may be available when analyzing real datasets, and, in this case, we leverage the supervision idea proposed by Legramanti et al. (2022) to account for such information in the modeling. We use  $\mathbf{c} = \{c_1, c_2, \dots, c_N\}$  to denote the categorical node attributes in our context where each  $c_i \in \{1, 2, \dots, C\}$ , and propose that

$$f(\mathbf{z} | \mathbf{c}) \propto \mathcal{W}_{N,K} \prod_{g=1}^K P(\mathbf{c}_g) \alpha^{(n_g)}, \quad (10)$$

where  $\mathbf{c}_g := \{c_i : z_i = g\}$  and where the distribution  $P(\mathbf{c}_g)$  is chosen as the Dirichlet-multinomial cohesion of Müller et al. (2011), which is given by:

$$P(\mathbf{c}_g) = \frac{\prod_{c=1}^C \Gamma(n_{g,c} + \omega_c)}{\Gamma(n_g + \omega_0)} \frac{\Gamma(\omega_0)}{\prod_{c=1}^C \Gamma(\omega_c)}. \quad (11)$$

Here,  $\omega_c$  is the cohesion parameter for each level  $c$  of the node attributes and  $\omega_0 = \sum_{c=1}^C \omega_c$ . Similar to Eq. (9), we can determine an urn scheme also for the supervised case as:

$$P(z_{N+1} = k | \mathbf{z}, \mathbf{c}, c_{N+1}) \propto \begin{cases} \frac{n_{k,c_{N+1}} + \omega_{c_{N+1}}}{n_k + \omega_0} (n_k + \alpha), & \text{for } k = 1, 2, \dots, K; \\ \frac{\omega_{c_{N+1}}}{\omega_0} \frac{W_{N+1,K+1}}{W_{N+1,K}} \alpha, & \text{for } k = K+1, \end{cases} \quad (12)$$

which, compared to Eq. (9), is inflated or deflated by a term that favors allocating the new node  $N+1$  to the group with higher fraction of the same node attribute as the  $c_{N+1}$ .

### 3.2 Inference and collapsing

In the case that the exogenous node attributes  $\mathbf{c}$  are available, adopting the supervised MFM introduced in Eq. (10) as well as the missing data imputation shown in Eq. (3) leads to the posterior distribution of our ZIP-LPCM being written as:

$$\begin{aligned} \pi(\mathbf{X}, \boldsymbol{\nu}, \mathbf{U}, \mathbf{z}, \beta, \mathbf{P}, \boldsymbol{\mu}, \boldsymbol{\tau} | \mathbf{Y}, \mathbf{c}) &\propto f(\mathbf{X} | \mathbf{Y}, \beta, \mathbf{U}, \boldsymbol{\nu}) f(\mathbf{Y} | \beta, \mathbf{U}, \boldsymbol{\nu}) f(\boldsymbol{\nu} | \mathbf{P}, \mathbf{z}) f(\mathbf{U} | \boldsymbol{\mu}, \boldsymbol{\tau}, \mathbf{z}) f(\mathbf{z} | \mathbf{c}) \\ &\quad \times \pi(\mathbf{P}) \pi(\boldsymbol{\mu}) \pi(\boldsymbol{\tau}) \pi(\beta) \\ &= f(\mathbf{X} | \beta, \mathbf{U}) f(\boldsymbol{\nu} | \mathbf{P}, \mathbf{z}) f(\mathbf{U} | \boldsymbol{\mu}, \boldsymbol{\tau}, \mathbf{z}) f(\mathbf{z} | \mathbf{c}) \pi(\mathbf{P}) \pi(\boldsymbol{\mu}) \pi(\boldsymbol{\tau}) \pi(\beta), \end{aligned} \quad (13)$$

where the  $f(\mathbf{Y} | \beta, \mathbf{U}, \boldsymbol{\nu})$ , the  $f(\boldsymbol{\nu} | \mathbf{P}, \mathbf{z})$  and the  $f(\mathbf{U} | \boldsymbol{\mu}, \boldsymbol{\tau}, \mathbf{z})$  are exactly the same as the ones shown in the complete likelihood in Eq. (6). Furthermore, similar to the  $f(\mathbf{Y} | \beta, \mathbf{U}, \boldsymbol{\nu})$ , the  $f(\mathbf{X} | \mathbf{Y}, \beta, \mathbf{U}, \boldsymbol{\nu})$  above can be obtained based on its sampling process in Eq. (3). The combination of the  $f(\mathbf{Y} | \beta, \mathbf{U}, \boldsymbol{\nu})$  and the  $f(\mathbf{X} | \mathbf{Y}, \beta, \mathbf{U}, \boldsymbol{\nu})$  in Eq. (13) is equivalent to the  $f(\mathbf{X} | \beta, \mathbf{U})$  which reads as follows:

$$f(\mathbf{X} | \beta, \mathbf{U}) = \prod_{i,j:i \neq j}^N f_{\text{Pois}}(x_{ij} | \exp(\beta - ||\mathbf{u}_i - \mathbf{u}_j||)).$$

However, in our context, not all the real networks that we work on provide exogenous node attributes. In the case that  $\mathbf{c}$  is not available, the unsupervised prior  $f(\mathbf{z})$  from Eq. (8) is instead proposed in Eq. (13) to replace the  $f(\mathbf{z} | \mathbf{c})$ .

We leverage conjugate prior distributions to marginalize a number of model parameters from the posterior distribution in (13). This methodology is also known as “collapsing” and has already been exploited in, for example, McDaid et al. (2012), Wyse and Friel (2012), Ryan et al. (2017), Rastelli, Latouche, et al. (2018), Legramanti et al. (2022), and Lu et al. (2024). By proposing the conjugate prior distributions:

$$\boldsymbol{\mu}_k | \tau_k \sim \text{MVN}_d(\mathbf{0}, 1/(\omega \tau_k) \mathbb{I}_d) \text{ for } k = 1, \dots, K, \quad (14)$$

$$\tau_k \sim \text{Gamma}(\alpha_1, \alpha_2/2) \text{ for } k = 1, \dots, K, \quad (15)$$

where  $(\omega, \alpha_1, \alpha_2)$  are hyperparameters to be specified a priori, the collapsed posterior distribution of the ZIP-LPCM is obtained as:

$$\pi(\mathbf{X}, \boldsymbol{\nu}, \mathbf{U}, \mathbf{z}, \beta, \mathbf{P} | \mathbf{Y}, \mathbf{c}) \propto f(\mathbf{X} | \beta, \mathbf{U}) f(\boldsymbol{\nu} | \mathbf{P}, \mathbf{z}) f(\mathbf{U} | \mathbf{z}) f(\mathbf{z} | \mathbf{c}) \pi(\mathbf{P}) \pi(\beta), \quad (16)$$

where the  $f(\mathbf{U}|\mathbf{z})$  is calculated according to the methodology explored in Ryan et al. (2017):

$$f(\mathbf{U}|\mathbf{z}) = \prod_{k=1}^K \left\{ \frac{\alpha_2^{\alpha_1}}{\Gamma(\alpha_1)} \frac{\Gamma(\alpha_1 + \frac{d}{2}n_k)}{\pi^{\frac{d}{2}n_k}} \left( \frac{\omega}{\omega + n_k} \right)^{\frac{d}{2}} \left[ \alpha_2 - \frac{\|\sum_{i:z_i=k} \mathbf{u}_i\|^2}{n_k + \omega} + \sum_{i:z_i=k} \|\mathbf{u}_i\|^2 \right]^{-\left(\frac{d}{2}n_k + \alpha_1\right)} \right\}. \quad (17)$$

Here,  $B(\cdot, \cdot)$  is the beta function and  $n_{gh} := \sum_{i,j:i \neq j}^N \mathbb{1}(z_i = g, z_j = h)$ , whereas  $n_k := \sum_{i=1}^N \mathbb{1}(z_i = k)$ , and  $\boldsymbol{\nu}_{gh}$  denotes the sum over all the  $\nu_{ij}|z_i = g, z_j = h, i \neq j$ .

Now that we have our target collapsed posterior distribution, we apply a partially collapsed Markov chain Monte Carlo approach (Van Dyk and Park, 2008; Park and Van Dyk, 2009) aiming to infer the latent clustering  $\mathbf{z}$ , the latent indicators of unusual zeros  $\boldsymbol{\nu}$ , the intercept  $\beta$  and the latent positions  $\mathbf{U}$  from the posterior in Eq. (16). This is accomplished by constructing a sampler which consists of multiple steps for each of the target variables and parameters, and for the number of occupied groups  $K$  simultaneously. The sampling of the imputed adjacency matrix  $\mathbf{X}$  straightforwardly follows from Eq. (3), and we leverage ideas similar to those in Lu et al. (2024) to infer  $\boldsymbol{\nu}$  and  $\mathbf{z}$ . Another layer of conjugacy is proposed for the inference procedure of the probability of unusual zeros  $\mathbf{P}$  that is required by the sampling step of  $\boldsymbol{\nu}$ , and we further develop a new truncated absorb-eject move tailored for the clustering without empty groups to facilitate the clustering inference. The sampling of  $\mathbf{U}$  and  $\beta$  are performed via two standard Metropolis-Hastings steps. More details of all these sampling steps are carefully discussed and provided next.

### 3.2.1 Inference for $\boldsymbol{\nu}$

Recall that each of the latent indicators of unusual zeros  $\{\nu_{ij}\}$  is assumed to follow the  $\text{Bern}(p_{z_i z_j})$  distribution. However, note that each  $\nu_{ij}$  must be zero by assumption when the corresponding observed  $y_{ij} > 0$ , so only those  $\{\nu_{ij} : y_{ij} = 0\}$  are required to be inferred during the inference. Conditional on that the observed interaction  $y_{ij}$  is a zero interaction, the probability of such an interaction being an unusual zero is:

$$P(\nu_{ij} = 1 | y_{ij} = 0, p_{z_i z_j}, \beta, \mathbf{u}_i, \mathbf{u}_j) = \frac{p_{z_i z_j}}{p_{z_i z_j} + (1 - p_{z_i z_j}) f_{\text{Pois}}(0 | \exp(\beta - \|\mathbf{u}_i - \mathbf{u}_j\|))}, \quad (18)$$

and, on the contrary, the conditional probability of it not being an unusual zero is  $P(\nu_{ij} = 0 | y_{ij} = 0, p_{z_i z_j}, \beta, \mathbf{u}_i, \mathbf{u}_j) = 1 - P(\nu_{ij} = 1 | y_{ij} = 0, p_{z_i z_j}, \beta, \mathbf{u}_i, \mathbf{u}_j)$ . This motivates the sampling of each  $\nu_{ij}$  to follow:

$$\nu_{ij} | y_{ij}, p_{z_i z_j}, \beta, \mathbf{u}_i, \mathbf{u}_j \sim \begin{cases} \text{Bern} \left( \frac{p_{z_i z_j}}{p_{z_i z_j} + (1 - p_{z_i z_j}) f_{\text{Pois}}(0 | \exp(\beta - \|\mathbf{u}_i - \mathbf{u}_j\|))} \right), & \text{if } y_{ij} = 0; \\ \mathbb{1}(\nu_{ij} = 1), & \text{if } y_{ij} > 0. \end{cases} \quad (19)$$

This distribution is actually the full-conditional distribution of  $\nu_{ij}$  under the posterior:

$$\pi(\boldsymbol{\nu}, \mathbf{U}, \mathbf{z}, \beta, \mathbf{P} | \mathbf{Y}, \mathbf{c}) \propto f(\mathbf{Y} | \beta, \mathbf{U}, \boldsymbol{\nu}) f(\boldsymbol{\nu} | \mathbf{P}, \mathbf{z}) f(\mathbf{U} | \mathbf{z}) f(\mathbf{z} | \mathbf{c}) \pi(\mathbf{P}) \pi(\beta), \quad (20)$$

which further marginalizes the augmented missing data imputed elements from Eq. (16), that is, collapsing the  $f(\mathbf{X} | \mathbf{Y}, \beta, \mathbf{U}, \boldsymbol{\nu})$  therein. The sampling in Eq. (19) requires the inference of the probability of unusual zeros defined for the interactions between and within

those occupied groups. This can be accomplished by further proposing the conjugate prior distributions:

$$p_{gh} \sim \text{Beta}(\beta_1, \beta_2) \text{ for } g, h = 1, \dots, K, \quad (21)$$

leading to a typical Gibbs sampling step for each non-redundant probability of unusual zeros:

$$p_{gh} | \boldsymbol{\nu}, \mathbf{z} \sim \text{Beta}(\boldsymbol{\nu}_{gh} + \beta_1, n_{gh} - \boldsymbol{\nu}_{gh} + \beta_2), \text{ for } g, h = 1, 2, \dots, K, \quad (22)$$

where  $(\beta_1, \beta_2)$  are hyperparameters. Note that the conditional probability of unusual zero provided that the corresponding observed interaction is a zero interaction shown in Eq. (18) is of key interest for practitioners to explore when observing a zero interaction. We refer to Lu et al. (2024) for a detailed discussion of the  $\boldsymbol{\nu}$  sampling in Eq. (19).

### 3.2.2 Inference for $\mathbf{z}$

Carrying out inference for the clustering variable  $\mathbf{z}$  based on the supervised MFM prior in Eq. (12) simultaneously infers the clustering allocations and automatically chooses the number of groups. However, the dimension of the model parameter matrix  $\mathbf{P}$  becomes problematic when the nodes are proposed to be assigned to a new empty group. Thus, following the prior distribution introduced in Eq. (21), we can marginalize the target posterior in Eq. (16) in a different way compared to the posterior in Eq. (20). Here, we collapse the parameter  $\mathbf{P}$  from Eq. (16) following, for example, McDaid et al. (2012) and Lu et al. (2024), and this leads to a posterior:

$$\pi(\mathbf{X}, \boldsymbol{\nu}, \mathbf{U}, \mathbf{z}, \beta | \mathbf{Y}, \mathbf{c}) \propto f(\mathbf{X} | \beta, \mathbf{U}) f(\boldsymbol{\nu} | \mathbf{z}) f(\mathbf{U} | \mathbf{z}) f(\mathbf{z} | \mathbf{c}) \pi(\beta), \quad (23)$$

where the collapsed likelihood function  $f(\boldsymbol{\nu} | \mathbf{z})$  reads as follows:

$$f(\boldsymbol{\nu} | \mathbf{z}) = \prod_{g=1, h=1}^K \left[ \frac{B(\boldsymbol{\nu}_{gh} + \beta_1, n_{gh} - \boldsymbol{\nu}_{gh} + \beta_2)}{B(\beta_1, \beta_2)} \right]. \quad (24)$$

The sampling of each  $z_i$  is based on the normalized probability proportional to its full-conditional distribution of the posterior in Eq. (23) (Legramanti et al., 2022), that is,

$$P(z_i = k | \boldsymbol{\nu}, \mathbf{U}, \mathbf{c}, \mathbf{z}^{-i}) \propto P(z_i = k | \mathbf{c}, \mathbf{z}^{-i}) f(\boldsymbol{\nu} | z_i = k, \mathbf{z}^{-i}) f(\mathbf{U} | z_i = k, \mathbf{z}^{-i}), \quad (25)$$

where the notation  $\mathbf{z}^{-i} := \mathbf{z} \setminus \{z_i\}$  contains all the clustering indicators except  $z_i$ , and the  $P(z_i = k | \mathbf{c}, \mathbf{z}^{-i})$  follows the supervised MFM urn scheme in Eq. (12) by assuming that the node  $i$  is removed from the network and then is treated as a new node to be assigned a group in the network. More specifically, we have

$$P(z_i = k | \mathbf{c}, \mathbf{z}^{-i}) \propto \begin{cases} \frac{n_{k, c_i}^{-i} + \omega_{c_i}}{n_k^{-i} + \omega_0} (n_k^{-i} + \alpha), & \text{for } k = 1, 2, \dots, K^{-i}; \\ \frac{\omega_{c_i}}{\omega_0} \frac{\mathcal{W}_{N-i+1, K-i+1}}{\mathcal{W}_{N-i+1, K-i}} \alpha & \text{for } k = K^{-i} + 1, \end{cases} \quad (26)$$

where  $(\cdot)^{-i}$  denotes the corresponding statistics obtained after removing node  $i$  from the network. If removing node  $i$  makes one of the groups empty, the remaining non-empty groups in  $\mathbf{z}^{-i}$  should be relabeled in ascending order by letting  $z_j = z_j - 1$  for all the

$\{z_j : j = 1, 2, \dots, N; j \neq i; z_j > z_i\}$  during the inference procedures. If  $\mathbf{c}$  is not available, the  $P(z_i = k | \mathbf{c}, \mathbf{z}^{-i})$  should be replaced with the  $P(z_i = k | \mathbf{z}^{-i})$  which instead follows the unsupervised MFM urn scheme in Eq. (9), that is, the specific form can be obtained by removing the  $(n_{k,c_i}^{-i} + \omega_{c_i})/(n_k^{-i} + \omega_0)$  and the  $\omega_{c_i}/\omega_0$  terms in Eq. (26). We refer to Miller and Harrison (2018), Geng et al. (2019), Legramanti et al. (2022), and Lu et al. (2024) for more details of the inference procedure of  $\mathbf{z}$ .

### 3.3 Truncated absorb-eject move

Since the latent clustering variable  $\mathbf{z}$  is updated one element at a time based on Eq. (25), the inference algorithm is susceptible to remain stuck in a local posterior mode. Thus, we propose to leverage an Absorb-Eject (AE) move proposed by Nobile and Fearnside (2007) to facilitate the clustering inference and to deal with such a sampling issue. However, the AE move, as suggested by Nobile and Fearnside (2007), may create empty groups, and this does not work well with our method, which requires non-empty groups. Thus we instead propose a Truncated Absorb-Eject (TAE) move which specifically addresses this issue by no longer creating empty groups, as we describe more in detail here below.

Similar to a typical AE move, we have two reversible moves in each iteration of the inference algorithm: a truncated *eject* move, denoted as  $eject^T$ , and an *absorb* move. In general, with probability  $P(eject^T)$ , the  $eject^T$  move is applied and, with probability  $1 - P(eject^T)$ , the *absorb* move is applied. As an exception, the  $eject^T$  move is applied with probability 1 if  $K = 1$ , while the *absorb* move is applied with probability 1 when  $K = N$ .

- $Eject^T$  move: first randomly pick one of the  $K$  non-empty groups, say group  $g$ . Then, we sample an ejection probability from a prior distribution,  $p_e \sim \text{Beta}(a, a)$ , and each node in group  $g$  has probability  $p_e$  to be reallocated to the new group labelled as  $K+1$ . Thus, on the contrary, each node stays in group  $g$  with probability  $1 - p_e$ . The proposed state is denoted as  $(\mathbf{z}', K' = K + 1)$  after the reallocation. If this process creates an empty group, that is, either the proposed group  $g$  or the proposed group  $K+1$  in  $\mathbf{z}'$  is an empty group, we propose to abandon this truncated AE move and remains at the current state,  $(\mathbf{z}, K)$ .

If the reallocation does not create an empty group, we propose:

$$\{\mathbf{z}, K\} \rightarrow \{\mathbf{z}', K' = K + 1\},$$

with the proposal probability

$$P(\{\mathbf{z}, K\} \rightarrow \{\mathbf{z}', K'\}) = \frac{\int_0^1 p_e^{n'_{K'}} (1 - p_e)^{n'_g} \pi_{\text{beta}}(p_e | a, a) dp_e}{1 - p_0} P(eject^T) \frac{1}{K}, \quad (27)$$

where  $n'_{K'}$  and  $n'_g$ , respectively, denotes the number of nodes in group  $g$  and in group  $K'$  of the proposed clustering  $\mathbf{z}'$ . Here, the reallocation probability  $p_e$  is collapsed leading to:

$$\begin{aligned} \int_0^1 p_e^{n'_{K'}} (1 - p_e)^{n'_g} \pi_{\text{beta}}(p_e | a, a) dp_e &= \int_0^1 p_e^{n'_{K'}} (1 - p_e)^{n'_g} \frac{\Gamma(2a)}{\Gamma(a)^2} (1 - p_e)^{a-1} p_e^{a-1} dp_e \\ &= \frac{\Gamma(2a)}{\Gamma(a)^2} \int_0^1 p_e^{n'_{K'} + a - 1} (1 - p_e)^{n'_g + a - 1} = \frac{\Gamma(2a)}{\Gamma(a)^2} \frac{\Gamma(a + n'_g) \Gamma(a + n'_{K'})}{\Gamma(2a + n_g)}, \end{aligned} \quad (28)$$

where  $n_g$  is the number of nodes in group  $g$  of the current clustering  $\mathbf{z}$ , and note here that  $n_g = n'_{K'} + n'_g$ . The  $p_0$  in Eq. (27) is the probability that all the selected nodes are reallocated in one group leaving another group empty, and is calculated following the similar way as Eq. (28), that is,

$$P(n'_{K'} = 0) = P(n'_g = 0) = \frac{p_0}{2} = \int_0^1 p_e^0 (1 - p_e)^{n_g} \pi_{\text{beta}}(p_e | a, a) dp_e = \frac{\Gamma(2a)}{\Gamma(a)} \frac{\Gamma(a + n_g)}{\Gamma(2a + n_g)}.$$

The prior parameter  $a$  for the reallocation probability  $p_e$  is set by checking the pre-computed look-up table of  $p_0$  with respect to  $a$  and  $n_g$ . According to our simulation studies, since we do not expect too many moves to be abandoned due to creating empty groups, we propose to set  $p_0 = 0.02$  in our experiments.

On the contrary, the reverse proposal probability is

$$P(\{\mathbf{z}', K'\} \rightarrow \{\mathbf{z}, K\}) = \frac{P(\text{absorb})}{K}.$$

Thus this  $\text{eject}^T$  move is accepted with probability

$$\min \left( 1, \frac{f(\boldsymbol{\nu}|\mathbf{z}') f(\mathbf{U}|\mathbf{z}') f(\mathbf{z}'|\mathbf{c})}{f(\boldsymbol{\nu}|\mathbf{z}) f(\mathbf{U}|\mathbf{z}) f(\mathbf{z}|\mathbf{c})} \frac{P(\{\mathbf{z}', K'\} \rightarrow \{\mathbf{z}, K\})}{P(\{\mathbf{z}, K\} \rightarrow \{\mathbf{z}', K'\})} \right), \quad (29)$$

where the  $f(\boldsymbol{\nu}|\mathbf{z})$ ,  $f(\mathbf{U}|\mathbf{z})$  and  $f(\mathbf{z}|\mathbf{c})$  terms are, respectively, evaluated based on Eqs. (24), (17) and (10). Otherwise, we remain at the current state  $(\mathbf{z}, K)$ .

- *Absorb* move: first randomly select two groups, say groups  $g, h$  with  $g < h$ , from the  $K$  groups and merge them together into cluster  $g$ . Then relabel all the groups in ascending order from group 1 to group  $K' := K - 1$ . We accept this *absorb* move with the same probability scheme in Eq. (29) but with different proposal probability:

$$P(\{\mathbf{z}, K\} \rightarrow \{\mathbf{z}', K'\}) = \frac{P(\text{absorb})}{K(K-1)},$$

while the reverse proposal probability is

$$P(\{\mathbf{z}', K'\} \rightarrow \{\mathbf{z}, K\}) = \frac{\frac{\Gamma(2a)}{\Gamma(a)^2} \frac{\Gamma(a+n_g)\Gamma(a+n_h)}{\Gamma(2a+n'_g)}}{1 - 2 \frac{\Gamma(2a)}{\Gamma(a)} \frac{\Gamma(a+n'_g)}{\Gamma(2a+n'_g)}} \frac{P(\text{eject}^T)}{K'(K'+1)}.$$

The proof of the above formula is similar to Eq. (28). We refer to Nobile and Fearnside (2007) for more details. If the absorb move is not accepted, we remain at the current state.

### 3.4 Partially collapsed Metropolis-within-Gibbs

Inferring the indicators of unusual zeros  $\boldsymbol{\nu}$  and the latent clustering indicators  $\mathbf{z}$  from different posteriors does not guarantee the convergence to the same target distribution. However, note that both posteriors in Eq. (20) and in Eq. (23) are different partially

collapsed forms of the posterior in Eq. (16) which is further a partially collapsed form of the posterior:

$$\pi(\mathbf{X}, \boldsymbol{\nu}, \bar{\mathbf{P}}, \mathbf{U}, \mathbf{z}, \beta | \mathbf{Y}, \mathbf{c}) \propto f(\mathbf{X} | \mathbf{Y}, \beta, \mathbf{U}, \boldsymbol{\nu}) f(\mathbf{Y} | \beta, \mathbf{U}, \boldsymbol{\nu}) f(\boldsymbol{\nu} | \bar{\mathbf{P}}, \mathbf{z}) f(\mathbf{U} | \mathbf{z}) f(\mathbf{z} | \mathbf{c}) \pi(\beta) \pi(\bar{\mathbf{P}}), \quad (30)$$

where  $\bar{\mathbf{P}}$  is defined as a  $N \times N$  matrix of which  $\mathbf{P}$  is a submatrix. The joint prior distribution  $\pi(\bar{\mathbf{P}}) \propto \prod_{g=1, h=1}^N f_{\text{Beta}}(\bar{p}_{gh} | \beta_1, \beta_2)$  is proposed to be the product up to the maximum reachable value of  $K$ , that is,  $N$ , and the  $\{\bar{p}_{gh}\}$  here are the entries of  $\bar{\mathbf{P}}$ . Thus we leverage the partially collapsed Gibbs approach proposed by Van Dyk and Park (2008) and Park and Van Dyk (2009) to ensure the correct target stationary distribution. Our Algorithm 1 further adopts the Metropolis-Hastings (M-H) steps of inferring  $\beta$  and  $\mathbf{U}$  leading to the Partially Collapsed Metropolis-within-Gibbs (PCMwG) algorithm. Note that the full-conditional distribution of  $\beta$  remains the same under either the posterior in Eq. (16) or the posterior in Eq. (30), that is,

$$p(\beta | \mathbf{X}, \mathbf{U}) \propto f(\mathbf{X} | \beta, \mathbf{U}) \pi(\beta) = \left[ \prod_{i,j: i \neq j}^N f_{\text{Pois}}(x_{ij} | \exp(\beta - \|\mathbf{u}_i - \mathbf{u}_j\|)) \right] \pi(\beta), \quad (31)$$

while the full-conditional distribution of each  $\mathbf{u}_i$  is also invariant under the two different posteriors:

$$\begin{aligned} p(\mathbf{u}_i | z_i = k, \beta, \mathbf{X}, \mathbf{U} \setminus \{\mathbf{u}_i\}) &\propto f(\mathbf{X} | \beta, \mathbf{U}) f(\mathbf{U} | \mathbf{z}^{-i}, z_i = k) \\ &\propto \left[ \prod_{j:j \neq i}^N f_{\text{Pois}}(x_{ij}, x_{ji} | \exp(\beta - \|\mathbf{u}_i - \mathbf{u}_j\|)) \right] \left[ \alpha_2 - \frac{\left\| \sum_{j:z_j=k} \mathbf{u}_j \right\|^2}{n_k + \omega} + \sum_{j:z_j=k} \|\mathbf{u}_j\|^2 \right]^{-(\frac{d}{2} n_k + \alpha_1)}. \end{aligned} \quad (32)$$

Here we set normal proposal distribution with variance  $\sigma_\beta^2$  for the M-H step of  $\beta$ , while we propose a multivariate normal proposal distribution for each latent position  $\mathbf{u}_i$  with covariance  $\sigma_U^2 \mathbb{I}_d$ . Both proposal distributions are centred at the previous state of the corresponding parameter or latent variables.

According to Van Dyk and Park (2008) and Park and Van Dyk (2009), the ordering of the steps in Algorithm 1 matters in that the sampling of  $\mathbf{X}$  must be performed after the sampling on  $\boldsymbol{\nu}$ ; also, the inference on  $\mathbf{P}$  must be performed after the step on  $\mathbf{z}$  and after the truncated AE move. If these requirements are not satisfied, the algorithm would yield a Markov chain targeting an unknown stationary distribution.

The posterior samples of interests are those of  $\boldsymbol{\nu}, \beta, \mathbf{U}, \mathbf{z}$  and  $K$ . Since the sampled values of each  $\nu_{ij}$  are either 0s or 1s, the posterior mean of the  $\nu_{ij}$ , denoted as  $\hat{\nu}_{ij}$ , provides an approximation of the conditional probability in Eq. (18). Such an approximation takes into account the uncertainty generated by  $\beta, \mathbf{U}, \mathbf{z}$  and  $\mathbf{P}$ , where  $p_{z_i z_j}$  varies along with different posterior clustering  $\mathbf{z}$  throughout the posterior samples. Similarly, the posterior mean is also calculated for the intercept  $\beta$  to obtain its summary statistic,  $\hat{\beta}$ .

As concerns the latent clustering variable, we may obtain a point estimate via

$$\hat{\mathbf{z}} = \arg \min_{\mathbf{z}'} \mathbb{E}_{\text{post}}[\mathcal{L}_{VI}(\mathbf{z}, \mathbf{z}') | \mathbf{Y}, \mathbf{c}]. \quad (33)$$

Here  $\mathcal{L}_{VI}$  is the Variation of Information (VI) loss function that measures the ratio of the individual and the mutual entropies of the clusterings (Meilă, 2007; Rastelli and Friel,

---

**Algorithm 1** A partially collapsed Metropolis-within-Gibbs sampler for ZIP-LPCM.

---

**Input:**  $\mathbf{Y}, \mathbf{c}, \sigma_\beta^2, \sigma_U^2, \alpha_1, \alpha_2, \omega, \alpha, \beta_1, \beta_2, \{\omega_c : c = 1, 2, \dots, C\}$ .  
 Initialise  $\mathbf{U}, \mathbf{z}, \beta, \mathbf{P}, \boldsymbol{\nu}, \mathbf{X}$ .

**for**  $t = 1$  to  $T$  **do**

- for**  $i, j = 1, 2, \dots, N$  and  $i \neq j$  **do**
  - 1. Sample  $\nu_{ij}$  from Eq. (19).
  - 2. Sample  $x_{ij}$  from Eq. (3) where  $\lambda_{ij}$  is obtained from Eq. (4).
- end for**
- 3.1 Propose  $\beta' \sim N(\beta, \sigma_\beta^2)$ .
- 3.2 Based on Eq. (31), accept  $\beta = \beta'$  with probability  $\min\left(1, \frac{p(\beta'| \mathbf{X}, \mathbf{U})}{p(\beta | \mathbf{X}, \mathbf{U})}\right)$ .  
 Otherwise, set  $\beta = \beta$ .
- for**  $i = 1, 2, \dots, N$  **do**
  - 4.1 Propose  $\mathbf{u}'_i \sim MVN(\mathbf{u}_i, \sigma_U^2 \mathbb{I}_d)$ .
  - 4.2 Based on Eq. (32), accept  $\mathbf{u}_i = \mathbf{u}'_i$  with probability
$$\min\left(1, \frac{p(\mathbf{u}'_i | z_i, \beta, \mathbf{X}, \mathbf{U} \setminus \{\mathbf{u}_i\})}{p(\mathbf{u}_i | z_i, \beta, \mathbf{X}, \mathbf{U} \setminus \{\mathbf{u}_i\})}\right).$$
  - Otherwise, set  $\mathbf{u}_i = \mathbf{u}'_i$ .
- end for**
- for**  $i = 1, 2, \dots, N$  **do**
  - 5. Each clustering variable  $z_i$  is inferred to be  $k$  with the normalized probability proportional to Eq. (25) for  $k = 1, 2, \dots, K^{-i} + 1$ .
- end for**
- 6. The truncated AE move is applied following Section 3.3.
- for**  $g, h = 1, 2, \dots, K$  **do**
  - 7. Infer  $p_{gh}$  from Eq. (22).
- end for**
- end for**

**Output:** Posterior samples of  $\boldsymbol{\nu}, \mathbf{X}, \beta, \mathbf{U}, \mathbf{z}, \mathbf{P}, K$  for each iteration  $t = 1, 2, \dots, T$ .

---

2018; Legramanti et al., 2022; Wade and Ghahramani, 2018). An estimated  $\hat{K}$  can thus be obtained based on the number of non-empty groups in  $\hat{\mathbf{z}}$ . A natural approach to summarize posterior samples of latent positions  $\mathbf{U}$  is to first apply Procrustes transformation (Borg and Groenen, 2005) on each posterior sample with respect to a reference  $\mathbf{U}$ , and then to calculate the posterior mean from the posterior samples of each  $\mathbf{u}_i$ . However, we find that, under LPCMs, the latent positions can still keep rotating within each individual group, especially when the groups are well separated. This creates further complications that are not resolved by the standard Procrustes transformations. For this reason, we opt for a more pragmatic approach and resort to obtain a point estimate of  $\mathbf{U}$  via

$$\hat{\mathbf{U}} = \arg \max_{\mathbf{U}} \text{post} [f(\mathbf{X} | \beta, \mathbf{U}) f(\boldsymbol{\nu} | \mathbf{P}, \mathbf{z}) f(\mathbf{U} | \mathbf{z}) f(\mathbf{z} | \mathbf{c}) \mathbb{1}(\mathbf{z} = \hat{\mathbf{z}})], \quad (34)$$

which is the posterior latent positions from the posterior state which maximizes the complete likelihood function of the posterior in Eq. (30) among those states whose corresponding posterior clustering is identical to  $\hat{\mathbf{z}}$ . In the case that no posterior clustering agrees with  $\hat{\mathbf{z}}$ , the posterior states after burn-in process are all considered. However, this is not the case of any experiments we illustrate in the next sections.

### 3.5 Choice of the number of latent dimensions

In our simulations and applications, we generally propose  $d = 3$  for the latent positions, aiming to provide 3-d visualizations of the complex networks. Latent spaces with higher dimensions cannot be visualized in practice and are generally not required to represent most typical real networks, so  $d = 2$  or  $d = 3$  dimensions are usually more than sufficient. In fact, in Hoff et al. (2002), the authors defined (for binary networks) a concept of representability that determines by which dimension  $d$  of the latent positions a network is representable. However, it is not easy to theoretically evaluate the value  $d$ , especially for complex networks, regardless of whether such a concept can be well extended to weighted networks. In any case, the authors argue that a small value of  $d$  can be sufficient to represent a large variety of common social networks. In this spirit, we perform our simulation studies in three dimensions, and, in the real data applications of Section 5, we compare the  $d = 3$  performance to the corresponding  $d = 2$  performance for each real network we focus on, giving some nuanced views for the different cases. We note that model choice for LPMs remains a critical research question that is currently being addressed more in detail by other contemporary literature. Our work contributes in this research direction by providing additional examples of the typical results that can be obtained in two and three dimensional LPMs.

## 4 Simulation studies

In this section, we show the performance of the newly proposed ZIP-LPCM + MFM model in both a supervised and unsupervised setting. For clarity, the inference algorithm of the unsupervised case is obtained by simply replacing Eq. (12) with Eq. (9) in Eq. (25) within the Algorithm 1 Step 5, and by replacing Eq. (10) with Eq. (8) in Step 6 of the algorithm. Further, we also make comparisons with the binary LPCM explored in Ryan et al. (2017), which we suitably extend to the non-negative weighted networks framework. The extension of the binary LPCM to the weighted case is done by replacing the Bernoulli logistic link function with the Poisson distribution equipped with the link function in Eq. (4), leading to the Poisson LPCM (Pois-LPCM). Note that the Pois-LPCM is a specific case of the ZIP-LPCM if we let the probability of unusual zeros for each pair of nodes be zero, corresponding to the situation that no missing data is assumed for the model. Both supervised and unsupervised MFM priors are also proposed for the Pois-LPCM along with the truncated AE move to achieve a more fair performance comparison. Thus the corresponding inference processes for the Pois-LPCM also follow Algorithm 1, the only differences being the removal of Steps 1, 2, 7, as well as all the  $\nu$  terms in Steps 5, 6, and also treating  $\mathbf{X}$  as  $\mathbf{Y}$  instead.

We propose two simulation studies. In simulation study 1, we randomly generate two artificial networks: one from a ZIP-LPCM (scenario 1) and one from a Pois-LPCM (scenario 2). We implement the supervised and the unsupervised versions of ZIP-LPCM and of Pois-LPCM on the artificial network in each scenario.

Instead, in simulation study 2, we focus on synthetic networks which are generated from the Zero-Inflated Poisson Stochastic Block Model (ZIP-SBM) explored in Lu et al. (2024). Also for this second simulation study, we consider two scenarios. In the first scenario we have a basic community structure, whereas in the second we also include hubs and thus disassortative patterns. Our goal is to analyze whether our newly proposed ZIP-LPCM is able to fit well to networks that generated from a different model, that is, the ZIP-SBM. Indeed, it is known that the ZIP-SBM is able to characterize specific structures which are not possible to represent with the latent positions of the ZIP-LPCM (due to the inherent transitivity of the latent space), and thus may be more flexible when compared to the ZIP-LPCM. However, the ZIP-SBM cannot capture much variability within the blocks, and cannot provide the latent space views of the networks, thus making the ZIP-LPCM a viable modeling choice. Note that the simulation settings of all the models in this section mimic the structures of the real networks from our real data applications, thus, the simulation study performance provide valuable benchmarks for the analyses of real networks.

## 4.1 Simulation study 1

We propose synthetic networks with  $N = 75$  nodes and  $K = 5$  clusters, where the true clustering  $\mathbf{z}^*$  has group sizes:  $n_1 = 5, n_2 = 10, n_3 = 15, n_4 = 20$  and  $n_5 = 25$ . The network that we generate for scenario 1 is randomly simulated from a ZIP-LPCM with settings:  $\beta = 3, \boldsymbol{\tau} = (\frac{1}{0.25}, \frac{1}{0.50}, \frac{1}{0.75}, 1, \frac{1}{1.25})$  and

$$\boldsymbol{\mu} = \left[ \begin{pmatrix} -1.5 \\ -1.5 \\ -1.5 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \\ 2 \end{pmatrix} \right], \mathbf{P} = \begin{pmatrix} 0.40 & 0.05 & 0.10 & 0.05 & 0.10 \\ 0.10 & 0.40 & 0.05 & 0.10 & 0.05 \\ 0.05 & 0.10 & 0.40 & 0.05 & 0.10 \\ 0.10 & 0.05 & 0.10 & 0.40 & 0.05 \\ 0.05 & 0.10 & 0.05 & 0.10 & 0.40 \end{pmatrix}.$$

The network in scenario 2 is simulated from a Pois-LPCM with the same set of parameters shown above, but excluding the probability of unusual zeros  $\mathbf{P}$ . In order to assess the performance of the estimation procedure, the above parameter values, which are used for generating the networks, are treated as the reference values to be compared to the corresponding posterior samples. We further use  $(\cdot)^*$  to denote these reference values, that is,  $(K^*, \mathbf{z}^*, \beta^*, \boldsymbol{\tau}^*, \boldsymbol{\mu}^*, \mathbf{P}^*)$  indicate the true parameter values that have generated the data. However, not all of these parameters are actually used, for example, the parameters  $\boldsymbol{\tau}$  and  $\boldsymbol{\mu}$  are marginalized out during the inference and thus they are not taken into account. Figure 1 illustrates the latent positions and the clustering used for generating the networks in two scenarios. The pattern of the latent positions for the second scenario are similar to the first scenario one but with denser edges because no missing data is assumed. We refer to <https://github.com/Chaoyi-Lu/ZIP-LPCM> for more details including the 3-d interactive plots of the latent positions as well as all the implementation code of the experiments in this paper. Figure 2 shows the heatmap plots of the adjacency matrices for both synthetic networks.

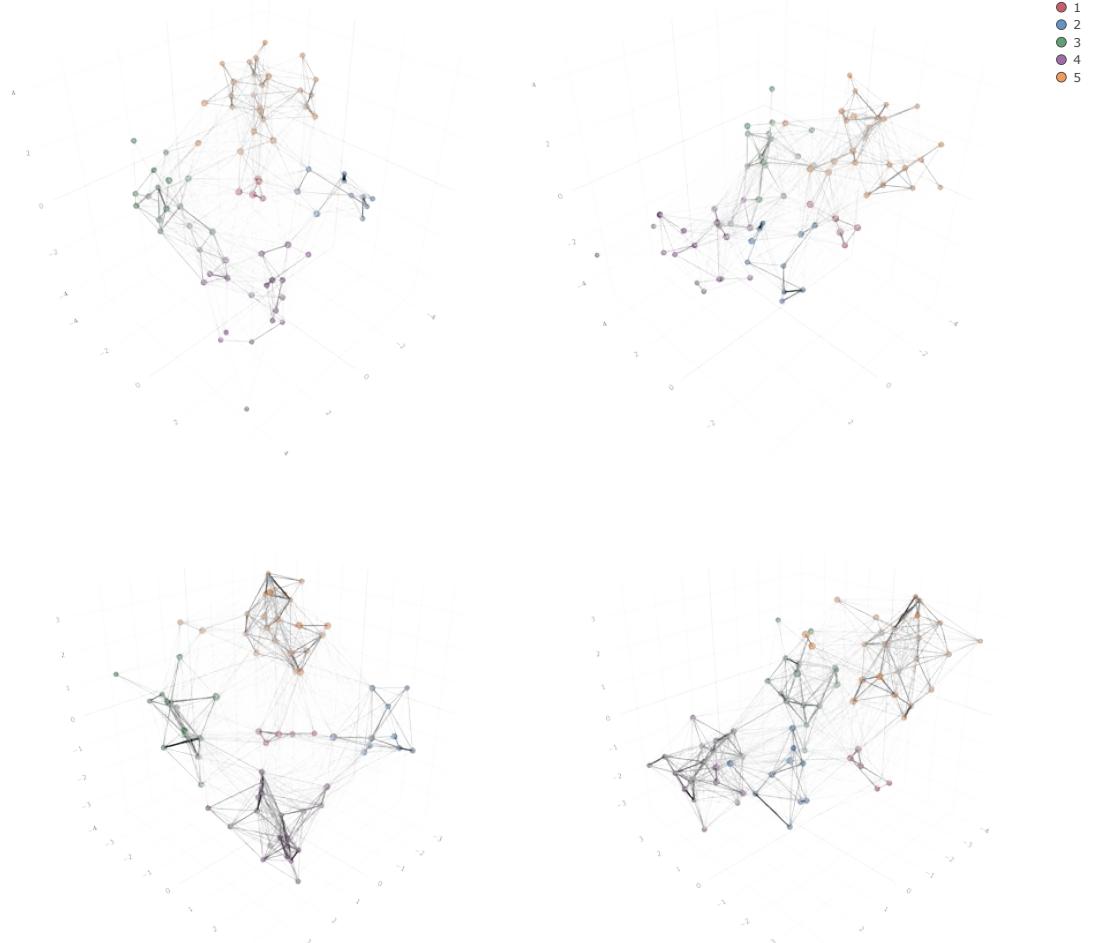


Figure 1: Simulation study 1 synthetic networks. The 1st row plots correspond to the scenario 1 network, while the 2nd row plots correspond to the scenario 2 network. The 1st column plots show the 3-dimensional plots of the latent positions with different node colors denoting the corresponding true clustering. Node sizes are proportional to node betweenness, whereas edge widths and colors are proportional to edge weights. The 2nd column plots are the rotated plots of the 1st column latent position plots that are rotated for 90° clockwise with respect to the vertical axis.

Prior settings similar to those of Handcock et al. (2007) and Ryan et al. (2017) are applied for all the experiments of this paper:  $\alpha = 3$  for the MFM and  $\alpha_1 = 1, \alpha_2 = 0.103$  in Eq. (15). The tuning parameter  $\omega$  in Eq. (14) is instead set as 0.01, encouraging more split clusters in the latent space. A common prior distribution of  $\bar{K}$  in MFM is assumed as we discussed in Section 3.1, that is,  $\bar{K} \sim \text{Pois}(1)|\bar{K} > 1$ . Based on the ZIP-SBM results illustrated in Lu et al. (2024), we also propose to set a Beta(1, 9) prior for the probability of unusual zeros by default in Eq. (21), but we also check in the simulation studies the sensitivity of this prior setting by considering four other different options, that is, Beta(1, 1), Beta(1, 3), Beta(1, 19) and Beta(1, 99).

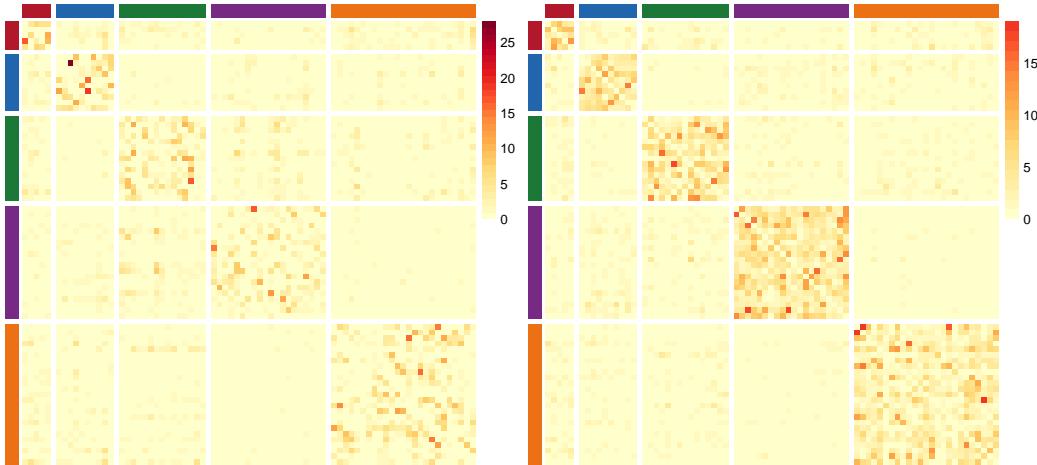


Figure 2: Simulation study 1. Synthetic networks' adjacency matrix heatmap plots. Darker entries correspond to higher edge weights. The side-bars indicate the reference clustering  $\mathbf{z}^*$ . Left plot: scenario 1 network, generated from a ZIP-LPCM. Right plot: scenario 2 network, generated from a Pois-LPCM.

As regards the node attributes, a contaminated version of the true clustering  $\mathbf{z}^*$  is used as the exogenous node attributes  $\mathbf{c}$  where the clustering of 20 out of 75 nodes are randomly reallocated. This setting aims to check whether the output are robust when noise exists in the node attributes. The prior of  $\beta$  is proposed to be a continuous uniform distribution defined on a large enough interval centered around zero so that the corresponding probability density function can always be canceled in the acceptance ratio of Step 3.2 of Algorithm 1. The cohesion parameter of the supervised MFM is canonically set as  $\omega_c = 1$  for  $c = 1, 2, \dots, C$ .

With regard to the MCMC posterior samples, we run the algorithm for 12,000 iterations, where the first 2000 iterations are discarded as burn-in for each setting that we consider. The initial clustering is proposed to be the trivial clustering where each node occupies a different group. Latent positions are initialized by a natural approach where classical multidimensional scaling (Gower, 1966) is applied on the geodesic distance matrix of the observed adjacency matrix, in the same style as Hoff et al. (2002). The proposal variances of M-H steps of  $\beta$  and  $\mathbf{U}$  are tuned so that the corresponding acceptance rates are approximately 0.23. The probability of applying an  $eject^T$  move is set as  $P(eject^T) = 0.5$  by default.

Recall that a point estimate of the posterior clustering and of the latent positions is, respectively, obtained by Eq. (33) and Eq. (34). The set of distances  $\{d_{ij} = \|\mathbf{u}_i - \mathbf{u}_j\| : i, j = 1, 2, \dots, N; i > j\}$  between each pair of nodes are invariant under any rotation or translation of  $\mathbf{U}$ , so that the set of  $\{\hat{d}_{ij}\}$  can be obtained via posterior mean of each  $d_{ij}$  in the posterior samples, accounting for the uncertainty of all other model parameters and latent variables. We treat the distances obtained from  $\mathbf{U}^*$  as the corresponding reference values to be compared to those  $\{\hat{d}_{ij}\}$ , that is,  $\{d_{ij}^* = \|\mathbf{u}_i^* - \mathbf{u}_j^*\| : i, j = 1, 2, \dots, N; i > j\}$ . Similarly, the probability of unusual zeros  $\mathbf{P}$  is, by definition, composed by  $\{p_{gh} : g, h = 1, 2, \dots, K\}$  for each pair of non-empty clusters and is dependent on the clustering. Due to the fact that the posterior clustering keeps mixing during the inference leading to

different number of clusters, the identifiability problem occurs in the posterior samples of  $\mathbf{P}$ . However, instead of such a group-level parameter, we focus on an individual-level  $N \times N$  matrix,  $\mathbf{p}$ , within which each  $ij$ th entry is the probability of unusual zero  $p_{z_iz_j}$ , and we obtain the corresponding  $\hat{p}_{z_iz_j}$  via posterior mean accounting for the uncertainty of the posterior clustering. The  $p_{z_iz_j}^*$  is proposed to be zero for each pair of nodes  $i, j$  in scenario 2 by considering that the Pois-LPCM is a specific case of the ZIP-LPCM.

The performance of eight different cases are illustrated in Table 1. The table shows that the supervised implementations are able to provide better clustering performance with lower uncertainty in both scenarios as expected, even though there is significant contamination existing in the exogenous node attributes. The Pois-LPCM cases fail to recover the true clustering and are shown to be unable to fit well to the network generated from the ZIP-LPCM in scenario 1. On the contrary, the ZIP-LPCM provides good performance in both scenario 1 and 2, especially when the parameter  $\beta_2$  of the prior of the probability of unusual zeros is set to be around 9. In the case that  $\beta_2 = 99$ , the prior encourages very small probability of unusual zeros, and this makes the ZIP-LPCM become close to the Pois-LPCM leading to small  $\{\hat{p}_{z_iz_j}\}$  shown as, for example,  $\mathbb{E}(\{|\hat{p}_{z_iz_j} - p_{z_iz_j}^*|\})[\text{sd}]$  where  $p_{z_iz_j}^* = 0$  in scenario 2 of Table 1. Moreover, the inferred  $\hat{\mathbf{z}}$  from the cases of “ZIP-LPCM Sup Beta(1,99)” and “Pois-LPCM Sup” is also very similar in scenario 1. Though the estimated clustering is not detailed here, the materials can be provided upon request or following the provided code in the **Code and Data** section to reproduce the output.

Table 1: Simulation study 1. Performance of eight different implementations where (i)  $\hat{K}$ : the number of clusters in  $\hat{\mathbf{z}}$ ; (ii)  $\text{VI}(\hat{\mathbf{z}}, \mathbf{z}^*)$ : the VI distance between the point estimate  $\hat{\mathbf{z}}$  and the true clustering  $\mathbf{z}^*$ ; (iii)  $\mathbb{E}_{\mathbf{z}}[\text{VI}(\hat{\mathbf{z}}, \mathbf{z}) \mid \mathbf{Y}]$ : the minimized expected posterior VI loss of the clustering with respect to  $\hat{\mathbf{z}}$ . This statistic measures the uncertainty of the posterior clustering around the  $\hat{\mathbf{z}}$ ; (iv)  $\mathbb{E}(\{|\hat{d}_{ij} - d_{ij}^*|\})[\text{sd}]$ : the mean of  $\{|\hat{d}_{ij} - d_{ij}^*| : i, j = 1, 2, \dots, N; i > j\}$  with the corresponding standard deviation (sd) shown in the square bracket; (v)  $\hat{\beta}$ : the posterior mean of  $\beta$ ; (vi)  $\mathbb{E}(\{|\hat{p}_{z_iz_j} - p_{z_iz_j}^*|\})[\text{sd}]$ : the mean of  $\{|\hat{p}_{z_iz_j} - p_{z_iz_j}^*| : i, j = 1, 2, \dots, N; i > j\}$  with sd in the square bracket. More details are included in Section 4.1. The best performance within each column are highlighted in bold font.

SCENARIO	$\hat{K}$		$\text{VI}(\hat{\mathbf{z}}, \mathbf{z}^*)$		$\mathbb{E}_{\mathbf{z}}[\text{VI}(\hat{\mathbf{z}}, \mathbf{z}) \mid \mathbf{Y}]$		$\mathbb{E}(\{ \hat{d}_{ij} - d_{ij}^* \})[\text{sd}]$		$\hat{\beta}$		$\mathbb{E}(\{ \hat{p}_{z_iz_j} - p_{z_iz_j}^* \})[\text{sd}]$	
	1	2	1	2	1	2	1	2	1	2	1	2
ZIP-LPCM Sup Beta(1,1)	7	5	0.68	<b>0.00</b>	0.551	0.031	0.359[0.266]	1.356[1.097]	2.95	2.93	0.193[0.158]	0.157[0.127]
ZIP-LPCM Sup Beta(1,3)	5	5	0.00	<b>0.00</b>	0.018	<b>0.008</b>	0.250[0.203]	1.347[1.088]	3.03	<b>2.96</b>	0.084[0.065]	0.116[0.081]
ZIP-LPCM Sup Beta(1,9)	5	5	0.00	<b>0.00</b>	0.009	0.011	<b>0.249[0.201]</b>	1.341[1.081]	<b>3.02</b>	<b>2.96</b>	0.033[0.034]	0.074[0.050]
ZIP-LPCM unSup Beta(1,9)	5	5	0.00	<b>0.00</b>	0.072	0.052	0.250[0.202]	1.339[1.080]	<b>3.02</b>	<b>2.96</b>	0.032[0.031]	0.069[0.043]
ZIP-LPCM Sup Beta(1,19)	5	5	0.00	<b>0.00</b>	<b>0.005</b>	0.021	0.262[0.210]	1.334[1.074]	3.03	2.93	<b>0.029[0.025]</b>	0.039[0.019]
ZIP-LPCM Sup Beta(1,99)	5	5	0.27	<b>0.00</b>	0.023	0.029	0.290[0.229]	1.334[1.074]	<b>3.02</b>	<b>2.96</b>	0.084[0.055]	<b>0.009[0.002]</b>
Pois-LPCM Sup	5	5	0.42	<b>0.00</b>	0.369	0.025	0.384[0.293]	<b>1.322[1.070]</b>	2.62	2.91	—	—
Pois-LPCM unSup	2	5	1.53	<b>0.00</b>	1.086	0.072	0.400[0.320]	1.333[1.072]	2.63	<b>2.96</b>	—	—

The results also highlight that the ZIP-LPCM implementations with higher prior mean of the probability of unusual zeros tend to overestimate  $K$  on ZIP-LPCM networks. For example, in scenario 1 the posterior clustering of the “ZIP-LPCM Sup Beta(1,1)” tends to split some of the groups into smaller subgroups, leading to  $\hat{K} = 7$  against  $K^* = 5$ , even though such an implementation is supervised and thus uses some extra information. On the contrary, the unsupervised Pois-LPCM underestimates  $K$ : the posterior clustering of the “Pois-LPCM unSup” merges several groups together and provides  $\hat{K} = 2$ . Considering that the Pois-LPCM is in fact an extreme case of the ZIP-LPCM wherein

all the probability of unusual zeros is set to be zero, this overestimated/underestimated  $K$  performance correspond to the fact that the probability of unusual zeros controls the sparsity of the network, where larger probability corresponds to sparser latent positions bringing more separated clusters, and vice-versa.

Each element  $\nu_{ij}$  of the indicators of unusual zeros  $\boldsymbol{\nu}$  is defined to be either 1 or 0. Hence, the posterior mean of each  $\nu_{ij}$  measures the proportion of the times that the corresponding  $y_{ij} > 0$ , the posterior samples of  $\nu_{ij}$  are all inferred to zero by default. Denoting  $\hat{\nu}_{ij}$  as the posterior mean of  $\nu_{ij}$ , then  $\hat{\nu}_{ij}$  becomes an approximation of the conditional probability of the unusual zeros in Eq. (18) accounting for the posterior uncertainty of all other model parameters and latent variables. The performance of  $\hat{\nu}_{ij}$  for the supervised ZIP-LPCM cases from scenario 1 are shown in Figure 3. We note that the Beta(1, 1) case is excluded because

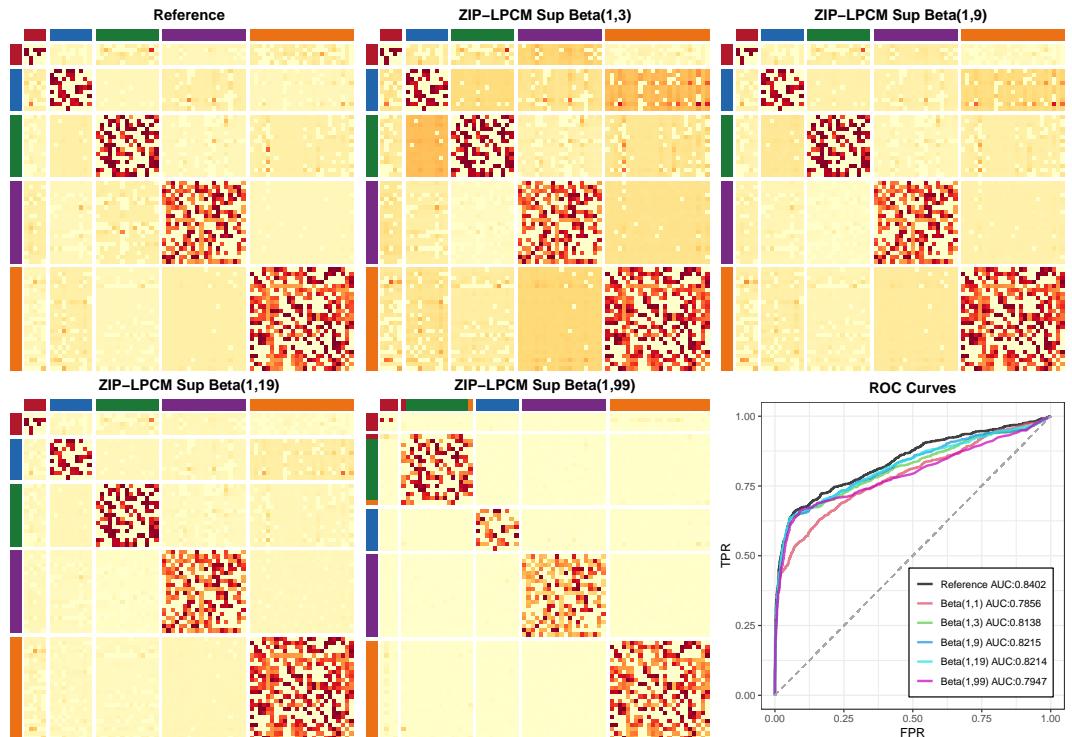


Figure 3: Simulation study 1 scenario 1. Performance of the posterior mean  $\hat{\nu}$ , which approximates the conditional probability in Eq. (18). The top-left plot is the heatmap of the reference values of Eq. (18), obtained by leveraging the reference model parameters used for simulating the network, whereby darker entry colors correspond to higher values. The other four heatmap plots describe  $\hat{\nu}$  as inferred by the corresponding priors indicated on top of the heatmap. The rows and columns of the matrices are rearranged and separated according to  $\hat{z}$  while the side-bars indicate the true clustering of each individual. The last plot shows the Receiver Operating Characteristic (ROC) curves for all the supervised ZIP-LPCM cases, where the reference  $\boldsymbol{\nu}^*$  is the response variable.

its summarized clustering is not satisfactory. The figure shows that the Beta(1, 9) case provide the best matching and the best ROC curves with respect to the references, while slightly tuning the  $\beta_2$  provides comparable performance. However, it is also interesting that the Beta(1, 99) prior also successfully and correctly prioritizes many zero interactions

which are more likely to be unusual zeros compared to other zeros. In fact, more non-zero interactions exist in a particular block, more accurate the pairwise distances between the nodes inside are, hence bringing more robust inference results of how likely the zero interactions in the block are unusual zeros. Note here that in general smaller  $p_{ij}$  does not imply smaller value of Eq. (18): that also depends on  $\beta$  and  $\mathbf{U}$ . Hence, from a practical standpoint we suggest the Beta(1, 9) as the default prior setting for the probability of unusual zeros, whereas priors with smaller means can be considered when practitioners expect a more clustered network or prefer to learn more towards true zeros rather than unusual zeros. By contrast, higher prior mean is encouraged if practitioners expect sparser network architecture bringing more subgroup features. However, moderate tuning of the prior parameters does not significantly affect the overall performance as shown in both Figure 3 and Table 1.

## 4.2 Simulation study 2

In this second simulation study, the network size  $N$  and the clustering  $\mathbf{z}^*$  of the synthetic networks are the same as those proposed in the first simulation study. The networks are randomly simulated from the ZIP-SBM which can be obtained by removing the latent position parts in Eq. (4) and in Eq. (5) from the ZIP-LPCM, and directly proposing a  $\text{Pois}(\lambda_{z_i z_j})$  in Eq. (2). The model parameter  $\boldsymbol{\lambda}$  is a  $K \times K$  matrix with entries  $\{\lambda_{gh}\}$ :  $g, h = 1, 2, \dots, K\}$  being the Poisson rates defined for the interactions between any two occupied groups.

In scenario 1 of simulation study 2, the values of  $\boldsymbol{\lambda}$  used for generating the network are indicated with  $\boldsymbol{\lambda}_1$  in the equation shown below, whereas the probability of unusual zeros  $\mathbf{P}$  is the same as the one in Section 4.1.

$$\boldsymbol{\lambda}_1 = \begin{pmatrix} 7.0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 4.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 3.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 2.0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 2.5 \end{pmatrix}, \boldsymbol{\lambda}_2 = \begin{pmatrix} 7.0 & 2.0 & 2.0 & 2.0 & 2.0 \\ 2.0 & 4.5 & 0.5 & 0.5 & 0.5 \\ 2.0 & 0.5 & 3.5 & 0.5 & 0.5 \\ 2.0 & 0.5 & 0.5 & 2.0 & 0.5 \\ 2.0 & 0.5 & 0.5 & 0.5 & 2.5 \end{pmatrix}, \mathbf{P}_2 = \begin{pmatrix} 0.40 & 0.60 & 0.20 & 0.60 & 0.20 \\ 0.20 & 0.40 & 0.05 & 0.10 & 0.05 \\ 0.60 & 0.10 & 0.40 & 0.05 & 0.10 \\ 0.20 & 0.05 & 0.10 & 0.40 & 0.05 \\ 0.60 & 0.10 & 0.05 & 0.10 & 0.40 \end{pmatrix}.$$

The values indicated with  $\boldsymbol{\lambda}_2$  and  $\mathbf{P}_2$  describe instead the setup for the second scenario of this simulation study, whereby the interaction rate matrix  $\boldsymbol{\lambda}_2$  includes a group of hubs, that is, nodes that have relatively high connection rates to all other groups. The heatmap plots of the networks' adjacency matrices from both scenarios are shown in Figure 4.

Figure 5 illustrates the inferred  $\hat{\mathbf{U}}$  obtained by the supervised ZIP-LPCM with the default Beta(1, 9) prior setting, whereas Table 2 illustrates the performance of all the implementations that we take into account in this second simulation study. The latent positions show well-separated clusters when they are inferred for ZIP-SBM networks, with significant hubs exiting in the center for scenario 2. Based on the  $\boldsymbol{\lambda}^*$ 's used for simulating the networks, it is also shown that smaller  $\{\lambda_{gh}^*\}$  correspond to sparser inferred latent positions between and within the clusters. The presence of the hubs leads to slightly more clustered latent positions in scenario 2, which in turn brings slightly higher  $\hat{\beta}$  in scenario 2 as shown in Table 2.

In the ZIP-LPCM, the  $\lambda_{ij}$  can be obtained based on  $\beta$  and  $\mathbf{U}$  following Eq. (4) for each pair of nodes  $i, j$ : we use this to construct the posterior samples of  $\lambda_{ij}$  for each

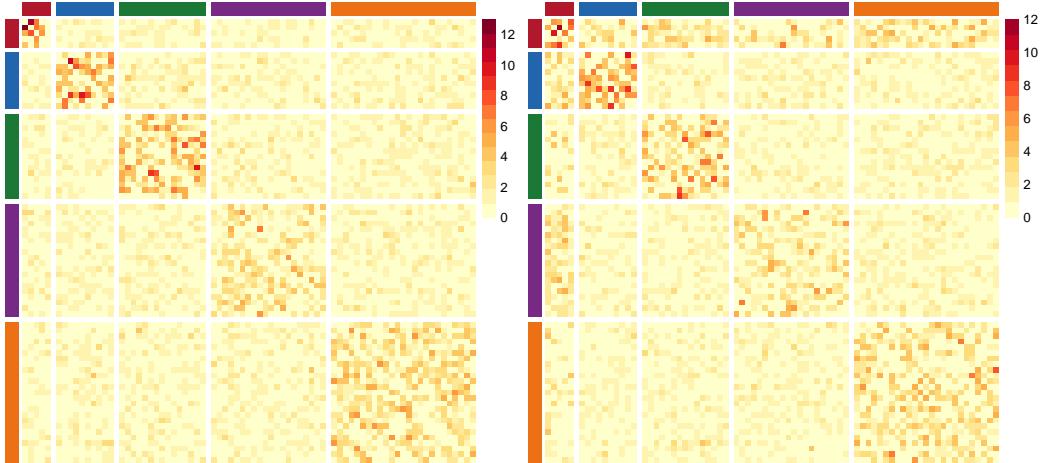


Figure 4: Simulation study 2. Synthetic networks' adjacency matrix heatmap plots. Darker entries correspond to higher edge weights. The side-bars indicate the reference clustering  $\mathbf{z}^*$ . Left plot: scenario 1 network, generated from a ZIP-SBM without hubs. Right plot: scenario 2 network, generated from a ZIP-SBM with hubs.

Table 2: Simulation study 2. Performance of eight different implementations where (i)  $\hat{K}$ : the number of clusters in  $\hat{\mathbf{z}}$ ; (ii)  $\text{VI}(\hat{\mathbf{z}}, \mathbf{z}^*)$ : the VI distance between the point estimate  $\hat{\mathbf{z}}$  and the true clustering  $\mathbf{z}^*$ ; (iii)  $\mathbb{E}_{\mathbf{z}}[\text{VI}(\hat{\mathbf{z}}, \mathbf{z}) \mid \mathbf{Y}]$ : the minimized expected posterior VI loss of the clustering with respect to  $\hat{\mathbf{z}}$ . This statistic measures the uncertainty of the posterior clustering around the  $\hat{\mathbf{z}}$ ; (iv)  $\hat{\beta}$ : the posterior mean of  $\beta$ ; (v)  $\mathbb{E}(\{|\hat{p}_{z_i z_j} - p_{z_i z_j}^*|\})[\text{sd}]$ : the mean of  $\{|\hat{p}_{z_i z_j} - p_{z_i z_j}^*|\} : i, j = 1, 2, \dots, N; i > j\}$  with the corresponding standard deviation (sd) shown in the square bracket; (vi)  $\mathbb{E}(\{|\hat{\lambda}_{ij} - \lambda_{ij}^*|\})[\text{sd}]$ : the mean of  $\{|\hat{\lambda}_{ij} - \lambda_{ij}^*|\} : i, j = 1, 2, \dots, N; i > j\}$  with the corresponding sd in the square bracket. More details are included in Section 4.2. The best performance within each column excluding the  $\hat{\beta}$  column are highlighted in bold font.

SCENARIO	$\hat{K}$		$\text{VI}(\hat{\mathbf{z}}, \mathbf{z}^*)$		$\mathbb{E}_{\mathbf{z}}[\text{VI}(\hat{\mathbf{z}}, \mathbf{z}) \mid \mathbf{Y}]$		$\hat{\beta}$		$\mathbb{E}(\{ \hat{p}_{z_i z_j} - p_{z_i z_j}^* \})[\text{sd}]$		$\mathbb{E}(\{ \hat{\lambda}_{ij} - \lambda_{ij}^* \})[\text{sd}]$	
	1	2	1	2	1	2	1	2	1	2	1	2
ZIP-LPCM Sup Beta(1,1)	5	5	0.00	0.00	0.00	0.17	1.42	1.60	0.076[0.069]	0.092[0.078]	0.152[0.261]	0.245[0.299]
ZIP-LPCM Sup Beta(1,3)	5	5	0.00	0.00	0.00	0.00	1.42	1.56	0.063[0.071]	0.073[0.067]	0.149[0.259]	0.243[0.306]
ZIP-LPCM Sup Beta(1,9)	5	5	0.00	0.00	0.00	0.00	1.40	1.55	0.051[0.048]	0.059[0.061]	0.149[0.265]	0.240[0.318]
ZIP-LPCM unSup Beta(1,9)	5	5	0.00	0.00	0.00	0.00	1.42	1.55	0.042[0.038]	0.060[0.058]	0.145[0.260]	<b>0.239[0.319]</b>
ZIP-LPCM Sup Beta(1,19)	5	5	0.00	0.00	0.00	0.00	1.39	1.54	<b>0.037[0.026]</b>	<b>0.056[0.071]</b>	<b>0.141[0.267]</b>	0.239[0.329]
ZIP-LPCM Sup Beta(1,99)	5	5	0.00	0.42	0.19	0.08	1.37	1.53	0.082[0.049]	0.112[0.126]	0.151[0.288]	0.287[0.413]
Pois-LPCM Sup	5	5	0.00	0.00	0.07	0.05	1.03	1.15	—	—	0.312[0.511]	0.461[0.564]
Pois-LPCM unSup	5	4	0.00	0.40	0.21	0.23	1.02	1.16	—	—	0.312[0.514]	0.465[0.573]

ZIP-LPCM implementation. The posterior mean of these samples forms the  $\hat{\lambda}_{ij}$  which contributes to the statistic,  $\mathbb{E}(\{|\hat{\lambda}_{ij} - \lambda_{ij}^*|\})[\text{sd}]$ , shown in Table 2. The corresponding reference parameter  $\lambda_{ij}^*$  is obtained according to  $\lambda_{z_i^* z_j^*}^*$ .

In general, our ZIP-LPCM still performs reasonably well in situations where the data are generated using a ZIP-SBM. However, we note that in this case the best performance is provided by the Beta(1,19) prior instead of the default Beta(1,9). The Pois-LPCM fails to recover the true clustering of the scenario 2 network when the inference is unsupervised. The same deteriorated performance is also observed for the supervised ZIP-LPCM with Beta(1,99) prior, the performance of which is considered to be close to that of the Pois-LPCM. After a careful analysis of the resulting posterior samples, we noticed that even though these two settings are able to recover the true clustering, it is easy for them

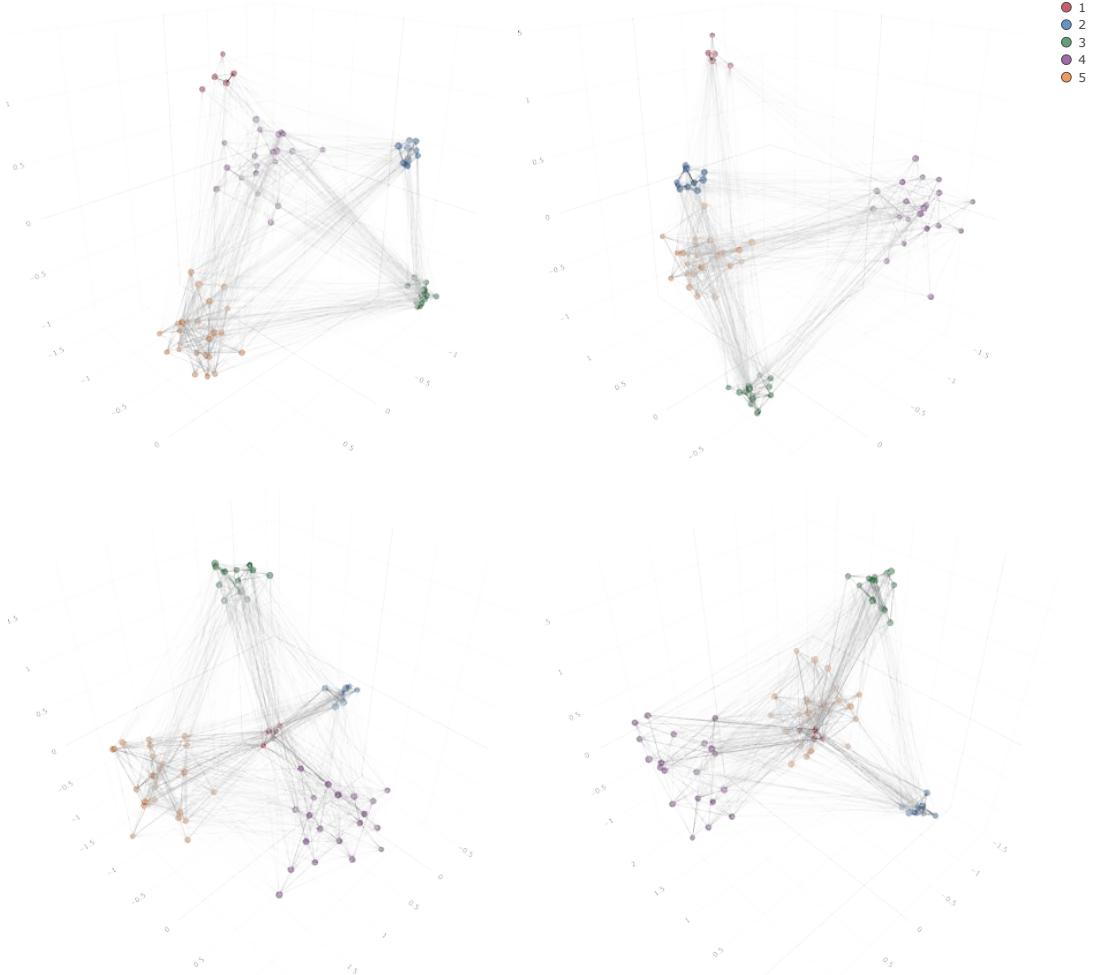


Figure 5: Simulation study 2. The 1st and the 2nd rows illustrate the inferred point estimate  $\hat{U}$  obtained by ZIP-LPCM Sup Beta(1,9) implementations for Scenario 1 and Scenario 2, respectively. The 2nd column plots are rotated version of the 1st column plots where each inferred latent position rotated for  $90^\circ$  clockwise with respect to the vertical axis. Different node colors correspond to different inferred groups according to the corresponding  $\hat{z}$ . Node sizes are proportional to node betweenness while edge widths and colors are proportional to edge weights.

to get stuck around some local posterior mode of the clustering, whereby the hubs are merged into another group. We find that these simulation results highlight that the zero-inflation feature of the model becomes critical to efficiently characterize the group differences and thus recover the correct partitioning of the nodes.

Figure 6 further illustrates the performance of  $\hat{\nu}$  for the ZIP-LPCM. Though the elements in the diagonal blocks are well approximated, the elements in the off-diagonal blocks are difficult to infer. This shows that, in some cases, it is difficult for the ZIP-LPCM in the latent space to capture specific network architectures produced by those

freely chosen model parameters of ZIP-SBM. This is especially significant for the asymmetric patterns of the upper-diagonal and the lower-diagonal blocks shown in Figure 6. However, we can still observe remarkably many elements in the off-diagonal blocks of  $\hat{\nu}$  from scenario 2 that have significantly higher values than others, especially for those blocks associated to the hubs. The patterns of these elements generally agree with the corresponding reference patterns, thus successfully and correctly highlighting non-negligible many zero interactions that are more likely to be unusual zeros compared to others.

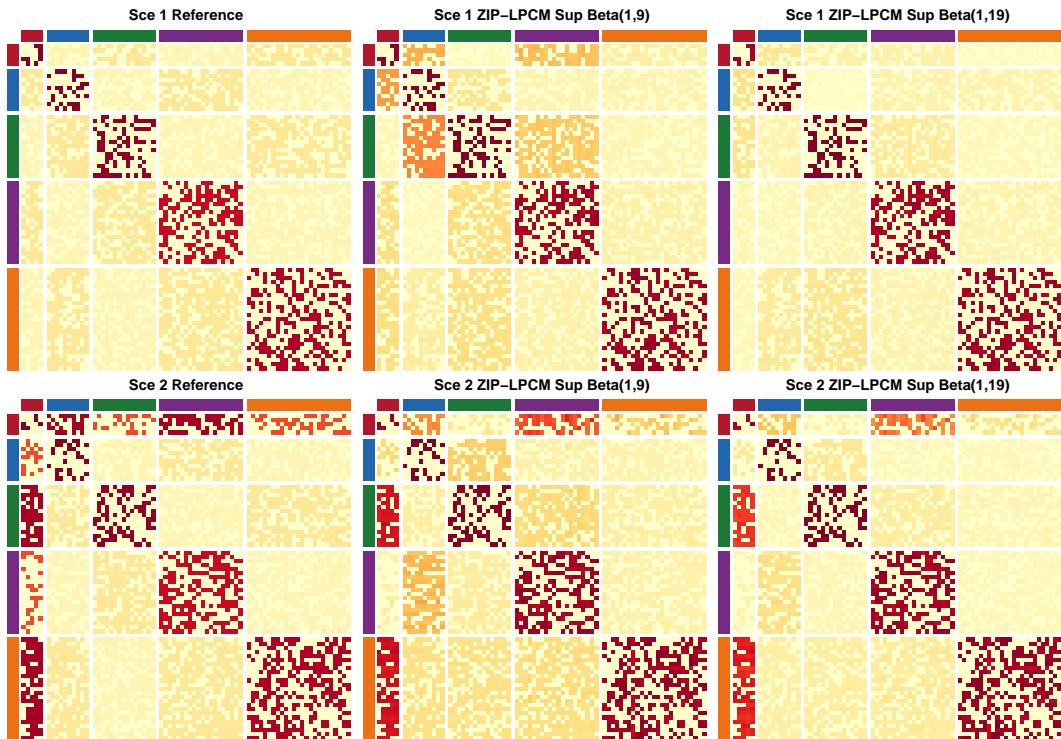


Figure 6: Simulation study 2. Performance of  $\hat{\nu}$  based on the Beta(1,9) and Beta(1,19) prior settings. The 1st and 2nd rows, respectively, correspond to scenarios 1 and 2. Darker colors indicate higher (approximate) probability of unusual zero conditional on the fact that the corresponding observed interaction is a zero interaction. The rows and columns of the matrices are rearranged and separated according to  $\hat{z}$  while the side-bars indicate the true clustering of each individual.

As a final note, we also applied the supervised ZIP-SBM with the Beta(1,9) prior and a Ga(1,1) Poisson rate prior, which extend the applications in Lu et al. (2024) to directed networks, on both ZIP-SBM networks from the two different scenarios. We indicate these as “ZIP-SBM Sup” here. Since the network data was generated from the ZIP-SBM, it is expected that the “ZIP-SBM Sup” implementations have natural advantages over ZIP-LPCM implementations in this case. However, we note that, though the results of the “ZIP-SBM Sup” implementations are even better than the best “ZIP-LPCM Sup Beta(1,19)” cases illustrated in Table 2 and Figure 4, the discrepancies in performance between them are not substantial. Especially for the inferred approximate conditional probability of unusual zeros  $\{\hat{\nu}_{ij}\}$ , the mean absolute error of which between the “ZIP-

SBM Sup” case and the “ZIP-LPCM Sup Beta(1,19)” case is 0.0403 in scenario 1, and is 0.0301 in scenario 2, while the corresponding standard deviation is 0.0320 and 0.0756, respectively. Considering that the ZIP-SBM is not able to visualize networks, these deterioration can be fairly neglected compared to the gains in the ability of network visualization for the ZIP-LPCM.

## 5 Real data applications

In this section, we illustrate the ZIP-LPCM performance on four different real networks. The experiment priors and proposal variance settings are similar to those applied in the simulation studies in Section 4 with the default prior setting of the probability of unusual zeros being Beta(1, 9). Since the real networks have different network sizes and complexity, a conservative 60,000 iterations are used for each real network, with the first 30,000 iterations treated as burn-in. We find that this leads to satisfactory mixing and convergence in all the applications considered.

In this section, we introduce an extra summary statistic to help with the illustration of the results. Recall that the posterior mean  $\hat{\nu}_{ij}$  approximates Eq. (18), which is the conditional probability of  $y_{ij}$  being an unusual zero provided that  $y_{ij}$  is observed as a zero. However, once an observed  $y_{ij} = 0$  is inferred as an unusual zero, the corresponding missing weight  $x_{ij} \sim \text{Pois}(\lambda_{ij})$  can be assumed following Eq. (3), where  $\lambda_{ij} = \exp(\beta - \|\mathbf{u}_i - \mathbf{u}_j\|)$ . This allows us to construct the conditional probability of an observed zero interaction that should actually be a non-zero interaction by using the product of  $P(x_{ij} > 0 | \beta, \mathbf{u}_i, \mathbf{u}_j) = 1 - f_{\text{Pois}}(0 | \lambda_{ij})$  and Eq. (18), confirmed by:

$$P(x_{ij} > 0 | y_{ij} = 0, \beta, \mathbf{u}_i, \mathbf{u}_j, p_{z_i z_j}) = P(x_{ij} > 0 | \beta, \mathbf{u}_i, \mathbf{u}_j)P(\nu_{ij} = 1 | y_{ij} = 0, \beta, \mathbf{u}_i, \mathbf{u}_j, p_{z_i z_j}).$$

The above probability can be approximated by  $\hat{P}(x_{ij} > 0 | y_{ij} = 0, \dots) = [1 - f_{\text{Pois}}(0 | \hat{\lambda}_{ij})]\hat{\nu}_{ij}$  accounting for the uncertainty of the posterior samples. From a practical standpoint, this statistic is usually the one that the practitioners are more interested in, compared to Eq. (18). Here, the  $\hat{\lambda}_{ij}$  can be obtained by the posterior mean of Eq. (4) accounting for the uncertainty of posterior samples of  $\beta$  and  $\mathbf{U}$ .

### 5.1 Sampson monks network

The social interactions among a group of 18 monks were recorded by Sampson (1968) during his stay in a monastery. A political “crisis in the cloister” occurred in that period, leading to the expulsion of four monks and the voluntary departure of several others. The interactions were recorded as follows: each monk was asked at three different time points whether they had positive relations to each of other monks and to rank the three monks they were closest to. The dataset has been studied on many previous works, including Hoff et al. (2002). We aggregated the three time points leading to a directed non-negative discrete weighted social network describing different levels of friendships between the monks.

The observed adjacency matrix  $\mathbf{Y}$  is shown as the first plot of Figure 7. Each monk was classified by Sampson to be within one of the three groups: “Turks”, “Outcasts” or “Loyal”, and such a clustering is treated as the reference clustering  $\mathbf{z}^*$  for this network.

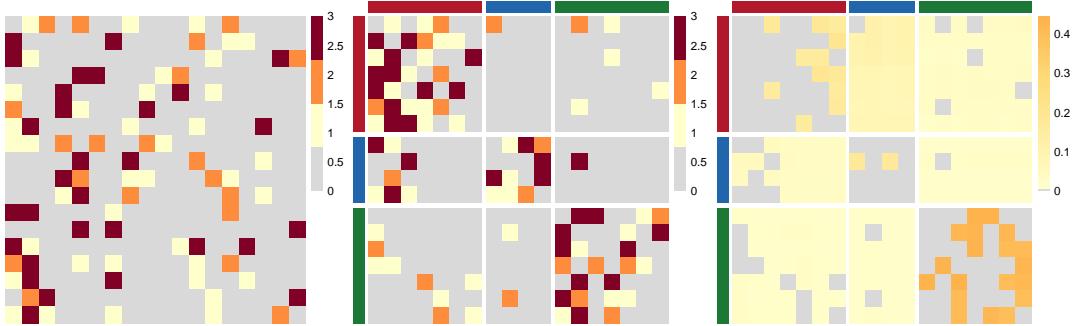


Figure 7: The heatmap plots for the Sampson monks real network where the greys are used for zero values in order to highlight other non-zero elements. Left plot: original observed adjacency matrix,  $\mathbf{Y}$ . Middle plot: plot of  $\mathbf{Y}$  where the rows and columns of the matrices are rearranged and separated according to  $\mathbf{z}^*$ . The different colors in the side-bars of this and the last plot correspond to different reference clustering of each individual. Right plot: inferred  $\hat{\mathbf{P}}(x_{ij} > 0 | y_{ij} = 0, \dots)$ .

The plot of  $\mathbf{Y}$  rearranged based on  $\mathbf{z}^*$  is shown as the second plot of Figure 7, where we use red, blue and green colors to represent the three reference groups, respectively. An extra true-or-false variable “Cloisterville” from De Nooy et al. (2018) was also added to this dataset to indicate whether or not each monk attended the minor seminary of “Cloisterville” before coming to the monastery. In our experiments, we leverage the combination of the  $\mathbf{z}^*$  and the “Cloisterville” information as the exogenous node attributes indicated with  $\mathbf{c}$ , to implement the supervised version of the ZIP-LPCM on this network. For example, the monks in the “Turks” group are separated into two different levels in  $\mathbf{c}$  depending on whether each of them attended the minor seminary or not. Similar for the other two groups leading to a total of  $C = 6$  levels in  $\mathbf{c}$ .

The inferred latent positions,  $\hat{\mathbf{U}}$ , shown in Figure 8 illustrate a typical SBM structure which resembles the ones shown in our simulation study 2 in Section 4.2. The inferred clustering  $\hat{\mathbf{z}}$  perfectly agrees with the reference clustering  $\mathbf{z}^*$ , even consider the more informative exogenous node attributes  $\mathbf{c}$  during inference. It is shown that the exogenous “Cloisterville” information does not bring more subgroup features to the clustering of this network under the ZIP-LPCM. Our unsupervised ZIP-LPCM implementation, which is not detailed here, also returns the same inferred clustering but with higher uncertainty of the posterior clustering.

The red “Turks” group is shown to be more tightly clustered than the blue “Outcasts” group, which is itself more clustered than the green “Loyal” group. According to the analysis of the conditional probability of observed zeros that should actually be non-zero interactions shown in the last plot of Figure 7, the within-group zero interactions of the “Loyal” group are more likely to be non-zeros compared to other zero interactions, though the corresponding inferred probability is around 0.4 which is not particularly high. The number of zero interactions from “Turks” to “Outcasts” are shown to be significantly less than that in the reverse direction, leading to slightly higher conditional probability of those zero-interactions being unusual zeros. However, considering the far latent distance between the two groups illustrated in Figure 8, the corresponding conditional probability of being non-zeros shown in the last plot of Figure 7 remains relatively small and negligible.

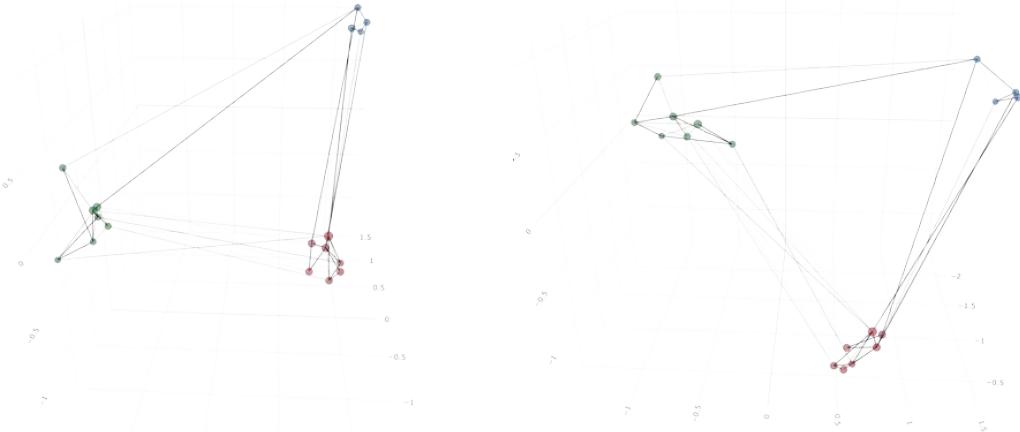


Figure 8: The Sampson monks real network. Left plot: inferred latent positions,  $\hat{\mathbf{U}}$ . The three inferred groups in  $\hat{\mathbf{z}}$  are distinguished by different colors, and perfectly agree with the reference clustering. Node sizes are proportional to node betweeness. Edge widths and colors are proportional to edge weights. Right plot: rotated version of the latent positions shown in the left plot where the whole latent space is rotated by  $90^\circ$  clockwise with respect to the vertical axis.

## 5.2 Windsurfers network

This is an undirected non-negative discrete weighted network (L. C. Freeman, Webster, et al., 1998; L. C. Freeman, S. C. Freeman, et al., 1988; Kunegis, 2013) recording the interpersonal contacts between 43 windsurfers in southern California during the fall of 1986. As the network size is larger than the Sampson monks network, the Beta(1, 19) prior of the probability of unusual zeros is proposed to encourage more clustering. Further, no reference clustering and exogenous node attributes are available for this network, so the unsupervised ZIP-LPCM is used.

Figure 9 shows that the whole network may generally be split into two main groups. However, it is interesting that our inferred clustering  $\hat{\mathbf{z}}$  returns  $\hat{K} = 3$  inferred groups, one of which is likely to be a core group since it sits at the center of another group. We use dark red color to represent the core group while the light red and light blue colors are used, respectively, for the corresponding non-core group and another group. In combination with the middle plot of Figure 10, it is shown that there are four windsurfers clustered into the red core group: they frequently interact with each other thus leading to very strong connections. This core group also actively interacts with all other windsurfers from the red non-core group, within which a few windsurfers seem to be the “close friends” of the core windsurfers. These findings indicate that the members from the core group may have a central or leader role or generally high reputation across the whole network.

It can also be observed in Figure 9 that four blue windsurfers are also well connected with each other and tend to form a core of the blue group, but their connection patterns tend to be significantly weaker than the red core group, and thus our model prefers not

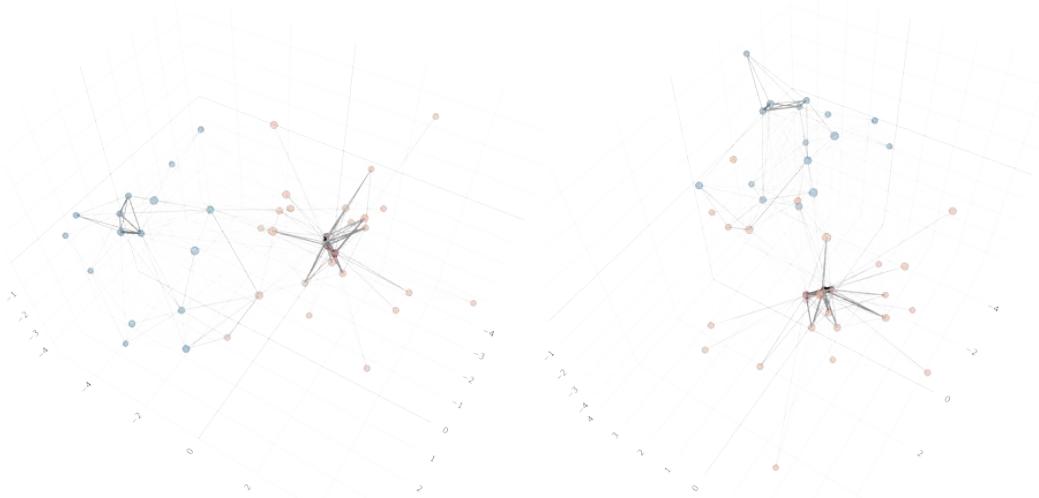


Figure 9: Windsurfers real network. Left plot: inferred latent positions,  $\hat{\mathbf{U}}$ . The three inferred groups in  $\hat{\mathbf{z}}$  are distinguished by different colors. Node sizes are proportional to node betweenness. Edge widths and colors are proportional to edge weights. Right plot: rotated version of the latent positions shown in the left plot where the whole latent space is rotated by  $90^\circ$  clockwise with respect to the vertical axis.

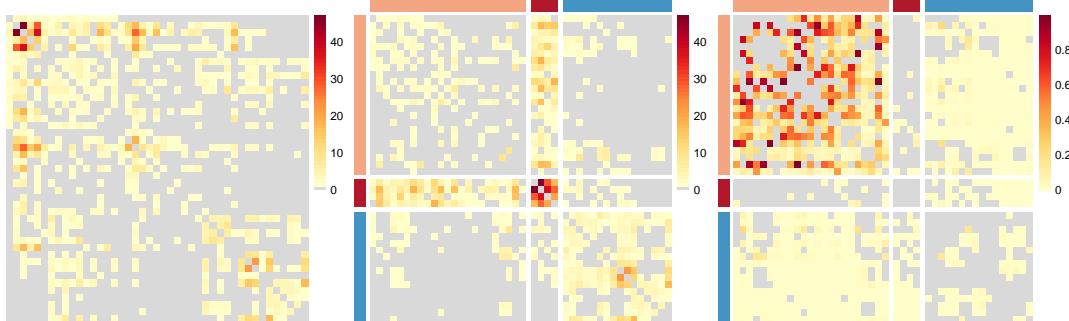


Figure 10: The heatmap plots for the windsurfers real network where the greys are used for zero values in order to highlight other non-zero elements. Left plot: original observed adjacency matrix,  $\mathbf{Y}$ . Middle plot: plot of  $\mathbf{Y}$  where the rows and columns of the matrices are rearranged and separated according to  $\hat{\mathbf{z}}$ . The different colors on the side-bars correspond to different inferred clustering of each individual. Right plot: inferred  $\hat{P}(x_{ij} > 0 | y_{ij} = 0, \dots)$ .

to distinguish them from the whole blue group. The zero interactions which are more likely to be non-zeros are mainly those within the red non-core group: some of them have very high probability to be non-zero interactions. This result indicates that either the corresponding interpersonal contact data was lost, or that those pairs of windsurfers might have other forms of interactions which were not recorded in this dataset.

### 5.3 Train bombing network

This dataset (Hayes, 2006; Kunegis, 2013) aims at reconstructing the contacts between suspected terrorists involved in the train bombing of Madrid on 11th March, 2004. The corresponding undirected network records 64 suspects, and the interaction strength be-

tween each pair of them indicates an aggregation of friendship and co-participating in training camps or previous attacks. For this dataset, we propose a Beta(1, 9) prior and we use the unsupervised setting.

The inferred latent positions and clustering shown in Figure 11 illustrate that a small red group containing five suspects forms the core of the whole network: strong connections between each pair of members inside this core group can be observed. The big blue

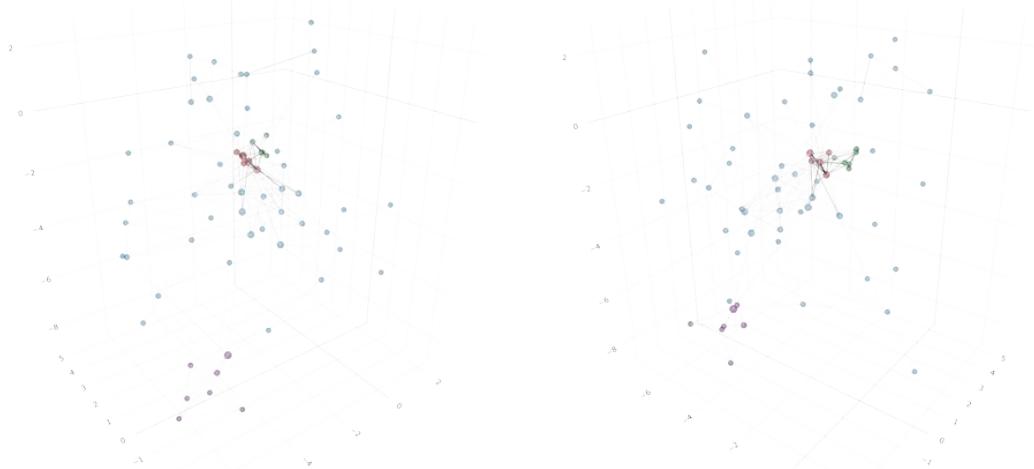


Figure 11: Train bombing real network. Left plot: inferred latent positions,  $\hat{U}$ . The inferred clustering distinguished by different colors. Node sizes are proportional to node betweenness. Edge widths and colors are proportional to edge weights. Right plot: rotated version of the latent positions shown in the left plot where the whole latent space is rotated by 90° clockwise with respect to the vertical axis.

group contains the non-core suspects and, based on the middle plot of Figure 12, some of them seem to be more “central” than others in light of the significantly stronger connections with each other and with the core nodes. By contrast, the other “non-central” blue

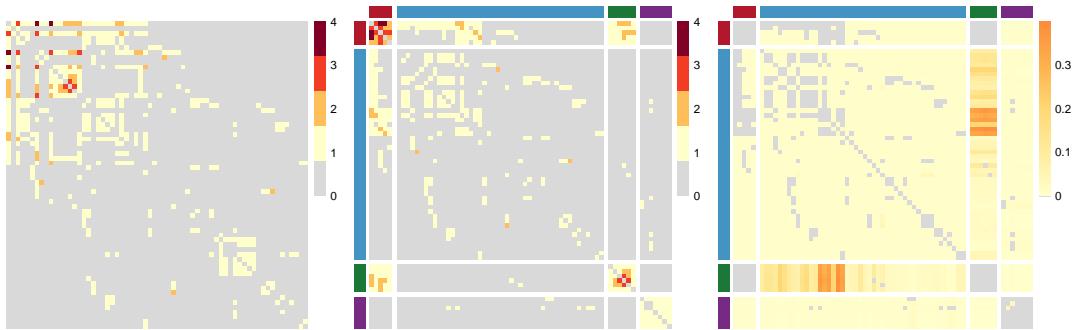


Figure 12: The heatmap plots for the train bombing real network where the greys are used for zero values in order to highlight other non-zero elements. Left plot: original observed adjacency matrix,  $\mathbf{Y}$ . Middle plot: the plot of  $\mathbf{Y}$  where the rows and columns of the matrices are rearranged and separated according to  $\hat{z}$ . The different colors in the side-bars of this and the last plot correspond to different inferred clustering of each individual. Right plot: inferred  $\hat{P}(x_{ij} > 0 | y_{ij} = 0, \dots)$ .

members do not interact with the core nodes and only interact with these “central” blue members. Though this “central” and “non-central” feature of the inferred blue group can be illustrated by the plots, our inference results suggest not to split them into two separate groups.

It is interesting that we also observe a small green group which contains five suspects who interact strongly with each other. This special group is shown to only interact well with the core red group and has very few connections with the non-core blue group. According to the corresponding inferred latent positions shown in Figure 11, these features are unusual and in agreement with the results shown in the last plot of Figure 12 where there are significantly many zero interactions, which are more likely to be non-zeros compared to other zero interactions, between this special green group and the non-core blue group. Finally, a small purple group seems to form a separate small network by themselves and only has few weak connections with the blue group.

#### 5.4 Summit co-attendance criminality network

The last real network that we propose in this paper was obtained from <https://sites.google.com/site/ucinetsoftware/datasets/covert-networks>, and was previously analysed in a number of works, including Legramanti et al. (2022) and Lu et al. (2024). This is an undirected network recording the co-attendance of summits for 84 criminal suspects operating in the north of Italy. The edge weights represent the number of summits that any two individuals co-attended. We consider a ground-truth partition of the nodes as in Lu et al. (2024), which we indicate with  $\mathbf{z}^*$ . This reference clustering is made of 10 groups, which describe the roles and affiliations of the suspects.

The network also contains additional exogenous node attributes, which we indicate with  $\mathbf{c}$ , and use for the supervised implementations. The additional node information is also a partition with  $C = 7$  clusters. Since these partitions contain more levels compared to the ones explored in the previous sections, we propose to apply a Beta(1, 6) as the prior of the probability of the unusual zeros, so that we encourage the network sparsity, thus bringing more subgroup features.

Figure 13 illustrates the inferred latent positions plotted along with a set of different node colors and shapes. The clustering in the 1st row plots is the reference clustering,  $\mathbf{z}^*$ , where different node colors correspond to different affiliations. Darker squares indicate individuals with more central roles within the organisation. Figure 14 shows the heatmaps for this network, with sidebars indicating  $\mathbf{z}^*$ , and row and column gaps indicating  $\hat{\mathbf{z}}$ . We note that there is a fairly strong agreement between the ground truth partition and the inferred one.

Combining both Figure 13 and 14, significant core-periphery structure can be observed. Based on the inferred latent positions, the core groups, consisting mainly of the criminal suspects with central roles shown as darker groups in the 2nd row plots of Figure 13, mostly sit at the center of the network, while the non-core groups are at the periphery. The core groups generally show a flat structure as they roughly lie within a flat plane, whereas the periphery non-core groups are more tridimensional and more pervasive. But the members of the non-core groups mainly have connections within their

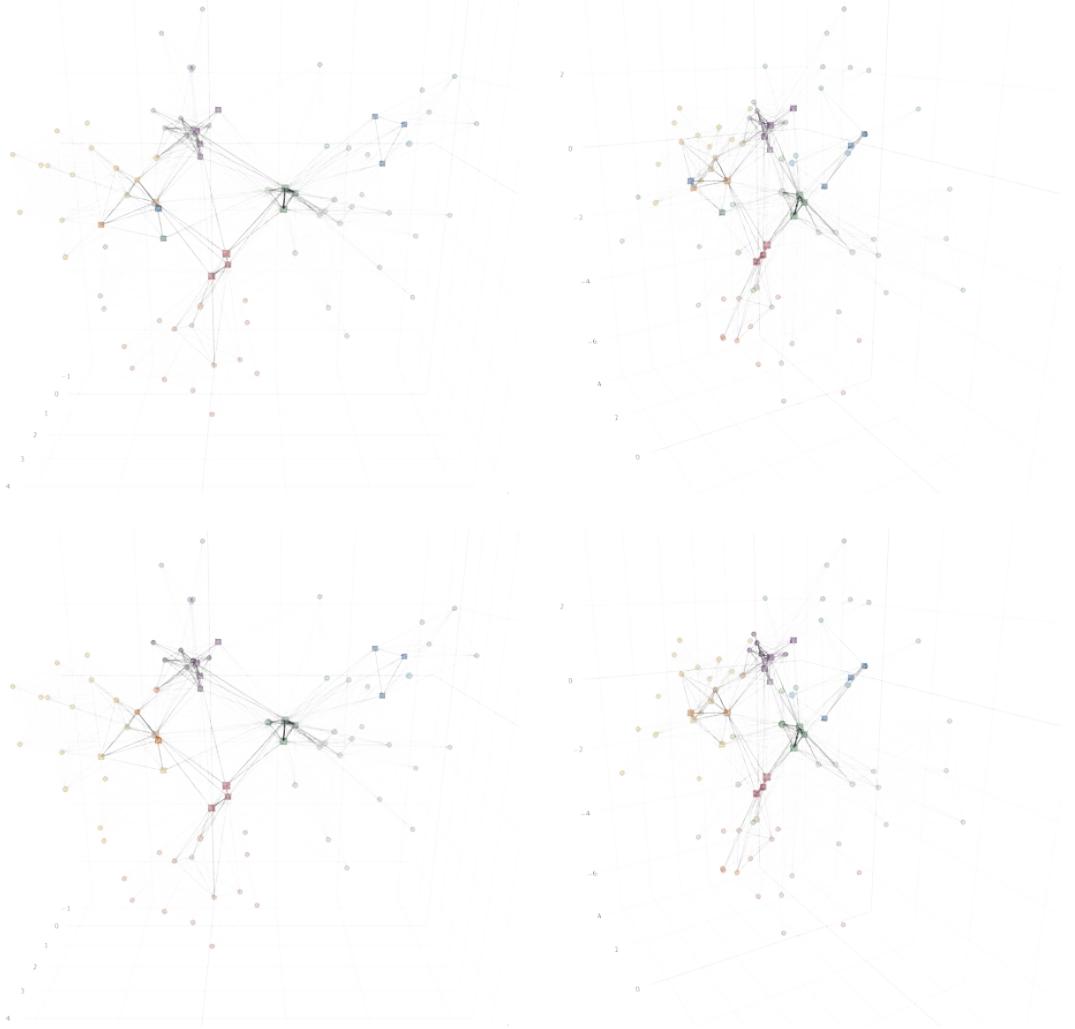


Figure 13: Summit co-attendance criminality network. The 1st row plots illustrate the inferred latent positions,  $\hat{U}$ , along with the different dark or light node colors indicating the reference clustering  $z^*$ . Darker square nodes indicate a more central role in the organization. The same  $\hat{U}$  is shown in the 2nd row plots but the node colors denote instead the inferred partition  $\hat{z}$ . The 2nd column plots are the rotated version of the latent positions shown in the 1st column plots where the whole latent space is rotated by  $60^\circ$  clockwise with respect to the vertical axis. Node sizes are proportional to node betweeness. Edge widths and colors are proportional to edge weights.

own group or with the corresponding core groups, indicating a lack of interactions between the non-core groups for this network. This is further confirmed by visualizing the first plot of Figure 14. The members within each core group are densely connected with each other, but significant sparsity between the cores can be observed, as exemplified by those relatively sparse between-group interactions of the cores at the center of the latent positions in Figure 13. However, most of the corresponding zeros within and between these core groups are not likely to be non-zeros according to the right plot of Figure 14.

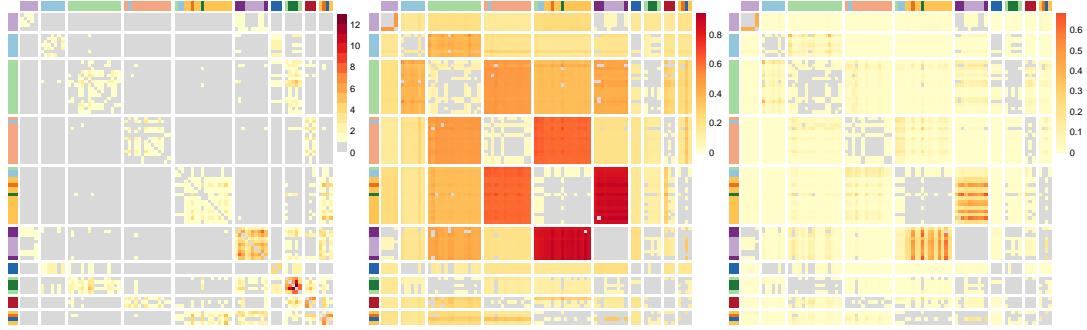


Figure 14: The output heatmap plots for the summit co-attendance criminality network where the greys are used for zero values in order to highlight other non-zero elements. The different colors in the side-bars correspond to the role-affiliation information,  $\mathbf{z}^*$ . The rows and columns of the matrices are rearranged and separated according to  $\hat{\mathbf{z}}$ , where the inferred groups containing central nodes are placed at the bottom while the others are placed at the top. Left plot: adjacency matrix  $\mathbf{Y}$ . Middle plot: inferred  $\hat{\nu}$ . Right plot: inferred  $\hat{P}(x_{ij} > 0 | y_{ij} = 0, \dots)$ .

It is worthwhile to note that our inferred clustering,  $\hat{\mathbf{z}}$ , is similar to the inferred clustering (denoted as  $\mathbf{z}^\dagger$  here) obtained by ZIP-SBM in Lu et al. (2024). However, one key difference between  $\hat{\mathbf{z}}$  and  $\mathbf{z}^\dagger$  is relevant to some of the central nodes, including two orange nodes, one blue and one green, whose inferred latent positions are close to each other and are near the center of the network as shown in Figure 13.

## 5.5 Comparisons to inference on 2-dimensional latent space

We finish this section of real data applications by comparing the inference output obtained in a 3-dimensional latent space to the output obtained in a 2-dimensional latent space for the real datasets. The analyses are performed in an analogous way, with the obvious exception that the dimension of the latent positions is instead set as  $d = 2$ . Figure 15 illustrates the inferred 2-dimensional latent positions for the real networks. In summary, these results show that the plot of the 2-d inferred latent positions is roughly like a screenshot of the corresponding 3-d inferred latent positions from a specific angle, for example, if we compare the top-left plot of Figure 15, which illustrates the 2-dimensional  $\hat{\mathbf{U}}$  of the windsurfers network, to the left plot of Figure 9. Similarly, the bottom-left plot of Figure 15, which shows the 2-dimensional  $\hat{\mathbf{U}}$  and the reference clustering  $\mathbf{z}^*$  of the criminality network, generally follow the pattern which is similar to that of the top-left plot of Figure 13 which is a screenshot of the 3-dimensional  $\hat{\mathbf{U}}$  of this network.

However, this does not mean that the third dimension is redundant, actually, there is good evidence that a 2-dimensional latent space is not sufficient to characterize the architectures of these complex networks. For example, in the summit co-attendance criminality network, the green nodes are distributed pervasively and tend to overlap with red and blue nodes in 2-d latent space as shown in the bottom-left plot of Figure 15. In comparison, this group of nodes can be well separated along the extra 3rd dimension in 3-d latent space according to Figure 13. Furthermore, it is also shown that the networks are generally inferred to be more aggregated in 2-d space compared to the corresponding 3-d cases, making it harder to well infer the network clustering.

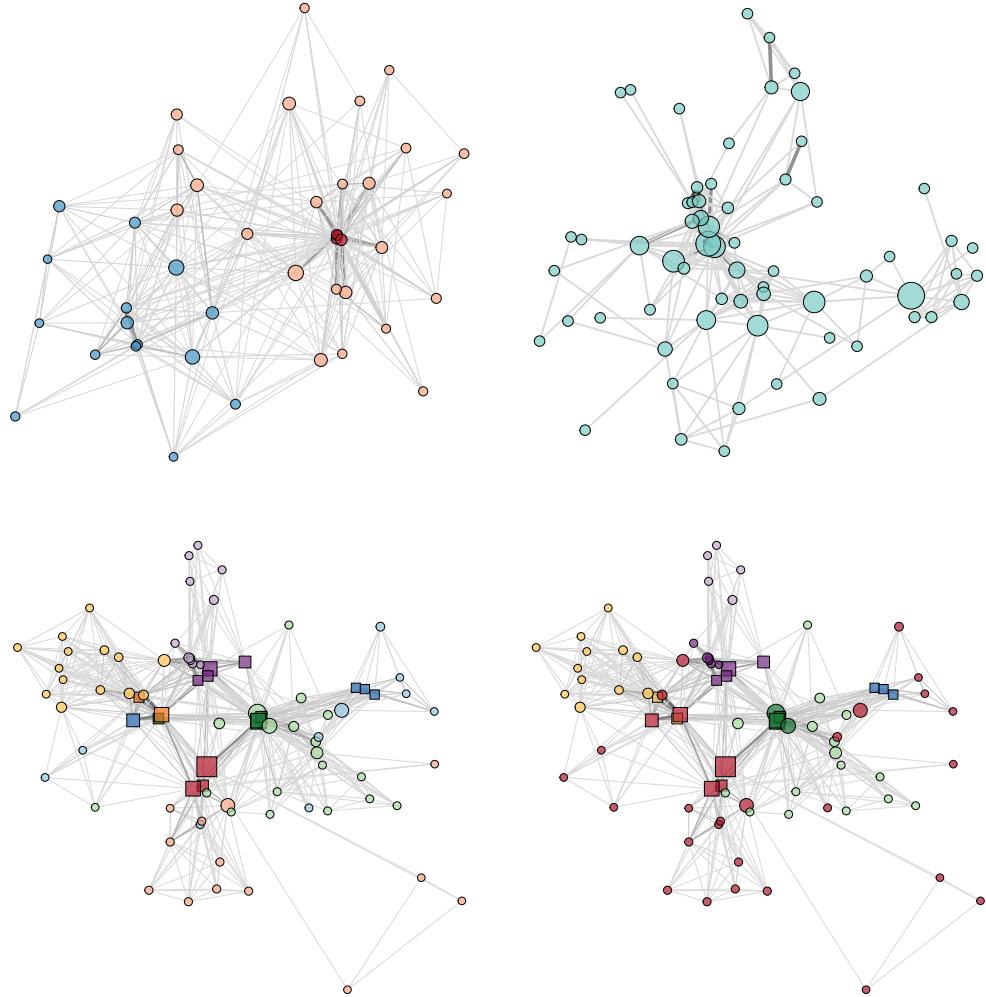


Figure 15: The windsurfers, train bombing and summit co-attendance criminality network in 2-dimensional latent space. The plot settings regarding node sizes, colors, types and edge colors, widths are similar to those applied in the previous subsections. Top-left: inferred 2-d latent positions' plot of the windsurfers network. Different node colors correspond to different inferred groups in  $\hat{z}$ . Top-right: inferred 2-d latent positions' plot of the train bombing network where the inferred clustering has only 1 group. Bottom plots: inferred 2-d latent positions of the summit co-attendance criminality network, where the different node colors in the left plot indicate the reference clustering  $z^*$ , while those in the right plot correspond to different inferred groups in  $\hat{z}$ .

Similarly for the train bombing network, the 2-dimensional  $\hat{\mathcal{U}}$  and the corresponding  $\hat{z}$  are plotted at top-right of Figure 15. The core red group and the special green group are well inferred in the 3-d space shown in Figure 11, but can no longer be well separated and be identified in the 2-d space, bringing a trivial inferred clustering which only has one singleton group. Our experiments show that the illustrated unsatisfactory clustering of the summit co-attendance criminality network and the train bombing network inferred in 2-d space also brings unreasonable performance of inferred probability of unusual

zeros. On the other hand, we point out that the inference performance of the windsurfers network and the Sampson monks network in 2-d space are shown to be comparable with those in 3-d space. Similar inferred latent positions' patterns and inferred clustering are observed in both 2-d and 3-d space, though the inferred probability of unusual zeros is generally higher in 2-d cases due to the more aggregated patterns of the 2-d latent positions. The plot of inferred 2-d latent positions of the Sampson monks network, which is not included in Figure 15, is similar to the plots shown in Figure 8. We refer to <https://github.com/Chaoyi-Lu/ZIP-LPCM> for more details of all the 2-d implementations for the four real networks.

The above findings show that 3-dimensional latent positions are capable of giving a more nuanced model-based representation compared to 2-d latent spaces. The choice of the number of latent space dimensions remains a challenging problem. According to our results, 3-dimensional latent spaces can allow for a better model fit while still maintaining visualizations that are easy to interpret. Thus, our work emphasizes once more the criticality of this model-choice research question within the scope of latent space modelling.

## 6 Conclusion and discussion

This paper describes an original zero-inflated Poisson latent position cluster model which leverages several recent ideas from the literature on computational statistics to analyze non-negative weighted networks with missing data. Our methodology combines zero-inflated Poisson distributions, clustering (via mixture of finite mixtures), and optional nodes attributes to be used within the clustering structure, when available. As regards the inferential procedure, our novel approach relies on a partially collapsed sampler which features a new truncated absorb-eject move, and leads to an automatic and computationally efficient selection of the optimal number of groups. One fundamental output of our proposed procedure is that it provides new model-based visualizations of the complex network data using a 3-dimensional latent space.

The results that we obtain using this methodology include the latent space visualizations, the clustering of the nodes, and the detection of unusual zeros, that is, missing or non-reported data. As we demonstrate via various examples on simulated datasets, the model has great flexibility and can generalize well to a variety of data patterns. In addition, the inferential procedure is scalable and is able to discover the accurate estimates for the model parameters and model structure. Applications on various real networks show that we are able to uncover multiple complex and interesting architectures for the social networks, which provide new perspectives on the analyses of these datasets.

Future work will focus on computational scalability, since we are interested in estimating networks with much larger sizes that are common in real world. As motivated by Legramanti et al. (2022), instead of a mixture of finite mixtures clustering prior, one could compare different forms of Gibbs-type priors (Gnedin and Pitman, 2006) to complement our framework. Another viable modification is to replace Eq. (4), which is one of the simplest forms to link between the  $\mathbb{E}(y_{ij})$  and the corresponding latent positions, with a more sophisticated choice, to better match the data distribution. Similarly, non-Poisson data analysis may also be considered. Finally, following Sewell and Chen (2015),

a natural and ambitious extension for this network model would be to consider a dynamic setting, for the interactions, the clustering, and the missing data specifications.

## Acknowledgements

The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number 12/RC/2289\_P2.

## Code and Data

The application code and data for the simulation studies and the real data applications shown in Sections 4 and 5 is available at <https://github.com/Chaoyi-Lu/ZIP-LPCM>.

## References

- Borg, I. and Groenen, P. J. (2005), *Modern multidimensional scaling: Theory and applications*, Springer Science & Business Media.
- De Nooy, W., Mrvar, A., and Batagelj, V. (2018), *Exploratory social network analysis with Pajek: Revised and expanded edition for updated software*, vol. 46, Cambridge University Press.
- Freeman, L. C., Freeman, S. C., and Michaelson, A. G. (1988), “On human social intelligence”, *Journal of Social and Biological Structures*, 11.4, 415–425.
- Freeman, L. C., Webster, C. M., and Kirke, D. M. (1998), “Exploring social structure using dynamic three-dimensional color images”, *Social Networks*, 20.2, 109–118.
- Geng, J., Bhattacharya, A., and Pati, D. (2019), “Probabilistic community detection with unknown number of communities”, *Journal of the American Statistical Association*, 114.526, 893–905.
- Ghosh, S. K., Mukhopadhyay, P., and Lu, J.-C. J. (2006), “Bayesian analysis of zero-inflated regression models”, *Journal of Statistical Planning and Inference*, 136.4, 1360–1375.
- Gnedin, A., Haulk, C., and Pitman, J. (2009), “Characterizations of exchangeable partitions and random discrete distributions by deletion properties”, *arXiv preprint arXiv:0909.3642*.
- Gnedin, A. and Pitman, J. (2006), “Exchangeable Gibbs partitions and Stirling triangles”, *Journal of Mathematical Sciences*, 138, 5674–5685.
- Gower, J. C. (1966), “Some distance properties of latent root and vector methods used in multivariate analysis”, *Biometrika*, 53.3-4, 325–338.
- Hall, D. B. (2000), “Zero-inflated Poisson and binomial regression with random effects: a case study”, *Biometrics*, 56.4, 1030–1039.
- Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007), “Model-based clustering for social networks”, *Journal of the Royal Statistical Society Series A: Statistics in Society*, 170.2, 301–354.
- Hayes, B. (2006), “Connecting the dots”, *American Scientist*, 94.5, 400–404.
- Hoff, P. D. (2003), “Random effects models for network data”, *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, Citeseer.

- Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002), “Latent space approaches to social network analysis”, *Journal of the American Statistical Association*, 97.460, 1090–1098.
- Kaur, H., Rastelli, R., and Friel, N. (2023), “Latent Position Network Models”, *The Sage Handbook of Social Network Analysis*, SAGE Publications, chap. 36, 526–541, ISBN: 9781529614671.
- Kunegis, J. (2013), “KONECT – The Koblenz Network Collection”, *Proc. Int. Conf. on World Wide Web Companion*, 1343–1350, URL: <http://dl.acm.org/citation.cfm?id=2488173>.
- Lambert, D. (1992), “Zero-inflated Poisson regression, with an application to defects in manufacturing”, *Technometrics*, 34.1, 1–14.
- Legramanti, S., Rigon, T., Durante, D., and Dunson, D. B. (2022), “Extended stochastic block models with application to criminal networks”, *The Annals of Applied Statistics*, 16.4, 2369.
- Lemonete, A. J., Moreno-Arenas, G., and Castellares, F. (2019), “Zero-inflated Bell regression models for count data”, *Journal of Applied Statistics*.
- Lu, C., Durante, D., and Friel, N. (2024), “Zero-inflated stochastic block modeling of efficiency-security tradeoffs in weighted criminal networks”, *arXiv preprint arXiv:2410.23838*.
- Ma, C. (2024), “Statistical latent space models for international classification of diseases (ICD) codes”, PhD thesis.
- Ma, Z., Ma, Z., and Yuan, H. (2020), “Universal latent space model fitting for large networks with edge covariates”, *Journal of Machine Learning Research*, 21.4, 1–67.
- McDaid, A. F., Murphy, T. B., Friel, N., and Hurley, N. J. (2012), “Clustering in networks with the collapsed stochastic block model”, *arXiv preprint arXiv:1203.3083*.
- Meilă, M. (2007), “Comparing clusterings—an information based distance”, *Journal of Multivariate Analysis*, 98.5, 873–895.
- Miller, J. W. and Harrison, M. T. (2018), “Mixture models with a prior on the number of components”, *Journal of the American Statistical Association*, 113.521, 340–356.
- Müller, P., Quintana, F., and Rosner, G. L. (2011), “A product partition model with regression on covariates”, *Journal of Computational and Graphical Statistics*, 20.1, 260–278.
- Nobile, A. (2005), “Bayesian finite mixtures: A note on prior specification and posterior computation (Technical report)”, *University of Glasgow*.
- Nobile, A. and Fearnside, A. T. (2007), “Bayesian finite mixtures with an unknown number of components: The allocation sampler”, *Statistics and Computing*, 17, 147–162.
- Nowicki, K. and Snijders, T. A. B. (2001), “Estimation and prediction for stochastic blockstructures”, *Journal of the American Statistical Association*, 96.455, 1077–1087.
- Park, T. and Van Dyk, D. A. (2009), “Partially collapsed Gibbs samplers: Illustrations and applications”, *Journal of Computational and Graphical Statistics*, 18.2, 283–305.
- Pitman, J. (2006), *Combinatorial stochastic processes: Ecole d'été de probabilités de saint-flour xxxii-2002*, Springer.
- Rastelli, R. and Friel, N. (2018), “Optimal Bayesian estimators for latent variable cluster models”, *Statistics and Computing*, 28, 1169–1186.
- Rastelli, R., Friel, N., and Raftery, A. E. (2016), “Properties of latent variable network models”, *Network Science*, 4.4, 407–432.

- Rastelli, R., Latouche, P., and Friel, N. (2018), “Choosing the number of groups in a latent stochastic blockmodel for dynamic networks”, *Network Science*, 6.4, 469–493.
- Ridout, M., Hinde, J., and Demétrio, C. G. (2001), “A score test for testing a zero-inflated Poisson regression model against zero-inflated negative binomial alternatives”, *Biometrics*, 57.1, 219–223.
- Ryan, C., Wyse, J., and Friel, N. (2017), “Bayesian model selection for the latent position cluster model for social networks”, *Network Science*, 5.1, 70–91.
- Salter-Townshend, M., White, A., Gollini, I., and Murphy, T. (2012), “Review of statistical network analysis: models, algorithms, and software”, *Statistical Analysis and Data Mining*, 5.4, 243–264.
- Salter-Townshend, M. and Murphy, T. B. (2013), “Variational Bayesian inference for the latent position cluster model for network data”, *Computational Statistics & Data Analysis*, 57.1, 661–671.
- Sampson, S. F. (1968), *A novitiate in a period of change: An experimental and case study of social relationships*, Cornell University.
- Sewell, D. K. and Chen, Y. (2015), “Latent space models for dynamic networks”, *Journal of the American Statistical Association*, 110.512, 1646–1657.
- Sewell, D. K. and Chen, Y. (2016), “Latent space models for dynamic networks with weighted edges”, *Social Networks*, 44, 105–116.
- Tanner, M. A. and Wong, W. H. (1987), “The calculation of posterior distributions by data augmentation”, *Journal of the American Statistical Association*, 82.398, 528–540.
- Van Dyk, D. A. and Park, T. (2008), “Partially collapsed Gibbs samplers: Theory and methods”, *Journal of the American Statistical Association*, 103.482, 790–796.
- Wade, S. and Ghahramani, Z. (2018), “Bayesian cluster analysis: Point estimation and credible balls (with discussion)”.
- Wyse, J. and Friel, N. (2012), “Block clustering with collapsed latent block models”, *Statistics and Computing*, 22, 415–428.

## Poster session

Martin Metodiev (Université Clermont Auvergne) *Easily Computed Marginal Likelihoods for Multivariate Mixture Models Using the THAMES Estimator*

Dimitrios Karlis (Athens University of Economics and Business) *Clustering and Random Coefficient Bivariate Integer Valued Autoregressive Time Series*

Sara Geremia (University of Trieste) *Uncovering core-periphery structures in collaboration networks*

Noemi Corsini (University of Cambridge) *Pantheon Data Mixed Modal Clustering*

Maya Guy (Université Côte d'Azur) *From Fragments to Families: Asteroid Clustering*

Seydina Ousmane Niang (Université Côte d'Azur) *Importance weighting variational graph autoencoder for nodes clustering of complex networks*

Pedro Menezes de Araujo (UCD) *A Model-Based Approach to Clustering Temporal and Cross-National Mortality Data*

Gertraud Malsiner-Walli (Vienna University of Economics and Business) *Visually inspecting the posterior of the partitions in Bayesian clustering*

Grigoryan Mariam (Inria MAASAI) *A Model-Based Clustering Approach for Chemical Toxicity Assessment Using Cell Painting Data*

Silvia Dallari (University of Bologna) *Random projection based mixtures of Poisson Log-Normal distributions*

Edoardo Redivo (University of Bologna) *Clustering Criteria for Evaluating and Estimating Entity Resolution Models*

# Easily Computed Marginal Likelihoods for Multivariate Mixture Models Using the THAMES Estimator

A presentation for the  
**Working Group on Model-Based Clustering**

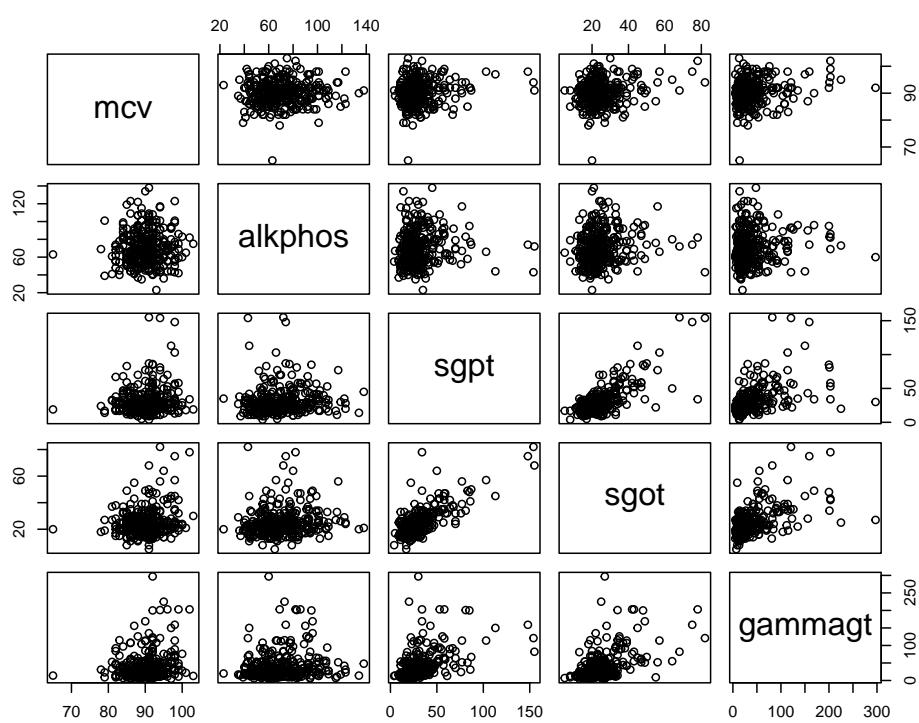
Martin Metodiev  
martin.metodiev@doctorant.uca.fr

Université Clermont Auvergne  
Laboratoire de Mathématiques Blaise Pascal (LMBP)  
Équipe Probabilités, Analyse et Statistique (PAS)

joint work with Nicholas J. Irons, Marie Perrot-Dockès, Pierre  
Latouche and Adrian E. Raftery

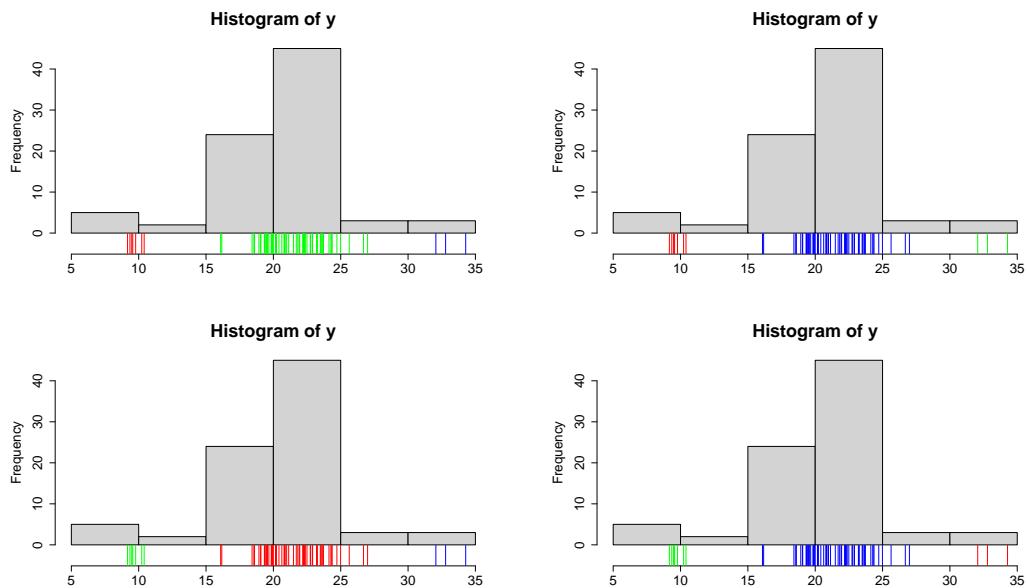
1 / 5

Setting : finding the right number of mixture components



2 / 5

## Problem : Label Switching



3 / 5

## The truncated harmonic mean estimator (THAMES)

- ▶ for Bayesians, the marginal likelihood is an **optimal** model choice measure
- ▶ The THAMES is an estimator of the marginal likelihood that is
  - ▶ precise (**unbiased** and **asymptotically normal** on the inverse likelihood scale),
  - ▶ generic (works on **any** mixture model with tractable likelihood),
  - ▶ simple (**symmetric** and **quickly computed**).

4 / 5

## R-package : **thamesmix** (on CRAN !)

---

thames_mixtures	<i>THAMES estimator of the reciprocal log marginal likelihood for mixture models</i>
-----------------	--

---

### Description

This function computes the THAMES estimate of the reciprocal log marginal likelihood for mixture models using posterior samples and unnormalized log posterior values.

### Usage

```
thames_mixtures(  
  logpost,  
  sims,
```

You only need 2 arguments to use the package thamesmix :

- ▶ an MCMC sample from the posterior
- ▶ a function that evaluates the unnormalized log-poster density  
(the logarithm of the prior times the likelihood)

# COMMUNITY-LEVEL CORE-PERIPHERY STRUCTURES IN RESEARCH NETWORKS

Sara Geremia<sup>1</sup>, Domenico De Stefano<sup>1</sup>, Michael Fop<sup>2</sup>

<sup>1</sup>University of Trieste, Italy

<sup>2</sup>University College Dublin, Ireland

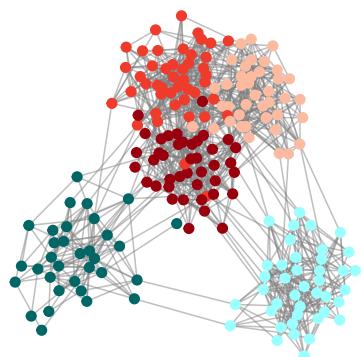
WGMBC - 21-25th July 2025



1

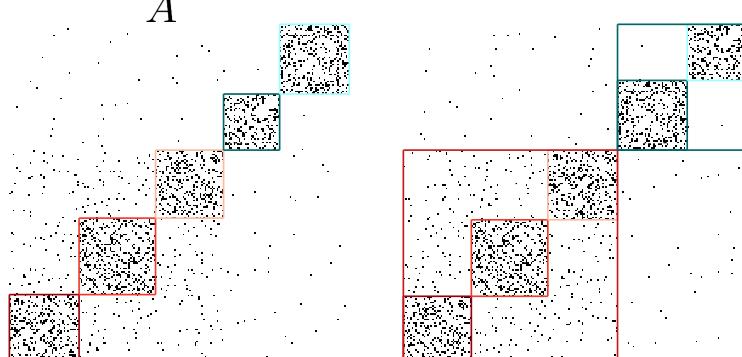
## COMMUNITY-LEVEL CORE-PERIPHERY STRUCTURE ADJACENCY MATRIX REPRESENTATION

Community  
Partition



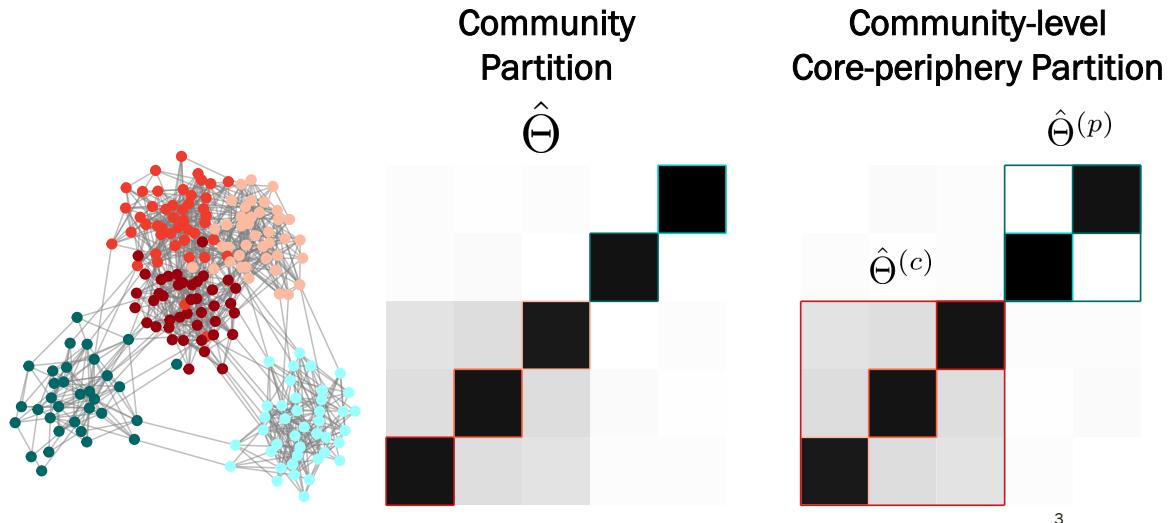
Community-level  
Core-periphery Partition

A



2

## COMMUNITY-LEVEL CORE-PERIPHERY STRUCTURE CONNECTIVITY MATRIX REPRESENTATION



## COMMUNITY-LEVEL CORE-PERIPHERY DETECTION A NOVEL SBM-BASED APPROACH



Define an objective function  $\phi$  that:



**Maximizes** core communities' inter-connectivity  $\hat{\Theta}^{(c)}$

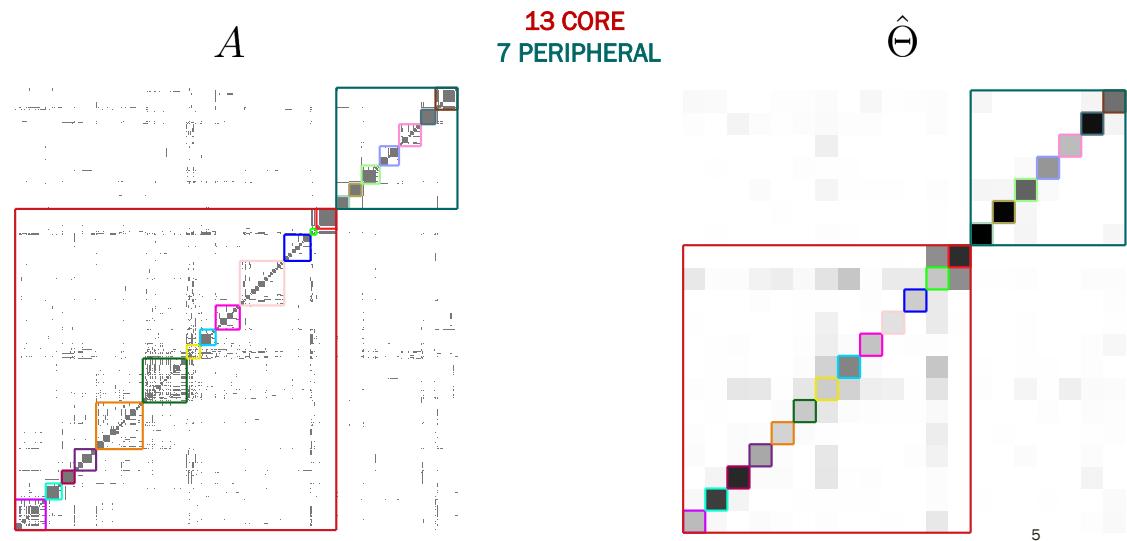


**Minimizes** peripheral communities' inter-connectivity  $\hat{\Theta}^{(p)}$

$$\phi = (c_1 + c_2 - c_3) - (p_1 + p_2 + p_3)$$

- The optimization of  $\phi$  is a combinatorial problem
- This problem is solved using a genetic algorithm

# COMMUNITY-LEVEL CORE-PERIPHERY STRUCTURE IN A RESEARCH NETWORK



# THANK YOU!

sara.geremia@phd.units.it



## Pantheon data mixed modal clustering

(joint work with Professor Giovanna Menardi)

Noemi Corsini

31<sup>st</sup> Working Group on Model-Based Clustering - Antibes, France

1 / 4

## Motivating application



The **Pantheon dataset** offers a rich lens on how historical and cultural relevance is distributed across human societies

- **5,500 years** of recorded human history
- **11,341 biographies** of globally recognized individuals featured in over 25 language editions on Wikipedia

**Goal** → Find cluster using mixed-type variables with a **strong association structure**

2 / 4

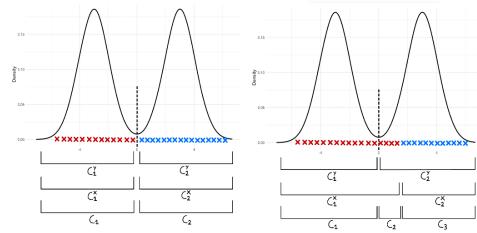


## Method

Traditional methods often require **encoding** categorical variables or **discretising** continuous ones. **Few approaches handle mixed data directly**

Leveraging the extension of modal clustering to the categorical setting, we propose a **working idea to extend it to mixed-type data**

- Handle continuous and categorical variables
- Automatically find the number of clusters



3 / 4



## Next steps

Integrating them is non-trivial since the density measures are

- **Formally** defined for continuous variables
- Only **conceptually** defined for categorical variables
- Tackle the problem separately and **then combine the results**

Move beyond the current approach towards a statistically sound method that **unify the two frameworks**

4 / 4



UNIVERSITY OF  
CAMBRIDGE

See you at the poster session!

Noemi Corsini

[nc670@cam.ac.uk](mailto:nc670@cam.ac.uk)

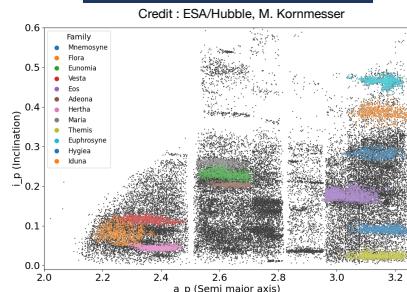
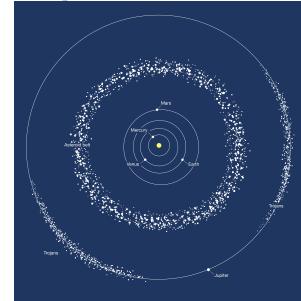
# From Fragments to Families: Asteroid Clustering

Maya GUY([maya.guy@msomg.fr](mailto:maya.guy@msomg.fr))<sup>1,2,3, 4</sup>, Vincent VANDEWALLE<sup>1,2,4</sup>, Benoit CARRY<sup>1,3</sup>

<sup>1</sup>Université Côte d'Azur, <sup>2</sup>Inria, <sup>3</sup>Lagrange, Observatoire de la Côte d'Azur, <sup>4</sup>Laboratoire J.A. Dieudonné

## What is a family and why study them?

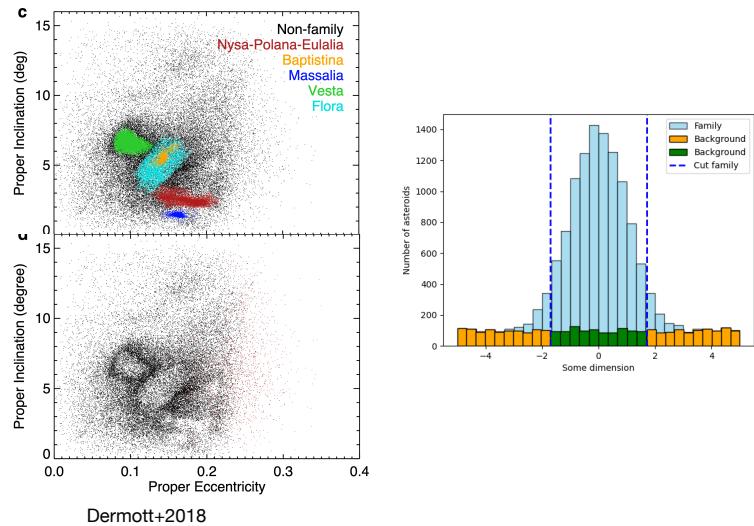
- ◆ Collision
- ◆ Thermal effect
- Evolutionnary process
  - All members share the same age
  - Each family have a different age
- Original size distribution



# Why a new method ?

Historical method :

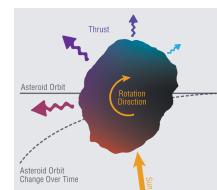
- Hierarchical clustering
- V-shape



# Physically inspired model

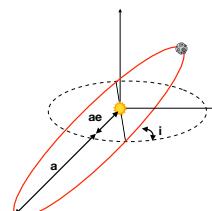
Yarkovsky effect

$$\frac{da}{dt} = \left( \frac{da}{dt} \right)_{\text{annuel}} + \left( \frac{da}{dt} \right)_{\text{diurne}} = \frac{(1-A)}{9na^2} \frac{6}{Dpc} \frac{S_{\odot}}{\Delta^2} (W_n \sin^2 \gamma - 2W_{\omega} \cos \gamma)$$



$$p(a, e, i, H) = \pi_{\text{background}} p(a, e, i, H | \text{background}) + \sum_{k=1}^K \underbrace{\pi_k p(a, e, i, H | k)}$$

Skew-t x Normal



⇒ Parameters estimated with EM algorithm

# Importance weighted directed graph variational auto-encoder for block modelling of complex networks

Seydina Ousmane NIANG

Phd student in applied mathematics  
Maasai research team  
INRIA Centre Université Côte d'Azur

✉ seydina-ousmane.niang@inria.fr

Supervision: Charles Bouveyron, P. Latouche,  
& M. Cornelie

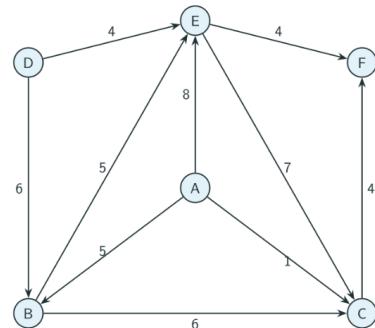


## Context

- Data: adjacency matrix  $A$ ,  $A_{ij}$  is the number of connexions from node  $i$  to  $j$ .

### Goal:

- Regroup entities into  $Q$  clusters
- Efficiently visualize the network structure



### Maximum-likelihood problem:

- Assume a family of generative models  $\{p_\theta\}_\Theta$  and fit them to the training data
- Pick the model  $p_{\hat{\theta}}$  with the highest likelihood

## Deep-ZLPBM

Data: adjacency matrix  $A$  such that  $A_{ii} = 0$  for all  $i$ .

We assume that:

- There are  $Q$  clusters
- Each entity  $i$  has a latent representation  $Z_i \sim \mathcal{N}(0, \gamma I_{Q-1})$  where  $\gamma$  is an hyper-parameter
- Each entity  $i$  has probability  $\eta_{iq}$  to be in cluster  $q$
- $\eta_i$  is a bijective transformation of  $Z_i$
- Connection probability between nodes only depends on the cluster belongings
- Given  $Z$ ,  $i$  and  $j$  have probability to be linked of  $\eta_i^\top \Pi \eta_j$
- if nodes  $i$  and  $j$  are linked, their connection is drawn from a Poisson  $\mathcal{P}(\eta_i^\top \Lambda \eta_j)$

$$\begin{cases} \mathbb{P}(A_{ij} = 0 | Z_i, Z_j) = 1 - \eta_i^\top \Pi \eta_j (1 - \exp(-\eta_i^\top \Lambda \eta_j)) \\ \mathbb{P}(A_{ij} = k | Z_i, Z_j) = \eta_i^\top \Pi \eta_j \frac{(\eta_i^\top \Lambda \eta_j)^k}{k!} \exp(-\eta_i^\top \Lambda \eta_j) \quad \forall k \geq 1 \end{cases}$$

## IW-ELBO (Importance-Weighted ELBO)

$$\log p(A | \Pi, \Lambda) = \log \mathbb{E}_{Z^{(1)}, \dots, Z^{(K)} \sim q} \left[ \frac{1}{K} \sum_{\ell=1}^K \frac{p(A, Z^{(\ell)} | \Pi, \Lambda)}{q(Z^{(\ell)} | A)} \right]$$

- Intractable
- Applying Jensen's inequality:

$$\log p(A | \Pi, \Lambda) \geq \mathbb{E}_{Z^{(1)}, \dots, Z^{(K)} \sim q} \left[ \log \left( \frac{1}{K} \sum_{\ell=1}^K \frac{p(A, Z^{(\ell)} | \Pi, \Lambda)}{q(Z^{(\ell)} | A)} \right) \right] := \mathcal{L}_K(\Pi, \Lambda, q)$$

- $\mathcal{L}_K$  is known as the **Importance-Weighted ELBO (IW-ELBO)**
- When  $K = 1$ ,  $\mathcal{L}_1$  is the usual evidence lower bound (ELBO)

**Theoretical results:**

- For all  $K \geq 1$ :

$$\mathcal{L}_1 \leq \mathcal{L}_K \leq \mathcal{L}_{K+1} \leq \log p(A | \Pi, \Lambda)$$

- As  $K \rightarrow \infty$ :

$$\mathcal{L}_K \rightarrow \log p(A | \Pi, \Lambda) \quad \text{if} \quad \frac{p(A, Z)}{q(Z | A)} \quad \text{is bounded}$$

## Results

$\pi = 0.1, \alpha = 4, \omega = 2$			
Method	Communities	Disassortative	Hub
Node2vec	$0.06 \pm 0.08$	$0.002 \pm 0.003$	$0.08 \pm 0.05$
Deepwalk	$0 \pm 0.00$	$0.0 \pm 0.00$	$0.0 \pm 0.00$
PSBM	$0.21 \pm 0.26$	$0.12 \pm 0.29$	$0.13 \pm 0.28$
Deep-ZLPBM ELBO	$0.24 \pm 0.15$	$0.27 \pm 0.13$	$0.15 \pm 0.09$
Deep-ZLPBM iw-ELBO	<b><math>0.89 \pm 0.13</math></b>	<b><math>0.91 \pm 0.03</math></b>	<b><math>0.62 \pm 0.16</math></b>

$\pi = 0.15, \alpha = 4, \omega = 2$			
Method	Communities	Disassortative	Hub
Node2vec	$0.26 \pm 0.17$	$0.009 \pm 0.007$	$0.01 \pm 0.009$
Deepwalk	$0 \pm 0.00$	$0.0 \pm 0.00$	$0.0 \pm 0.00$
PSBM	$0.38 \pm 0.2$	$0.21 \pm 0.24$	$0.21 \pm 0.26$
Deep-ZLPBM ELBO	$0.81 \pm 0.11$	$0.41 \pm 0.11$	$0.39 \pm 0.16$
Deep-ZLPBM iw-ELBO	<b><math>0.98 \pm 0.01</math></b>	<b><math>1 \pm 0.00</math></b>	<b><math>0.75 \pm 0.06</math></b>

$\pi = 0.3, \alpha = 4, \omega = 2$			
Method	Communities	Disassortative	Hub
Node2vec	$0.89 \pm 0.15$	$0.04 \pm 0.03$	$0.33 \pm 0.11$
Deepwalk	$0 \pm 0.00$	$0.0 \pm 0.00$	$0.0 \pm 0.00$
PSBM	$0.51 \pm 0.16$	$0.46 \pm 0.18$	$0.41 \pm 0.19$
Deep-ZLPBM ELBO	<b><math>1 \pm 0.00</math></b>	<b><math>1 \pm 0.00</math></b>	$0.73 \pm 0.14$
Deep-ZLPBM iw-ELBO	<b><math>1 \pm 0.00</math></b>	<b><math>1 \pm 0.00</math></b>	<b><math>0.91 \pm 0.07</math></b>

Table 1: Benchmark with competitors

## Conclusion and perspectives

So far we can state:

- Deep-ZLPBM efficiently performs partial node clustering and representation learning in valued networks
- Optimizing the importance-weighted ELBO (iw-ELBO) provides significant performance gains compared to the standard ELBO optimization



# CLUSTERING MORTALITY STAGES ACROSS COUNTRIES WITH MIXTURES OF CATEGORICAL DISTRIBUTIONS

---

Student: Pedro Menezes de Araújo

Supervisors: Prof. Claire Gormley, Prof. Brendan Murphy

University College Dublin

HOST INSTITUTIONS

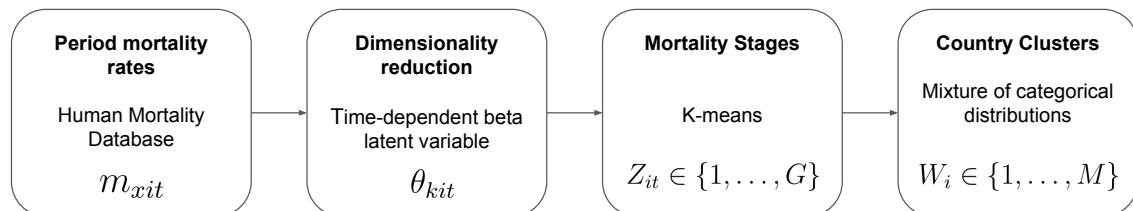


## INTRODUCTION

- Clustering mortality data is useful for tasks such as **forecasting** or identifying **inequality or divergence** in mortality trends.
- Most studies that cluster countries overlook the dynamics of mortality evolution, offering **little insight into the stages each country has undergone**.
- We proposed a two-step approach where we first **find mortality stages** and then **cluster the countries based on their stage dynamics**.

## PROPOSED METHODOLOGY

- For country  $i = 1, \dots, 30$ , time  $t = 1960, \dots, 2010$ , and age group  $x \in \{0, 5, 10, \dots, 110+\}$ .



2

## MIXTURE OF CATEGORICAL DISTRIBUTIONS

- Model for the observed mortality stages:

$$p(Z_{it} = g | W_i = m) = \frac{\exp\{S_{mg}(t)\}}{\sum_{g=1}^G \exp\{S_{mg}(t)\}}, \quad S_{mg}(t) = 0.$$

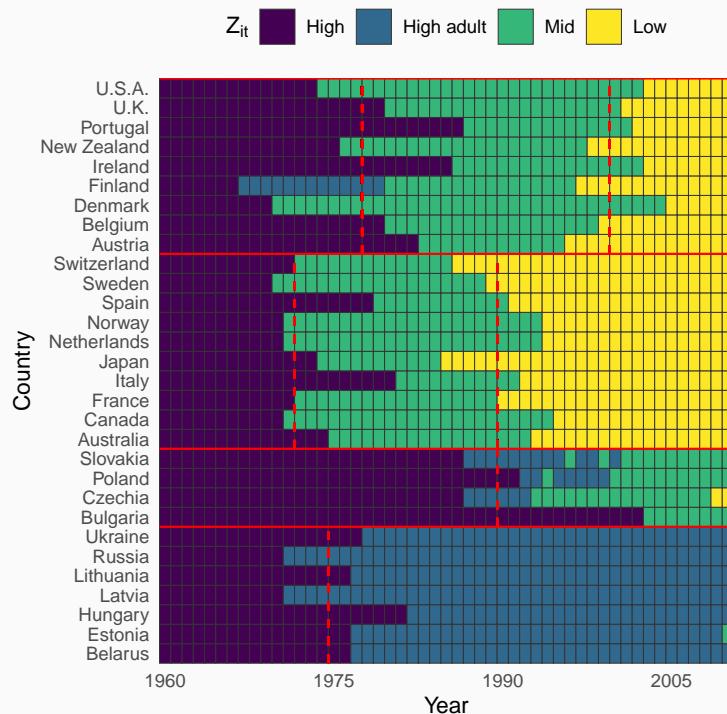
- We use a 'natural' proper representation for the penalised splines:

$$S_{mg}(t) = \beta_{mg} + \mathbf{v}_t^\top \mathbf{u}_{mg}, \quad \mathbf{u}_{mg} \sim N(0, \lambda \mathbf{\Lambda}_+).$$

- $\mathbf{v}_t$  and  $\mathbf{\Lambda}_+$  are constructed based on the QR decomposition of the basis functions.
- We use Bayesian inference (Gibbs sampling).

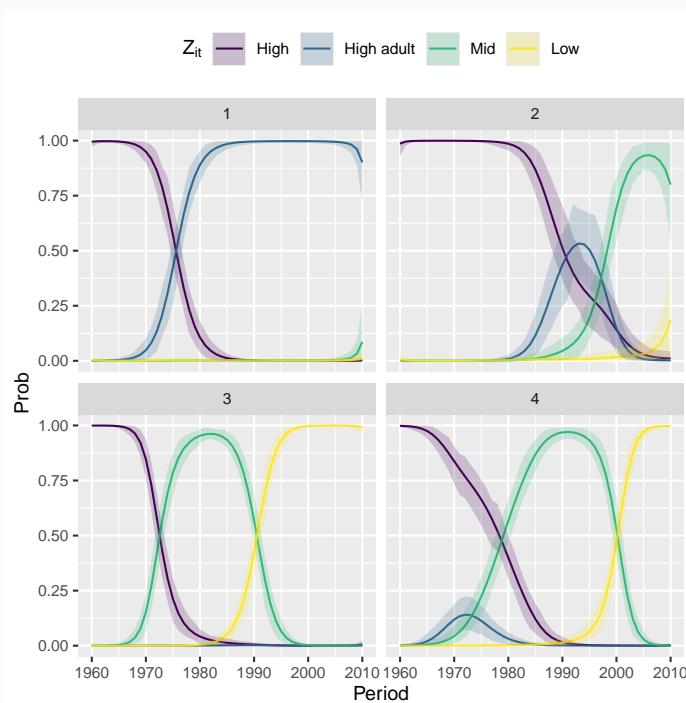
3

## STAGE ( $Z_{it}$ ) AND COUNTRY ( $W_i$ ) CLUSTERS



4

## COUNTRY CLUSTERS STAGE PROBABILITIES



5

## CONCLUSION AND NEXT STEPS

- We found four stages and four country clusters based on their stage dynamics.
- We plan to estimate  $Z_{it}$  and  $W_i$  jointly instead of a two-step approach.
- Investigate further model selection (choice of  $\lambda$ , number of basis functions,  $M$ ,  $G$ , etc).

Poster Flash Session

## Visually inspecting the posterior partition space



Gertraud Malsiner-Walli

joint work with Thomas Rusch and Bettina Grün

Vienna University of Economics and Business

Working Group on Model-Based Clustering Nice, July 21-25, 2025

## Framework

- In the **Bayesian finite mixture model**:  
the unknown component indicators  $\mathbf{S}_i \in \{\mathbf{1}, \dots, \mathbf{K}\}$  are included as latent variables.
- $(S_1, \dots, S_N)$  induce a **partition**  $\mathcal{C}$  of the  $N$  data points:

$$\mathcal{C} = \{C_1, \dots, C_{K_+}\}, \text{ where } C_k = \{i : S_i = k\}.$$

- When using MCMC sampling, in each iteration of the sampler,  $S_1, \dots, S_N$  induce a partition  $\mathcal{C}$  of the data.  $\Rightarrow$  produces 10,000s of clusterings!
- How to summarize the **posterior of the partitions**?

## Problem

---

- **Investigating the posterior partition distribution** is complicated by the discrete, unordered nature and massive dimension of the partition space (e.g.  $B_{20} = 51724158235372$ )
  - ⇒ Obtaining a point estimate for clustering? (GreddyEPL, *salso*)
  - ⇒ Describing uncertainty: credible balls? (?)
  - ⇒ Check multimodality? Wade and Balocchi (2025+)
- **Aim of the analysis:** Can we learn something about the posterior distribution of the partitions?
  - ⇒ Different final clustering solutions: Are they in the ‘center’ of the posterior?
  - ⇒ Are there several modes?
  - ⇒ Can we characterize HD-regions around each mode?

Slide: 3

## Proposal

---

- We are applying **multidimensional scaling (MDS)** (Kruskal, 1964, Borg and Groenen, 2007):
  - ⇒ to represent each partition as a point in a 2-dimension space
  - ⇒ make the partition distribution visible.
- Similarity measure: adjusted Rand index (ARI), Distance measure: 1–ARI.
- We use the R packages *smacof*, *smacofx*.

Slide: 4

# Application

---

- **Galaxy data** ( $N = 82$ ): How many clusters are there?
  - We specify the priors of a prominent run (Richardson & Green, 1997).
  - We apply MDS to the sampled partitions and identified a **second mode** not detected by the different final clustering solutions.
- ⇒ **MDS can be useful:**
- to investigate features of the posterior partition distribution (HDR, multimodality, location of the final clustering solutions),
  - to check the appropriateness of prior definitions (e.g. mismatch between mode and HDR).

# A Model-Based Clustering Approach for Toxicity Assessment Using Cell Painting Data

Mariam Grigoryan<sup>1</sup>, Vincent Vandewalle<sup>1</sup>, David Rouquieré<sup>2</sup>

1. University Côte d'Azur, Inria (MAASAI), CNRS  
2. Bayer Crop Science, Toxicology Data Science Department

Working Group on Model-Based Clustering  
July 22th, 2025

## Context and Motivation

- **Cell Painting** is a high-content imaging assay that stains cellular compartments (nucleus, cytoplasm, etc.) across 5 fluorescent channels.
- Each cell is profiled with thousands of morphological features.
- Toxicity is often gradual and dose-dependent, not always defined by clear thresholds.
- Traditional supervised models require predefined toxicity labels, which may be incomplete.
- We propose an unsupervised model to detect subtle phenotypic shifts and identify the concentration where responses begin to diverge from baseline.

## Dataset Overview

- **Source:** Axiom dataset (OASIS consortium)
- **Cell line:** HepaRG (human liver cells)
- **Experimental design:**
  - DMSO wells as negative controls
  - Compounds tested at 8 different concentrations and 2 replicates.
- **Features:** ~4500 morphological profiles per cell.

## Proposed Model

- We use a generative clustering approach to model morphological heterogeneity.
- Cell populations are represented by clusters inferred via Gaussian mixture model.
- We estimate the proportion of each cluster per well and track shifts across concentrations.



Figure: Pipeline for Model Application in Cell Painting Data

## Statistical Evaluation

- Test if cluster proportions differ between treated and DMSO wells:

$$H_0^{(s,c)} : \mathcal{L}(\hat{\pi}_c^s) = \mathcal{L}(\hat{\pi}_0) \quad vs. \quad H_1^{(s,c)} : \mathcal{L}(\hat{\pi}_c^s) \neq \mathcal{L}(\hat{\pi}_0)$$

- Permutation testing is applied to assess the significance of observed shifts.
- Adaptive SEQstep procedure is then used to correct for multiple comparisons.

## Case Study: Histamine Receptor Compounds

- We show results for two compounds targeting the same pathway but with distinct DILI classifications:
  - **Ebrotidine** (DILI most concern)
  - **Loratadine** (DILI no concern)
- Ebrotidine induces early and significant shifts in cluster proportions at increasing concentrations, consistent with its DILI risk classification.
- Loratadine shows stable morphological profiles across doses.

## Conclusion and Future Work

- Our unsupervised model identifies early morphological responses to hepatotoxic compounds.
- Future work: scale to large compound libraries and assess the distinction between desired pharmacological responses and adverse effects.

# Random Projection based Mixtures of Poisson Log-Normal Distributions

Silvia Dallari

joint with Laura Anderlucci and Angela Montanari

University of Bologna

32nd Summer Working Group on Model-Based Clustering,  
Nice - July 22<sup>nd</sup> 2025

## Framework

To model the dependence among count data with potential group structures, multivariate Poisson-Lognormal (MPLN) mixtures have been proposed (Silva *et al.*, 2019, Subedi & Browne, 2020, and Chiquet *et al.*, 2021).

For each sample:

### Latent layers:

$$\mathbf{z}|\mathbf{s} = i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

$$\mathbf{s} \sim \mathcal{M}(1, \boldsymbol{\pi} = (\pi_1, \dots, \pi_k))$$

### Observation layer:

$$y_j|z_j \sim \mathcal{P}(\exp(c_j + z_j))$$

$$(j = 1, \dots, p)$$

- $\pi_i > 0$  and  $\sum_{i=1}^k \pi_i = 1$ ;
- $\mathbf{s}$ : allocation variable;

- $\boldsymbol{\mu}_i$  and  $\boldsymbol{\Sigma}_i$ : parameters of the  $i$ -th  $p$ -variate normal density;
- $c_j$ : offset.

## Proposal

### Problem

When data are high-dimensional, the estimate of  $\Sigma_i$  in the latent space may become **singular**.

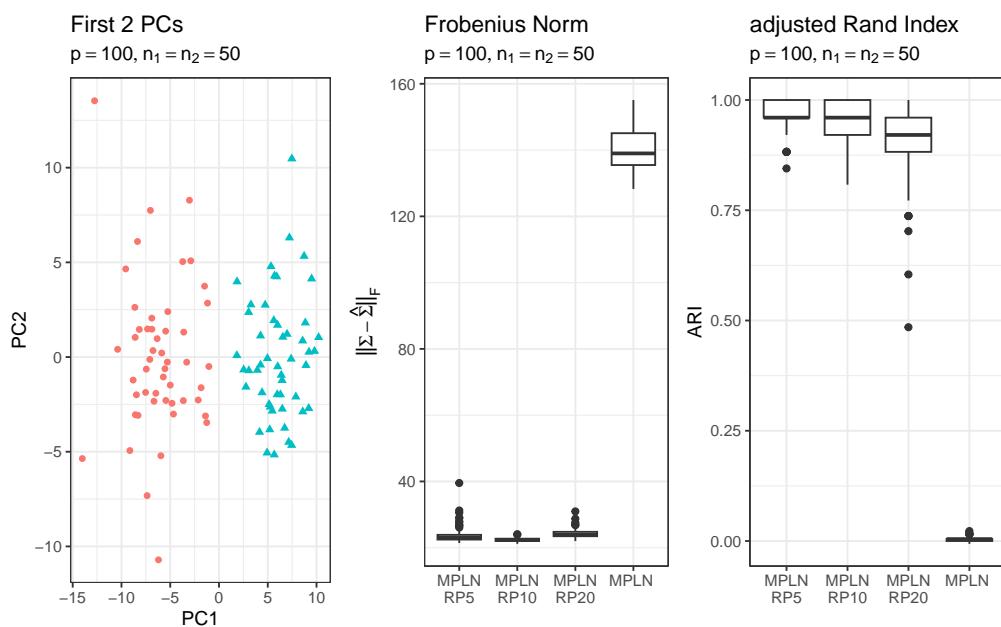


### Solution

Building on the solution of Marzetta *et al.* (2011) in the Gaussian framework, we propose to modify the M-step estimate of  $\Sigma_i$  in the variational EM algorithm by using a **Random Projection** based ensemble estimate.

## Simulation Study

Data have been generated using the PLNmodels R package of Chiquet *et al.* (2021), with 100 repetitions for each scenario.



## Main References

- Chiquet, J., Mariadassou, M., Robin, S. (2021). The Poisson-Lognormal Model as a Versatile Framework for the Joint Analysis of Species Abundances. *Frontiers in Ecology and Evolution*, 9.
- Dallari, S., Anderlucci, L., Montanari, A. (2025). High-Dimensional Gaussian Mixtures with Random Projection Based Covariance Estimates. *Italian Statistical Society Series on Advances in Statistics*. Springer, Cham.
- Marzetta, T.L., Tucci, G.H., Simon, S.H. (2011). A Random Matrix-Theoretic Approach to Handling Singular Covariance Estimates. *IEEE Transactions on Information Theory*, 57(9), pp.6256-6271.
- Silva, A., Rothstein, S.J., McNicholas, P.D., Subedi, S. (2019). A multivariate Poisson-log normal mixture model for clustering transcriptome sequencing data. *BMC Bioinformatics*, 20 (394).
- Subedi, S. & Browne, R.P. (2020). A family of parsimonious mixtures of multivariate Poisson-lognormal distributions for clustering multivariate count data. *Stat*, 9(1), e310.

# Clustering Criteria for Evaluating and Estimating Entity Resolution Models

Edoardo Redivo

*University of Bologna*

July 22, 2025

Working Group on Model-Based Clustering 2025

1 / 5

## Entity resolution: two perspectives

Single dataset  $\mathbf{X} \in \mathbb{R}^{n \times p}$ , where multiple observations might come from the same individual, lacking a **unique identifier**.

## **Clustering task**

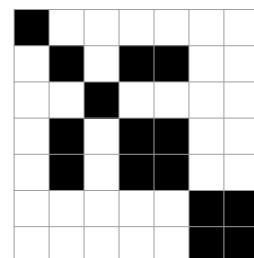
- Cluster observations from the same individual.
  - Linkage structure: *partition* of the integers  $\{1, \dots, n\}$ .

X

## *Binary classification task*

- Classify pairs of observations as being coreferent or not.
  - Linkage structure: coreference matrix  $\Delta \in \mathbb{R}^{n \times n}$ .

A



## Evaluating entity resolution models

### Clustering criteria

### Binary classification metrics

- The two evaluation perspectives are connected when comparing the true ( $\Delta$ ) and estimated ( $\hat{\Delta}$ ) coreference matrix.

$$\text{Rand Index (RI)} = \text{Accuracy} = \frac{\text{TP} + \text{TN}}{\binom{n}{2}},$$

where  $\text{TP} = \sum_{i < j} \Delta_{ij} \hat{\Delta}_{ij}$  and  $\text{TN} = \sum_{i < j} (1 - \Delta_{ij})(1 - \hat{\Delta}_{ij})$ .

- Similar criteria to those used for evaluation, expressed as loss functions  $L$ , are used for the point estimation of partitions in Bayesian inference.

$$\hat{\Delta}^* \approx \underset{\hat{\Delta}}{\operatorname{argmin}} \frac{1}{R} \sum_{r=1}^R L(\Delta^{(r)}, \hat{\Delta}).$$

- Binder's loss is a weighted sum of the errors in the binary classification:

$$L_{\text{Binder}}(\Delta, \hat{\Delta}) = a \text{FN} + b \text{FP},$$

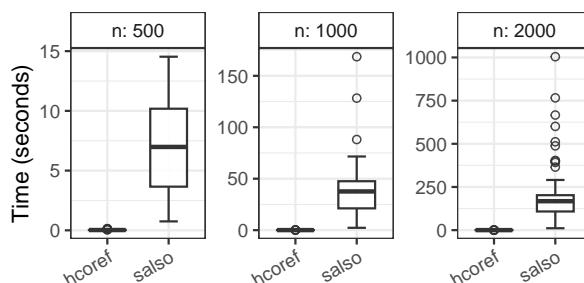
when  $a = b = 1$ , it is directly related to the Rand Index:

$$L_{\text{Binder}}(\Delta, \hat{\Delta}) = \binom{n}{2} (1 - \text{RI}).$$

3 / 5

## Faster point estimation for ER partitions

- State-of-the-art solutions for  $\hat{\Delta}^*$  are greedy search algorithms such as `salso` and `GreedyEPL`.
- A new algorithm is proposed, `hcoref`, that applies a search only within subsets of the observations, identified via single linkage hierarchical clustering.
- In ER tasks, both  $n$  and  $K$  are large, and no maximum number of clusters can be set. In these settings `hcoref` greatly reduces runtime, without loss in accuracy.



4 / 5

## CEM algorithm for the Fellegi-Sunter model

- The Fellegi-Sunter model is a latent class analysis applied to comparison data:

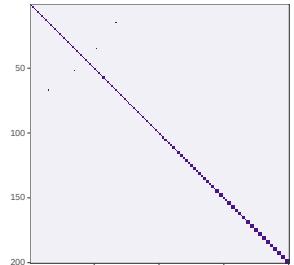
$$\gamma_{ijh} = \begin{cases} 1 & x_{ih} = x_{jh} \\ 0 & \text{otherwise} \end{cases}$$

$$\gamma_{ijh} | \Delta_{ij} = 1 \stackrel{\text{iid}}{\sim} \text{Bernoulli}(m_h)$$

$$\gamma_{ijh} | \Delta_{ij} = 0 \stackrel{\text{iid}}{\sim} \text{Bernoulli}(u_h)$$

$$\Delta_{ij} \stackrel{\text{iid}}{\sim} \text{Bernoulli}(\lambda)$$

- A CEM algorithm is proposed, incorporating constraints on  $\Delta$ .
- The E-and-C step can be solved with a greedy algorithm based on hierarchical clustering, similarly to hcoref.



Estimated coreference matrix  $\hat{\Delta}$  from simulated data.

- Settings

$n = 200, p = 7,$   
 $K = 140.$

- Metrics

$\text{FP} = 3, \text{FN} = 11,$   
 $\text{ARI} = 91\%.$

## Software session

Pierre-Alexandre Mattei (Inria) *Learning mixtures with SGD instead of EM?*

Dan Sewell (University of Iowa) *JANE Community detection via the latent space network model*

Francesco Amato (Université Lumière Lyon) *clustMMM: a software to cluster longitudinal mixed data*