Statistical modeling and analysis of radio-induced adverse effects based on in vitro and in vivo data

Polina Arsenteva

THÈSE DE DOCTORAT DE L'ÉTABLISSEMENT
UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ
PRÉPARÉE À L'INSTITUT DE MATHÉMATIQUES DE BOURGOGNE

École doctorale n° 553: Carnot-Pasteur

Doctorat en mathématiques appliquées

STATISTICAL MODELING AND ANALYSIS OF RADIO-INDUCED ADVERSE EFFECTS BASED ON IN VITRO AND IN VIVO DATA

Présentée par

Polina ARSENTEVA

devant le jury composé de

Mohamed Amine BENADJAOUD	IRSN	**Co-directeur**
Anne-Laure BOULESTEIX	LMU Munich	**Examinatrice**
Hervé CARDOT	UBFC	**Directeur**
Alvaro KÖHN-LUQUE	University of Oslo	**Examinateur**
Catherine MATIAS	CNRS, Sorbonne Université	**Présidente**
Igor MELNYKOV	University of Minnesota	**Rapporteur**
Stefan MICHIELS	INSERM, Gustave Roussy	**Rapporteur**
Patrick TARDIVEL	UBFC	**Examinateur**

ACKNOWLEDGMENTS

I would like to start by thanking my family for their infinite support, above all my parents, without whom none of my accomplishments would have been possible. I also thank my friends and colleagues for sharing joys and sorrows of a Ph.D., and for having made these three years a true pleasure.

I wish to sincerely thank Igor Melnykov and Stefan Michiels for having accepted being reporters of my thesis, and for their most valuable feedback. I express my appreciation to Anne-Laure Boulesteix, Alvaro Köhn-Luque, Catherine Matias and Patrick Tardivel for being part of the jury.

I would also like to thank all the collaborators from the IRSN, in particular the biologists Olivier Guipaud, Fabien Milliat and Vincent Paget, for our extremely fruitful joint effort.

Finally, I would like to express my utmost gratitude to my advisors Mohamed Amine Benadjaoud and Hervé Cardot for having put their faith in me, for everything they taught me, for their unfailing support and for all the time and effort that they have invested in me. I feel incredibly lucky for having them as mentors at this early stage of my career.

Abstract

In this work we address the problem of adverse effects induced by radiotherapy on healthy tissues. The goal is to propose a mathematical framework to compare the effects of different irradiation modalities, to be able to ultimately choose those treatments that produce the minimal amounts of adverse effects for potential use in the clinical setting. The adverse effects are studied in the context of two types of data: in terms of the in vitro omic response of human endothelial cells, and in terms of the adverse effects observed on mice in the framework of in vivo experiments. In the in vitro setting, we encounter the problem of extracting key information from complex temporal data that cannot be treated with the methods available in literature. We model the fold changes, the object that encodes the difference in the effect of two experimental conditions, in the way that allows to take into account the uncertainties of measurements as well as the correlations between the observed entities. We construct a distance, with a further generalization to a dissimilarity measure, allowing to compare the fold changes in terms of all the important statistical properties. Finally, we propose a computationally efficient algorithm performing clustering jointly with temporal alignment of the fold changes. The key features extracted through the latter are visualized using two types of network representations, for the purpose of facilitating biological interpretation. In the in vivo setting, the statistical challenge is to establish a predictive link between variables that, due to the specificities of the experimental design, can never be observed on the same animals. In the context of not having access to joint distributions, we leverage the additional information on the observed groups to infer the linear regression model. We propose two estimators of the regression parameters, one based on the method of moments and the other based on optimal transport, as well as the estimators for the confidence intervals based on the stratified bootstrap procedure.

Key words: Radiotherapy, Complex temporal data, Joint clustering with alignment, Omic network inference, Data fusion, Wasserstein distance.

RÉSUMÉ

Dans ce travail nous abordons le problème des effets indésirables induits par la radiothérapie sur les tissus sains. L'objectif est de proposer un cadre mathématique pour comparer les effets de différentes modalités d'irradiation, afin de pouvoir éventuellement choisir les traitements qui produisent le moins d'effets indésirables pour l'utilisation potentielle en clinique. Les effets secondaires sont étudiés dans le cadre de deux types de données : en termes de réponse omique in vitro des cellules endothéliales humaines, et en termes d'effets indésirables observés sur des souris dans le cadre d'expérimentations in vivo. Dans le cadre in vitro, nous rencontrons le problème de l'extraction d'informations clés à partir de données temporelles complexes qui ne peuvent pas être traitées avec les méthodes disponibles dans la littérature. Nous modélisons le fold change radio-induit, l'objet qui code la différence d'effet de deux conditions expérimentales, d'une manière qui permet de prendre en compte les incertitudes des mesures ainsi que les corrélations entre les entités observées. Nous construisons une distance, avec une généralisation ultérieure à une mesure de dissimilarité, permettant de comparer les fold changes en termes de toutes leurs propriétés statistiques importantes. Enfin, nous proposons un algorithme computationnellement efficace effectuant le clustering joint avec l'alignement temporel des fold changes. Les caractéristiques clés extraites de ces dernières sont visualisées à l'aide de deux types de représentations de réseau, dans le but de faciliter l'interprétation biologique. Dans le cadre in vivo, l'enjeu statistique est d'établir un lien prédictif entre des variables qui, en raison des spécificités du design expérimental, ne pourront jamais être observées sur les mêmes animaux. Dans le contexte de ne pas avoir accès aux lois jointes, nous exploitons les informations supplémentaires sur les groupes observés pour déduire le modèle de régression linéaire. Nous proposons deux estimateurs des paramètres de régression, l'un basé sur la méthode des moments et l'autre basé sur le transport optimal, ainsi que des estimateurs des intervalles de confiance basés sur le bootstrap stratifié.

Mots clés : Radiothérapie, Données temporelles complexes, Clustering joint avec alignement, Inférence de réseau omic, Fusion de données, Distance de Wasserstein.

CONTENTS

INTRODUCTION

Radiotherapy is a one of the main types of cancer treatment, along with surgery and chemotherapy, received by approximately 60% of all cancer patients (Warren et al., 2008). It is based on using ionizing radiation to either kill cancer cells or block their ability to divide. Similarly to other types of treatment, radiotherapy may induce adverse effects, that is undesirable changes to healthy tissues situated around the irradiated tumor. Modern technological advances give us access to a vast range of different modes of irradiation, that may vary in terms of dose, volume, energy, etc. It is of high interest for potential clinical applications to be able to choose such modes of radiotherapy that minimize the amount of potential adverse effects.

The ROSIRIS research program (RadiobiOlogie des Systèmes Intégrés pour l'optimisation des traitements utilisant des rayonnements ionisants et évaluation du RISque associé), initiated at IRSN in 2012, aims to improve biological knowledge of the radio-induced adverse effects by an approach that incorporates knowledge in micro and nanodosimetry, radiobiology, systems biology and radiopathology. The program consists of three axes:

- **Axis 1 (scale of a particle):** biological, chemical and physical analysis of radiation effect on the level of a particle, including micro and nanodosimetry models, and quantitative measures of signaling damages to DNA.
- **Axis 2 (scale of a cell):** analysis of in vitro response to different modalities of irradiation through such measures as clonogenic survival, cellular death, senescence, transcriptional signature, etc.
- **Axis 3 (scale of an organ/animal):** analysis of in vivo radio-induced adverse effects through such histological as well as survival and weight loss measures, performed on mice.

This PhD thesis focuses on axes 2 and 3 of the ROSIRIS project, namely on modeling phenotypic changes in cells induced by different irradiation modalities on the one hand, and the adverse effects observed on mice after irradiation on the other hand, with the ultimate goal of constructing a link between the two in order to predict the latter with the former.

The most widely used tool in radiobiology to compare two types of treatment is relative biological effectiveness, or RBE (Valentin, 2003). Currently, RBE is derived from the linear quadratic model which relates the absorbed dose to the fraction of surviving cells by performing a clonogenic assay (Munshi et al., 2005). This approach provides important information but remains too simplistic to accurately predict radio-induced adverse effects. Indeed, RBE measurements based on cellular clonogenic ability do not take into account the phenotypic changes in the surviving cells in question. The goal of this PhD is to propose multi-parametric alternatives to RBE as means to quantitatively and qualitatively compare different irradiation modalities.

In the framework of the axis 2 of the ROSIRIS project, the Laboratory of Radiobiology of Medical Exposures (LRMed) has generated numerous experimental molecular data measuring changes in omic profiles following irradiation of human endothelial cells (cells lining the inner surface of blood vessels). These data are rich in information but also complex to analyze from the statistical point of view. The first part of this thesis will focus on developing a mathematical framework extracting key features from these complex datasets, in order to assess the impact of different irradiation modalities on cell dysfunction, and to use these features for adverse effects prediction. The proposed approach will be demonstrated on datasets measuring transcriptomic profiles for multiple time points after irradiation for two different energy levels.

In parallel with this in vitro work, numerous in vivo experiments on mice have made it possible to collect histological and physiological responses to irradiation. The data from these experiments, due to the destructive nature of measurements, often do not allow to directly establish a link between these different types of responses. In the second part of the thesis, the problem of connecting the variables from in vivo experiments that are never jointly observed will be addressed, with the aim of constructing a predictive model for the adverse effects. In particular, the chosen mathematical framework will be used to predict septal thickening with the expression of pro-inflammatory genes in the context of a study on the effect of irradiated volume on adverse effects appearing in the lungs.

Part 1

MODELING AND ANALYSIS OF CELLULAR RESPONSE TO IRRADIATION BASED ON IN VITRO DATA

CHAPTER 1

INTRODUCTION

1.1. Motivation and context

In modern biomedical research, in vitro experiments are a popular choice to study the effect of a treatment. For instance, in the context of studying the response to different modalities of irradiation, in vitro allows to perform experiments on human cells and subsequently use the findings for predicting adverse effects in patients. Consequently, data resulting from such studies are often encountered in literature. One the most popular choices is measuring omic bulk response, e.g. gene or protein expression. The main quantity of interest for such experiments is a fold change, which represents the difference between the treated (case) and the non-treated (control) conditions. Additionally, the interest often lies in studying the dynamic of the response after treatment, which is why the fold changes have a temporal character. The set of fold changes typically contains multiple hundreds of biological entities, the goal is thus to compare the response to a treatment of multiple entities over time.

Altogether, we have one or multiple of such complex datasets to analyze and compare. The analysis in such cases inevitably involves reducing the datasets to a small number of representative features, or groups characterized by typical behavior templates and key actors, which methodologically translates into the task of clustering. It allows to address different features of the data systematically, based on a clustering-induced hierarchy, and facilitates interpretation of the findings. Furthermore, it is known that biological entities such as genes and proteins are causally connected to one another, forming regulatory networks. In this respect, the temporal aspect of the data can be leveraged by integrating the alignment into clustering, thus on the one

hand aiding the clustering itself, and on the other hand gaining information on temporal cascades and the predictive nature of the considered entities. Finally, the key features extracted with clustering and alignment have to be visualized in a comprehensive manner to render the results accessible and interpretable, which is achieved through network inference.

In this work, we propose a data-driven mathematical and computational framework extracting key features from complex in vitro omic datasets with specific characteristics. We introduce new estimators of fold changes as well as a new distance that allow to account for the information on uncertainties and correlations available in the considered datasets. We developed a procedure performing simultaneous alignment and clustering, a multivariate computationally efficient equivalent to the approaches proposed by Sangalli et al. (2010) and Kazlauskaite et al. (2019). We present a number of additional features, among them a penalty designed to reinforce separation of positively and negatively expressed entities, which is pertinent in the radiobiological setting. Lastly, we propose a number of tools for fold changes network visualization and summary, inspired by gene regulatory networks (Riccadonna et al., 2016; Nguyen and Braun, 2018).

1.2. Data characteristics

In this work, we consider datasets obtained from in vitro experiments in a generic setting in order to study the effect of a certain treatment. Figure 1.2.1 illustrates such experimental setting in the context of studying the response of cells to irradiation. Datasets of interest share the following characteristics, that constitute their complexity and lead to a non-trivial statistical problem:

- **Presence of two experimental conditions.** This feature is necessary to study the effect of the treatment, the experimental conditions are then case (treated) and control (non-treated). The focus is put on studying the differences between the responses for two conditions.
- **Presence of multiple time points.** This is the case if the interest lies in studying the dynamic of the response to the considered treatment. In case of the example illustrated in Figure 1.2.1, the goal is to quantify the differences in responses between the case (irradiated) and the control (non-irradiated) 2 days, 4 days, 1 week, 2 week and 3 weeks after the moment when the case culture is irradiated. A major statistical challenge arises from the measurements being of destructive nature, compromising the cells and making them unsuitable for

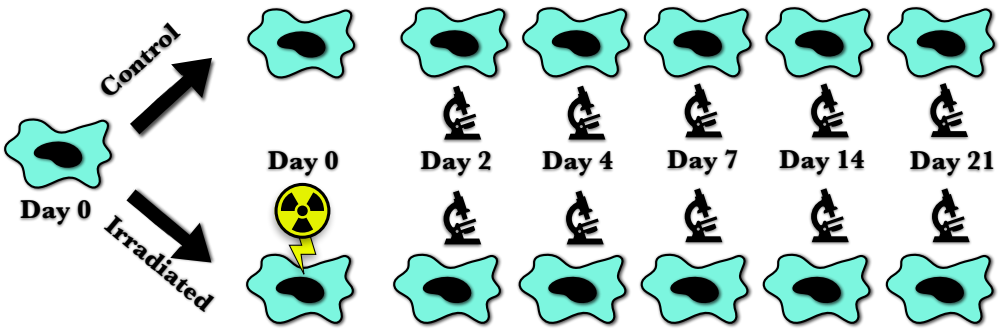


Figure (1.2.1) A schematic representation of the experimental design used to obtain the data considered in this paper. For a given experiment, the condition measurements for all considered entities are taken simultaneously from the cells of one culture flask (technical replicate). Due to the destructive nature of the measurements, every time point is observed on a separate culture flask.

repeated measurements. In our example, the measurements for Day 4 and later cannot be performed on the same culture flask as for Day 2. Hence, the measurements for all time points have to be taken on separate flasks. This implies that we do not have access to temporal correlations in the dataset, and cannot observe an actual temporal signal and treat the data as longitudinal/time series. In practice, cellular populations used to take measurements for each time point are separated from one population just prior to the experiments. Hence, the flasks used for different time points contain different cells, which implies that we can reasonably assume the independence with respect to the temporal dimension.

- **Presence of multiple biological entities.** This feature arises since goal is to study the response in terms of certain omic. The data used as an example in this work is transcriptomic, which mean that we measure gene expression. There are typically hundreds of genes that are measured simultaneously from the same cellular culture. This implies that in such a setting we have access to correlations between the considered biological entities.
- **Presence of multiple replicates.** Replicates are necessary in order to be able to account for measurement uncertainties, which is especially crucial given that different time points are observed independently. In this case, the mean values over replicates represent an actual signal, but should not be considered alone for the inference since it does not include the information on uncertainties.

The features mentioned above present a statistical challenge from perspectives. First, time series based approaches cannot be used for inference due to the destructive

nature of measurements, and the fact that time series inference is based on temporal covariances. Second, function and stochastic process-based approaches are often unsuitable in the case were the number of biological entities is significantly larger than the number of time points. Lastly, the approach has to take into account the available information on uncertainties and correlations between entities, which excludes simple techniques such as treating only mean signals. In this work, we propose a statistical framework that addresses these challenges, while leveraging data characteristics to render the computations time efficient.

1.3. Existing research

After exhaustive research of the approaches available in literature, we were unable to find any that had been designed specifically for the kind of data that we treat in this work. Consequently, the term "state of the art" is not strictly applicable in this context. Since, in a nutshell, the main goal of this work is clustering of temporal data, the main mathematical frameworks of interest include functional data analysis, and stochastic processes frameworks such as Gaussian and auto-regressive processes.

Functional data analysis. Introduced dy James O. Ramsay in Ramsay and Silverman (2005), the term refers to modeling data, often temporal data, representing the dynamic of a process over a continuum. Classical examples of application including temperature measurements and growth curves, functional methods are designed for longitudinal data, based on the idea of reducing the dimension while preserving functional patterns by projecting the data onto a functional basis. A lot of work exists involving clustering of functional data, among them works performing clustering jointly with alignment (Sangalli et al., 2010, 2009), and those targeting gene expression data (Luan and Li, 2003). While effective when applied to datasets similar to those they were designed for, these approaches are unsuitable for our purposes since they expect the temporal data to be actual signal, which is not the case treated here. Indeed, in our case the only observed signal is the average response, working with it alone implies ignoring much of the information contained in the replicates. Given that functional approach was originally intended as the main modeling framework for out data, we proposed our own functional approach to modeling omic fold changes that takes into account independently measured time points and replicates, based on Ramsay and Silverman (2005) and Zhang (2013). The model is presented in Appendix B, an illustrative example is given in Figure 1.3.1. It has been decided not to pursue this venue, having concluded that the functional approach is not the best suited for

data with such small number of unequally spaced time points, and with a much bigger number of individuals considered. Sparse approaches have been developed (Yao et al., 2005; Müller et al., 2008) to address data with these characteristics but cannot be employed in our case since the time points are not random.

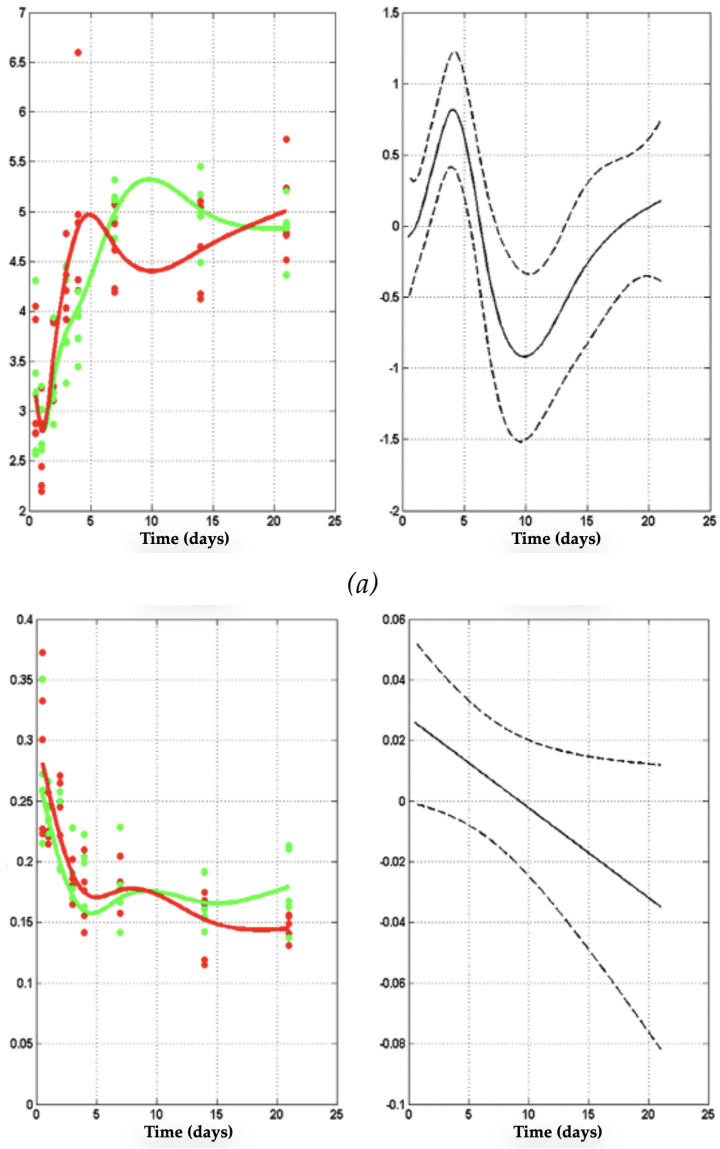


Figure (1.3.1) Example of gene expression curves inferred from one of transcriptomic datasets with functional data analysis. (a) and (b) represent responses of two different genes. Images on the left contain control (green) and case (red) responses, and those on the right contain the fold changes.

Gaussian processes. Based on placing a Gaussian prior on functions modeling the temporal data in question, this framework has been used in the past by Heinonen et al. (2015) to model the same kind of data as here, but with different goals, in particular for

detecting time intervals with differential expression between two experimental conditions. A work by Kazlauskaite et al. (2019) presents a Gaussian processes-based approach that can be considered as one the closest available alternatives to our method, since it also performs clustering of temporal data jointly with alignment. However, as it was designed for longitudinal data with many time points and few curves to align and cluster, it does not perform well on the kind of data treated in our case, which is demonstrated in Section 3.2.

Other methods. Alternative approaches that do not fall under the categories of functional data analysis or Gaussian processes include different variations of time series inference (Genolini et al., 2016; Heerah et al., 2021) and stochastic differential equations (Delattre et al., 2016; Donnet et al., 2010). In particular, the approach suggested by Genolini et al. (2016) is similar to ours on account of also being distance-based (Fréchet distance) and performing clustering while considering alignments. Nonetheless, it is based on individual curves only, and cannot take into account the information on joint distributions of multiple entities. The framework proposed by Heerah et al. (2021) is based on auto-regressive processes, with a goal of inferring causalities between entities based on the temporal dynamic of the response. Although the approach does not include clustering, we test it on our simulated data in order to assess the predictive qualities of our time warping tool.

Temporal alignment and directed network inference. The framework that we propose for temporal fold changes alignment (referring to it as time warping, with a slight abuse of language, since it was inspired by time warping in functional data analysis) is used not only for nested clustering, but also to infer directed fold changes networks for visualization and biological hypotheses generation purposes. Current research in omic network inference revolves mainly around gene regulatory networks (Riccadonna et al., 2016; Nguyen and Braun, 2018). For instance, the approach proposed by Riccadonna et al. (2016) uses dynamic time warping for directed network inference. It should be noted, however, that such methods are designed for data similar to those generated by GeneNetWeaver (Schaffter et al., 2011), which is based on relating genes with respect to the covariances only, whereas our goal is to take the differences in means into account as well.

CHAPTER 2

EXTRACTING KEY FEATURES FROM IN VITRO DATASETS

In this chapter, we introduce multivariate estimators of temporal fold changes containing all the available information on measurements uncertainties as well as the joint distributions of the entities. Next, we propose a distance between these estimators that allows to take this information fully into account. We define a transformation that we will refer to as "time warp" and will use to align fold change estimators with respect to time, and then propose a generalized version of the distance to make it applicable for aligned fold changes. Finally, we present our procedure performing fold changes clustering jointly with alignment with its properties. In addition, we describe an application of stochastic block model combined as an extension and an alternative to our main clustering procedure.

2.1. Fold change modeling and estimation

In the multivariate setting, in order to avoid introducing non-existent information by smoothing the temporal response, we consider time as discrete. For a given dataset, we define the response variable as Y_{ikj}^t for an entity (gene) $i \in \{1, 2, \dots, n_e\}$, under the experimental condition $k = 0$ if control (non-irradiated) and $k = 1$ if case (irradiated), at a time point $t \in \{t_1, t_2, \dots, t_p\}$ and for a replicate $j \in \{1, 2, \dots, n_r\}$. Here we assume there are n_r observations for both experimental condition and every time point without loss of generality, given that $n_r \geq 2$. Two constraints with respect to the covariances between the responses follow from the specificity of the experimental design described in Section 1.2:

(1) The measures of expressions for all genes for a given experimental condition and time point are collected from the same plate, which allows to estimate cross section covariances between genes, i.e. $\mathrm{Cov}\left(Y_{ikj}^t, Y_{i'kj}^t\right)$ for $i \neq i'$.

(2) We do not have access to the temporal covariance structure due to the destructive technique used in collecting measures from a plate for a given time point. Thus, measures for different time points are produced individually on different cells and are not correlated, i.e. given distinct time points $t \neq t'$, for any replicate pair $(j, j') \in \{1, 2, \dots, n_r\}^2$ and entity pair $(i, i') \in \{1, 2, \dots, n_e\}^2$ we have $\mathrm{Cov}\left(Y_{ikj}^t, Y_{i'kj'}^{t'}\right) = 0$.

Classical estimators of the fold changes in the multivariate setting are the pointwise estimators: a set of empirical individual fold changes is denoted by $\Gamma = (\Gamma_1, \dots, \Gamma_{n_e})$ where $\Gamma_i = \left(\Gamma_i^{t_1}, \dots, \Gamma_i^{t_p}\right)$ such that $\Gamma_i^t = \dfrac{\sum_{j=1}^{n_r} Y_{i1j}^t - \sum_{j=1}^{n_r} Y_{i0j}^t}{n_r} = \overline{Y_{i1}^t} - \overline{Y_{i0}^t}$, representing the difference between the means of the control and the case response. However, these estimators do not take into account the information of uncertainties and correlations present in the data. We propose a new definition of fold changes estimators in order to fully take into account all the information about their estimated distributions:

DEFINITION 2.1.1. *The estimator of the fold change of entity i is denoted by $\widehat{\Gamma}_i$, assumed to be a random Gaussian vector and is defined as follows:*

$$\widehat{\Gamma}_i | \Gamma_i, \Sigma_{\Gamma_i} \sim \mathcal{N}\left(\Gamma_i, \Sigma_{\Gamma_i}\right) \text{ such that } \Gamma_i^t = \overline{Y_{i1}^t} - \overline{Y_{i0}^t} \text{ (the pointwise estimator),}$$

$$\Sigma_{\Gamma_i} = \begin{bmatrix} \sigma^2_{\Gamma_i^{t_1}} & & 0 \\ & \ddots & \\ 0 & & \sigma^2_{\Gamma_i^{t_p}} \end{bmatrix}, \text{ and } \sigma^2_{\Gamma_i^t} = \frac{\sum_{j=1}^{n_r} \left[(Y_{i1j}^t - \overline{Y_{i1}^t})^2 + (Y_{i0j}^t - \overline{Y_{i0}^t})^2\right]}{n_r - 1}.$$

REMARK 2.1.1. *The fact that the covariance matrix in Definition 2.1.1 is diagonal is a direct consequence of the second covariance constraint mentioned above.*

2.2. Introducing a new distance between fold change estimators

Since the task at hand is clustering of the estimators of fold changes, and thus distribution clustering, there is a need to choose an appropriate distance. First, we expand the Definition 2.1.1 to a pair of fold changes by specifying their joint distribution:

DEFINITION 2.2.1. *The estimator of a pair of fold changes of entities i and i' is denoted as $\left[\widehat{\Gamma}_i^\intercal \widehat{\Gamma}_{i'}^\intercal\right]^\intercal$, assumed to be a random Gaussian vector and is defined as follows:*

$$\begin{bmatrix}\widehat{\Gamma}_i \\ \widehat{\Gamma}_{i'}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\Gamma_i \\ \Gamma_{i'}\end{bmatrix}, \begin{bmatrix}\Sigma_{\Gamma_i} & \mathrm{P}_{\Gamma_i\Gamma_{i'}} \\ \mathrm{P}_{\Gamma_i\Gamma_{i'}}^\intercal & \Sigma_{\Gamma_{i'}}\end{bmatrix}\right),$$

where the quantities Γ_i, $\Gamma_{i'}$, Σ_{Γ_i} and $\Sigma_{\Gamma_{i'}}$ describing marginal distributions of $\widehat{\Gamma}_i$ and $\widehat{\Gamma}_{i'}$ are defined according to Definition 2.1.1, and the cross-covariance matrix

$$\mathrm{P}_{\Gamma_i\Gamma_{i'}} = \begin{bmatrix}\rho_{\Gamma_i\Gamma_{i'}^{t_1}} & & 0 \\ & \ddots & \\ 0 & & \rho_{\Gamma_i\Gamma_{i'}^{t_p}}\end{bmatrix}$$

$$\textit{with } \rho_{\Gamma_i\Gamma_{i'}^t} = \frac{\sum_{j=1}^{n_r}\left[(Y_{i1j}^t - \overline{Y_{i1}^t})(Y_{i'1j}^t - \overline{Y_{i'1}^t}) + (Y_{i0j}^t - \overline{Y_{i0}^t})(Y_{i'0j}^t - \overline{Y_{i'0}^t})\right]}{n_r - 1}.$$

The chosen distance is constructed based on L^2-distance between normally distributed fold changes estimators $\widehat{\Gamma}_i$ and $\widehat{\Gamma}_{i'}$, the latter is constructed as follows (Givens and Shortt, 1984):

$$d_2^2\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right) = \mathbb{E}\|\widehat{\Gamma}_i - \widehat{\Gamma}_{i'}\|_2^2 = \|\Gamma_i - \Gamma_{i'}\|_2^2 + \mathrm{Tr}(\Sigma_{\Gamma_i}) + \mathrm{Tr}(\Sigma_{\Gamma_{i'}}) - 2\mathrm{Tr}(\mathrm{P}_{\Gamma_i\Gamma_{i'}}) \tag{2.2.1}$$

where $\|\cdot\|_2$ is the the Euclidean norm.

DEFINITION 2.2.2. *The squared L^2-distance between fold changes estimators $\widehat{\Gamma}_i$ and $\widehat{\Gamma}_{i'}$, with the joint distribution given in Definition 2.2.1, will be denoted as $\widehat{\mathbf{d_2^2}}$ and defined as follows:*

$$\widehat{\mathbf{d_2^2}}\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right) = \sum_{l=1}^{p}\left(\Gamma_i^{t_l} - \Gamma_{i'}^{t_l}\right)^2 + \sum_{l=1}^{p}\sigma^2_{\Gamma_i^{t_l}} + \sum_{l=1}^{p}\sigma^2_{\Gamma_{i'}^{t_l}} - 2\sum_{l=1}^{p}\rho_{\Gamma_i\Gamma_{i'}^{t_l}}.$$

Comparison with Wasserstein distance. Wasserstein distance served as inspiration for $\widehat{\mathbf{d_2^2}}$ and is constructed similarly to $\widehat{\Gamma}_{i'}$ under Gaussian assumption. In the general case, squared 2-Wasserstein distance between two random variables with marginal distributions P_1 and P_2 can be expressed as follows (Verdinelli and Wasserman, 2019):

$$W_2^2(P_1, P_2) = \inf_J \int \|x - y\|^2 dJ(x, y).$$

In other words, it performs optimal transport of the marginal P_2 to the P_1 by choosing the joint distribution J that produces the optimal mapping. In the Gaussian case, W_2^2 can be rewritten based on the distance presented in (2.2.1). The difference from $\widehat{\mathbf{d_2^2}}$ is

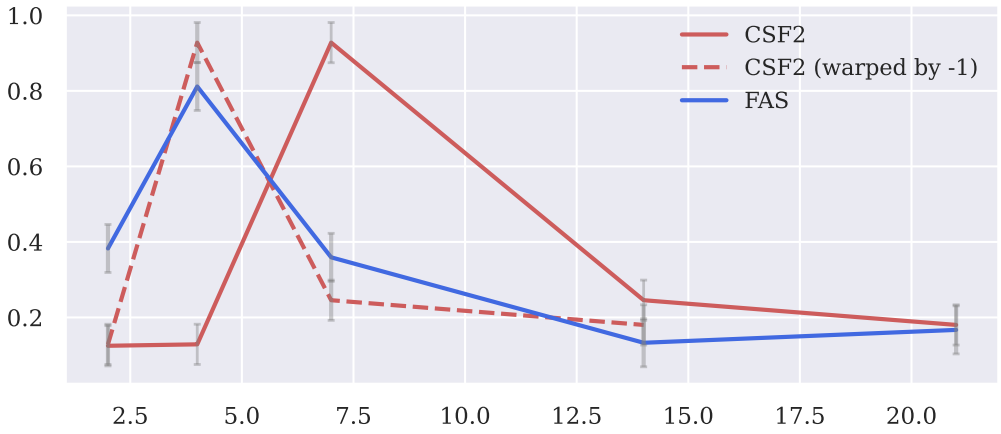


Figure (2.3.1) The effect of time warping illustrated on a figure, where means with standard deviation of a pair of normalized transcriptomic fold changes are plotted. It can be observed that after warping the fold change of the gene CSF2 backwards (continuous line represents the original mean and the dashed one represents the warped mean), its mean practically coincides with that of the fold change of the gene FAS.

that W_2^2 seeks to minimize $\mathbb{E}\|\widehat{\Gamma}_i - \widehat{\Gamma}_{i'}\|_2^2$ by finding $\mathrm{P}_{\Gamma_i\Gamma_{i'}}$ that achieves this minimum. Widely used to compare marginal distributions, it cannot, however, take into account the information on the joint distribution if it is available.

2.3. Fold change alignment

In this section, we introduce all the mathematical quantities necessary to perform the temporal alignment of the fold changes, which will be further applied jointly with clustering. The idea behind alignment is illustrated in Figure 2.3.1. In this example, fold changes are very similar up to a time shift, which means that alignment should significantly reduce the distance between them and thus force them to belong to the same cluster. First, we define a transformation of a pair of time vectors that will be referred to as a time warp, in analogy with a similar concept in functional data analysis.

DEFINITION 2.3.1. *Let $\mathcal{T}$ and $\mathcal{T}_s$ be sets of time vectors for considered omic datasets. A time warp $\mathcal{W}_s$ of step $s \in \mathbb{Z}$ is a transformation of two time vectors, defined as follows:*

$$\mathcal{W}_s : \mathcal{T}^2 \to \mathcal{T}_s^2$$

$$\begin{pmatrix} \mathbf{t^0} \\ \mathbf{t^0} \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{t^{11}} \\ \mathbf{t^{12}} \end{pmatrix} \text{ where:}$$

$$\mathbf{t^0} = \{t_l\}_{l=1}^{p}, \quad \mathbf{t^{11}} = \begin{cases} \{t_l\}_{l=1}^{p-s} \textit{ if } s > 0 \\ \{t_l\}_{l=1-s}^{p} \textit{ if } s < 0 \\ \mathbf{t^0} \textit{ if } s = 0 \end{cases} \quad \textit{and} \quad \mathbf{t^{12}} = \begin{cases} \{t_l\}_{l=1+s}^{p} \textit{ if } s > 0 \\ \{t_l\}_{l=1}^{p+s} \textit{ if } s < 0 \\ \mathbf{t^0} \textit{ if } s = 0 \end{cases}.$$

In this definition, we distinguish three major warping types: backward warp ($s < 0$), forward warp ($s > 0$) and identity warp ($s = 0$). Next, we define a warped fold changes pair in terms of the original fold changes:

DEFINITION 2.3.2. *Let $\mathbf{t^0} \in \mathcal{T}$ be a p-dimensional time vector, and $s \in \mathbb{Z}$ a warp step. We denote as $\left[\widehat{\Gamma_i \circ \mathcal{W}_s}^{\mathsf{T}} \widehat{\Gamma_{i'} \circ \mathcal{W}_s}^{\mathsf{T}}\right]^{\mathsf{T}}$ an s-warped fold changes pair $\left[\widehat{\Gamma}_i^{\mathsf{T}} \widehat{\Gamma}_{i'}^{\mathsf{T}}\right]^{\mathsf{T}}$ such that:*

$$\begin{bmatrix} \widehat{\Gamma_i \circ \mathcal{W}_s} \\ \widehat{\Gamma_{i'} \circ \mathcal{W}_s} \end{bmatrix} = \begin{bmatrix} \left(\widehat{\Gamma}_i^{\mathbf{t_1^1}}, \ldots, \widehat{\Gamma}_i^{\mathbf{t_{p-|s|}^1}}\right)^{\mathsf{T}} \\ \left(\widehat{\Gamma}_{i'}^{\mathbf{t_{p-|s|+1}^1}}, \ldots, \widehat{\Gamma}_{i'}^{\mathbf{t_{2(p-|s|)}^1}}\right)^{\mathsf{T}} \end{bmatrix}$$

where $\mathbf{t^1} = \mathcal{W}_s(\mathbf{t^0})$.

REMARK 2.3.1. *In order to be able to refer directly to an individual warped fold change, we denote it as $\widehat{\Gamma_i \circ \mathcal{W}_s}$ with a slight abuse of notation, since the warping transformation is applied to a pair of fold changes.*

For every fold changes pair only the first fold change is being moved since it allows for a more convenient manipulation of warping results while being able to examine all warping possibilities if considering both forward and backward type warping. According to the definitions presented above, the first fold change in the pair is being warped with a subsequent cutoff of extraneous parts. The second fold change in the pair does not move, however its parts that do not correspond to remaining post-warping points of the first one are also being cut off. The calculations are detailed in the proof of Proposition 2.3.1.

We introduce a new dissimilarity measure between the random fold changes estimators that is a generalization of the distance $\widehat{\mathbf{d_2^2}}$ in order to take all the covariances into account in the case where time warping is applied:

DEFINITION 2.3.3. *Let $s \in \mathbb{Z}$, and X and Y be p-dimensional Gaussian random variables with a joint distribution*

$$\begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \Sigma_X & \mathrm{P}_{XY} \\ \mathrm{P}_{XY}^\intercal & \Sigma_Y \end{bmatrix} \right) \text{ such that } [\mathrm{P}_{XY}]_{ij} = 0 \text{ if } j \neq i - s.$$

We define a dissimilarity measure $\widehat{\mathbf{diss}}_\mathbf{s}$ between X and Y as follows:

$$\widehat{\mathbf{diss}}_\mathbf{s}(X, Y) = \|\mu_X - \mu_Y\|^2 + \mathrm{Tr}(\Sigma_X) + \mathrm{Tr}(\Sigma_Y) - 2 \sum_{l=1}^{p-|s|} [\mathrm{P}_{XY}]_{\left(l+s\mathbb{1}_{\mathbb{Z}_+^*}(s), l+s\mathbb{1}_{\mathbb{Z}_-^*}(s)\right)}.$$

Applying the dissimilarity measure to the warped fold changes, we get the expression resembling the value of $\widehat{\mathbf{d}}_\mathbf{2}^\mathbf{2}$ in the case of non-warped fold changes, with extraneous part getting cut off as a result of warping:

PROPOSITION 2.3.1. *Let $s \in \mathbb{Z}$ be a warp step, and $\left[\widehat{\Gamma_i \circ \mathcal{W}_s}^\intercal \widehat{\Gamma_{i'} \circ \mathcal{W}_s}^\intercal\right]^\intercal$ an s-warped fold changes pair. The value of dissimilarity $\widehat{\mathbf{diss}}_\mathbf{s}$ between the fold changes $\widehat{\Gamma_i \circ \mathcal{W}_s}$ and $\widehat{\Gamma_{i'} \circ \mathcal{W}_s}$ can be expressed in the following form:*

$$\widehat{\mathbf{diss}}_\mathbf{s}\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right) = \sum_{l=l^*}^{p^*} \left(\Gamma_i^{t_l} - \Gamma_{i'}^{t_{l+s}}\right)^2 + \sum_{l=l^*}^{p^*} \sigma^2_{\Gamma_i^{t_l}} + \sum_{l=l^*}^{p^*} \sigma^2_{\Gamma_{i'}^{t_{l+s}}} - 2 \sum_{l=1+|s|}^{p-|s|} \rho_{\Gamma_i \Gamma_{i'}^{t_l}},$$

where $l^* = 1 - s\mathbb{1}_{\mathbb{Z}_-^*}(s)$ and $p^* = p - s\mathbb{1}_{\mathbb{Z}_+^*}(s)$.

Proof We will denote the joint distribution of the fold changes pair $\left[\widehat{\Gamma_i \circ \mathcal{W}_s}^\intercal \widehat{\Gamma_{i'} \circ \mathcal{W}_s}^\intercal\right]^\intercal$ in the following way:

$$\begin{bmatrix} \widehat{\Gamma_i \circ \mathcal{W}_s} \\ \widehat{\Gamma_{i'} \circ \mathcal{W}_s} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \Gamma_i \circ \mathcal{W}_s \\ \Gamma_{i'} \circ \mathcal{W}_s \end{bmatrix}, \begin{bmatrix} \Sigma_{\Gamma_i} \circ \mathcal{W}_s & \mathrm{P}_{\Gamma_i \Gamma_{i'}} \circ \mathcal{W}_s \\ (\mathrm{P}_{\Gamma_i \Gamma_{i'}} \circ \mathcal{W}_s)^\intercal & \Sigma_{\Gamma_{i'}} \circ \mathcal{W}_s \end{bmatrix} \right).$$

Using Definitions 2.3.2 and 2.2.1, the means can be expressed depending on the warp type:

$$\begin{bmatrix} \Gamma_i \circ \mathcal{W}_s \\ \Gamma_{i'} \circ \mathcal{W}_s \end{bmatrix} = \begin{cases} \left[\Gamma_i^{t_1} \dots \Gamma_i^{t_{p-s}} \Gamma_{i'}^{t_{1+s}} \dots \Gamma_{i'}^{t_p}\right]^\intercal & \text{if } s > 0 \\ \left[\Gamma_i^{t_{1-s}} \dots \Gamma_i^{t_p} \Gamma_{i'}^{t_1} \dots \Gamma_{i'}^{t_{p+s}}\right]^\intercal & \text{if } s < 0 \\ [\Gamma_i^\intercal \Gamma_{i'}^\intercal]^\intercal & \text{if } s = 0 \end{cases},$$

Similarly, we can express the elements of the covariance matrix:

$$\Sigma_{\Gamma_i}\circ\mathcal{W}_s = \begin{cases} \begin{bmatrix} \sigma^2_{\Gamma_i^{t_1}} & & 0 \\ & \ddots & \\ 0 & & \sigma^2_{\Gamma_i^{t_{p-s}}} \end{bmatrix} & \text{if } s>0 \\ \begin{bmatrix} \sigma^2_{\Gamma_i^{t_{1-s}}} & & 0 \\ & \ddots & \\ 0 & & \sigma^2_{\Gamma_i^{t_p}} \end{bmatrix} & \text{if } s<0 \\ \Sigma_{\Gamma_i} \text{ if } s=0 \end{cases}, \quad \Sigma_{\Gamma_{i'}}\circ\mathcal{W}_s = \begin{cases} \begin{bmatrix} \sigma^2_{\Gamma_{i'}^{t_{1+s}}} & & 0 \\ & \ddots & \\ 0 & & \sigma^2_{\Gamma_{i'}^{t_p}} \end{bmatrix} & \text{if } s>0 \\ \begin{bmatrix} \sigma^2_{\Gamma_{i'}^{t_1}} & & 0 \\ & \ddots & \\ 0 & & \sigma^2_{\Gamma_{i'}^{t_{p+s}}} \end{bmatrix} & \text{if } s<0 \\ \Sigma_{\Gamma_{i'}} \text{ if } s=0 \end{cases}.$$

$$\text{and} \quad K\circ\mathcal{W}_s = \begin{cases} \begin{bmatrix} 0 & & 0 \\ \rho_{\Gamma_i\Gamma_{i'}^{t_{1+s}}} & \ddots & \\ 0 & \rho_{\Gamma_i\Gamma_{i'}^{t_{p-s}}} & 0 \end{bmatrix} & \text{if } s>0 \\ \begin{bmatrix} 0 & \rho_{\Gamma_i\Gamma_{i'}^{t_{1-s}}} & 0 \\ & \ddots & \rho_{\Gamma_i\Gamma_{i'}^{t_{p+s}}} \\ 0 & & 0 \end{bmatrix} & \text{if } s<0 \\ \Sigma_{\Gamma_{i'}} \text{ if } s=0 \end{cases}$$

It can be noted that since the non-zero elements of the matrix $K\circ\mathcal{W}_s$ have been moved from the diagonal to either sub-diagonal or super-diagonal of order s, its trace is now equal to zero. Since the condition on the joint distribution given in Definition 2.3.3 is satisfied, we can calculate the value of $\widehat{\text{diss}}_s$ element by element, starting with the square norm of the difference between means:

$$\|\Gamma_i\circ\mathcal{W}_s - \Gamma_{i'}\circ\mathcal{W}_s\|^2 = \begin{cases} \sum_{l=1}^{p-s}\left(\Gamma_i^{t_l} - \Gamma_{i'}^{t_{l+s}}\right)^2 & \text{if } s>0 \\ \sum_{l=1-s}^{p}\left(\Gamma_i^{t_l} - \Gamma_{i'}^{t_{l+s}}\right)^2 & \text{if } s<0 \\ \sum_{l=1}^{p}\left(\Gamma_i^{t_l} - \Gamma_{i'}^{t_{l+s}}\right)^2 & \text{if } s=0 \end{cases}.$$

Next, the trace of the covariance matrix of the first warped fold change in the pair:

$$\mathrm{Tr}(\Sigma_{\Gamma_i} \circ \mathcal{W}_s) = \begin{cases} \sum_{l=1}^{p-s} \sigma^2_{\Gamma_i^{t_l}} \text{ if } s > 0 \\ \sum_{l=1-s}^{p} \sigma^2_{\Gamma_i^{t_l}} \text{ if } s < 0 \\ \sum_{l=1}^{p} \sigma^2_{\Gamma_i^{t_l}} \text{ if } s = 0 \end{cases} .$$

Similarly for the second fold change:

$$\mathrm{Tr}(\Sigma_{\Gamma_{i'}} \circ \mathcal{W}_s) = \begin{cases} \sum_{l=1+s}^{p} \sigma^2_{\Gamma_i^{t_l}} = \sum_{l=1}^{p-s} \sigma^2_{\Gamma_i^{t_{l+s}}} \text{ if } s > 0 \\ \sum_{l=1}^{p+s} \sigma^2_{\Gamma_i^{t_l}} = \sum_{l=1-s}^{p} \sigma^2_{\Gamma_i^{t_{l+s}}} \text{ if } s < 0 \\ \sum_{l=1}^{p} \sigma^2_{\Gamma_i^{t_l}} = \sum_{l=1}^{p} \sigma^2_{\Gamma_i^{t_{l+s}}} \text{ if } s = 0 \end{cases} .$$

Finally, the last term containing the cross-covariances is calculated:

$$\sum_{l=1}^{p-|s|} \left[\mathrm{P}_{\Gamma_i\Gamma_{i'}} \circ \mathcal{W}_s\right]_{\left(l+s\mathbb{1}_{\mathbb{Z}_+^*}(s), l+s\mathbb{1}_{\mathbb{Z}_-^*}(s)\right)} = \begin{cases} \sum_{l=1+s}^{p-s} \rho_{\Gamma_i\Gamma_{i'}^{t_l}} \text{ if } s > 0 \\ \sum_{l=1-s}^{p+s} \rho_{\Gamma_i\Gamma_{i'}^{t_l}} \text{ if } s < 0 \\ \sum_{l=1}^{p} \rho_{\Gamma_i\Gamma_{i'}^{t_l}} \text{ if } s = 0 \end{cases} .$$

Hence, we obtain the value of the dissimilarity by writing the expression for any step $s \in \mathbb{Z}$. □

REMARK 2.3.2. *It can be noted that* $\widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right) = \widehat{\mathbf{d}_\mathbf{2}^\mathbf{2}}\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right)$ *in the case of the identity warp* $s = 0$.

REMARK 2.3.3. *We chose to construct time warping in a way that some parts of the fold changes that move outside of the temporal domain considered, get cut off. This choice was deemed as preferable to alternatives based on extending the fold changes instead of cutting them, since they imply adding unobserved information. However, it potentially introduces a bias in comparison between warped and unwarped sequences, hence it is important to normalize the dissimilarities with respect to the number of post-warping time points in order to render them comparable .*

2.4. Joint clustering with alignment

The main idea behind our approach to key features selection for a given dataset is reducing the fold changes to a small number of behavior types up to a time shift, which

translates into clustering of aligned fold changes. In order to combine dissimilarities between fold changes with optimal alignments, we introduce the following matrix:

DEFINITION 2.4.1. *Let $\mathcal{S} \subset \mathbb{Z}$ be a finite set of considered warp steps, such that $\mathcal{S} = \{-s_{max}, \dots, s_{max}\}$, given a maximal warping step $s_{max} \in \mathbb{N}$. The Optimal Warping Dissimilarity matrix, denoted $\mathcal{OWD}$, is a matrix containing the values of the dissimilarity measure $\widehat{\mathbf{diss}}_{\mathbf{s}}$ for all pairs of fold changes in case of their optimal pairwise alignment over the set of all possible warps with steps in $\mathcal{S}$, or formally:*

$$\mathcal{OWD} = \left[\min_{s \in \mathcal{S}} \left[\widehat{\mathbf{diss}}_{\mathbf{s}} \left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s} \right) \right] \right]_{1 \leq i, i' \leq n_e}.$$

According to Definition 2.3.1, alignments are uniquely defined not for a given fold change, but for a given pair of fold changes. In other words, a fold change may have different optimal warps when paired with different fold changes, and thus the elements of the $\mathcal{OWD}$ matrix are not directly comparable. Hence, the task can only be approached by iterating between clustering and alignment until convergence. We choose the framework of clustering with k-medoids (Kaufmann and Rousseeuw, 1987) since it is based on comparing elements to a medoid, which is an actual member of the population, thus allowing to keep track of correlations throughout clustering. In comparison, many other methods such as k-means or its extensions are unsuitable for the task for the following reasons:

- **K-means applied directly to the $\mathcal{OWD}$ matrix:** while this approach is potentially applicable to choose clusters, it is not suitable for alignment since it does not allow to choose fold change warps uniquely.
- **K-means through constructing a fold change barycenter:** after the first iteration, the information on joint distributions is lost in this case since the barycenters are not observed from the data.

Using k-medoids allows to make alignment clustering-dependent: while comparing elements to medoids for clustering, their warps can also be chosen in a unique way with respect to medoids. We perform clustering using "k-means like" version of k-medoids (Park and Jun, 2009) based on a series of random initializations of type k-means++ (Arthur and Vassilvitskii, 2007). Pseudocode for the state-of-the-art version of joint clustering and alignment, applied in the context discussed in this work, is presented in Algorithm 1. Similar frameworks are used in Sangalli et al. (2010) and Kazlauskaite et al. (2019).

Algorithm 1 Joint clustering and alignment algorithm: classical version

Require: Fold changes $\widehat{\Gamma} = \left(\widehat{\Gamma}_1, \ldots, \widehat{\Gamma}_{n_e}\right)$, $K \in \mathbb{N}$, $it_{max} \in \mathbb{N}$, $n_{init} \in \mathbb{N}$, $\epsilon > 0$.

$TC \leftarrow \infty$
for $init \in \{1, \ldots, n_{init}\}$ **do**
  Initialize centroids $C = (C_1, \ldots, C_K) \subset \{1, \ldots, n_e\}$ with kmeans++
  $TC_{it} \leftarrow \infty$
  $\Delta TC \leftarrow \infty$
  $it \leftarrow 1$
  while $\Delta TC > \epsilon$ **and** $it < it_{max}$ **do**
    $TC_{it}^{new} \leftarrow 0$
    1. Assign step:
    for $i \in \{1, \ldots, n_e\}$ **do**
      $d_{min} \leftarrow \infty$
      for $k \in \{1, \ldots, K\}$ **do**
        $s_k \leftarrow \arg\min_{s \in \mathcal{S}} \widehat{\mathbf{diss}}_{\mathbf{s}} \left(\widehat{\Gamma_i \circ \mathcal{W}}_s, \widehat{\Gamma_{C_k} \circ \mathcal{W}}_s\right)$ ▷ Align FCs with the centroid
        $d_k \leftarrow \widehat{\mathbf{diss}}_{\mathbf{s_k}} \left(\widehat{\Gamma_i \circ \mathcal{W}}_{s_k}, \widehat{\Gamma_{C_k} \circ \mathcal{W}}_{s_k}\right)$
        if $d_k < d_{min}$ **then** ▷ Assign FCs to centroids
          $d_{min} \leftarrow d_k$
          $Cl_i \leftarrow k$
        end if
      end for
    end for
    2. Update step:
    for $k \in \{1, \ldots, K\}$ **do**
      $d_{min} \leftarrow \infty$
      for $i \in cluster_k = \{i \in \{1, \ldots, n_e\} | Cl_i = k\}$ **do** ▷ Candidate for a centroid
        $d_{cluster_k} \leftarrow 0$
        for $i' \in cluster_k$ **do**
          $s_{i'i} \leftarrow \arg\min_{s \in \mathcal{S}} \widehat{\mathbf{diss}}_{\mathbf{s}} \left(\widehat{\Gamma_{i'} \circ \mathcal{W}}_s, \widehat{\Gamma_i \circ \mathcal{W}}_s\right)$ ▷ Align FCs with the candidate
          $d_{cluster_k} \leftarrow d_{cluster_k} + \widehat{\mathbf{diss}}_{\mathbf{s_{i'i}}} \left(\widehat{\Gamma_{i'} \circ \mathcal{W}}_{s_{i'i}}, \widehat{\Gamma_i \circ \mathcal{W}}_{s_{i'i}}\right)$
        end for
        if $d_{cluster_k} < d_{min}$ **then** ▷ Choose new centroid
          $d_{min} \leftarrow d_{cluster_k}$
          $C_k^{new} \leftarrow i$
        end if
      end for
      $TC_{it}^{new} \leftarrow TC_{it}^{new} + d_{min}$
    end for
    3. Calculate the change in total cost:
    $\Delta TC \leftarrow TC_{it} - TC_{it}^{new}$
    if $\Delta TC > \epsilon$ **then**
      $C \leftarrow (C_1^{new}, \ldots, C_K^{new})$
      $TC_{it} \leftarrow TC_{it}^{new}$
    end if
    $it \leftarrow it + 1$
  end while
  if $TC_{it} < TC$ **then**
    $C \leftarrow (C_1, \ldots, C_K)$ ▷ centroids labels
    $Cl \leftarrow (Cl_1, \ldots, Cl_{n_e})$ ▷ cluster labels
    $\mathcal{W} \leftarrow \left(s_{1Cl_1}, \ldots, s_{n_eCl_{n_e}}\right)$ ▷ warps
    $TC \leftarrow TC_{it}$
  end if
end for
return C, Cl, $\mathcal{W}$

We propose a modification of the algorithm, that reduces the computation time by leveraging the low temporal dimensionality of the data in the multivariate setting. Since the number of time points is typically small, the number of possible warps has to be even smaller and known in advance, and since the distributions of the fold change pairs under different warps are known, it is possible to calculate all alignment options before performing clustering, while reducing computation time and in a non-memory-intensive way. Consequently, we introduce the following quantity, that will be used in the modified version of the algorithm:

DEFINITION 2.4.2. *Let $\mathcal{S} \subset \mathbb{Z}$ be a finite set of considered warp steps, such that $\mathcal{S} = \{-s_{max}, \dots, s_{max}\}$, given a maximal warping step $s_{max} \in \mathbb{N}$. The Optimal Warp matrix, denoted $\mathcal{OW}$, is a matrix containing, for all pairs of fold changes, the values in $\mathcal{S}$ corresponding to the warp steps allowing to achieve their optimal pairwise alignment with respect to the dissimilarity measure $\widehat{\mathbf{diss}}_{\mathbf{s}}$, or formally:*

$$\mathcal{OW} = \left[\arg\min_{s\in\mathcal{S}} \left[\widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right)\right]\right]_{1\leq i,i'\leq n_e}.$$

PROPOSITION 2.4.1. *The following statements are true for matrices $\mathcal{OWD}$ and $\mathcal{OW}$:*

(1) $\mathcal{OWD}$ is symmetric.

(2) $\mathcal{OW}$ is anti-symmetric.

Proof Let $(i, i') \in \{1, \dots, n_e\}^2$ be an entity pair. The statements of the proposition are equivalent to saying that, for any warp step $s \in \mathcal{S}$, we have:

(2.4.1)
$$\begin{cases} \min_{s\in\mathcal{S}} \left[\widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right)\right] = \min_{s\in\mathcal{S}} \left[\widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_{i'} \circ \mathcal{W}_s}, \widehat{\Gamma_i \circ \mathcal{W}_s}\right)\right] \\ \arg\min_{s\in\mathcal{S}} \left[\widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right)\right] = -\arg\min_{s\in\mathcal{S}} \left[\widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_{i'} \circ \mathcal{W}_s}, \widehat{\Gamma_i \circ \mathcal{W}_s}\right)\right] \end{cases}.$$

Let us denote $s_* = \arg\min_{s\in\mathcal{S}} \left[\widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right)\right]$. To prove both parts of the proposition, it suffices to show that the following is true:

(2.4.2)
$$\widehat{\mathbf{diss}}_{\mathbf{s}_*}\left(\widehat{\Gamma_i \circ \mathcal{W}_{s_*}}, \widehat{\Gamma_{i'} \circ \mathcal{W}_{s_*}}\right) = \widehat{\mathbf{diss}}_{-\mathbf{s}_*}\left(\widehat{\Gamma_{i'} \circ \mathcal{W}_{-s_*}}, \widehat{\Gamma_i \circ \mathcal{W}_{-s_*}}\right)$$

Using the expression of the dissimilarity given in Proposition 2.3.1, we can develop the left-hand side of (2.4.2):

(2.4.3)
$$\widehat{\mathbf{diss}}_{\mathbf{s}_*}\left(\widehat{\Gamma_i \circ \mathcal{W}_{s_*}}, \widehat{\Gamma_{i'} \circ \mathcal{W}_{s_*}}\right) = \sum_{l=l^*}^{p^*}\left(\Gamma_i^{t_l} - \Gamma_{i'}^{t_{l+s_*}}\right)^2 + \sum_{l=l^*}^{p^*}\sigma^2_{\Gamma_i^{t_l}} + \sum_{l=l^*}^{p^*}\sigma^2_{\Gamma_{i'}^{t_{l+s_*}}} - 2\sum_{l=1+|s_*|}^{p-|s_*|}\rho_{\Gamma_i\Gamma_{i'}^{t_l}}.$$

Algorithm 2 Joint clustering and alignment algorithm based on $\mathcal{OWD}$ and $\mathcal{OW}$ matrices

Require: Fold changes $\widehat{\Gamma} = \left(\widehat{\Gamma}_1, \ldots, \widehat{\Gamma}_{n_e}\right)$, $K \in \mathbb{N}$, $it_{max} \in \mathbb{N}$, $n_{init} \in \mathbb{N}$, $\epsilon > 0$.

Compute $\mathcal{OWD}$ and $\mathcal{OW}$
$TC \leftarrow \infty$
for $init \in \{1, \ldots, n_{init}\}$ **do**
  Initialize centroids $C = (C_1, \ldots, C_K) \subset \{1, \ldots, n_e\}$ with kmeans++
  $TC_{it} \leftarrow \infty$
  $\Delta TC \leftarrow \infty$
  $it \leftarrow 1$
  while $\Delta TC > \epsilon$ **and** $it < it_{max}$ **do**
    $TC_{it}^{new} \leftarrow 0$
    1. Assign step:
    for $i \in \{1, \ldots, n_e\}$ **do**
      $Cl_i \leftarrow \arg\min_{k \in \{1, \ldots, K\}} \mathcal{OWD}_{iC_k}$ ▷ Assign aligned FCs to centroids
    end for
    2. Update step:
    for $k \in \{1, \ldots, K\}$ **do**
      $d_{min} \leftarrow \infty$
      for $i \in cluster_k = \{i \in \{1, \ldots, n_e\} | Cl_i = k\}$ **do** ▷ Candidate for a centroid
        $d_{cluster_k} \leftarrow \sum_{i' \in cluster_k} \mathcal{OWD}_{ii'}$
        if $d_{cluster_k} < d_{min}$ **then** ▷ Choose new centroid
          $d_{min} \leftarrow d_{cluster_k}$
          $C_k^{new} \leftarrow i$
        end if
      end for
      $TC_{it}^{new} \leftarrow TC_{it}^{new} + d_{min}$
    end for
    3. Calculate the change in total cost:
    $\Delta TC \leftarrow TC_{it} - TC_{it}^{new}$
    if $\Delta TC > \epsilon$ **then**
      $C \leftarrow (C_1^{new}, \ldots, C_K^{new})$
      $TC_{it} \leftarrow TC_{it}^{new}$
    end if
    $it \leftarrow it + 1$
  end while
  if $TC_{it} < TC$ **then**
    $C \leftarrow (C_1, \ldots, C_K)$ ▷ centroids labels
    $Cl \leftarrow (Cl_1, \ldots, Cl_{n_e})$ ▷ cluster labels
    $\mathcal{W} = \left(\mathcal{OW}_{1Cl_1}, \ldots, \mathcal{OW}_{n_e Cl_{n_e}}\right)$ ▷ warps
    $TC \leftarrow TC_{it}$
  end if
end for
return C, Cl, $\mathcal{W}$

where $l^* = 1 - s_* \mathbb{1}_{\mathbb{Z}_-^*}(s_*)$ and $p^* = p - s_* \mathbb{1}_{\mathbb{Z}_+^*}(s_*)$.

Similarly, we develop the right-hand side:

(2.4.4)

$$\widehat{\mathbf{diss}}_{-\mathbf{s}_*}\left(\Gamma_{i'} \widehat{\circ \mathcal{W}_{-s_*}}, \Gamma_i \widehat{\circ \mathcal{W}_{-s_*}}\right) = \sum_{l=l_*}^{p_*} \left(\Gamma_{i'}^{t_l} - \Gamma_i^{t_{l-s_*}}\right)^2 + \sum_{l=l_*}^{p_*} \sigma^2_{\Gamma_{i'}^{t_l}} + \sum_{l=l_*}^{p_*} \sigma^2_{\Gamma_i^{t_{l-s_*}}} - 2 \sum_{l=1+|-s_*|}^{p-|-s_*|} \rho_{\Gamma_{i'} \Gamma_i^{t_l}}.$$

where $l_* = 1 - (-s_*)\mathbb{1}_{\mathbb{Z}^*_-}(-s_*)$ and $p_* = p - (-s_*)\mathbb{1}_{\mathbb{Z}^*_+}(-s_*)$. These quantities can be rewritten as $l_* = 1 + s_*\mathbb{1}_{\mathbb{Z}^*_+}(s_*) = l^* + s_*$ and $p_* = p + s_*\mathbb{1}_{\mathbb{Z}^*_-}(s_*) = p^* + s_*$. It can also be noticed that $|-s_*| = |s_*|$, and $\rho_{\Gamma_{i'}\Gamma_i^{t_l}} = \rho_{\Gamma_i\Gamma_{i'}^{t_l}}$ by the symmetry of the covariance. Hence, we can rewrite (2.4.4) as follows:

$$\widehat{\mathbf{diss}}_{-\mathbf{s}_*}\left(\Gamma_{i'}\widehat{\circ\mathcal{W}}_{-s_*}, \Gamma_i\widehat{\circ\mathcal{W}}_{-s_*}\right) = \sum_{l=l^*+s_*}^{p^*+s_*}\left(\Gamma_i^{t_{l-s_*}} - \Gamma_{i'}^{t_l}\right)^2 + \sum_{l=l^*+s_*}^{p^*+s_*}\sigma^2_{\Gamma_i^{t_{l-s_*}}} + \sum_{l=l^*+s_*}^{p^*+s_*}\sigma^2_{\Gamma_{i'}^{t_l}} - 2\sum_{l=1+|s_*|}^{p-|s_*|}\rho_{\Gamma_i\Gamma_{i'}^{t_l}}. \tag{2.4.5}$$

It can be noticed that the expression in (2.4.5) is identical to (2.4.3), which concludes the proof. □

REMARK 2.4.1. *$\mathcal{OW}$ allows to interpret the main warping types. For a given fold changes pair, if the optimal warp is the identity warp, they are referred to as simultaneous. If not, then one fold change in the pair is warped forward with respect to the other, whereas the other fold change is being warped backwards with respect to the first. In this case, the fold change that is warped forward is referred to as 'predictive' of other one, whereas the latter is labeled as 'predicted', or 'regulated'.*

The modified version of the previous algorithm, presented in Algorithm 2, is based on integrating time warping in the clustering process through pre-calculated matrices $\mathcal{OWD}$ and $\mathcal{OW}$. The former replaces a standard dissimilarity matrix, the latter is used to extract final warps. The comparison between the two algorithms leads to the following result:

THEOREM 2.4.1. *The following is true about the joint clustering and alignment algorithms:*

(1) Algorithm 2 converges in a finite number of iterations.

(2) Algorithms 1 and 2 are equivalent, in the sense that for the same input they produce the same output.

(3) Algorithms 1 and 2 have polynomial time complexities, that are given in the proof. Moreover, the degree of the largest polynomials of the time complexity of Algorithm 1 is greater than that of Algorithm 2, meaning that the latter is less complex.

Proof To prove the first statement, it suffices to show that the total cost always decreases, that is, for every iteration it, $TC_{it} \geq TC_{it+1}$.

We denote $(Cl_1, \dots, Cl_{n_e})$ and $\left(Cl_1^*, \dots, Cl_{n_e}^*\right)$ cluster labels at iterations it and $it+1$ respectively. For a given initialization, the cost at iteration it can be written as follows:

$$TC_{it} = \sum_{k=1}^{K} \sum_{i \in cluster_k} \mathcal{OWD}_{iC_k}, \tag{2.4.6}$$

given, for every cluster label $k \in \{1, \dots, K\}$, $cluster_k = \{i \in \{1, \dots, n_e\} | Cl_i = k\}$ the current composition of the cluster, and C_k the corresponding centroid. Similarly, the cost at iteration $it + 1$ can be expressed:

$$TC_{it+1} = \sum_{k=1}^{K} \sum_{i \in cluster_k^*} \mathcal{OWD}_{iC_k^*}, \tag{2.4.7}$$

given, for every cluster label $k \in \{1, \dots, K\}$, $cluster_k^* = \{i \in \{1, \dots, n_e\} | Cl_i^* = k\}$ the current composition of the cluster, and C_k^* the corresponding centroid. Additionally, for a given cluster label k, we denote the migrating sub-clusters:

- the sub-cluster of elements that left cluster k at $it + 1$:

$$cluster_k^{*C} = \{i \in \{1, \dots, n_e\} | Cl_i = k \text{ and } Cl_i^* \neq k\},$$

- the sub-cluster of elements that joined cluster k at $it + 1$:

$$cluster_k^{C} = \{i \in \{1, \dots, n_e\} | Cl_i \neq k \text{ and } Cl_i^* = k\}.$$

Noticing that $cluster_k = \left(cluster_k^* \cup cluster_k^{*C}\right) \setminus cluster_k^C$, the quantity TC_{it} can be decomposed as follows:

$$TC_{it} = \underbrace{\sum_{k=1}^{K} \sum_{i \in cluster_k^*} \mathcal{OWD}_{iC_k}}_{\text{A}} + \underbrace{\sum_{k=1}^{K} \sum_{i \in cluster_k^{*C}} \mathcal{OWD}_{iC_k}}_{\text{B}} - \underbrace{\sum_{k=1}^{K} \sum_{i \in cluster_k^{C}} \mathcal{OWD}_{iC_k}}_{\text{C}}. \tag{2.4.8}$$

First, it follows from the "Update" step by construction that

$$A = \sum_{k=1}^{K} \sum_{i \in cluster_k^*} \mathcal{OWD}_{iC_k} \geq \sum_{k=1}^{K} \sum_{i \in cluster_k^*} \mathcal{OWD}_{iC_k^*} = TC_{it+1}.$$

Next, we consider the quantities B and C. It can be noticed, by construction of the "Assign" step, that for every $i \in cluster_k^{*C}$ there exists a unique $k* \in \{1, \dots, K\} \setminus k$ such that $i \in cluster_{k*}^C$ and $\mathcal{OWD}_{iC_{k^*}} \leq \mathcal{OWD}_{iC_k}$. In other words, there is a bijection between the indices in B and C, such that the corresponding elements of the sum in B are larger than those in C. Therefore, B $-$ C ≥ 0, and $TC_{it} = \text{A} + \text{B} - \text{C} \geq TC_{it+1}$.

Thus, the total cost sequence is decreasing, and, noticing that total cost is positive, it can be concluded that the sequence has a limit. Finally, there is a finite number of cluster configurations possible, therefore the sequence of total costs contains a finite number of values. Hence, the algorithm converges in a finite number of iterations.

The second statement follows directly from Definitions 2.4.1 and 2.4.2. In particular, we have:

- Line 12 of Algorithm 2 is equivalent to lines 11-19 of Algorithm 1, since:

$$\underset{k\in\{1,\dots,K\}}{\arg\min}\ \mathcal{OWD}_{iC_k} = \underset{k\in\{1,\dots,K\}}{\arg\min}\left(\min_{s\in\mathcal{S}}\left[\widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_i\circ\mathcal{W}_s},\widehat{\Gamma_{C_k}\circ\mathcal{W}_s}\right)\right]\right) = \underset{k\in\{1,\dots,K\}}{\arg\min}\,(d_k),$$

  where d_k is the quantity from Algorithm 1 of the final value after the for loop terminating at line 19.
- Lines 18-22 of Algorithm 2 are equivalent to lines 25-33 of Algorithm 1, since:

$$\sum_{i'\in cluster_k}\mathcal{OWD}_{ii'} = \sum_{i'\in cluster_k}\mathcal{OWD}_{i'i} = \sum_{i'\in cluster_k}\widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_{i'}\circ\mathcal{W}_s},\widehat{\Gamma_i\circ\mathcal{W}_s}\right),$$

  due to the symmetry of $\mathcal{OWD}$, the final quantity being equivalent to the value of $d_{cluster_k}$ at line 33 of Algorithm 1.
- Line 37 of Algorithm 2 is equivalent to line 48 of Algorithm 1.

We obtain the following complexities for different parts of the algorithms:

(1) Assign step: $\mathcal{O}(n_eK|\mathcal{S}|)$ for Algorithm 1, and $\mathcal{O}(n_eK)$ for Algorithm 2.

(2) Update step: $\mathcal{O}(n_e^2K|\mathcal{S}|)$ for Algorithm 1, and $\mathcal{O}(n_e^2K)$ for Algorithm 2.

Thus, adding the complexity of calculating the matrices $\mathcal{OWD}$ and $\mathcal{OW}$ beforehand, we obtain in total $\mathcal{O}(n_{init}it_{max}n_eK|\mathcal{S}|(1+n_e))$ for Algorithm 1, and $\mathcal{O}(n_{init}it_{max}n_eK(1+n_e)+n_e^2|\mathcal{S}|)$ for Algorithm 2. The degree of the largest polynomial of the former is 6, and that of the latter is 5, hence Algorithm 2 is less complex. □

REMARK 2.4.2. *In practice, the improvement of Algorithm 2 in terms of the runtime can be very important because a large value often has to be chosen for n_{init}. Since such clustering algorithms tend to be rather initialization sensitive, it is beneficial to perform such a number of random initializations that covers a sufficiently big range of initial combinations, which becomes important with higher values of n_e.*

REMARK 2.4.3. *In line 37 of Algorithm 2, the indexing order of $\mathcal{OW}$ is important, since this matrix is anti-symmetric, as shown in Proposition 2.3.1. This specific indexing implies that the fold changes are being warped with respect to their centroids, which remains static.*

2.5. Stochastic block model: an alternative approach to clustering

Stochastic block model[1] is a class of models for random graphs which can be considered as a clustering technique since it assumes that the graph has a latent community structure. Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ be a random graph, where $\mathcal{N}$ is the node set representing biological entities, and $\mathcal{E}$ is the edge set. In the framework of stochastic block models, a latent variable that determines the block structure of the network is introduced. Having fixed the total number of communities K, we define a binary matrix $U = (U_{iq})_{i\in\{1,\dots,n_e\},q\in\{1,\dots,K\}}$ such that $U_{iq} = 1$ indicates the entity i belonging to the community q. Matrix U is a realization of a random variable following multinomial probability distribution with the parameter $\lambda \in \{\lambda \in \mathbb{R}_+^K | \sum_q \lambda_q = 1\}$ such that $\mathbb{P}(U_{iq} = 1) = \lambda_q$, with $U_i \perp\!\!\!\perp U_{i'}$ for $(i, i') \in \{1, \dots, n_e\}^2$ such that $i \neq i'$. The connectedness of nodes is described by a random adjacency matrix $X = (X_{ii'})_{(i,i')\in\{1,\dots,n_e\}^2}$, such that $X_{ii'} = 1$ if i and i' form an edge and $X_{ii'} = 0$ otherwise. Given the groups that the vertices belong to, the appearance of the edge between them follows a Bernoulli distribution with the parameter $\pi = (\pi_{qq'})_{(q,q')\in\{1,\dots,K\}^2}$ as follows: $\mathbb{P}(X_{ii'} = 1 | U_{iq}U_{i'q'} = 1) = \pi_{qq'}$.

In practice, the block structure of the network is inferred from the given adjacency matrix. The likelihood $P(X|\lambda, \pi)$ can be explicitly written, but in practice it is not calculable due to the number of possible partitions to be explored growing exponentially with the number of entities in the network. Inference is performed using variational expectation-maximization algorithm (or VEM). The method introduces a certain distribution over the latent variable space Q and is based on decomposing the complete log-likelihood as follows:

$$P(X|\lambda, \pi) = \mathbb{E}_Q \left[\log P(X, U|\lambda, \pi) - \log Q(U)\right] + D_{KL}\left(Q(U) \| P(U|X, \lambda, \pi)\right),$$

where the last term is the Kullback-Leibler divergence of the distribution Q from $P(U|X, \lambda, \pi)$, the actual unknown distribution of latent variables. The inference consists of integrating into EM a new optimization problem: choosing a distribution Q over the latent variable space that minimizes the gap between the tractable quantity $\mathbb{E}_Q \left[\log P(X, U|\lambda, \pi) - \log Q(U)\right]$ and the complete log-likelihood. Hence, the following double maximization problem is resolved:

$$\max_{\lambda,\pi,Q} \mathbb{E}_Q \left[\log P(X, U|\lambda, \pi) - \log Q(U)\right] = \max_{\lambda,\pi} P(X|\lambda, \pi) - \min_{Q} D_{KL}\left(Q(U) \| P(U|X, \lambda, \pi)\right).$$

[1] see Lee and Wilkinson (2019) for more details on stochastic block models

In the variational context, latent variables U_i are assumed to be independent but no longer identically distributed. By introducing an additional set of parameters $\tau = (\tau_1, \ldots, \tau_{n_e})$ with $\tau_i \in \mathbb{R}_+^K$ and $\sum_q \tau_{iq} = 1$, we obtain a multinomial distribution of the latent variables under Q such that $\mathbb{P}_Q(U_{iq} = 1) = \tau_{iq}$.

2.5.1. Clustering-based SBM. In addition to our main method, we propose detection of communities from the fold changes network using stochastic block model as a model-based alternative to k-medoids clustering. While accomplishing the task equivalent to clustering, it exploits properties of the graph constructed from the dissimilarities in order to infer communities (or blocks) rather than exploiting the dissimilarities directly. The stochastic block model of omic fold changes is inferred using the tools provided by the Python package SparseBM (Frisch et al., 2021). The authors of the package suggest choosing the best result in terms of ICL (Integrated Completed Likelihood, a criterion commonly used in stochastic block models inference) produced based on numerous random initializations of the parameters (λ, π, τ). The reason why a big number of those is needed is that the VEM algorithm is very initialization-sensitive: different initial parameter combinations lead to falling into different local optima, which are not necessarily close to the global optimum. We propose initializing the parameters based on clustering performed with $\widehat{\mathbf{d_2^2}}$-based k-medoids, thus combining two approaches to community detection along with adding the network aspect and minimizing the effect of bad parameter initialization. Such approach allows to achieve a network representation with communities close to those identified by k-medoids, while going even further in terms of cost minimization.

The focus is put on the variational latent variable parameter τ and the edges presence parameter π. For the fold changes $\{\widehat{\Gamma}_1, \ldots, \widehat{\Gamma}_{n_e}\}$ partitioned into K clusters, we proceed as follows:

- $\tau^{\mathbf{init}}$: a number $\epsilon \in \mathbb{R}_+$ is chosen such that $\epsilon \ll 1$, then:

$$\tau_{iq}^{init} = (1 - \epsilon \times (K - 1)) \times \mathbb{1}_{Cl_i = q} + \epsilon \times \mathbb{1}_{Cl_i \neq q},$$

  which is equivalent to starting off the algorithm close to the local optimum corresponding to the hard clustering obtained with k-medoids;
- $\pi^{\mathbf{init}}$: this parameter is chosen randomly so that $\pi_{qq} \gg \pi_{qq'}$ for $q \neq q'$, which forces the algorithm to adjust initial clusters by increasing connectivity within each of them.

CHAPTER 3

SIMULATION STUDIES

A series of simulation studies was performed in order to evaluate the proposed approach to key features extraction in the framework similar to that of temporal omic fold changes with respect to the existing alternatives. The first group of simulations focuses on the choice of a distance between random variables on the one hand and of clustering algorithms on the other hand, whereas the second group of simulations is aimed at studying different configurations of the proposed approach, such as time warping and stochastic block model inference. Fold changes were simulated based on 4 behavior types representing 2 or 4 clusters. These behavior types were meant to reproduce the characteristics of the real fold changes that were expected to be distinguished by the proposed procedure. The simulations we performed are characterized by a relatively low level of model-imposed features. In particular, we only assume that the fold changes have estimated probability distributions described by means and covariances, thus we simulate directly the fold changes estimators. The temporal complexity of the data is independent of the framework we propose, and is only inspired by the functional patterns we observe in real data.

3.1. Simulation design

Let us consider a set of $n_e = 300$ simulated fold changes over $p = 8$ time points. Simulated fold changes are defined by their means and their covariance matrix. Using the same notation as previously in the context of real datasets, let the means be represented by $\Gamma = (\Gamma_1, \dots, \Gamma_{n_e})$, where $\Gamma_i = \left(\Gamma_i^{t_1}, \dots, \Gamma_i^{t_p}\right)$ for $i \in \{1, \dots, n_e\}$. The covariance matrices will be denoted by $\Psi = (\Psi_{ii'})_{(i,i') \in \{1,\dots,n_e\}^2}$, where:

$$\Psi_{ii'} = \begin{bmatrix} \psi_{\Gamma_i\Gamma_{i'}^{t_1}} & & 0 \\ & \ddots & \\ 0 & & \psi_{\Gamma_i\Gamma_{i'}^{t_p}} \end{bmatrix}, \text{ such that } \psi_{\Gamma_i\Gamma_{i'}^{t}} = \begin{cases} \sigma^2_{\Gamma_i^t} \text{ if } i = i' \\ \rho_{\Gamma_i\Gamma_{i'}^{t}} \text{ otherwise} \end{cases} \quad \text{for } t \in \{t_1, \dots, t_p\}.$$

Simulation design includes two scenarios with respect to means (henceforth referred to as M1 and M2), and 6 scenarios with respect to covariance matrix (C1, C2, C3, C4, C5 and C6).

3.1.1. Scenarios with respect to means.

3.1.1.1. *Scenario M1.* This approach to simulating means has been used in the simulation study focusing on the choice of distance and clustering algorithm (also referred to as simulation study 1). Here, time points are chosen to be unequally spaced, corresponding to those from the real omic datasets, in particular: $(t_1, \dots, t_p) = (0.5, 1, 2, 3, 4, 7, 14, 21)$. For each simulated entity index $i \in \{1, \dots, n_e\}$, the simulated fold change mean is $\Gamma_i = (f(t_1), \dots, f(t_p))$, where function $f \in \{f_1, f_2, f_3, f_4\}$ is chosen among four functions representing a distinct behavior type, or a cluster, according to the set of cluster labels, which in its turn is chosen depending on the scenario with respect to the covariances. In particular, f is chosen uniformly among $\{f_1, f_2, f_3, f_4\}$ if the scenario implies 4 distinct clusters, and among $\{f_1, f_2\}$ if the number of simulated clusters should be 2. The four generative models are defined as follows:

- $f_1 \colon [0, 21] \to \mathbb{R}$
$$x \mapsto \frac{a}{2}x^2 + bx + c$$

where $a \sim \mathcal{N}(0.05, 0.005^2)$, $b \sim \mathcal{N}(-10a_0, 4a_0^2)$ with $a_0 \overset{d}{=} a$, and $c \sim \mathcal{N}(2, 1)$;

- $f_2 \colon [0, 21] \to \mathbb{R}$
$$x \mapsto \frac{a}{3}x^3 - \frac{a(r_1 + r_2)}{2}x^2 + (ar_1r_2 + c)x + d$$

where $a \sim \mathcal{N}(-0.01, 0.001^2)$, $r_1 \sim \mathcal{N}(5, 1)$, $r_2 \sim \mathcal{N}(15, 1)$, $c \sim \mathcal{N}(6a_0, 4a_0^2)$ with $a_0 \overset{d}{=} a$, and $d \sim \mathcal{N}(3, 1)$;

- $f_3 \colon [0, 21] \to \mathbb{R}$

$$x \mapsto \frac{a}{3}x^3 - \frac{a(r_1 + r_2)}{2}x^2 + (ar_1r_2 + c)x + d$$

where $a \sim \mathcal{N}(0.01, 0.001^2)$, $r_1 \sim \mathcal{N}(5, 1)$, $r_2 \sim \mathcal{N}(15, 1)$, $c \sim \mathcal{N}(6a_0, 4a_0^2)$ with $a_0 \overset{d}{=} a$, and $d \sim \mathcal{N}(3, 1)$;

- $f_4 \colon [0, 21] \to \mathbb{R}$

$$x \mapsto \frac{a}{4}x^4 - \frac{a(r_1 + r_2 + r_3)}{3}x^3 + \frac{a\left(r_1r_2 + r_3(r_1 + r_2)\right)}{2}x^2 - (ar_1r_2r_3 + b)x + c$$

where $a \sim \mathcal{N}(5 \times 10^{-3}, (5 \times 10^{-5})^2)$, $r_1 \sim \mathcal{N}(2, 0.2^2)$, $r_2 \sim \mathcal{N}(10, 0.5^2)$, $r_3 \sim \mathcal{N}(18, 0.2^2)$, $b \sim \mathcal{U}([-0.05, 0.05])$, and $c \sim \mathcal{N}(2, 0.5^2)$.

3.1.1.2. *Scenario M2.* This approach to simulating means has been used in the simulation study focusing on the effect of time warping as well as stochastic block model (or simulation study 2). Studying time warping required introducing temporal shifts to simulated data, leading to the appearance of scalability issues in the context of the original simulation model. In order to solve this problem, time points have been changed to almost equidistant: $(t_1, \ldots, t_p) = (0.5, 3, 6, 9, 12, 15, 18, 21)$; and the fourth generative model has been changed from polynomial to sinusoidal. The remaining functions have only undergone minor adjustments, and the general procedure stays unchanged. The four generative models are presented below:

- $f_1 \colon [0, 21] \to \mathbb{R}$

$$x \mapsto \frac{a}{2}(x - s)^2 + b(x - s) + c$$

where $s \sim \mathcal{U}([-10, 10])$, $a \sim \mathcal{N}(0.05, 0.002^2)$, $b \sim \mathcal{N}(-11a_0, 4a_0^2)$ with $a_0 \overset{d}{=} a$, and $c \sim \mathcal{N}(2, 0.5^2)$;

- $f_2 \colon [0, 21] \to \mathbb{R}$

$$x \mapsto \frac{a}{3}(x - s)^3 - \frac{a(r_1 + r_2)}{2}(x - s)^2 + (ar_1r_2 + c)(x - s) + d$$

where $s \sim \mathcal{U}([-10, 10])$, $a \sim \mathcal{N}(-0.003, (10^{-5})^2)$, $r_1 \sim \mathcal{N}(8, 1)$, $r_2 \sim \mathcal{N}(12, 1)$, $c \sim \mathcal{N}(6a_0, 4a_0^2)$ with $a_0 \overset{d}{=} a$, and $d \sim \mathcal{N}(3, 0.5^2)$;

- $f_3 \colon [0, 21] \to \mathbb{R}$

$$x \mapsto \frac{a}{3}(x-s)^3 - \frac{a(r_1+r_2)}{2}(x-s)^2 + (ar_1r_2 + c)(x-s) + d$$

where $s \sim \mathcal{U}([-10, 10])$, $a \sim \mathcal{N}(0.003, (10^{-5})^2)$, $r_1 \sim \mathcal{N}(8, 1)$, $r_2 \sim \mathcal{N}(12, 1)$, $c \sim \mathcal{N}(6a_0, 4a_0^2)$ with $a_0 \overset{d}{=} a$, and $d \sim \mathcal{N}(2, 0.5^2)$;

- $f_4 \colon [0, 21] \to \mathbb{R}$

$$x \mapsto a \sin(b(x-s)) + c$$

where $s \sim \mathcal{U}([-7, 7])$, $a = |a_0|$ with $a_0 \sim \mathcal{N}(2, 1)$, $b \sim \mathcal{U}([0.3, 0.5])$, and $c \sim \mathcal{N}(2, 0.5^2)$.

It can be noticed that s is the time shift parameter, which is chosen to be uniformly distributed.

3.1.2. Scenarios with respect to covariance matrix.

3.1.2.1. *Independent case.* The elements of Ψ are chosen as follows:

$$\psi_{\Gamma_i\Gamma^t_{i'}} \overset{d}{=} \begin{cases} \mathcal{N}(0, 2^2) \text{ if } i = i' \\ 0 \text{ otherwise} \end{cases}.$$

This case corresponds to scenarios C1 and C2, the difference being the number of simulated clusters (4 and 2 respectively). All remaining scenarios simulate 2 clusters.

3.1.2.2. *Block-dependent case, low covariance (C3).* We denote cl_1 and cl_2 as sets containing simulated fold change labels belonging to cluster 1 and 2 respectively. The matrix Ψ is defined as a squared and scaled matrix $\Psi' = (\Psi'_{ii'})_{(i,i') \in \{1,\dots,n_e\}^2}$, where:

$$\Psi'_{ii'} = \begin{bmatrix} \psi'_{\Gamma_i\Gamma^{t_1}_{i'}} & & 0 \\ & \ddots & \\ 0 & & \psi'_{\Gamma_i\Gamma^{t_p}_{i'}} \end{bmatrix}, \text{ such that } \psi'_{\Gamma_i\Gamma^t_{i'}} \overset{d}{=} \begin{cases} |\mathcal{N}(0, 2^2)| \text{ if } \mathbb{1}_{cl_1}(i) = \mathbb{1}_{cl_1}(i') \\ 0 \text{ otherwise} \end{cases}.$$

The final covariance matrix is then $\Psi = \dfrac{\Psi'^2}{\max\{\psi'_{\Gamma_i\Gamma^t_{i'}} | i \neq i', t \in \{t_1, \dots, t_p\}\}}$.

3.1.2.3. *Block-dependent case, high covariance (C4).* The matrix Ψ takes the same form as in the previous case, but with a different scaling: $\Psi = \dfrac{\Psi'^2}{20}$.

3.1.2.4. *Positive vs. negative case, low covariance (C5).* We assume here without loss of generality that cluster labels are ordered in the way that entity labels in $cl_1 =$

$\{1, \ldots, \lfloor \frac{n_e}{2} \rfloor\}$ correspond to cluster 1, and labels in $cl_2 = \{\lceil \frac{n_e}{2} \rceil, \ldots, n_e\}$ correspond to cluster 2. Similarly to the previous cases, the matrix Ψ is defined as a squared and scaled , with the elements of the latter chosen as follows:

$$\psi_{\Gamma_i \Gamma_{i'}^t} \stackrel{d}{=} \begin{cases} \mathcal{U}([0,1]) \text{ if } \mathbb{1}_{cl_1}(i) = \mathbb{1}_{cl_1}(i') = 1 \\ \mathcal{U}([-1,0]) \text{ if } \mathbb{1}_{cl_2}(i) = \mathbb{1}_{cl_2}(i') = 1 \\ 0 \text{ otherwise} \end{cases} .$$

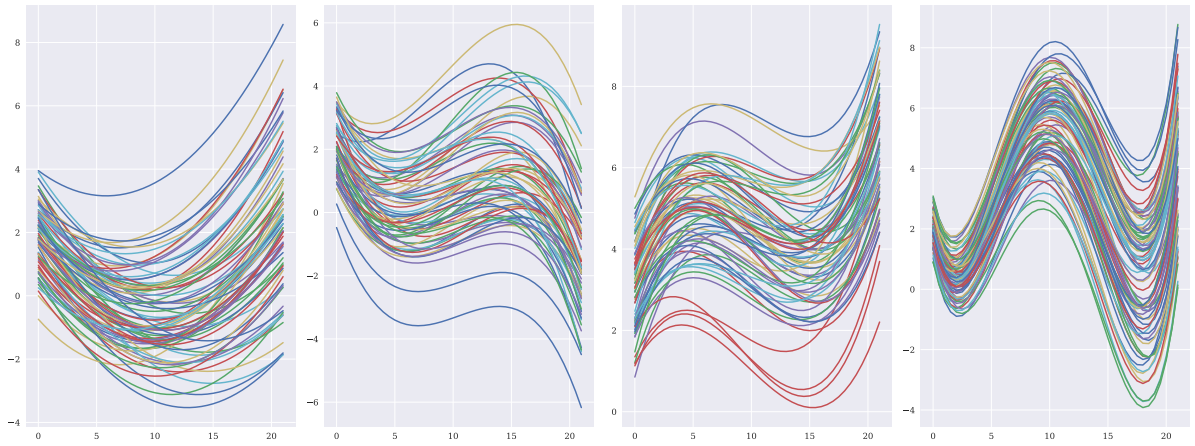
The final covariance matrix is then $\Psi = \frac{\Psi'^2}{100}$.

3.1.2.5. *Positive vs. negative case, high covariance (C6).* The matrix Ψ takes the same form as in the previous case, but with a different scaling: $\Psi = \frac{\Psi'^2}{50}$.

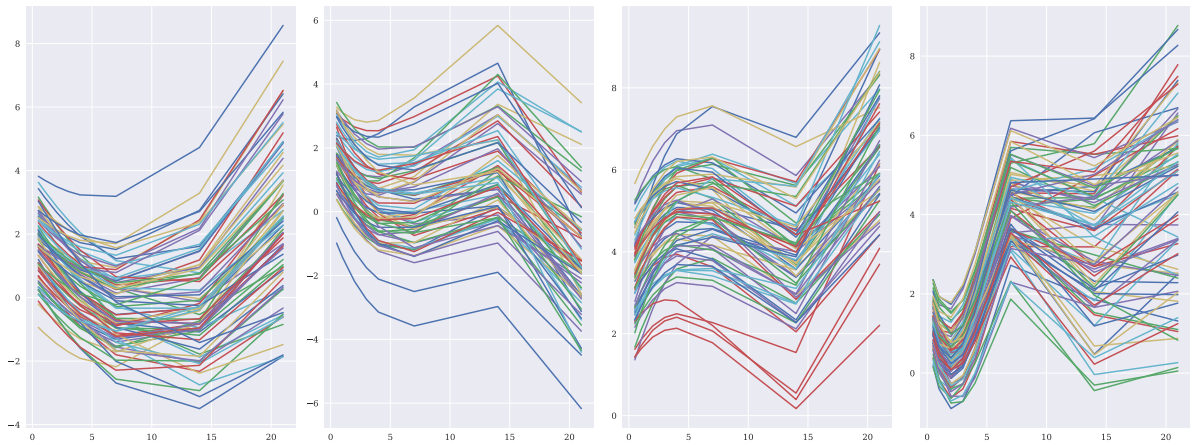
3.2. Results of simulation studies

3.2.1. Study 1. In order to validate our choice of $\widehat{\mathbf{d_2^2}}$ for the distance and k-medoids for the clustering algorithm, we compared their success with other candidates in different simulation scenarios reflecting different properties of the data. Clustering with k-means algorithm based on Wasserstein distance is chosen as the main competitor. In addition, Hellinger distance and hierarchical clustering were considered. The means of the fold changes were simulated according to scenario M1 and are presented in Figure 3.2.1. The four approaches were applied 10 times for each simulated set of fold changes in order to account for the variability arising from different random initializations[1] of all of them, with the exception of hierarchical clustering which is entirely deterministic. The adjusted rand index (ARI) and the V-measure index were chosen as metrics to quantify the success of clustering. The ARI is the corrected-for-chance version of the Rand index, the latter being defined as $RI = \frac{TP+TN}{TP+FP+FN+TN}$, where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives (see Chacón and Rastrojo (2023) for details). The V-measure is an average of homogeneity and completeness: homogeneity measuring the extent to which the individuals in a cluster are similar, and completeness measuring the extent to which similar individuals are put together by the algorithm (both are calculated using the Shannon's entropy, see Rosenberg and Hirschberg (2007) for details).

[1] the random initializations were the same for all methods to



(a)



(b)

Figure (3.2.1) Simulation study 1. a) Functional representation of simulated means. b) Multivariate representation of simulated means. Columns represent clusters.

The results of the study are summarized in Figure 3.2.2. Only the means of ARI and V-measure of the obtained clusters are presented since the standard deviation turned out to be negligible (the largest one had the order of 10^{-4}). Firstly, a simulation with all 4 clusters was performed where only the independent scenario was considered, i.e. $\mathrm{P}_{\Gamma_i\Gamma_{i'}} = 0_{p,p}$ for any considered fold changes pair with joint distribution $\begin{bmatrix} \widehat{\Gamma}_i \\ \widehat{\Gamma}_{i'} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \Gamma_i \\ \Gamma_{i'} \end{bmatrix}, \begin{bmatrix} \Sigma_{\Gamma_i} & \mathrm{P}_{\Gamma_i\Gamma_{i'}} \\ \mathrm{P}^{\intercal}_{\Gamma_i\Gamma_{i'}} & \Sigma_{\Gamma_{i'}} \end{bmatrix} \right)$ where $i \neq i'$. This implies that, since the main advantage of $\widehat{\mathbf{d}_2^2}$ with respect to Wasserstein distance is the ability to take K into account, any differences between the results obtained in the independent scenario would be due to the choice of algorithm rather than distance. It can be observed that $\widehat{\mathbf{d}_2^2}$ k-medoids generally produce better results in terms of both metrics compared to

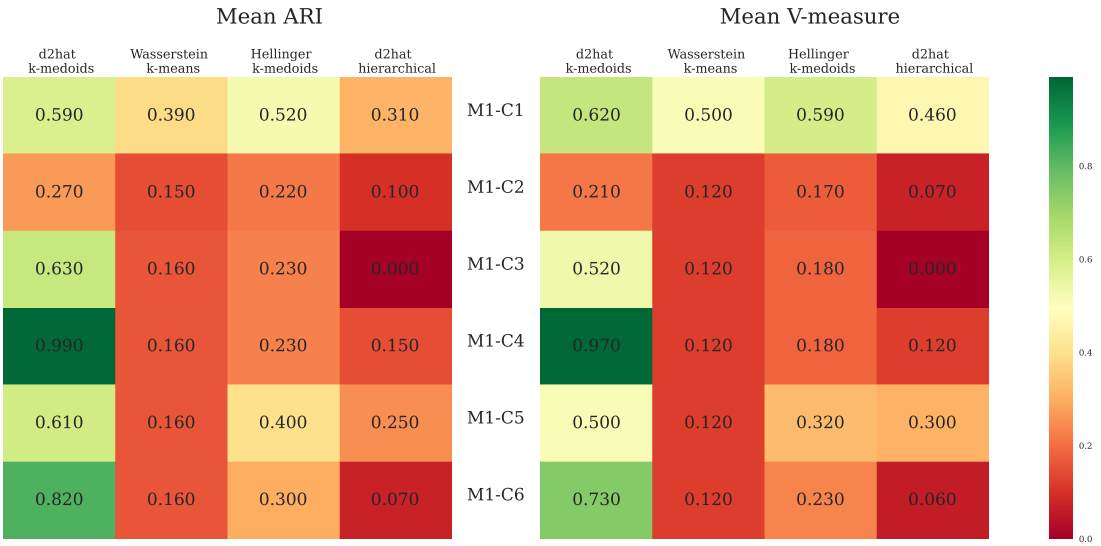
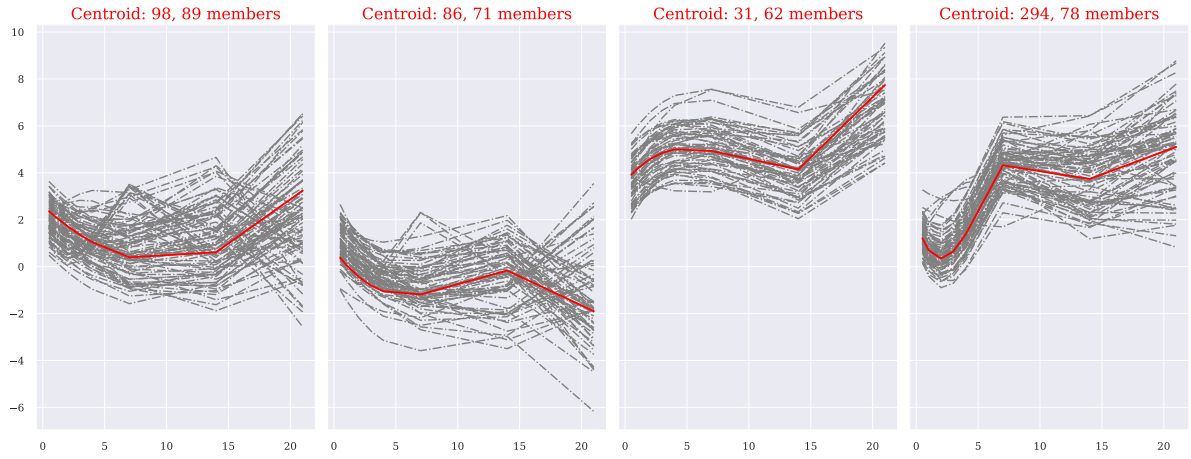


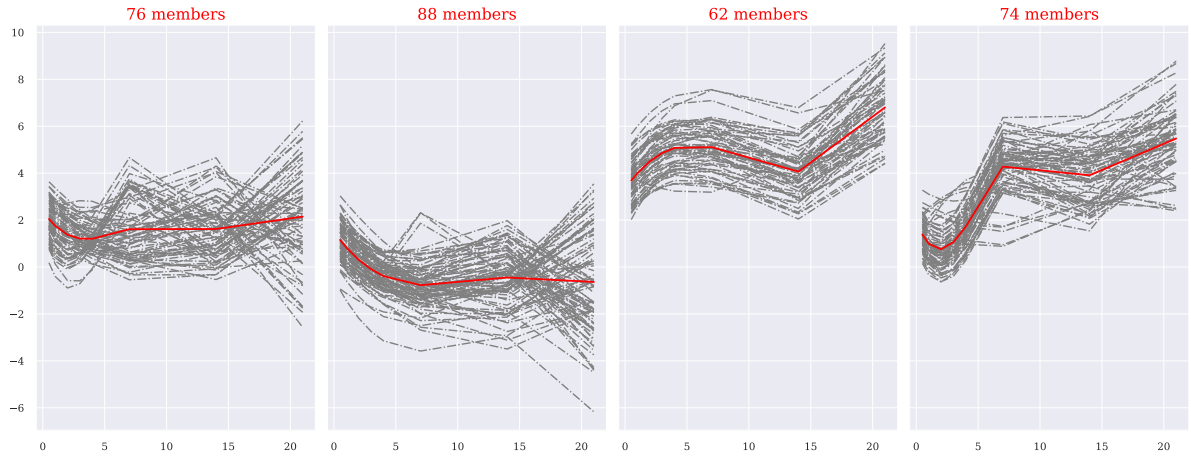
Figure (3.2.2) Results of the simulation study comparing distances and clustering algorithms.

Wasserstein k-means and hierarchical clustering, which means that k-medoids algorithm is more adapted to perform clustering of fold changes. An illustrative example of clustering of simulated fold changes by $\widehat{\mathbf{d}_2^2}$ k-medoids and Wasserstein k-means is presented in Figure 3.2.3 (contingency tables in Table 3.2.1). This example demonstrates the specific differences in outcomes of applying k-medoids and k-means to cluster fold changes in this context. It can be noted that the first two clusters are particularly often confused by both algorithms, but despite a big amount of fold changes in these clusters being wrongly classified, k-medoids manage to choose centroids that are highly representative of the average behavior in the corresponding clusters. Meanwhile, a severe deformation of the centroids can be observed in case of k-means, which may lead to mistakes in interpreting the average behavior of the fold changes in a given cluster. Hellinger distance with k-medoids algorithm appears not as effective but still comparable to the $\widehat{\mathbf{d}_2^2}$ k-medoids, which proves once again that it is beneficial to use k-medoids regardless of the distance choice.

Secondly, with the goal of studying the effect of taking the correlations between the entities into account and thus validating the choice of a distance, two simulation scenarios with respect to cross-covariances were designed. In order to be able to carry out both scenarios, simulations had to be restricted to two clusters out of four. The two most often confused clusters mentioned above were chosen. The first scenario produces block-dependent fold changes, and the second produces positively or negatively correlated fold changes depending on the cluster. Both scenarios were simulated in two configurations, namely with higher or lower overall correlation levels, and were



(a)



(b)

Figure (3.2.3) Simulation study 1. a) An example of clusters obtained with $\widehat{\mathbf{d}_2^2}$ k-medoids. b) An example of clusters obtained with Wasserstein k-means.

Clustering

Simulation	1	2	3	4
1	17	8	0	0
2	8	14	0	0
3	2	0	20	1
4	3	1	0	24

(a) $\widehat{\mathbf{d}_2^2}$ *k-medoids*

Clustering

Simulation	1	2	3	4
1	10	14	0	2
2	8	14	0	0
3	1	0	20	2
4	6	1	0	21

(b) Wasserstein k-means

Table (3.2.1) Contingency tables (%) for $\widehat{\mathbf{d}_2^2}$-based k-medoids and Wasserstein k-means clustering performed on the simulated data.

compared to the baseline scenario with independent fold changes. Figure 3.2.2 shows that the more important the correlations between the fold changes within one cluster, the more successful the clustering obtained with $\widehat{\mathbf{d}_2^2}$-based k-medoids. The results produced by Wasserstein k-means are invariant to the level of cross-covariances,

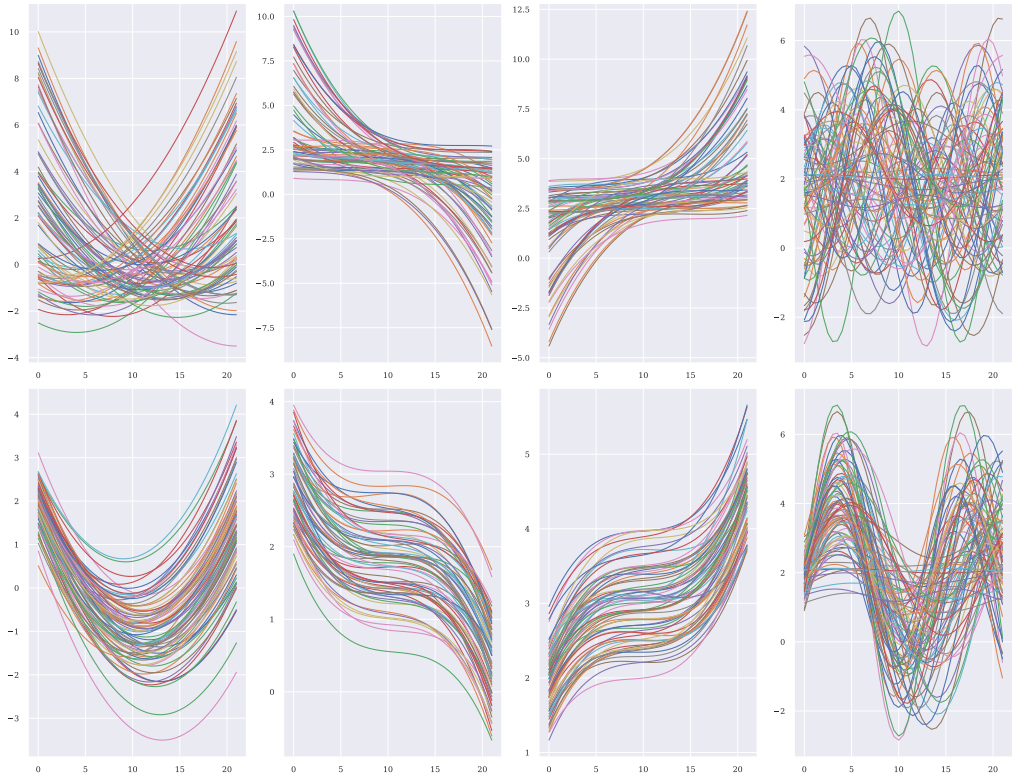
which implies that it is not adapted for the problems where the latter are to be taken into account. Hellinger distance-based k-medoids approach shows better results than Wasserstein k-means but it does not appear to be very sensitive to correlations (in fact, whatever the difference between scenarios can be observed is most likely due to its sensitivity to standard deviation rather than correlations). Finally, hierarchical clustering shows poor clustering results in all cases: although there seems to be no obvious pattern in its response to changes in cross-covariances, it is clearly not adapted for the problem at hand.

	ARI						**V-measure**					
	$\widehat{\mathbf{d}_2^2}$ k-medoids	$\widehat{\mathbf{diss}_s}$ k-medoids	SBM based on $\widehat{\mathbf{diss}_s}$ k-medoids	Random SBM	$\mathcal{D}$-UMAP + k-means	$\mathcal{OWD}$-UMAP + k-means	$\widehat{\mathbf{d}_2^2}$ k-medoids	$\widehat{\mathbf{diss}_s}$ k-medoids	SBM based on $\widehat{\mathbf{d}_2^2}$ k-medoids	Random SBM	$\mathcal{D}$-UMAP + k-means	$\mathcal{OWD}$-UMAP + k-means
Mean	0.22	0.61	0.59	0.38	0.57	0.88	0.39	0.67	0.61	0.43	0.68	0.88
Standard deviation	0.03	0	0	0.1	0.01	0.01	0.04	0	0	0.1	0.01	0.01
Time warping	✗	✔	✔	✔	✗	✔	✗	✔	✔	✔	✗	✔

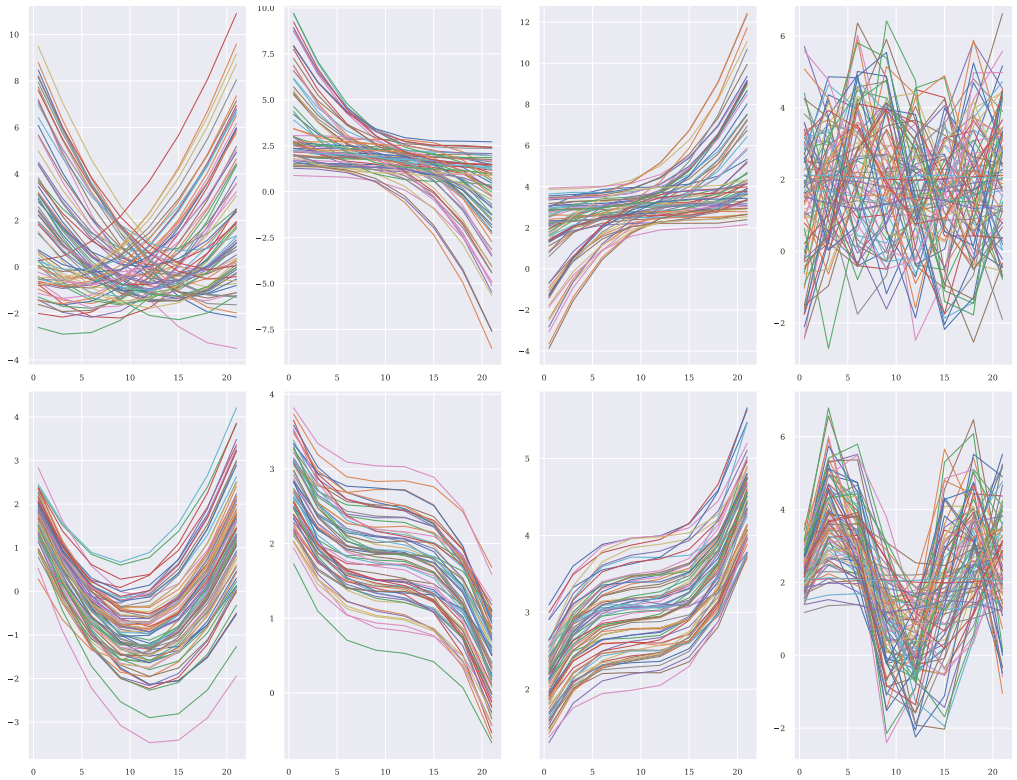
Table (3.2.2) Results of the simulation study of the effect of incorporating time warping and stochastic block model into clustering.

3.2.2. Study 2. Similarly to the first study, 10 clustering initializations were performed for six approaches: $\widehat{\mathbf{d}_2^2}$-based k-medoids without time warping, $\widehat{\mathbf{diss}_s}$-based k-medoids with time warping, stochastic block model inference with parameters initialized from the results obtained with the latter (as described in Section 2.5), stochastic block model with parameters initialized randomly in a non-constrained parameter space, k-means clustering of the UMAP projection coordinates of the matrix $\mathcal{D}$ without time warping, and k-means clustering of the UMAP projection coordinates of the matrix $\mathcal{OWD}$ with time warping. The means of the fold changes were simulated according to scenario M2 and are presented in Figure 3.2.4.

The results in the form of means and standard deviations of ARI and V-measure for every approach are presented in Table 3.2.2. Firstly, the observed important difference between the means of the results obtained with and without time warping indicates the importance of the latter when dealing with the data associated with remarkable time shifts that need to be accounted for. A special example of what could go wrong



(a)



(b)

Figure (3.2.4) Simulation study 2. a) Functional representation of simulated means. Top: unaligned means. Bottom: aligned means. b) Multivariate representation of simulated means. Top: unaligned means. Bottom: aligned means. Columns represent clusters.

Simulation \ Clustering	1	2	3	4
1	6	12	1	7
2	7	14	0	1
3	0	0	8	15
4	0	9	0	19

(a) without time warping

Simulation \ Clustering	1	2	3	4
1	25	1	0	0
2	0	21	1	0
3	0	2	21	0
4	0	13	1	15

(b) with time warping

Table (3.2.3) Contingency tables (%) for $\widehat{\mathbf{d}_2^2}$-based k-medoids without time warping and $\widehat{\mathbf{diss}}_\mathbf{s}$-based k-medoids with time warping performed on the simulated data.

if alignment is not incorporated into clustering is presented in Figure 3.2.5 (contingency tables in Table 3.2.3). One can notice that the approach without time warping managed to identify only two behavior types: the monotonously decreasing and the monotonously increasing. The oscillating character of the fourth cluster is especially indistinguishable in this case, whereas the approach with time warping managed to identify all four behavior types.

Secondly, the difference between the means obtained for stochastic block model initialized randomly and from $\widehat{\mathbf{diss}}_\mathbf{s}$-based k-medoids clustering, with a significant standard deviation observed in case of the former as opposed to its absence for the latter, shows that our approach to model parameter initialization serves as a way to avoid getting stuck in local optima, which are expected to be even more present in the real data. The example presented in Figure 3.2.6 (contingency tables in Table 3.2.4) illustrates how bad parameter initialization can guide the model inference in a completely wrong direction. Nevertheless, it appears that the clusters identification with stochastic block model may be less efficient than that with $\widehat{\mathbf{diss}}_\mathbf{s}$-based k-medoids algorithm, which means that the corresponding results obtained for the real data should be carefully compared.

Lastly, the approach that consists in first performing a UMAP projection of the dissimilarity matrix and then applying any classical clustering (in this case k-means) to the coordinates, that will be used in the next section to compare our approach to one of the state-of-the-art alternatives, is also considered in this simulation study. Here we observe the same tendency as for k-medoids with respect to adding time warping to clustering for data with horizontal shifts (Figure 3.2.7, contingency tables in Table 3.2.5). It also appears that this method performs remarkably on this simulated dataset. Having not observed the same effect on real data, it can be concluded that the choice of

Simulation \ Clustering	1	2	3	4
1	25	1	0	0
2	0	20	0	2
3	0	1	20	2
4	0	10	1	16

(a) $\widehat{\mathbf{diss}}_{\mathbf{s}}$ *k-medoids-based SBM*

Simulation \ Clustering	1	2	3	4
1	19	1	6	0
2	0	19	1	2
3	0	14	0	9
4	0	11	0	17

(b) random SBM

Table (3.2.4) Contingency tables (%) for stochastic block model inference based on $\widehat{\mathbf{diss}}_{\mathbf{s}}$ *k-medoids clustering and on a random initialization performed on the simulated data.*

Clustering \ Simulation	1	2	3	4
1	19	76	0	0
2	0	8	0	14
3	0	0	23	0
4	0	0	0	28

(a) without time warping

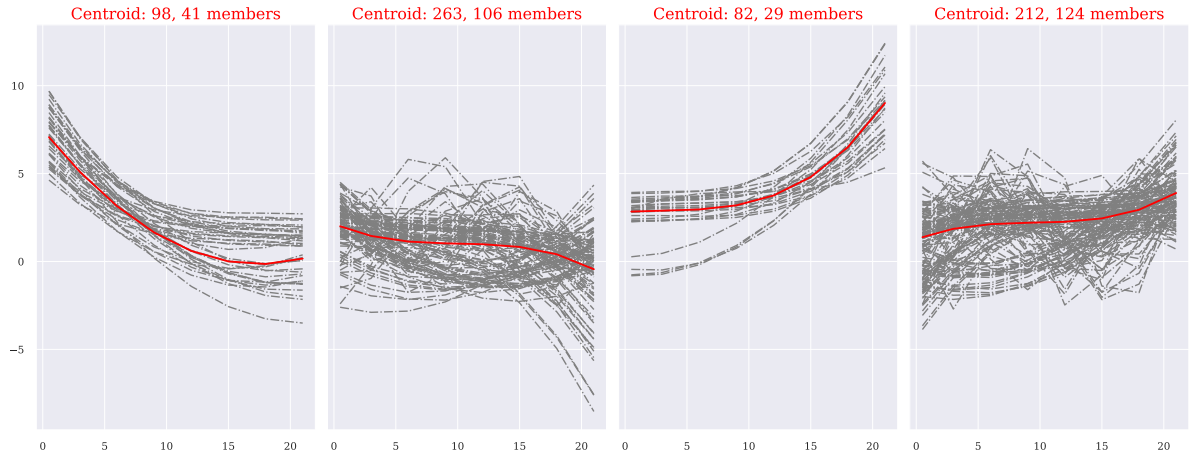
Clustering \ Simulation	1	2	3	4
1	26	1	0	0
2	0	22	0	0
3	0	0	23	0
4	0	5	0	22

(b) with time warping

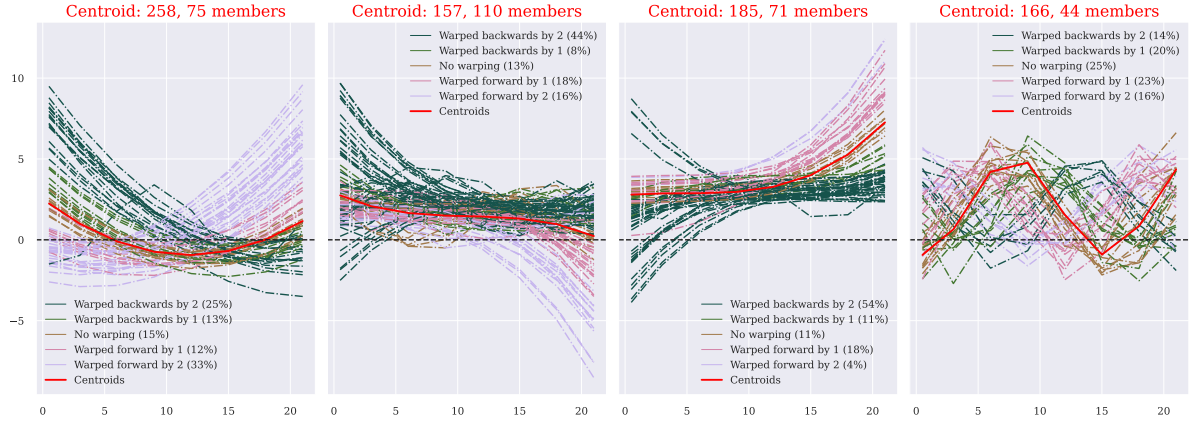
Table (3.2.5) Contingency tables (%) for k-means clustering of the UMAP projection of the the matrix $\mathcal{D}$ *without time warping and* $\mathcal{OWD}$ *with time warping performed on the simulated data.*

the approach to clustering out of the three presented here should be made depending on the dataset, as they may perform very differently from one dataset to another.

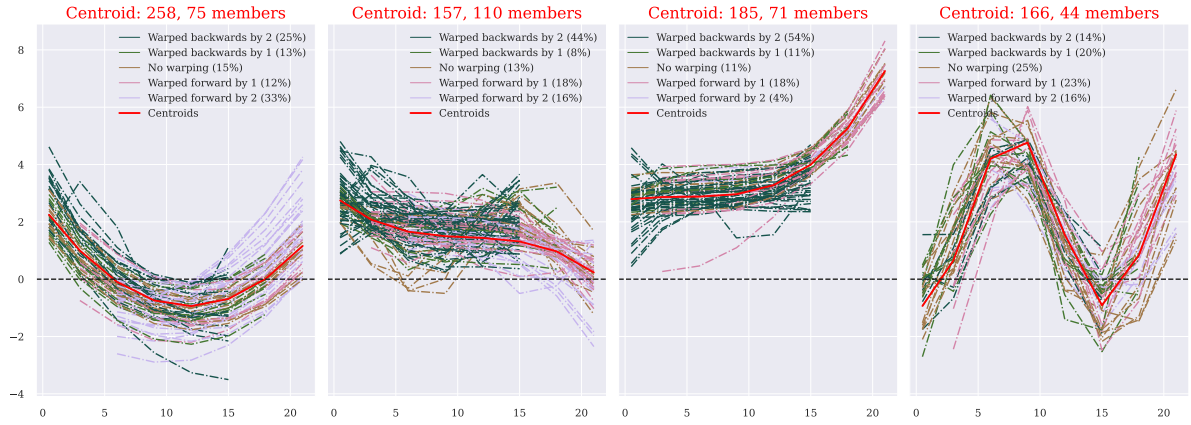
Finally, we considered two approaches that could be regarded as logical state-of-the-art alternatives to our method, and tested them on the fold changes with time shifts simulated in the framework of the second simulation study. The first one by Kazlauskaite et al. (2019) consists of a Gaussian process-based model (GPLVM) that learns temporal sequences' generative model as well as their alignments. Unlike our approach, it does not produce explicit cluster labels or number of clusters, instead it summarizes the learned information by producing a projection of the aligned sequences on a two-dimensional manifold (heatmap). The clusters can thus be obtained by applying any clustering procedure for two-dimensional data on the coordinates. In order to compare the efficacy of this approach to ours on omic fold changes-like data from simulation study 2, we first compared the obtained heatmap to a UMAP projection of the $\mathcal{OWD}$ matrix obtained with dissimilarity $\widehat{\mathbf{diss}}_{\mathbf{s}}$ (Figure 3.2.8). By examining the cluster separation on both figures, where each color corresponds to one of the four simulated clusters, one can notice that the clusters can be much better distinguished



(a)

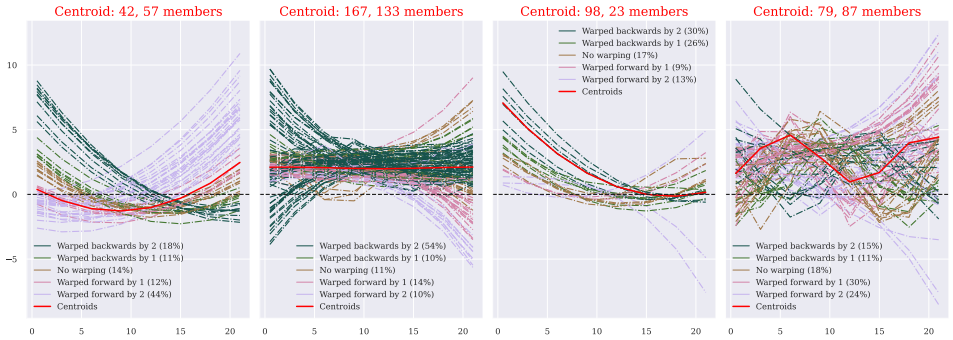


(b)

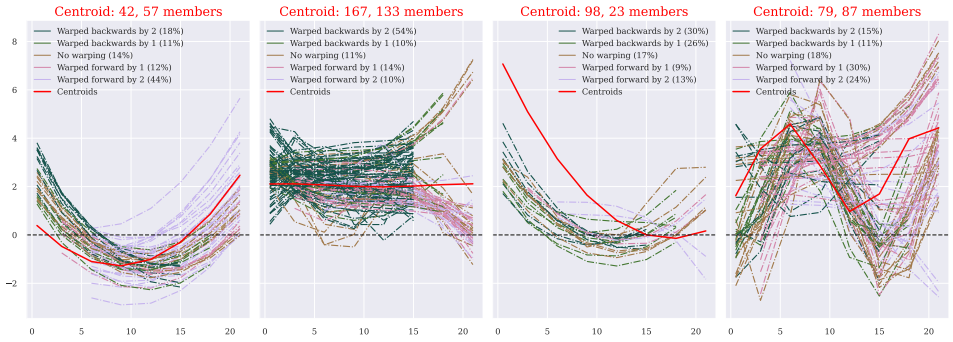


(c)

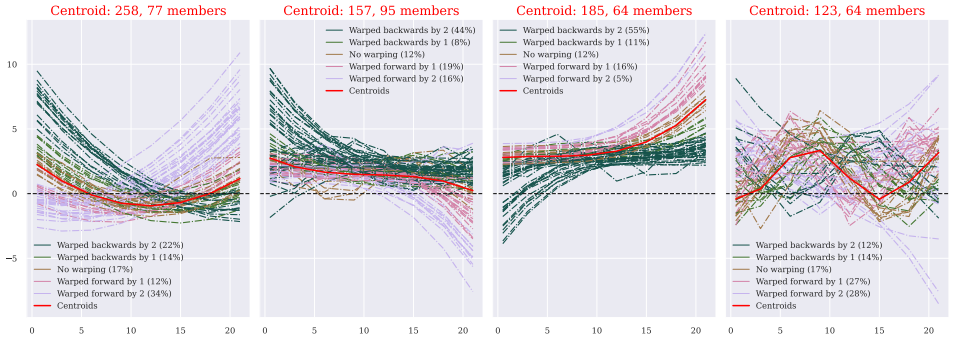
Figure (3.2.5) Results of the simulation study 2: the effect of time warping. a) An example of clusters obtained with $\widehat{\mathbf{d}_2^2}$ k-medoids without time warping. b) An example of clusters obtained with $\widehat{\mathbf{diss}}_\mathbf{s}$ k-medoids with time warping (unaligned fold changes). c) An example of clusters obtained with $\widehat{\mathbf{diss}}_\mathbf{s}$ k-medoids with time warping (aligned fold changes).



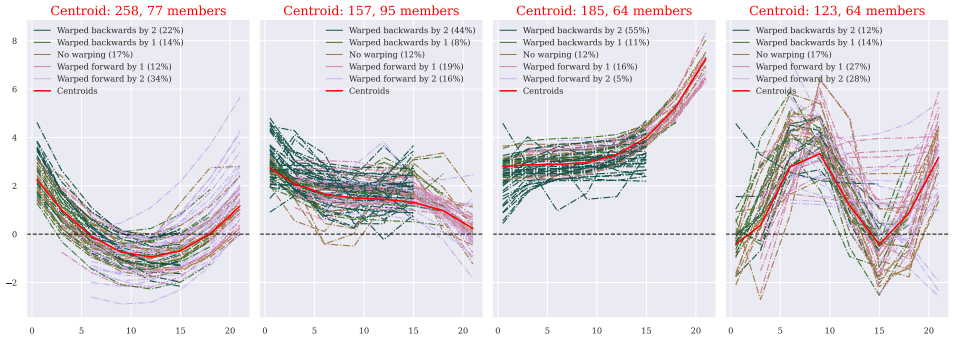
(a)



(b)

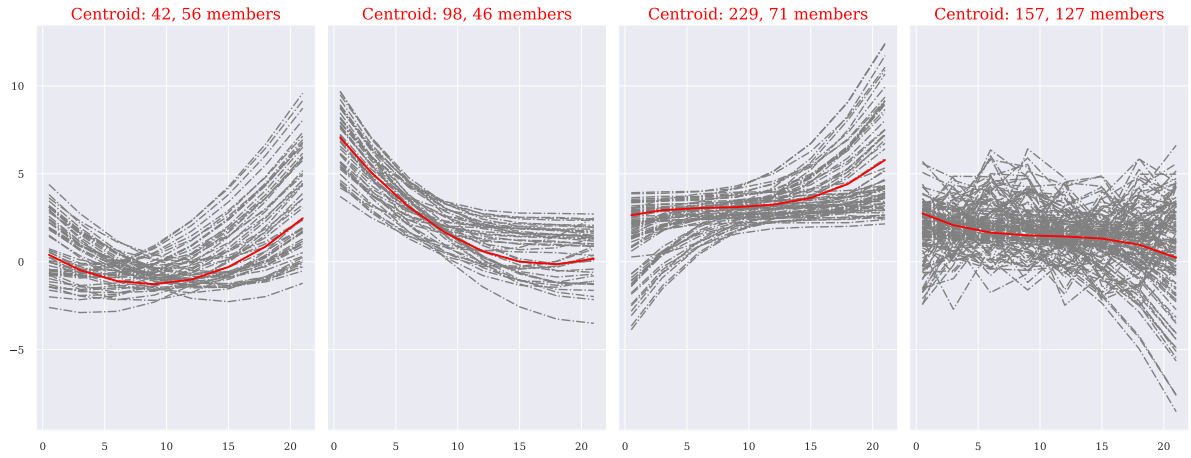


(c)

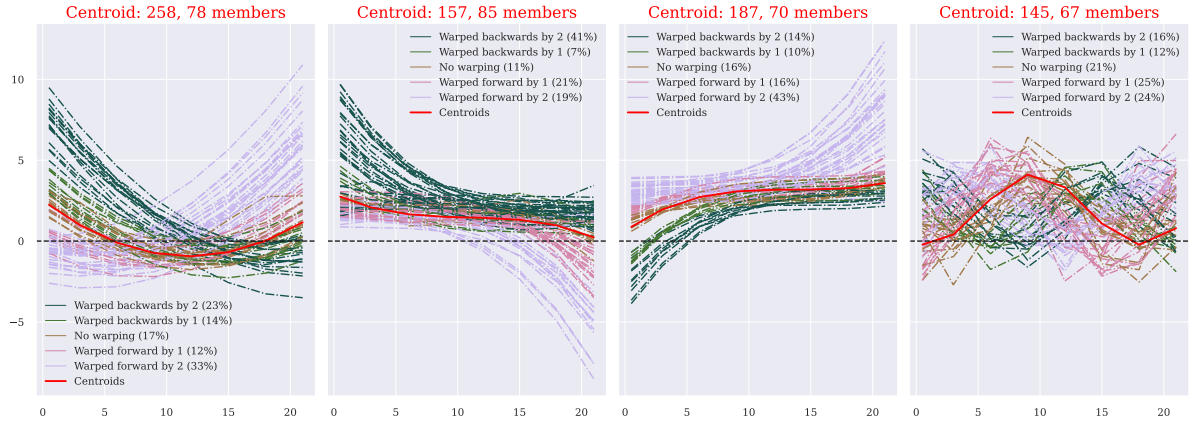


(d)

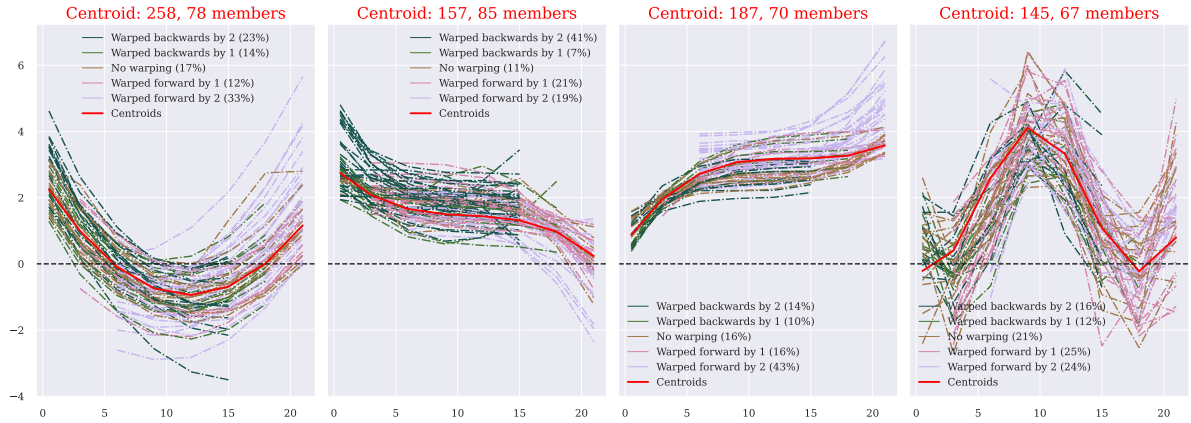
Figure (3.2.6) Results of the simulation study 2: the effect of stochastic block model inference. a) An example of blocks identified based on a random initialization (unaligned). b) An example of blocks identified based on a random initialization (aligned). c) An example of blocks identified based on $\widehat{\mathbf{diss}}_{\mathbf{s}}$ *k-medoids initialization (unaligned). d) c) An example of blocks identified based on* $\widehat{\mathbf{diss}}_{\mathbf{s}}$ *k-medoids initialization (aligned).*



(a)



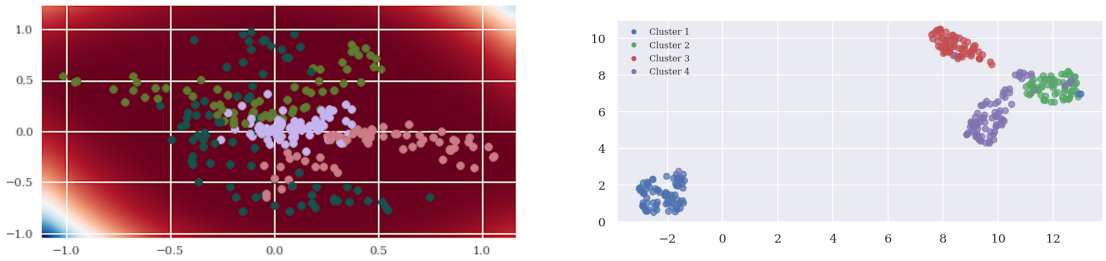
(b)



(c)

Figure (3.2.7) Results of the simulation study 2: the effect of clustering based on the UMAP projection of the distance matrix. a) Without time warping, projection of the D-matrix. b) With time warping, projection of the OW D-matrix (unaligned). c) With time warping, projection of the OW D-matrix (aligned).

in the case of $\widehat{\mathbf{diss}}_{\mathbf{s}}$-based approach. Additionally, clustering performed on the coordinates confirm this conclusion: ARI for GPLVM measures 0.14 with spectral clustering and 0.24 with k-means, whereas for $\widehat{\mathbf{diss}}_{\mathbf{s}}$-based approach both clustering algorithms perform at 0.83. We note here that the method proposed by Kazlauskaite et al. (2019) was originally designed for data rather different from ours, namely it is expected that the number of time points is significantly larger than that of sequences. In the omic datasets that we work with, the opposite is observed, which also leads to GPLVM taking much longer to perform the computations (several hours compared to less than a minute with a thousand repetitions for our method).



(a) *(b)*

Figure (3.2.8) Results of the simulation study 2: a) projection of fold change coordinates on a manifold of dimension 2 produced by GPLVM (Gaussian processes, Kazlauskaite et al. 2019), b) UMAP projection of the OWD matrix.

The second considered alternative method was proposed by Heerah et al. (2021) and consists in modeling pairs of time series with an auto-regressive model and performing a statistical test in order to assess whether they Granger-cause one another. The implementation is available through the R package 'irg'. This approach is not designed for clustering, it can however be used to validate the part of the procedure dedicated to alignment by comparing optimal time warps of every pair identified by our method to the Granger-causality selected as significant by 'irg'. Despite the differences in the data expected by the two methods, we observe a high correspondence in the identified causalities (62% or 79% depending on whether we consider bidirectional connections or not, which is not obvious given the design of simulated data). This result supports the idea that time warping can serve as means of identification of causal relationships between entities, potentially leading to underlying biological pathways.

CHAPTER 4

NETWORK INFERENCE FOR KEY FEATURES VISUALIZATION

In this chapter, we present two tools for the visualization of the objects encoding the key features of considered dataset described in Chapter 2: microscopic and mesoscopic fold change networks. Here we introduce the mathematical formalism behind these network representations. The networks provide different levels of insight into the data in question, their analysis and interpretation are discussed in Chapter 5.

4.1. Microscopic network

A classical version of the network of omic fold changes (also referred to as microscopic to distinguish from the mesoscopic network representation discribed in the next section) is modeled by a random graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the node set, and $\mathcal{E}$ is the edge set. The set $\mathcal{N}$ consists of biological entities (ex. genes) and is of size n_e, whereas the set $\mathcal{E}$ represents connections between these entities. The graph is described by the binary adjacency matrix $X = (X_{ii'})_{(i,i')\in\{1,\dots,n_e\}^2}$ such that $X_{ii'} = 1$ denotes the existence of an edge between entities i and i', and 0 denotes the absence of one.

The adjacency matrix for omic fold changes is constructed from a similarity measure that is formulated based on the dissimilarity between fold changes estimators. In its simplest form it is calculated based on the Optimal Warping Dissimilarity matrix:

$$Sim\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right) = \frac{\max\limits_{(a,b)\in\{1,\dots,n_e\}^2} \mathcal{OWD}_{ab} - \mathcal{OWD}_{ii'}}{\max\limits_{(a,b)\in\{1,\dots,n_e\}^2} \mathcal{OWD}_{ab}}.$$

It can also be reformulated in a minimax form, which makes apparent its link with the dissimilarity:

$$Sim\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right) = \min_{s\in\mathcal{S}} \left[\frac{\max\limits_{(a,b)\in\{1,\dots,n_e\}^2} \widehat{\mathbf{diss_s}}\left(\widehat{\Gamma_a \circ \mathcal{W}_s}, \widehat{\Gamma_b \circ \mathcal{W}_s}\right) - \widehat{\mathbf{diss_s}}\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right)}{\widehat{\mathbf{diss_s}}\left(\widehat{\Gamma_a \circ \mathcal{W}_s}, \widehat{\Gamma_b \circ \mathcal{W}_s}\right)} \right].$$

It can be noted that the formulation of the similarity measure in the case without time warping is achieved through the same definition by considering the degenerate warping space $\mathcal{S} = \{0\}$. In all cases, this similarity measure is bounded by 0 from below and 1 from above, the value in case of $i = i'$ being 1 and in case of $(i, i') = \underset{(a,b)\in\{1,\dots,n_e\}^2}{\arg\max} \widehat{\mathbf{d_2^2}}\left(\widehat{\Gamma}_a, \widehat{\Gamma}_b\right)$ being 0, which makes the measure more easily interpretable and comparable for different omic datasets.

Let us denote a set of all unique entity pairs as $pairs = \{(i, i') \in \{1, \dots, n_e\}^2 | i < i'\}$. We define the empirical cumulative distribution function $\widehat{F}_{n_{pairs}}$ of similarity over the observed fold change pairs as follows:

$$\widehat{F}_{n_{pairs}}(x) = \frac{1}{n_{pairs}} \sum_{(i,i')\in pairs} \mathbb{1}_{Sim\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right)\leq x},$$

where $n_{pairs} = (n_e^2 - n_e)/2$, and $x \in [0, 1]$ is a similarity level. In other words, $\widehat{F}_{n_{pairs}}(x)$ represents the proportion of fold change pairs less or as similar as x. For $\mathbf{p} \in [0, 1]$, representing sparsity level of the network, an empirical $\mathbf{p}$-quantile is constructed as follows:

$$\mathbf{q} = \inf\{x : \widehat{F}_{n_{pairs}}(x) \geq \mathbf{p}\}.$$

In order to define the elements of the adjacency matrix $X = (X_{ii'})_{(i,i')\in\{1,\dots,n_e\}^2}$, we distinguish two cases. If $\mathcal{G}$ is undirected, which is used in particular to infer stochastic block model in Section 2.5, its elements are equal to

$$X_{ii'} = \mathbb{1}_{Sim\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right)\geq \mathbf{q}}. \tag{4.1.1}$$

This definition implies that all entities that are at least as similar as the quantile corresponding to the chosen network sparsity level will be considered as connected, and not connected otherwise. In case if $\mathcal{G}$ is directed, the elements are defined as

$$X_{ii'} = \mathbb{1}_{Sim\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right)\geq \mathbf{q}} \times \mathbb{1}_{\mathcal{OW}_{ii'}\geq 0}. \tag{4.1.2}$$

The additional term means that only predictive and simultaneous relations with respect to pairwise warps from $\mathcal{OW}$ remain in the directed case. Matrix $\mathcal{OW}$ can be

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	0	0	1	0	0	0	0	0	0	1	1
2	0	0	1	0	0	0	0	0	0	0	1
3	0	0	0	0	1	1	0	0	0	0	0
4	0	0	1	0	0	0	1	1	1	0	0
5	0	0	0	0	0	1	0	0	1	0	1
6	0	0	0	0	0	0	0	0	0	0	1
7	0	0	0	1	0	0	0	0	1	0	1
8	0	0	1	0	0	0	0	0	0	1	1
9	1	0	1	0	1	0	1	1	0	1	0
10	0	0	0	0	0	0	0	0	1	0	1
11	0	0	1	0	0	0	1	1	1	0	0

Table (4.1.1) Example of an adjacency matrix for a directed network.

naturally used in this context since it is anti-symmetric, as stated in Proposition 2.4.1. The adjacency matrix is thus no longer symmetric: the symmetry is preserved for non-significant (absent) connections and simultaneous (bidirectional) connections, the remaining connections being anti-symmetric.

In the package *ScanOFC*, we propose a visualization for the microscopic fold changes network. Figure 4.1.1 illustrates an example of such network, calculated based on the adjacency matrix, presented in Table 4.1.1. In this toy example there are 11 entities, labeled with numbers from 1 to 11. We suppose that the entities are distributed in 3 clusters: $cluster_1 = \{1, 2, 3\}$, $cluster_2 = \{4, 5, 6\}$ and $cluster_3 = \{7, 8, 9, 10, 11\}$, with the corresponding centroids $C_1 = 1$, $C_2 = 4$ and $C_3 = 7$. It can be noted that the network representation has a block structure, with blocks corresponding to clusters, and with the centroid nodes bigger than the others. The visualization also distinguishes between the two types of connections: green edges correspond to predictive connections, whereas gray to simultaneous ones.

4.2. Mesoscopic network

This section introduces a type of network representation that combines key features obtained with both fold changes clustering and network inference described above. Thus, the notations used here will be a combination of those used in Section 2.4 on clustering and those introduced in the previous section. The mesoscopic network is modeled by a random graph $\mathcal{G}_\mathcal{M} = (\mathcal{N}_\mathcal{M}, \mathcal{E}_\mathcal{M})$, where $\mathcal{N}_\mathcal{M}$ is the node set, and $\mathcal{E}_\mathcal{M}$ is the edge set. As in the microscopic case, the node set $\mathcal{N}_\mathcal{M}$ contains biological entities,

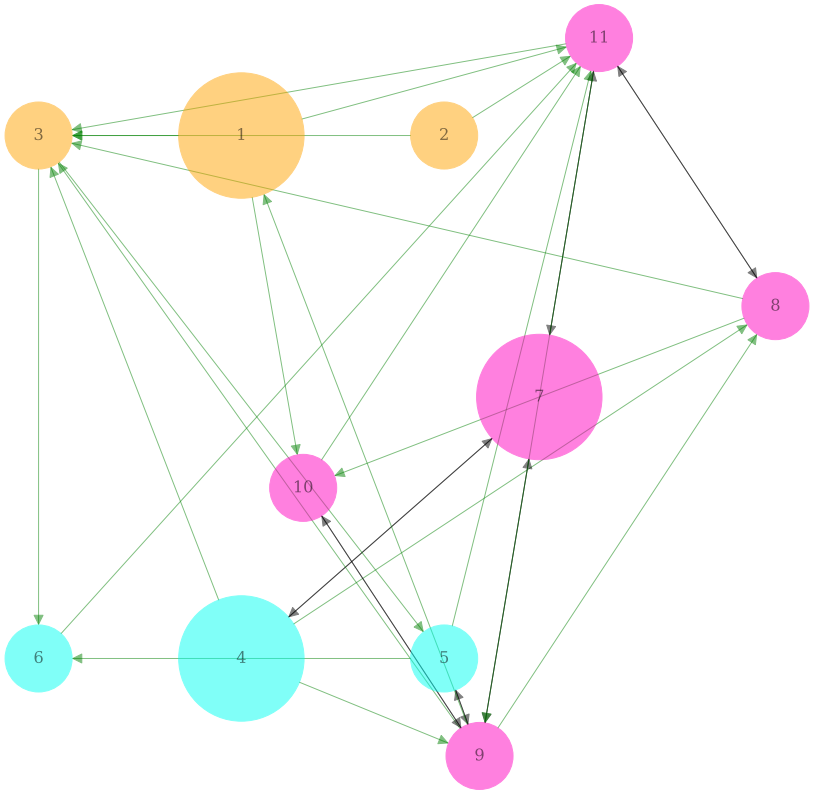


Figure (4.1.1) A visualization of the microscopic network based on the adjacency matrix presented in Table 4.1.1.

but in this case it is of size K, corresponding to the number of clusters, labeled by centroids $C = (C_1, \dots, C_K)$. The remaining information on clustering is contained in the associated node set weight function, defined as follows:

$$\begin{aligned} w_{\mathcal{N}} \colon \mathcal{N}_{\mathcal{M}} &\to \mathbb{N} \\ C_k &\mapsto \#cluster_k \\ &= \#\{i \in \{1, \dots, n_e\} | Cl_i = k\} \\ &= \sum_{i \in \{1, \dots, n_e\}} \mathbb{1}_{Cl_i = k}. \end{aligned}$$

The edge set contains the main information on connections between the clusters in the form of distributions, encoded in the associated edge set weight function. The definition of the weight function differs in the undirected and the directed cases. The undirected case is defined below:

$$\begin{aligned} w_{\mathcal{E}}^{ud} \colon \mathcal{E}_{\mathcal{M}} &\to \mathbb{N} \\ (C_k, C_{k'}) &\mapsto w_{kk'}. \end{aligned}$$

where $w_{kk'}$ defines edge thickness and encodes the total number of connections between the given pair of clusters:

$$w_{kk'} = \sum_{\substack{i \in cluster_k \\ i' \in cluster_{k'}}} \mathbb{1}_{Sim\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right) \geq \mathbf{q}}. \tag{4.2.1}$$

The directed case requires an extended version of the weight function, adding arrowheads on both sides of the edge:

$$w_{\mathcal{E}}^{d} \colon \mathcal{E}_{\mathcal{M}} \to [0,1] \times \mathbb{N} \times [0,1]$$
$$(C_k, C_{k'}) \mapsto (a_k, w_{kk'}, a_{k'}) \,.$$

where a_k and $a_{k'}$ define the sizes of the arrowheads towards C_k and $C_{k'}$ encoding the proportions of the predictive connections in the corresponding directions. The formal definitions are given below:

$$a_k = \frac{1}{w_{kk'}} \sum_{\substack{i \in cluster_k \\ i' \in cluster_{k'}}} \mathbb{1}_{\mathcal{OW}_{i'i} > 0} \times \mathbb{1}_{Sim\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right) \geq \mathbf{q}}, \tag{4.2.2}$$

$$a_{k'} = \frac{1}{w_{kk'}} \sum_{\substack{i \in cluster_k \\ i' \in cluster_{k'}}} \mathbb{1}_{\mathcal{OW}_{ii'} > 0} \times \mathbb{1}_{Sim\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right) \geq \mathbf{q}}. \tag{4.2.3}$$

It can be noticed that the quantities given in (4.2.1), (4.2.2) and (4.2.3) can be expressed only based on the elements of the adjacency matrix, defined in (4.1.2), instead of both the similarity and the $\mathcal{OW}$ matrix. In particular, the expression for the edge thickness can be expressed as

$$w_{kk'} = \sum_{\substack{i \in cluster_k \\ i' \in cluster_{k'}}} \mathbb{1}_{\min(X_{ii'}, X_{i'i}) = 1} = \sum_{\substack{i \in cluster_k \\ i' \in cluster_{k'}}} \min(X_{ii'}, X_{i'i}), \tag{4.2.4}$$

using the fact that if, for a given pair of entities $i \in cluster_k$ and $i' \in cluster_{k'}$ for $k \neq k'$, we have $Sim\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right) \geq \mathbf{q}$, then either $X_{ii'}$ or $X_{i'i}$ or both are equal to 1. We apply a similar reasoning to rewrite the expression for the arrowhead sizes, utilizing the symmetry of $Sim(\cdot, \cdot)$ and the anti-symmetry of $\mathcal{OW}$:

$$a_k = \frac{1}{w_{kk'}} \sum_{\substack{i \in cluster_k \\ i' \in cluster_{k'}}} \mathbb{1}_{X_{ii'} = 0} \times \mathbb{1}_{X_{i'i} = 1} = \frac{1}{w_{kk'}} \sum_{\substack{i \in cluster_k \\ i' \in cluster_{k'}}} X_{i'i}(1 - X_{ii'}), \tag{4.2.5}$$

(4.2.6) $$a_{k'} = \frac{1}{w_{kk'}} \sum_{\substack{i \in cluster_k \\ i' \in cluster_{k'}}} \mathbb{1}_{X_{ii'}=1} \times \mathbb{1}_{X_{i'i}=0} = \frac{1}{w_{kk'}} \sum_{\substack{i \in cluster_k \\ i' \in cluster_{k'}}} X_{ii'}(1 - X_{i'i}).$$

Indeed, considering the equations (4.2.2) and (4.2.5), saying that $Sim\left(\widehat{\Gamma}_i, \widehat{\Gamma}_{i'}\right) \geq \mathbf{q}$ and $\mathcal{OW}_{i'i} > 0$ is equivalent to saying that $Sim\left(\widehat{\Gamma}_{i'}, \widehat{\Gamma}_i\right) \geq \mathbf{q}$ and $\mathcal{OW}_{ii'} < 0$, which is true if and only if $X_{ii'} = 0$ and $X_{i'i} = 1$, based on (4.1.2).

To illustrate the principle behind the mesoscopic network representation and using the adjacency matrix-based equations (4.2.4), (4.2.5) and (4.2.6), in Figure 4.2.1 we present the visualization of the latter based on the matrix in Table 4.1.1, produced with the package *ScanOFC*. The information on the distributions, encoded in the edges characteristics, is also provided in a form of labels. For example, the edge between the clusters 1 and 2, labeled with the corresponding centroids 1 and 4, has "33%-3-67%" to describe the connectivity distribution. Indeed, the microscopic representation in Figure 4.1.1 clarifies this label, noticing that there are 3 directed (green) connections of the members of cluster 1 with those of cluster 2, with 2 directed towards cluster 2, and 1 towards cluster 1.

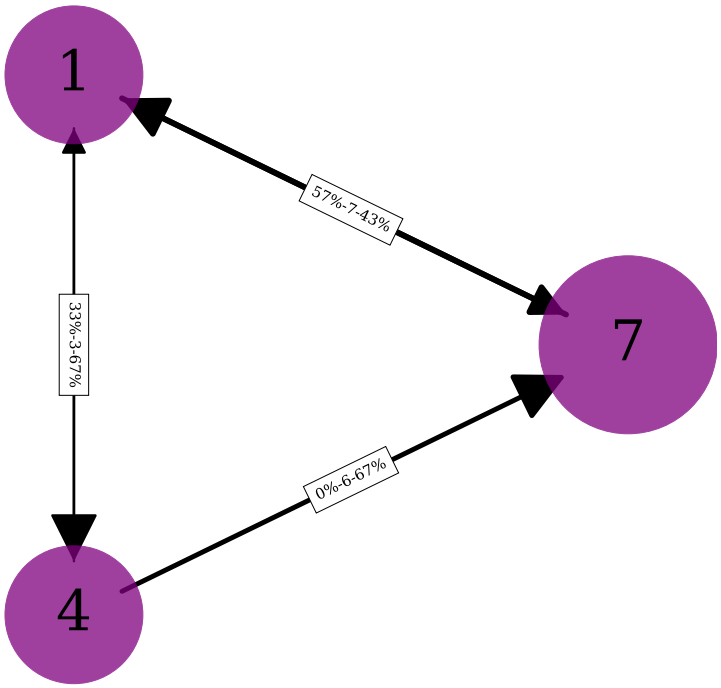


Figure (4.2.1) A visualization of the mesoscopic network based on the adjacency matrix presented in Table 4.1.1. Edge labels read as follows: "% of connections right → left - Total number of connections - % of connections left → right".

CHAPTER 5

APPLICATION TO REAL DATA

Returning to the radiobiological context, we apply our methodology to two in vitro datasets obtained through experiments studying the effects of two irradiation modalities differing in energy levels. For both datasets, the transcriptomic response of HUVEC cells (Human Umbilical Vein Endothelial Cells) is measured over time, with real time qPCR under control and under a single irradiation dose of 20 Gy at 0 h at 2.5 Gy/min. One dataset corresponds to irradiation using a LINAC, with the energy level of 4 MV, while the other encodes the response to SARRP (irradiation at 220 kV). First, we introduce data preprocessing and a new penalty that were motivated by the data studied in radiobiological context. Then, we present the analysis of the results obtained by applying our methods to these datasets. In particular, we compare various key features extracted for the two irradiation types, and perform the enrichment analysis of clusters and cluster subgroups with cellular processes in order to demonstrate the utility of the proposed tools.

5.1. Additional features motivated by data

5.1.1. Data preprocessing. Before performing clustering, certain transformations have to be applied to the raw data in order to amplify those characteristics that are of particular interest, and reduce those that can be ignored. We perform data scaling with respect to the following criteria:

- **Scaling by standard deviation:** performed in order to account for uncertainties, so that the observations with high uncertainty caused by individual variability appear with lower weight compared to those with low uncertainty.

Standard deviation estimates are calculated as follows: for $i \in \{1, 2, \dots, n_e\}$ we denote $\sigma_{\Gamma_i} = (\sigma_{\Gamma_i^{t_1}}, \dots, \sigma_{\Gamma_i^{t_p}})$ where $\sigma_{\Gamma_i^t} = \sqrt{\sigma^2_{\Gamma_i^t}}$.

- **Scaling by the fold change norm:** performed with the purpose of diminishing the effect of scale differences between the fold changes. The norm of $\widehat{\Gamma}_i$ associated with the distance $\widehat{\mathbf{d_2^2}}$ or the dissimilarity $\widehat{\mathbf{diss}}_\mathbf{s}$ can be expressed as follows:

$$Norm(\widehat{\Gamma}_i) = \sqrt{\|\Gamma_i\|_2^2 + \mathrm{Tr}(\Sigma_{\Gamma_i})} = \sqrt{\|\Gamma_i\|_2^2 + \sum_{l=1}^{p} \sigma^2_{\Gamma_i^{t_l}}}. \tag{5.1.1}$$

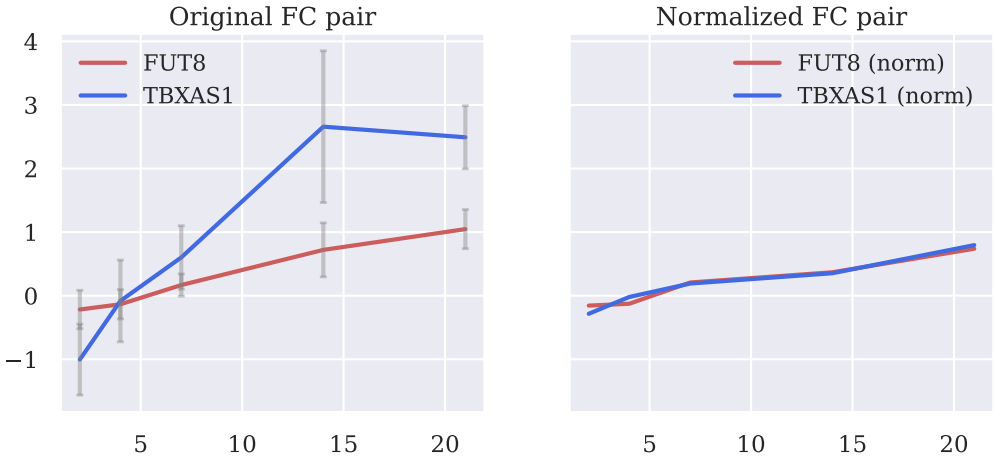


Figure (5.1.1) The effect of raw data scaling illustrated on a figure, where means with standard deviation of a pair of transcriptomic fold changes are plotted, inferred from the original data on the left and from scaled data on the right. As a result of scaling, the fold changes of genes TBXAS1 and FUT8 are rendered significantly closer than the original. For instance, it can be observed that the original fold changes are both characterized by almost monotonous growth during the whole period after irradiation. On the one hand, the curve of gene TBXAS1 is more concave, which can be neglected, and the difference is reduced by the scaling with respect to the norm. On the other hand, the scaling with respect to the standard deviation reduces the peak in the mean of gene TBXAS1 observed at day 14, which is also negligible due to very high standard deviation at that point.

The two scaling transformations described above are applied in a consecutive manner: the fold change norm scaling is calculated based on the result of the scaling by standard deviation, which implies that the norm of the final output is equal to 1. Thus, we obtain a processed dataset, from which new pairs of random fold changes estimators are constructed, and finally the pairwise distances are calculated. An illustrative example for the effect of preprocessing on the fold changes can be found in Figure 5.1.1.

5.1.1.1. *Fold change estimation from preprocessed data.* After applying the preprocessing to the response Y_{ikj}^t of an entity i at the time point t for a replicate j under the experimental condition k, the response becomes:

$$\widetilde{Y}_{ikj}^t = \frac{Y_{ikj}^t}{\sigma_{\Gamma_i^t} \times Norm\left(\Sigma_{\Gamma_i}^{-1}\widehat{\Gamma}_i\right)}, \text{ where } \sigma_{\Gamma_i^t} = \sqrt{\sigma_{\Gamma_i^t}^2}. \tag{5.1.2}$$

We obtain the following expression by applying the norm defined in (5.1.1) to the fold change $\widehat{\Gamma}_i$ after the scaling by standard deviation:

$$Norm\left(\Sigma_{\Gamma_i}^{-1}\widehat{\Gamma}_i\right) = \sqrt{\sum_{l=1}^{p}\left[\left(\frac{\Gamma_i^{t_l}}{\sigma_{\Gamma_i^{t_l}}^2}\right)^2 + 1\right]}. \tag{5.1.3}$$

The joint distribution of a fold change pair obtained from the preprocessed data can be rewritten in the following way:

$$\begin{bmatrix}\widehat{\widetilde{\Gamma}}_i \\ \widehat{\widetilde{\Gamma}}_{i'}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\widetilde{\Gamma}_i \\ \widetilde{\Gamma}_{i'}\end{bmatrix}, \begin{bmatrix}\Sigma_{\widetilde{\Gamma}_i} & \mathrm{P}_{\widetilde{\Gamma}_i\widetilde{\Gamma}_{i'}} \\ \left(\mathrm{P}_{\widetilde{\Gamma}_i\widetilde{\Gamma}_{i'}}\right)^\intercal & \Sigma_{\widetilde{\Gamma}_{i'}}\end{bmatrix}\right) \text{ such that:}$$

- Means for $x \in \{i, i'\}$:

$$\begin{aligned}\widetilde{\Gamma}_x &= \left(\frac{\sum_{j=1}^{n_r}(Y_{i1j}^{t_1} - Y_{i0j}^{t_1})}{n_r\sigma_{\Gamma_x^{t_1}}Norm\left(\Sigma_{\Gamma_x}^{-1}\widehat{\Gamma}_x\right)}, \cdots, \frac{\sum_{j=1}^{n_r}(Y_{i1j}^{t_p} - Y_{i0j}^{t_p})}{n_r\sigma_{\Gamma_x^{t_p}}Norm\left(\Sigma_{\Gamma_x}^{-1}\widehat{\Gamma}_x\right)}\right) \\ &= \frac{1}{Norm\left(\Sigma_{\Gamma_x}^{-1}\widehat{\Gamma}_x\right)}\left(\frac{\Gamma_x^{t_1}}{\sigma_{\Gamma_x^{t_1}}}, \cdots, \frac{\Gamma_x^{t_p}}{\sigma_{\Gamma_x^{t_p}}}\right),\end{aligned} \tag{5.1.4}$$

- Covariance matrices for $x \in \{i, i'\}$: $\Sigma_{\widetilde{\Gamma}_x} = \begin{bmatrix}\sigma_{\widetilde{\Gamma}_x^{t_1}}^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_{\widetilde{\Gamma}_x^{t_p}}^2\end{bmatrix}$,

$$\text{with } \sigma_{\widetilde{\Gamma}_x^t}^2 = \frac{\sum_{j=1}^{n_r}\left[(\widetilde{Y}_{i1j}^t - \overline{\widetilde{Y}_{i1}^t})^2 + (\widetilde{Y}_{i0j}^t - \overline{\widetilde{Y}_{i0}^t})^2\right]}{n_r - 1} = \frac{\sigma_{\Gamma_x^t}^2}{\sigma_{\Gamma_x^t}^2\left(Norm\left(\Sigma_{\Gamma_x}^{-1}\widehat{\Gamma}_x\right)\right)^2} = \frac{1}{\left(Norm\left(\Sigma_{\Gamma_x}^{-1}\widehat{\Gamma}_x\right)\right)^2}. \tag{5.1.5}$$

- Cross-covariance matrix: $\mathrm{P}_{\widetilde{\Gamma}_i\widetilde{\Gamma}_{i'}} = \begin{bmatrix} \rho_{\widetilde{\Gamma}_{i'}^{t_1}\widetilde{\Gamma}_x^{t_1}} & & 0 \\ & \ddots & \\ 0 & & \rho_{\widetilde{\Gamma}_{i'}^{t_p}\widetilde{\Gamma}_x^{t_p}} \end{bmatrix}$,

(5.1.6)
$$\text{with } \rho_{\widetilde{\Gamma}_i^t\widetilde{\Gamma}_{i'}^t} = \frac{\sum_{j=1}^{n_r}\left[(\widetilde{Y}_{i1j}^t - \overline{\widetilde{Y}_{i1}^t})(\widetilde{Y}_{i'1j}^t - \overline{\widetilde{Y}_{i'1}^t}) + (\widetilde{Y}_{i0j}^t - \overline{\widetilde{Y}_{i0}^t})(\widetilde{Y}_{i'0j}^t - \overline{\widetilde{Y}_{i'0}^t})\right]}{n_r - 1}$$
$$= \frac{\rho_{\Gamma_i^t\Gamma_{i'}^t}}{\sigma_{\Gamma_{i'}^t}\sigma_{\Gamma_{i'}^t} Norm\left(\Sigma_{\Gamma_i}^{-1}\widehat{\Gamma}_i\right) Norm\left(\Sigma_{\Gamma_{i'}}^{-1}\widehat{\Gamma}_{i'}\right)}.$$

All the subsequent analyses on real data are performed by applying the methodology from Chapters 2 and 4 directly to the scaled fold changes.

5.1.2. Sign penalty. From the biological perspective it is important to make a clear distinction between positively and negatively expressed entities. As a means to reinforce this distinction in the obtained clusters, we introduce a penalty term that increases the dissimilarity for those pairs of entities with different signs for one or more corresponding instances. For a warp step s and entity index pair $(i, i') \in \{1, \ldots, n_e\}^2$, the penalty term represents the proportion of time points where the means of the two considered fold changes have different signs:

$$Pen\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right) = \frac{1}{p - |s|}\sum_{l=1}^{p-|s|} \mathbb{1}_{\mathbb{R}_-}\left((\Gamma_i \circ \mathcal{W}_s)_{t_l} \times (\Gamma_{i'} \circ \mathcal{W}_s)_{t_l}\right).$$

By analogy with the distance matrix, a penalty matrix can be formulated:

$$[Pen_{ii'}]_{1\leq i,i'\leq n_e} \text{ such that } Pen_{ii'} = Pen\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right).$$

Finally, the penalized dissimilarity is defined with a penalization hyperparameter $\lambda \geq 0$:

$$Pen\left(d\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right)\right) = \widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right) + \lambda \times Pen_{ii'}.$$

This penalized dissimilarity is integrated into the aligned clustering procedure by replacing $\widehat{\mathbf{diss}}_{\mathbf{s}}\left(\widehat{\Gamma_i \circ \mathcal{W}_s}, \widehat{\Gamma_{i'} \circ \mathcal{W}_s}\right)$ in Definitions 2.4.1 and 2.4.2.

5.2. Results

We estimated transcriptomic fold changes of 157 genes for the LINAC dataset and 152 for the SARRP dataset. Gene expression was measured at 2, 4, 7, 14 and 21 days after irradiation. The estimation was performed based on 3 or 4 replicates. After the

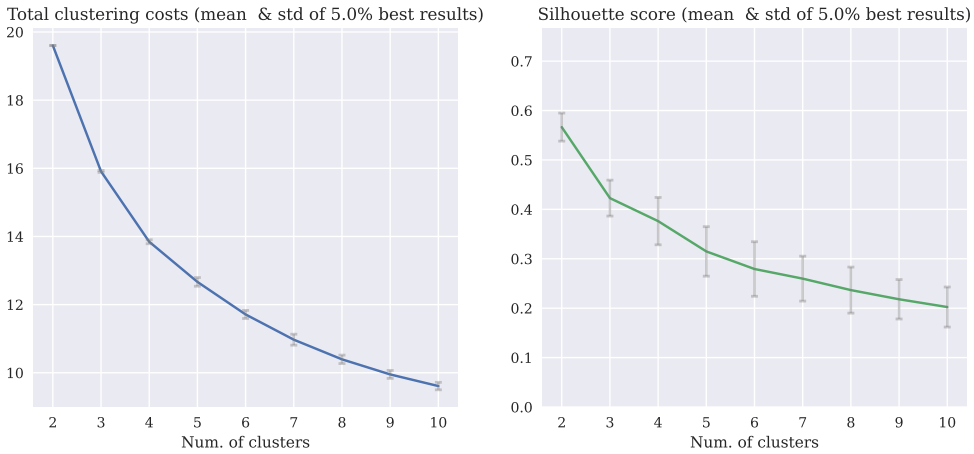
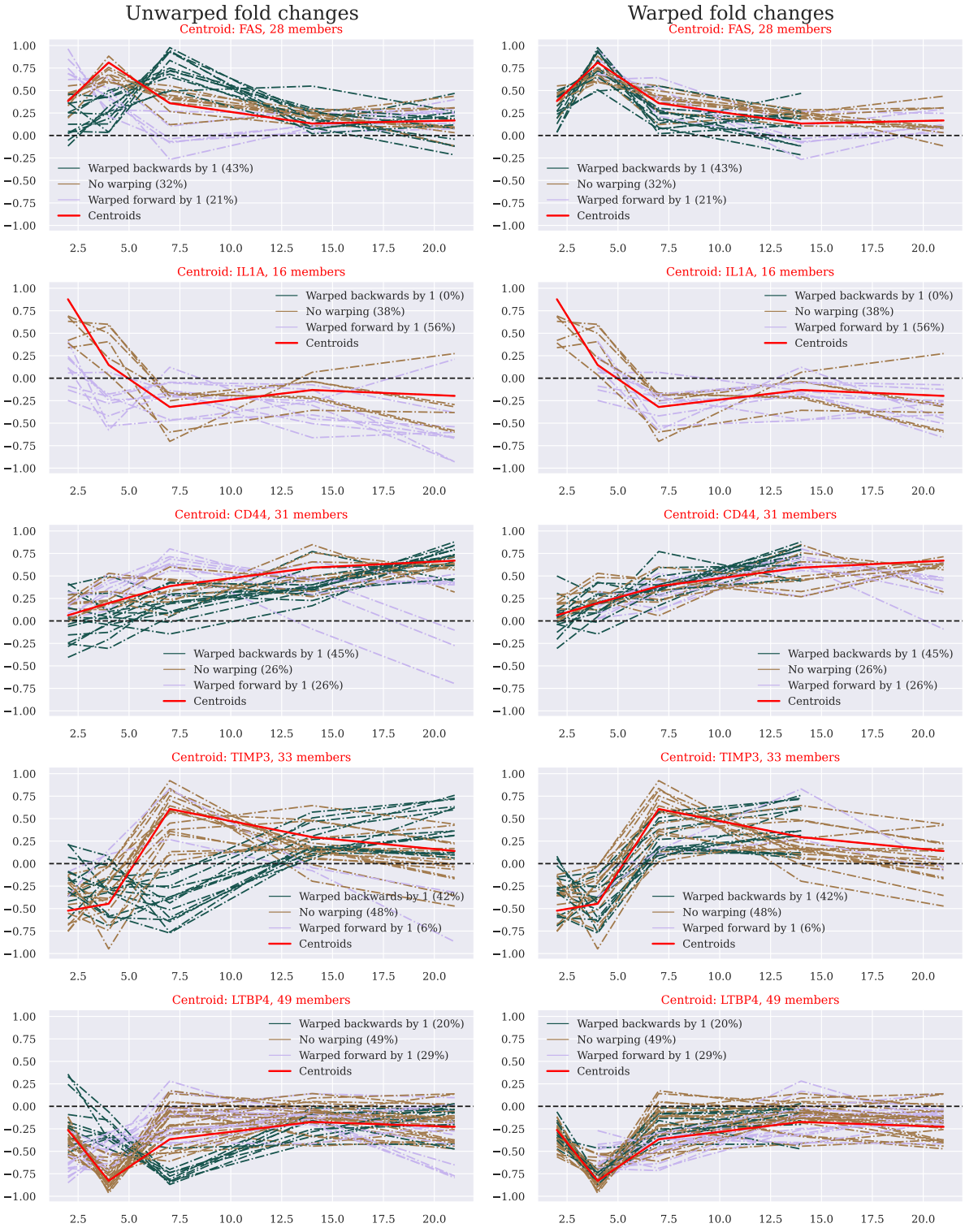


Figure (5.2.1) Means of cost and silhouette score with standard deviation of the 20% best outcomes of clustering of the LINAC dataset with $\widehat{\mathbf{diss}}_{\mathbf{s}}$ *k-medoids for the numbers of clusters in the set of fold changes ranging from 2 to 10. The most distinguishable shoulder for the total cost can be observed for 3 clusters, whereas the silhouette score declines for larger number of clusters. Both criteria suggest that smallest number of clusters should be chosen.*

fold changes were estimated from the log-transformed data and then preprocessed according to the procedures described in Sections 2.1 and 5.1.1, we performed clustering coupled with alignment based on the sign-penalized optimal warping dissimilarity matrix computed for both irradiation types.

5.2.1. Model choice evaluated on real data. It has been decided to choose 5 clusters produced by k-medoids clustering for these data based on the appearance of clusters expected from the biological point of view. Classical selection criteria such as total cost and silhouette score (Rousseeuw, 1987) appear to favor the smallest number of clusters (Figure 5.2.1). It appears that 5 is the smallest number of clusters that manage to produce well-separated behavior types. We compared clustering of the LINAC fold changes into 4 and 5 groups and concluded that 5 cluster version separates cluster 1 and 3 that are mixed together in cluster 4 of the 4 cluster version (Figures 5.2.2 and 5.2.3). This separation is justified biologically since it is important to distinguish the fold changes that are up-regulated three weeks after irradiation (cluster 3 of 5-cluster version) from those that are up-regulated early but lose the expression by two weeks after irradiation (cluster 1 of 5-cluster version). Such a distinction cannot be ensured merely by the means of having multiple alignment groups, given that having only 5 time points in the dataset the maximal warping step has to be set at 1.



(a) *(b)*

Figure (5.2.2) Clustering of the LINAC dataset with $\widehat{\mathbf{diss}}_{\mathbf{s}}$ *k-medoids in 5 clusters. a) Means of original normalized fold changes (unaligned). b) Means of warped normalized fold changes (aligned). The following behavior types can be distinguished (top to bottom): up-regulated and tending towards zero, up-regulated initially and down-regulated later on, steady growth, down-regulated initially and up-regulated later on, down-regulated and tending towards zero.*

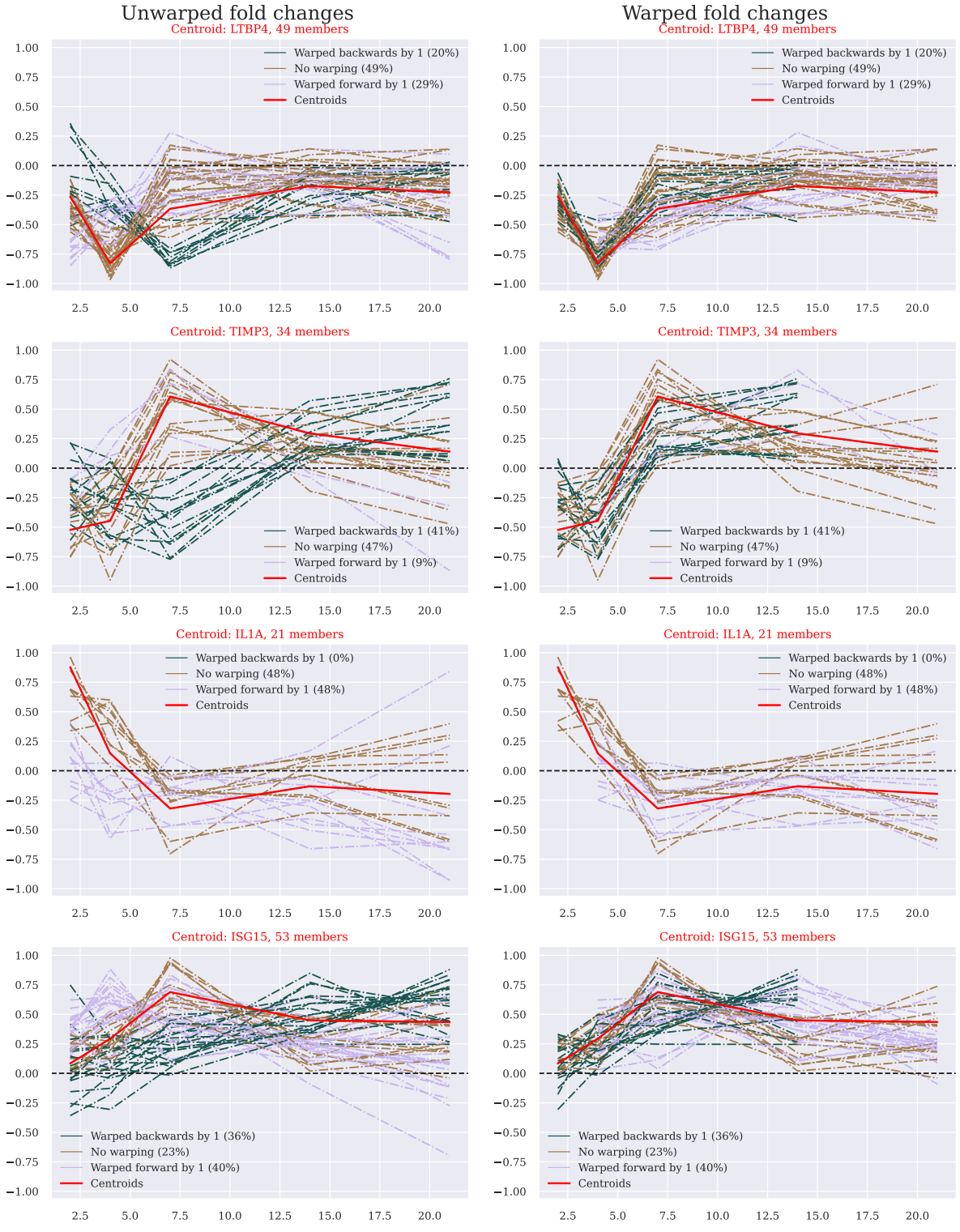


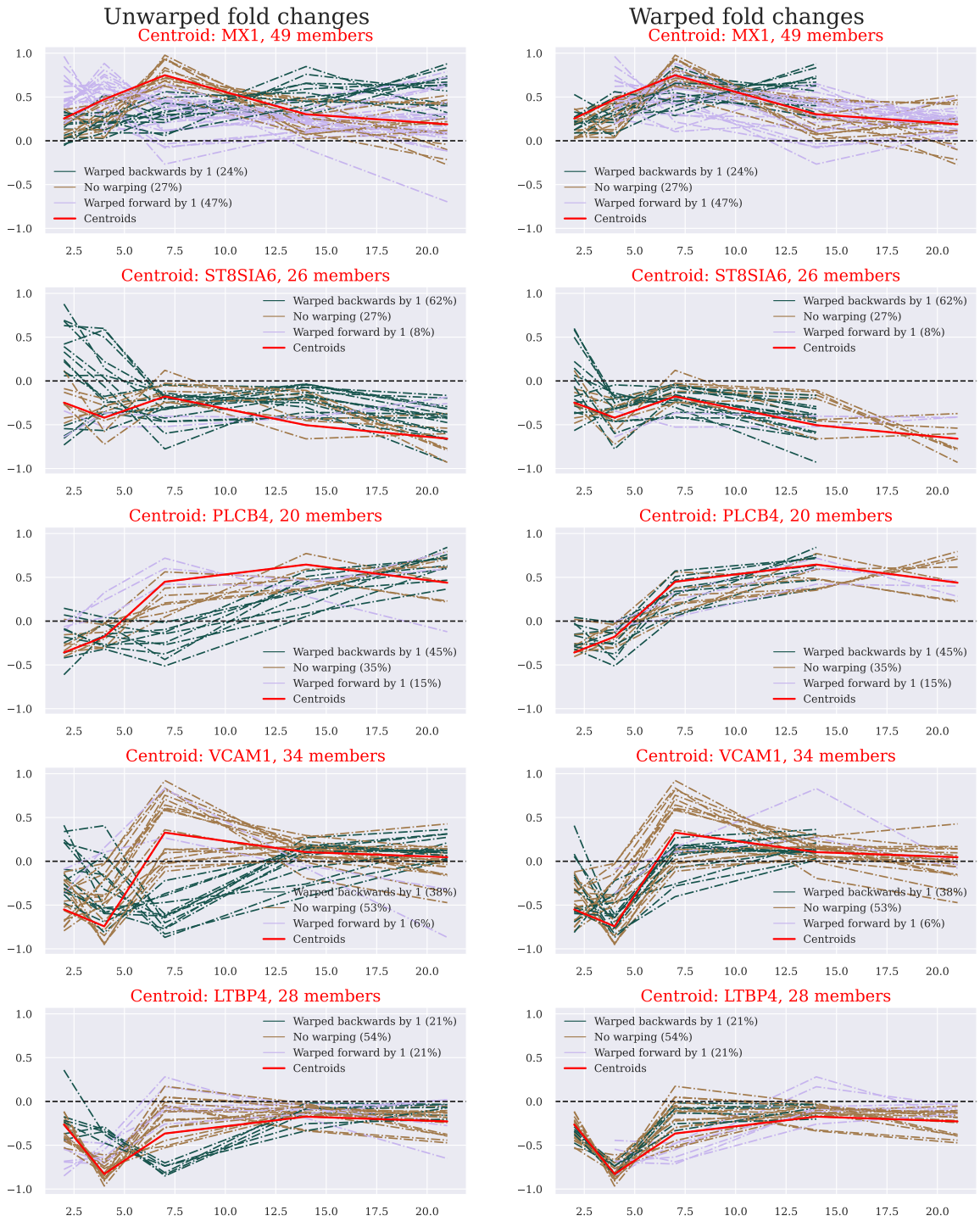
Figure (5.2.3) Clustering of the LINAC dataset with $\widehat{\mathbf{diss}}_{\mathbf{s}}$ *k-medoids in 4 clusters. a) Means of original normalized fold changes (unaligned). b) Means of warped normalized fold changes (aligned). The following behavior types can be distinguished (top to bottom): down-regulated and tending towards zero, down-regulated initially and up-regulated later on, up-regulated initially and down-regulated later on, up-regulated all along.*

LINAC clusters \ SARRP clusters	**1**	**2**	**3**	**4**	**5**
1	13	4	7	2	0
2	1	4	4	2	4
3	11	0	14	2	2
4	3	4	2	11	11
5	0	1	1	8	38

Table (5.2.1) Contingency table for clusters obtained for LINAC and SARRP.

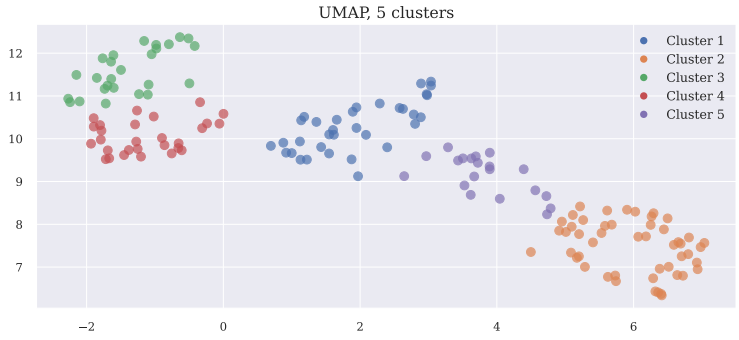
Regarding the choice of approach to clustering, it has been concluded that on this transcriptomic dataset $\widehat{\text{diss}}_s$ k-medoids allow identification of pertinent behavior types with better separation than k-means applied to the UMAP projection of the coordinates of the dissimilarity matrix, or stochastic block model. Despite the success of the UMAP-based method on simulated data, its real data clustering produces a poor separation of some clusters with respect of the expected behavior types, namely clusters 1 with 3, and 5 with 2 and 4 (Figures 5.2.4 and 5.2.5 for the UMAP projection and clustering, and Figure 5.2.6 for specific examples of fold changes that are clustered differently by two methods). One potential explanation may be the UMAP's lack of robustness when applied to highly irregular data, which has already been mentioned in literature (Wang et al., 2022; Hozumi et al., 2021). In line with this idea, the authors of the UMAP have pointed out its lower potential on small sample sizes of highly noisy data due to its tendency to assume locally manifold structure (McInnes et al., 2020). We believe that this assumption is not satisfied for our transcriptomic datasets, mainly due to the discontinuity introduced by time warping in the context of warps varying significantly from one pair to another. This effect is remarkably weaker for the simulated data sets, which explains the success of the UMAP in the simulation framework. As a result, $\widehat{\text{diss}}_s$ k-medoids approach has been chosen as the main, whereas the other two approaches are used to validate the results.

5.2.2. Cluster and network analysis of real data. The five clusters that were obtained are presented in Figures 5.2.7 (SARRP) and 5.2.2 (LINAC). The colorcode and the legend on the plots allows to identify which warp group (warped backward with respect to the centroid, simultaneous with the centroid and warped forward with respect to the centroid) each gene belongs to. Comparing the unaligned and the aligned

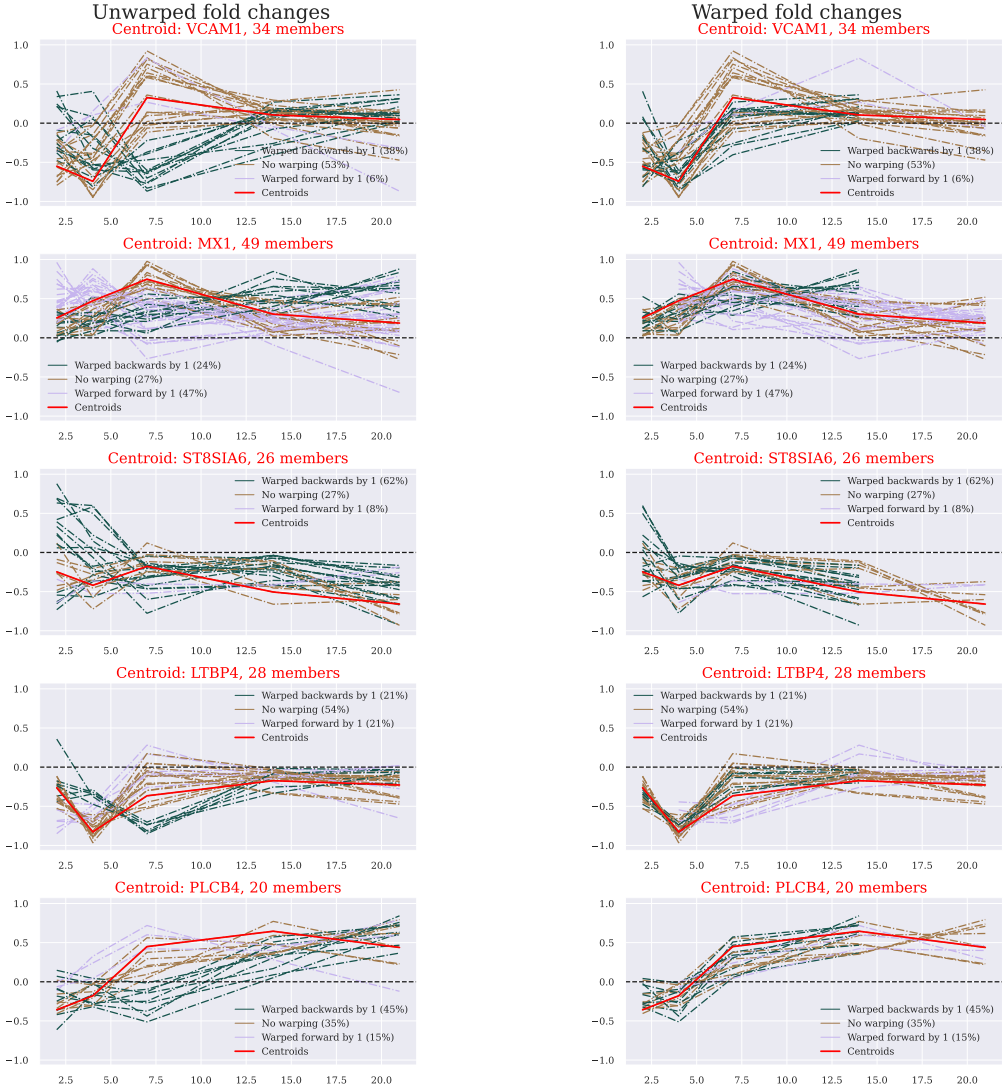


(a) *(b)*

Figure (5.2.4) Clustering of the LINAC dataset with k-means applied to a UMAP projection of the $\widehat{\mathbf{diss}}_{\mathbf{s}}$-matrix in 5 clusters. a) Means of original normalized fold changes (unaligned). b) Means of warped normalized fold changes (aligned). In comparison with $\widehat{\mathbf{diss}}_{\mathbf{s}}$ k-medoids, one can observe in particular: 1) the positive expression during the first week is less present in cluster 2 (ST8SIA6), which makes it similar to cluster 5 (LTBP4), 2) cluster 1 (MX1) contains many elements that are generally positively expressed without necessarily following the trend of the centroid.

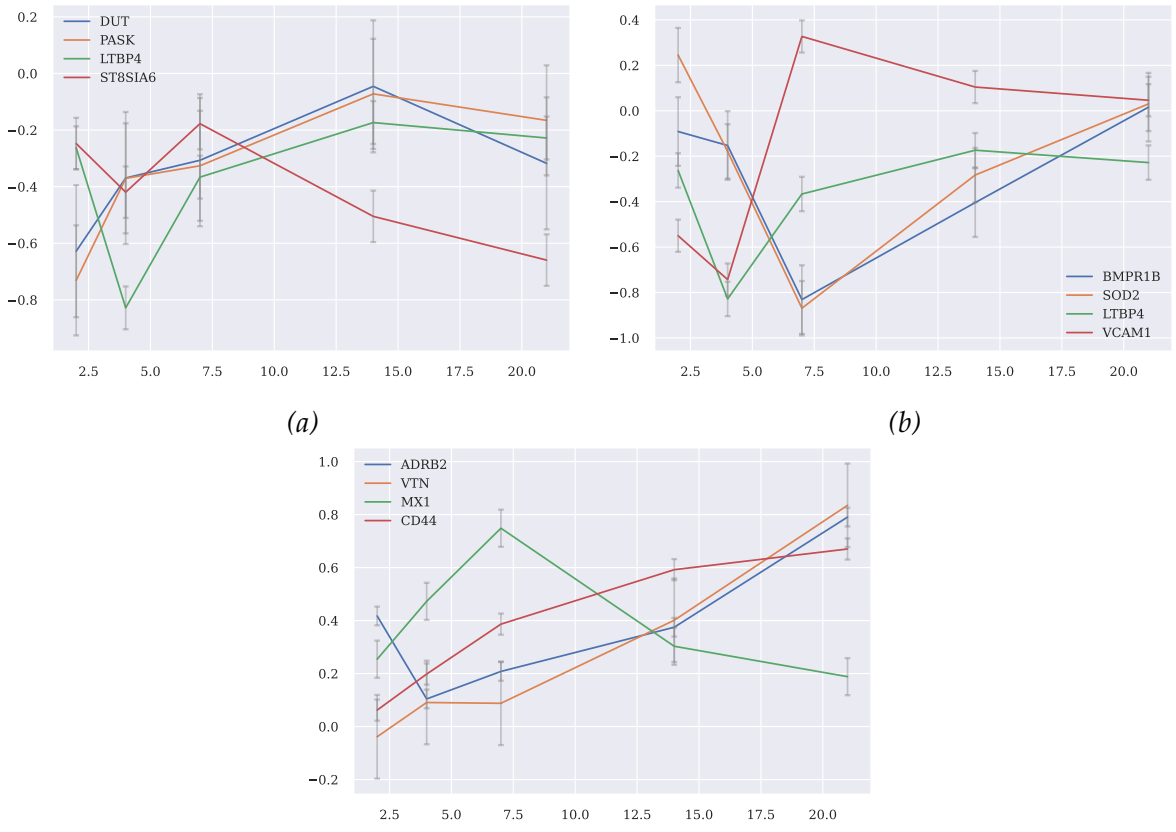


(a)



(b) *(c)*

Figure (5.2.5) UMAP projection of the $\widehat{\mathbf{diss}}_{\mathbf{s}}$-matrix from the LINAC dataset with the subsequent k-means clustering presented. Cluster indices are in the same order for the UMAP projection and the clustering, but different from the previous clustering results. The UMAP projection explains the the observations mention in Figure 5.2.4: 1) the distinction between clusters 3 (ST8SIA6) and 4 (LTBP4) is ambiguous, 2) a large block in cluster 2 (MX1) could be equivalently assigned to cluster 5 (PLCB4).

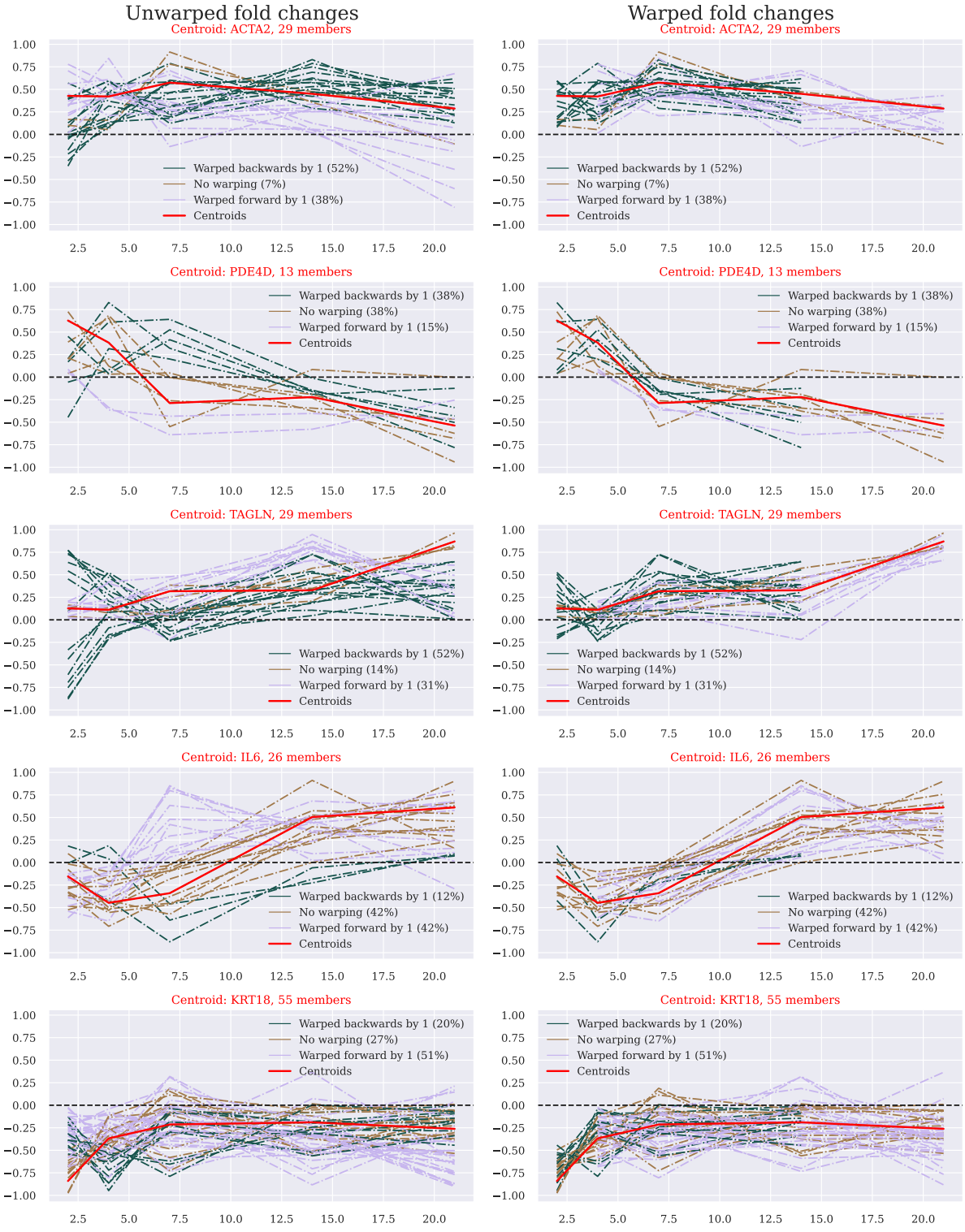


(a) *(b)*

(c)

Figure (5.2.6) Examples of means with standard deviation of the fold changes from the LINAC dataset that have been differently clustered with $\widehat{\mathbf{diss}}_{\mathbf{s}}$ *k-medoids and with k-means applied to a UMAP projection of the* $\widehat{\mathbf{diss}}_{\mathbf{s}}$*-matrix. a) DUT and PASK have been put in cluster 2 by UMAP (centroid ST8SIA6), while being warped to the right they are much closer to the centroid of cluster 5 (LTBP4) by* $\widehat{\mathbf{diss}}_{\mathbf{s}}$ *k-medoids. b) Warped to the left, BMPR1B and SOD2 are both mainly negative, therefore belong more to the cluster 5 (centroid LTBP4 by k-medoids) than cluster 4 (centroid VCAM1 by UMAP). c) ADRB2 and VTN show profiles corresponding to steady growth, and are extremely close to centroid of the cluster 3 (CD44), where they have been successfully put by k-medoids. Such classification is more reasonable than the one suggested by UMAP (cluster 1, centroid MX1).*

versions allows to see more clearly how each group has been transformed in order to get aligned with the centroid. The aligned version is the one used for clustering, allowing to identify global behavior types up to a time shift, whereas the unaligned version allows to identify temporal cascades inside every cluster, i.e. the forward-warped predict simultaneous that predict the backward-warped. It has to be noted that these plots only contain the means of preprocessed fold changes, giving a rough idea of the genes' behavior but can be at times misleading since clustering is performed on full fold changes, containing not only means but all the information on correlations and uncertainties that can be inferred from the replicates.



(a) . *(b)*

Figure (5.2.7) Clustering of the SARRP dataset with $\widehat{\mathbf{diss}}_{\mathbf{s}}$ *k-medoids in 5 clusters. a) Means of original normalized fold changes (unaligned). b) Means of warped normalized fold changes (aligned). The following behavior types can be distinguished (top to bottom): up-regulated and tending towards zero, up-regulated initially and down-regulated later on, steady growth, down-regulated initially and up-regulated later on, down-regulated and tending towards zero.*

The clusters are ordered to match between both conditions with respect to the response types represented by each cluster. Indeed, we manage to obtain very similar response types for both conditions: two clusters 1 and 3 characterized by up-regulation, being strongly up-regulated early and late respectively, a generally down-regulated cluster 5 that is roughly symmetric to cluster 1, and two clusters that manifest change of sign, with 2 being up-regulated early and down-regulated late, and 4 doing the opposite. For both irradiation types cluster 2 appears to be much smaller then the others, while cluster 5 contains almost a third of all fold changes. It can be observed that clusters 1 and 3 show much less striking distinction in case of SARRP compared to LINAC. It can also be noted that clusters 4 and 5 appear to have very overall consistent behavior across conditions, which is more visible in the unaligned case rather than aligned, since their centroids belong to different time groups (in case of cluster 4, the LINAC centroid TIMP3 corresponds to earlier expression then the SARRP centroid IL6, and the opposite is observed in case of cluster 5).

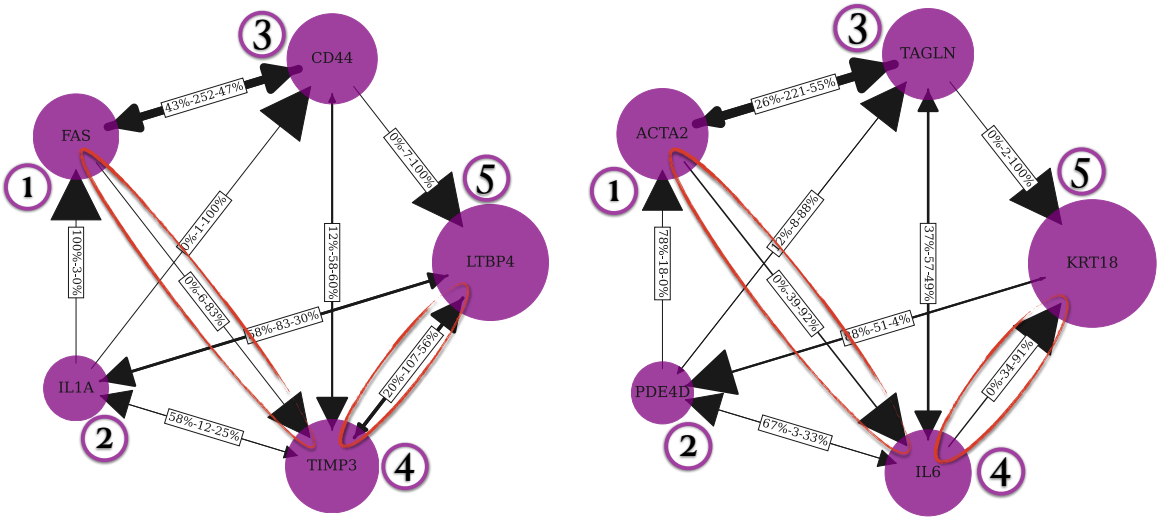


Figure (5.2.8) Mesoscopic views of the LINAC-specific (a) and SARRP-specific (b) fold changes network. Nodes represent clusters, labeled with their centroid genes, with node sizes corresponding to respective cluster sizes, and edges summarizing the connections between clusters based on the original adjacency matrix.

The next step of the statistical analysis consists in inferring the transcriptomic fold changes network according to the procedure described in Chapter 4. Package *ScanOFC* proposes two major visualization tools for network analysis, based on the models presented in Chapter 4: mesoscopic and microscopic views. The former is a way of concisely representing the key features of the network, with clusters as nodes,

and edges being derived from connectivity distribution between clusters. It allows in particular to obtain additional distributional information with respect to cluster migrations thanks to leveraging similarities, information that is inaccessible through studying the contingency table alone (Table 5.2.1). Depending on the goal, one can choose to work with full networks, or a part of it. For instance, mesoscopic graphs of Figure 5.2.8 were build based on condition-specific networks, meaning that the connections that constitute the adjacency matrix, and therefore the inter-cluster connectivity distribution, forming the edges of the mesoscopic graph, only appear in the corresponding irradiation condition and not the other, which allows to compare two conditions strictly based on their differences. There are multiple edges that appear in both graphs, the most important being the one between 1 (FAS/ACTA2) and 3 (CD44/TAGLN). Given that the networks are condition-specific, it suggests that there is a big number of genes that travel between clusters 1 and 3, which are rather similar in general for both conditions. The most striking difference between the graphs lies in edges that are present/important for one condition and absent/not important for the other. This seems to be particularly the case between clusters 4 (TIMP3/IL6) and 5 (LTBP4/KRT18), the phenomenon that is hard to interpret since it is not clear whether it is similar genes that change clusters, or genes potentially conserving clusters that change their similarity. The situation can be clarified by studying another mesoscopic graph presented in Figure 5.2.9, a hybrid representation of two conditions: the genes in each node are those associated with the LINAC clustering, while the network itself (and thus the connections) is the one specific to SARRP. This hybrid graph unsurprisingly demonstrates a bigger overall number of connections since the clustering is not the one natural for the network, and many connections that are otherwise intra-cluster appear here as inter-cluster. In particular, the connection between clusters 4 and 5 is even stronger than that between clusters 1 and 3, which indicates that there is a group of fold changes in LINAC's clusters 4 and 5 that are extremely similar for both conditions, with connections that disappear in SARRP, which is most likely caused by them becoming intra-cluster connections. All of the above suggests that migrations between clusters 4 and 5 are particularly important in detecting the differences between the two irradiation types.

The second proposed tool for network visualization sheds additional light on fold change distribution with respect to cluster migration. Figures 5.2.10 and 5.2.11 are two examples of multiple modes of representing a microscopic network with different features. The first figure illustrates the most natural representation mode of the LINAC

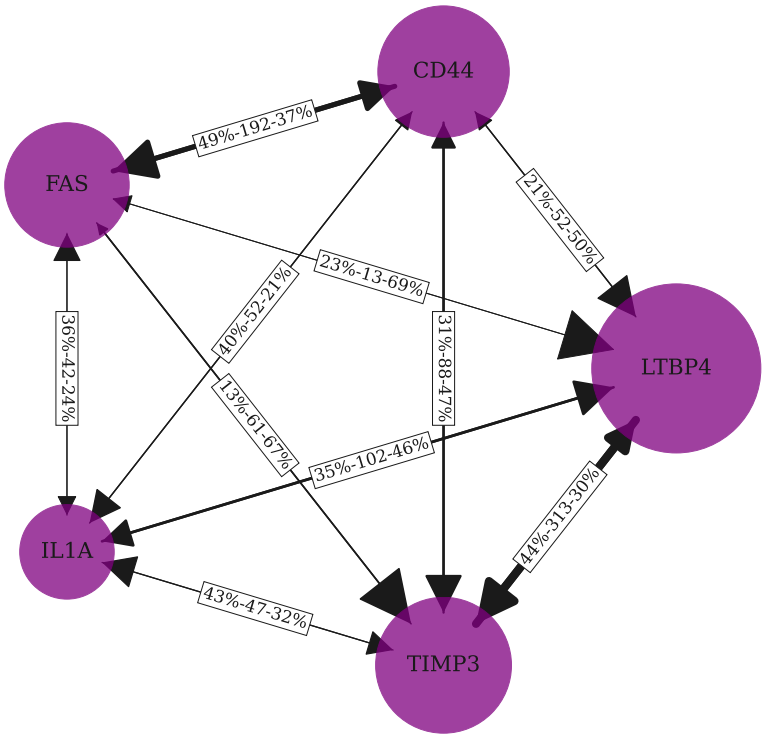


Figure (5.2.9) Mesoscopic view of the hybrid fold changes network. Nodes represent clusters (LINAC), labeled with their centroid genes, with node sizes corresponding to respective cluster sizes, and edges summarizing the connections between clusters based on the original adjacency matrix (SARRP).

network, whereas the second figure shows a hybrid network that serves specifically to compare two types of irradiation. For a consistent comparison both are based on the network containing only the links that are shared by both datasets, as opposed to the condition-specific graphs on Figure 5.2.8. Moreover, both networks have a block structure, but the blocks are constructed differently. The blocks of the network on Figure 5.2.10, denoted each by a distinct color, correspond to clusters inferred from the LINAC dataset. Within each block, the fold changes are placed around their centroid (bigger node) according to the Kamada-Kawai method. The network on Figure 5.2.11 has a hybrid block structure. In terms of layout, the blocks correspond to clusters inferred from the LINAC dataset, the node positions are the same as those in the LINAC network view (Figure 5.2.10). The cluster colors match those of the LINAC network but are assigned according to the SARRP clustering. The bigger nodes are the centroids with respect to the SARRP clustering. For example, on Figure 5.2.11 gene KRT18 is a bigger node, colored in pink, which is the color of cluster 5 on Figure 5.2.10, but located with cluster 4. This means that this gene is in cluster 4 for LINAC, but for SARRP it migrated to cluster 5, and also became its centroid.

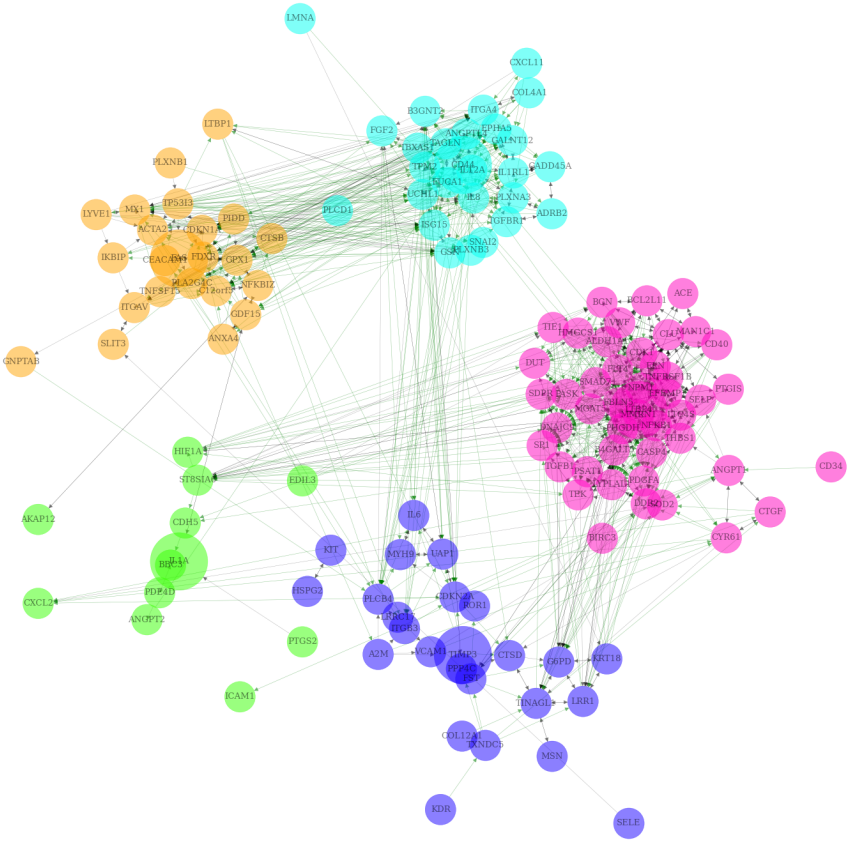


Figure (5.2.10) Microscopic views of the LINAC directed fold changes network.

By comparing the graphs, a few observations can be made that support the conclusions made out of the mesoscopic representations. In particular, it is made clear that approximately a half of LINAC's cluster 4 becomes a part of cluster 5 for SARRP. It implies that this very group of genes (pink nodes at the cluster 4 position) is responsible for a big number of previously mentionned connections turning intra-cluster. Moreover, cluster 5 is the only cluster whose centroid for SARRP is in cluster 4 for LINAC. However, cluster 5 seems to be much more stable than other clusters overall given its superior size. It can also be noted that clusters 1 and 3 seem to exchange genes mainly between each other, which is consistent with the idea of them being more similar for SARRP than for LINAC.

5.2.3. Study of fold change norms. One way to approach the biological effectiveness of the LINAC irradiation relative to SARRP is to identify those genes that are characterized by the biggest difference in expression between the two conditions. In order to do this, the fold changes have to be compared in their original scales, that is before the scaling by the fold change norms step of the preprocessing. However, comparing fold change scales without taking into account their behavior types makes the

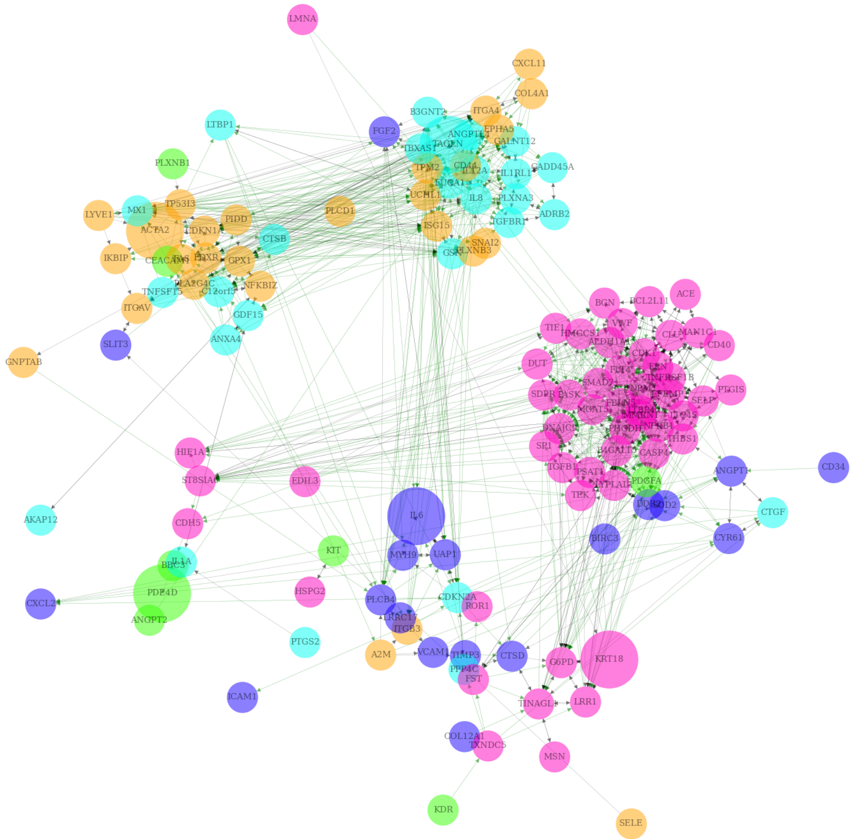


Figure (5.2.11) Microscopic views of the hybrid fold changes network.

results hard to interpret. In order to ensure interpretability, the differences in scales should be studied separately within each cluster, while taking into account the information about the fold change warping groups.

We introduce the following notation for a norm of a fold change of a gene i after scaling by standard deviation, distinguishing the irradiation condition by adding a corresponding superscript:

$$\|\widehat{\Gamma}_i^{condition}\| = Norm\left(\Sigma^{-1}_{\Gamma_i^{condition}} \widehat{\Gamma}_i^{condition}\right).$$

Consider a pair of clusters with labels $(k, k') \in \{1, \ldots, K\}^2$, indicating the labels with respect to LINAC and SARRP clusterings respectively. Treating the general case where k and k' can be unequal allows to consider both fold changes with a stable behavior pattern for both conditions, and those that migrate from one cluster to another. The goal is to identify all fold changes whose norms are at least as different as a given threshold $\tau_{kk'} \in \mathbb{R}^+$, or formally the following set:

$$\Omega^{\tau}_{kk'} = \left\{ i \in cluster_k^{LINAC} \cap cluster_{k'}^{SARRP} \mid \left| \|\widehat{\Gamma}_i^{LINAC}\| - \|\widehat{\Gamma}_i^{SARRP}\| \right| > \tau_{kk'} \right\}.$$

In parallel with the fold change norms per cluster, we consider the information that we can obtain with respect to their warp groups. The warps between the fold changes of the same genes estimated from different datasets cannot be explicitly calculated. This difficulty can be overcome by comparing the centroids of the given pair of clusters and fixing their respective warp groups based on their appearance in final clusterings. It can be often easily done for the matching clusters and less so for clusters with different behavior types. For example, after examining clusterings for SARRP and LINAC of figures 5.2.7 and 5.2.2 respectively, specifically by matching the corresponding centroids' peaks of positive/negative expression, it can be concluded that for cluster 1, LINAC's centroid is expressed one time step earlier than that of SARRP, and the opposite can be observed for cluster 5.

Given $\mathfrak{s}$ the time shift that has to be applied to C_k^{LINAC} in order to match it with $C_{k'}^{SARRP}$, we define the time shift between the fold changes of the gene i from LINAC and SARRP datasets as follows:

$$\mathfrak{s}_i = \left(\mathcal{OW}^{LINAC}_{iC_k^{LINAC}} + \mathfrak{s}\right) - \mathcal{OW}^{SARRP}_{iC_{k'}^{SARRP}}. \tag{5.2.1}$$

To get a global view of the changes in fold change norms taking place between LINAC and SARRP, we first consider the distributions of norms across clusters. The most general observation that can be made is the average difference between the norms for all fold changes being equal to 1.39. The fact that this value is positive implies that the fold changes for all genes tend to be more strongly expressed under LINAC than under SARRP, which is consistent with the biological expectations since LINAC is associated with higher irradiation energy and is thus supposed to produce stronger response in the irradiated case. Next, we focus on the average differences between the fold change norms for every cluster combination, presented in Table 5.2.2. The diagonal, which represents the fold changes that stay in the same cluster for both conditions, preserves the global trend, containing only positive values. It is on the level of cluster 1 that we observe the biggest difference. This fact explains the slight difference in shapes of cluster 1 for the two conditions: the fold changes demonstrate a significant difference between early and late response to irradiation in the case of LINAC, and a more flat shape in the case of SARRP. Whereas in the scale-normalized case it appears as if SARRP produced more response in the late stage, it is in fact due to LINAC producing more important response in the beginning, and thus the important positive difference in norm. The fact that the difference is the most pronounced for cluster 1 implies that the difference in the effect of these two types of irradiation is the

best represented by the fold changes that are positively expressed with strong early expression.

LINAC clusters \ SARRP clusters	**1**	**2**	**3**	**4**	**5**
1	4.04	2.05	-0.06	7.57	-
2	3.11	1.01	-3.89	-0.52	-0.39
3	2.64	-	2.43	-3.9	0.54
4	1.47	-0.94	-1.88	2.01	-1.29
5	-	5.6	-0.86	3.02	1.39

Table (5.2.2) Mean differences in scales (fold change norms) per cluster between LINAC and SARRP.

When studying the non-diagonal elements of the table, corresponding to the fold changes that change clusters across conditions, we can in multiple cases observe opposite signs for symmetric cluster pairs. In particular, the value is positive for the fold changes that are in cluster 5 for LINAC and in cluster 4 for SARRP, and negative for those that are in cluster 4 for LINAC and in cluster 5 for SARRP. This indicates that cluster 5 is generally characterized by higher norms and therefore more important expression. To further analyze the table, we can make a distinction between small and big groups. In the former category we can distinguish the fold changes with very high norms: those of genes CSF3 and SLIT3 clustered in 1 for LINAC and in cluster 4 for SARRP, and those of the gene PDGFA clustered in 5 for LINAC and in cluster 2 for SARRP. In both cases it implies that the change in scale explains the migration: more significant positive early expression of CSF3 and SLIT3, and more significant negative early expression of PDGFA, under LINAC. These three genes can serve as potentially promising candidates to study the radiation response in the experimental setting.

Regarding the big cluster combinations, we can particularly distinguish the fold changes migrating from 5 under LINAC to 4 under SARRP, and those migrating from 3 under LINAC to 1 under SARRP. Combining this information with typical behavior per cluster, it can be deduced that the first group is characterized by a stronger early negative expression, whereas the second one by a stronger late expression, while irradiated with LINAC.

	Intra-cluster				Inter-cluster	
	Cluster 1	Cluster 3	Cluster 4	Cluster 5	Clusters 5 (LINAC) $\rightarrow$ 4 (SARRP)	Clusters 3 (LINAC) $\rightarrow$ 1 (SARRP)
Mean norm differences per cluster	4.04	2.43	2.01	1.39	3.02	2.64
Gene names in $\Omega^{\tau}_{kk'}$ with $\tau = 4$	ACTA2 LYVE1 FDXR CDKN1A IKBIP	ADRB2 GADD45A IL1RL1	LRRC17 COL12A1 TIMP3 IL6	ELN PHGDH THBS1 CLU ACE CDK1 SELP DNAJC9 TNFRSF1B VWF	CD34 ANGPT1 PMAIP1 BMPR1B	CD44 SNAI2 UCHL1 ISG15

Table (5.2.3) Results of the analysis of $\Omega^{\tau}_{kk'}$ for the cluster combinations (both intra-cluster, or preserving cluster, and inter-cluster, or migrating) with the biggest number of elements and important mean differences in scales (norms). The genes with the fold changes that are simultaneous for both conditions with respect to the time shift defined in (5.2.1), that is if $\mathfrak{s}_i = 0$ for a gene i, are colored in blue.

The results of the analysis of the set $\Omega^{\tau}_{kk'}$ for certain combinations of clusters are presented in Table 5.2.3. The genes listed in the table are characterized by big differences in scale between the two irradiation conditions and can therefore be considered as potential key predictors of irradiation response. Among these genes can be found those that are already known as key actors in such cellular processes as senescence (CDKN1A, IL6, GADD45A), endothelial-mesenchymal transition (ACTA2, VWF, CD34), and endothelial activation (SELP, CD44). By considering the information about the time shift groups, we can construct gene cascades that can be potentially indicative of gene pathways. An example of such cascade is presented in Figure 5.2.12. Here the considered genes are among the ones written in blue in Table 5.2.3, indicating being simultaneous across conditions with respect to the definition of the time shift presented in (5.2.1), which makes the differences in scale more apparent and allows to manually construct the cascade in a natural way.

Another way to propose candidates for potential gene pathways is directly using the method *pathway_search* of the package *ScanOFC*. The method infers all the shortest

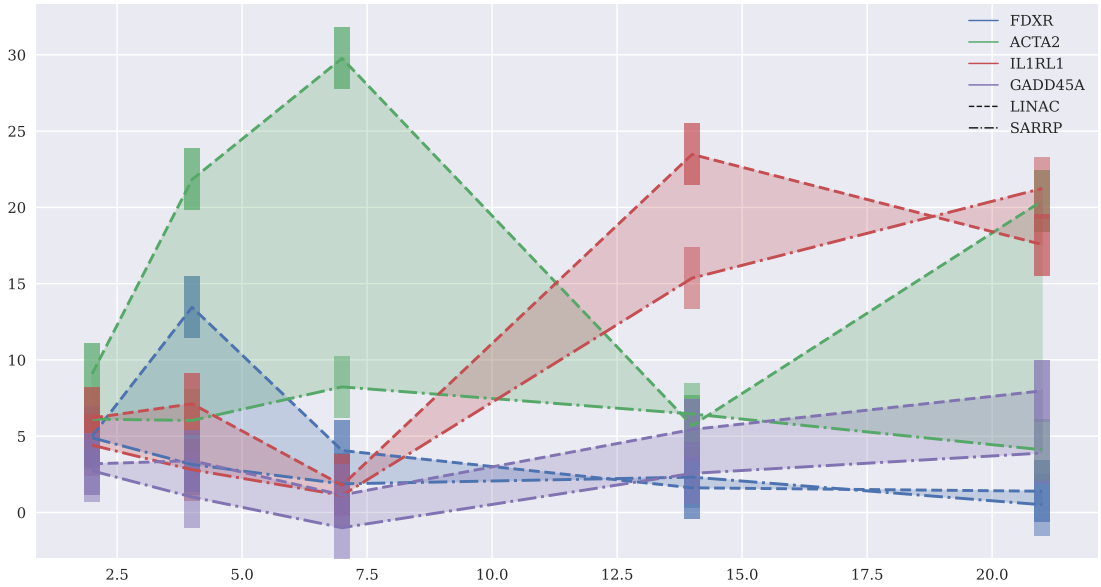


Figure (5.2.12) An example of a gene cascade FDXR → ACTA2 → IL1RL1 → GADD45A constructed out of genes found among the most different in scales between the two conditions, based on warp groups with respect to centroids.

paths in the given graph using the corresponding functions from the package *NetworkX*, and then sorts them with respect to their lengths and a criterion that is derived from the fold changes network framework. The criterion is a sum of two scores referred to as the warp score and the cluster score. The warp score corresponds to the number of connections in the considered path such that the associated warp is strictly positive. Bigger warp score implies more predictive relationships between genes and fewer simultaneous ones. The cluster score corresponds to the number of connections in the path between genes from different clusters. Maximizing the cluster score allows to obtain connections between different behavior types. Table 5.2.4 contains all the paths extracted from the network constructed by multiplying the adjacency matrices of LINAC and SARRP with 67% sparsity restricted to the genes that were previously identified as the most differentially expressed for the two considered conditions. The clustering used for the calculation of the cluster score takes both LINAC and SARRP clusterings into account, by distinguishing migration groups as separate clusters.

A typical example of a path that can be obtained using *pathway_search* is presented in Figure 5.2.13. The path demonstrates a consecutive up-regulation pattern, starting with the genes that get up-regulated immediately after irradiation, followed by those that are down-regulated initially and get up-regulated later, and ending with those that show strong down-regulation that disappears towards the end. Such paths are realistic candidates for radio-induced gene regulatory pathways, which would have

to be verified experimentally. It has to be noted that certain regulatory patterns observed by biologists cannot be captured in this framework, such as down-regulation stimulating up-regulation, since these two trends are of opposing nature, whereas our method can only capture relationships based on proximity.

	Gene path	Warp score	Cluster score	Total score
3-paths	ANGPT1 → SELP → ELN → IKBIP	3	2	5
	CD44 → UCHL1 → ANGPT1 → SELP	3	2	5
	UCHL1 → ANGPT1 → SELP → ELN	3	2	5
	UCHL1 → ANGPT1 → SELP → DNAJC9	3	2	5
	FDXR4 → ACTA2 → ISG15 → ADRB2	2	2	4
4-paths	UCHL1 → ANGPT1 → SELP → ELN → IKBIP	4	3	7
	CD44 → UCHL1 → ANGPT1 → SELP → ELN	4	2	6
	CD44 → UCHL1 → ANGPT1 → SELP → DNAJC9	4	2	6
5-paths	CD44 → UCHL1 → ANGPT1 → SELP → ELN → IKBIP	5	3	8

Table (5.2.4) All gene paths obtained from the intersection of LINAC and SARRP fold changes networks restricted to the genes present in Table 5.2.3.

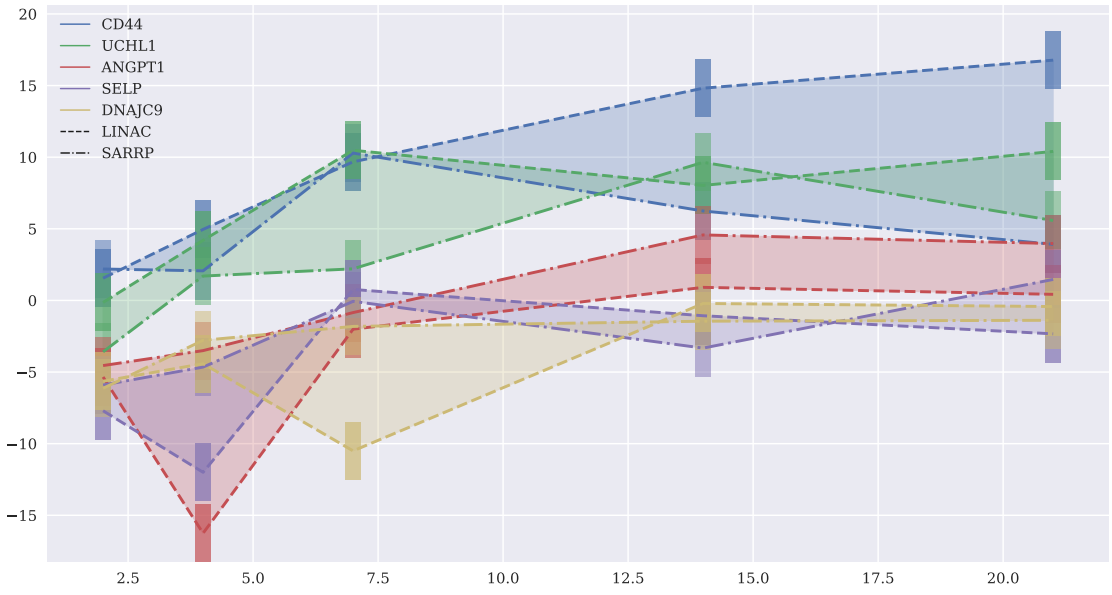


Figure (5.2.13) An example of a 4-path CD44 → UCHL1 → ANGPT1 → SELP → DNAJC9 presented in Table 5.2.4.

5.2.4. Enrichment analysis with Pathway Studio. In order to gain insight into the biological interpretation of the quantities and patterns identified using our methodology, we performed the enrichment analysis of clusters and cluster subgroups using Pathway Studio. The analysis can be summarized in 4 steps:

(1) Choose a list of entities to analyze. In our case, the lists were either genes that can be found in each cluster for LINAC and SARRP separately, or in cluster subgroups with respect to their migrations depending on the irradiation condition.
(2) Choose the biological features of interest for the enrichment. We focused on cellular processes, protein targets, protein regulators, expression targets and transcription factors.
(3) Extract the results and save in a form of a spreadsheet. The latter contains the list of biological features found in the literature that match with the given list of entities, and for each feature the list of entities that are known to be implicated and their total number, and some statistics such as the associated p-value.
(4) Combine the information from the spreadsheets for all clusters/subgroups, and visualize the summary in a form of a distribution diagram.

Among other analyses, we performed the enrichment with cellular processes of the intersections and differences between the clusters obtained for LINAC and SARRP in order to link the phenomenon of cluster migration with cellular processes. For every subgroup, the considered processes were filtered out with respect to the p-value at the level of 0.01, and with respect to the overlap at the level that was chosen in order to get a sufficient amount of information for larger clusters and avoid getting excessive information for smaller clusters. The results summarizing cluster migrations and the corresponding cellular processes the dominated in the analysis are presented in Figure 5.2.14, the original barplot with all of the detected cellular processes and their distributions across subgroups is presented in Figure 5.2.15. It can be observed that a number of processes are highly represented in multiple subgroups, it is the case for example of cell proliferation and adhesion. We are particularly interested in those that are highly represented in only one subgroup and not the others, thus allowing to conclude that the cellular process in question potentially characterize this subgroup. For each subgroup we manage to obtain such cellular processes. Some of them are in coherence with the enrichment analysis performed with Pathfinder, presented in Section C, in particular:

- the term 'cell death' detected by Pathway Studio in cluster 1 in common for LINAC and SARRP is related to the apoptosis term detected in cluster 1 separately for LINAC and SARRP by Pathfinder.

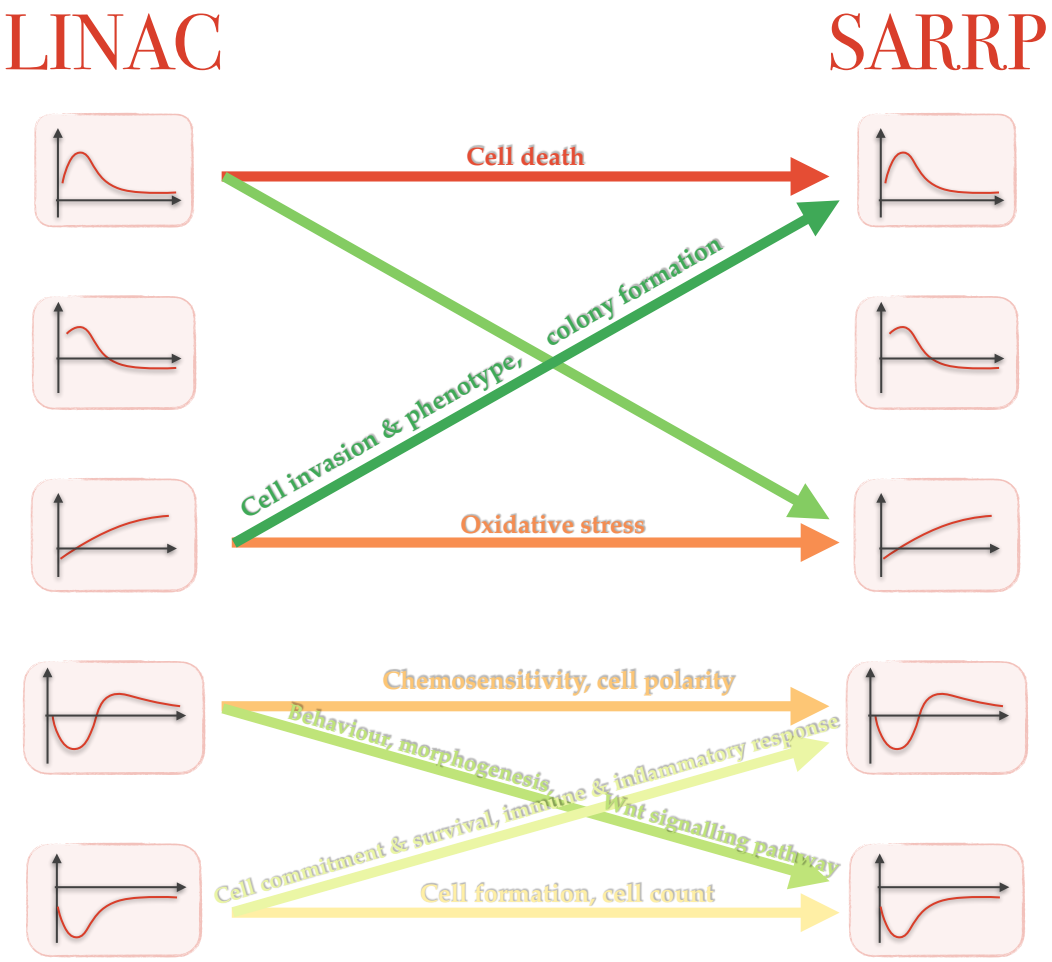


Figure (5.2.14) Illustration of cluster migrations between LINAC and SARRP, together with the dominating cellular processes identified for each of the subgroups of interest as a result of the enrichment analysis performed with Pathway Studio. The clusters are presented in order (from cluster 1 at the top to cluster 5 at the bottom), indicated by templates summarizing typical behavior types.

- terms 'chemosensitivity' and 'cell polarity' detected by Pathway Studio in cluster 4 in common for LINAC and SARRP is associated with chemotaxis detected in cluster 4 for SARRP by Pathfinder.
- the term 'morphogenesis' detected by Pathway Studio in LINAC's cluster 4 migrating to SARRP's cluster 5 can be linked with the 'regulation of cell shape' term detected in cluster 5 for LINAC by Pathfinder.

It should be noted, however, that the majority of terms that appear using Pathway Studio are not comparable to those that appear with Pathfinder, mainly due to the former being very general and the latter being much more specific.

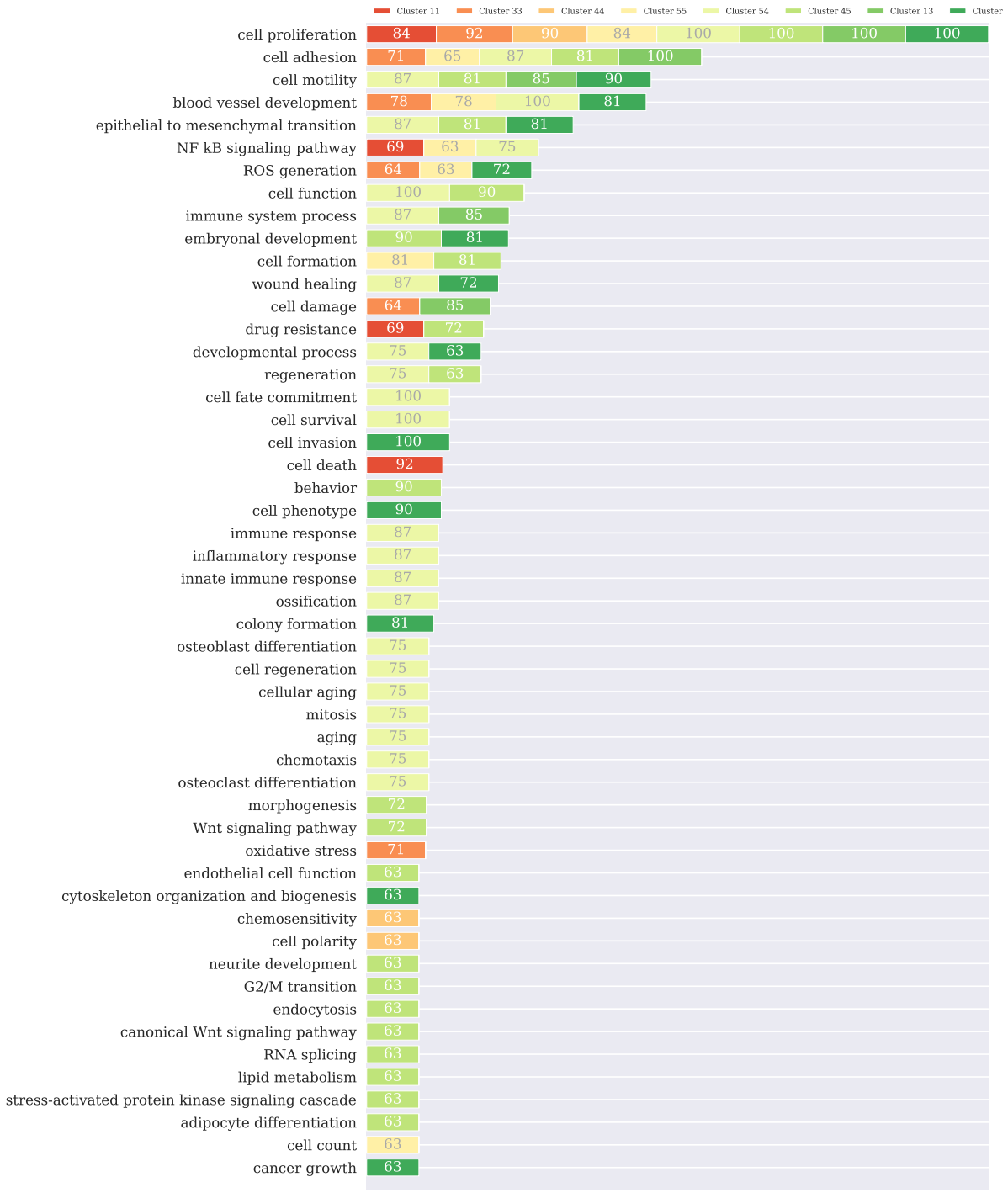


Figure (5.2.15) Summary of the enrichment with cellular processes of the subgroups of clusters obtained for LINAC and SARRP with respect to intersections and differences. Cellular processes, listed on the left, are sorted from the most represented to the least. The enriched subgroups, indicated with different colors, are those with the most important number of elements, including stable clusters (e.g. 'Cluster 11' for the genes that are in cluster 1 for both conditions), and migration clusters (e.g. 'Cluster 54' for the genes that are in cluster 5 for LINAC and in cluster 4 for SARRP). The numbers on the bars indicate the percentage of overlap with the given process for the given subgroup.

5.3. ScanOFC: Statistical framework for Clustering with Alignment and Network inference of Omic Fold Changes

A Python library containing tools for inference of multivariate omic fold changes from the data, for their subsequent clustering with alignment, and inference and visualisation of a network. The library is available at https://github.com/parsenteva/scanofc. Here is an overview of the main files:

- **scanofc.py**
  Main script, contains 3 classes: FoldChanges, Clustering and NetworkInference. See Appendix A for package documentation.
- **simulation_examples.ipynb**
  A Jupyter notebook containing examples from simulation studies showcasing frequently observed patterns and some of the potential interesting outcomes.
- **simulation_study_1.py**
  Main script of the first series of simulation studies focusing on the choice of distance and clustering algorithm.
- **simulation_study_2.py**
  Main script of the second series of simulation studies focusing on the effect of alignment, and two clustering alternatives: stochastic block model inference and clustering of the coordinates of the UMAP projection of the distance matrix.
- **scanofc_tutorial.ipynb**
  A Jupyter notebook demonstrating how to use ScanOFC on two real datasets.
- **scanofc_suppl_functions.py**
  Supplementary functions used in the tutorial.

Part 2

MODELING AND PREDICTION OF RADIO-INDUCED ADVERSE EFFECTS BASED ON IN VIVO DATA

CHAPTER 6

INTRODUCTION

6.1. Motivation and context

In order to study the effects of a certain treatment on a living organism, *in vivo* experiments are often conducted. In the context of a complex organism response, scientists may be interested in studying multiple variables describing the effect from different perspectives. In particular, such variables of interest often include a macroscopic biomarker only available through in vivo data on the one hand, and a microscopic biomarker that can also be observed on a cellular level. The interest in this case lies in predicting the former with the latter. For instance, in the context of studying the adverse effects induced by radiotherapy on healthy tissues, the potential outcomes of interest may manifest in the form of lesions, that are quantified in a form a certain macroscopic biomarker, and the levels of a some predictor such as gene expression. These measures often require sacrificing the animal. As a result, the quantities of interest cannot be observed on the same animals. Since the goal is to establish relationships between these variables, a problem of statistical data fusion arises.

In this work we propose an approach to estimate the relationship between the variables that are not simultaneously observed under the experimental setting described above, based on a conditional model assuming linear relationship within every component (experimental condition). An estimator derived with the method of moments as well as optimal transport solution using Wasserstein distance are considered for the real data problem in question.

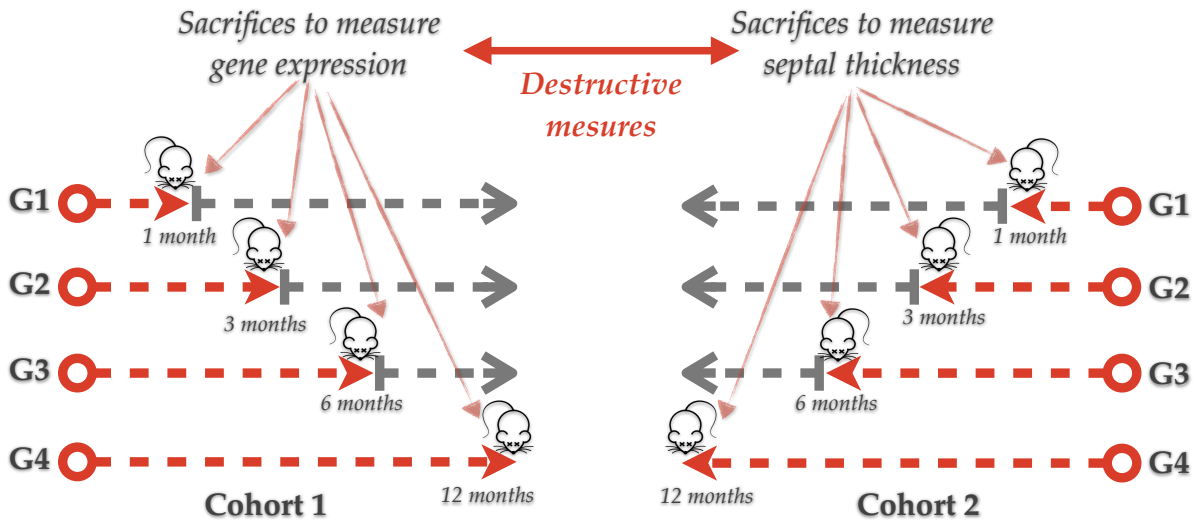


Figure (6.2.1) Schematic representation of the design of an in vivo experiment studying the effect of irradiated volume.

6.2. Example of an in vivo experiment

An illustrative example of an in vivo experiment where the variables of interest are never observed simultaneously is presented in Figure 6.2.1. In this experiment, mice are irradiated on the lungs with different volume, with a goal of studying the role of irradiated volume in the appearing of radio-induced adverse effects. The latter are assessed by measuring septal thickening, a macroscopic biomarker of radio-induced adverse effects in the lungs. The other variable that is measured with the purpose of predicting the adverse effect related variable is the expression of multiple pro-inflammatory genes. As shown in Figure 6.2.1, there are two independent cohorts in the study, one is used to measure gene expression, whereas the other for measuring septal thickness. This is a results of both measures being of destructive nature, which does not allow them to be taken on the same animals.

Comparing distributions of the measurements arising from the two cohorts, such as those presented in Figure 6.2.2, one may suspect a correlation or even a linear relationship between the variables. In order to assert whether such a relationship exists, one has to connect two variables that are never observed simultaneously, which translates into solving a data fusion problem. This can be done by taking into account another variable that is commonly present in such studies. This is a variable indicating belonging to a certain group for every observation, which is observed for both

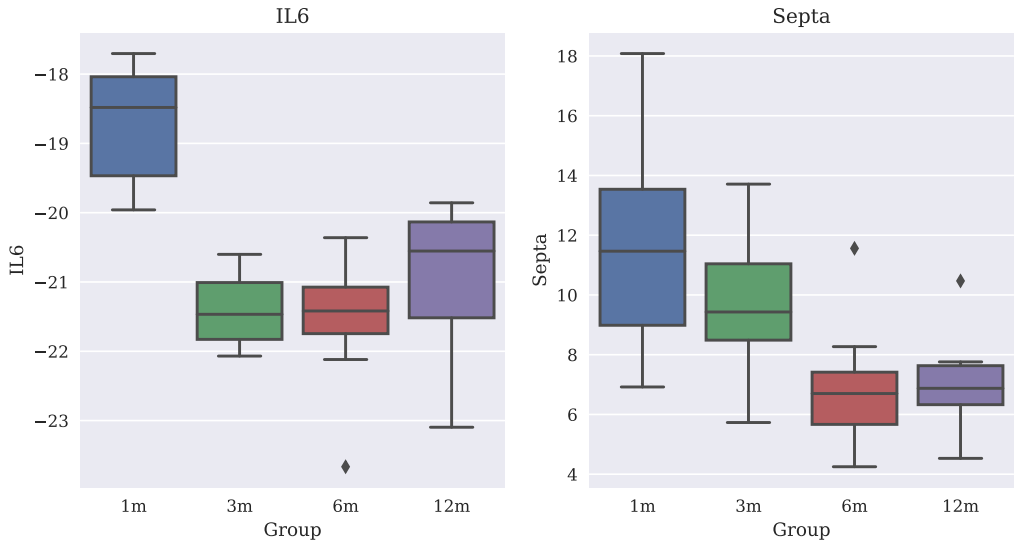


Figure (6.2.2) Distribution of the data, collected from the irradiated patch under SBRT with 3 mm beam size: the expression of the gene IL6 on the left, and septal thickness on the right. The measurements were made 1, 3, 6 and 12 months after irradiation.

cohorts. In this example, there are four groups indicating time points (1, 3, 6 and 12 months after irradiation) when the corresponding animals are sacrificed and the measurements are taken. Thus, this group variable can be used as an additional variable in order to link the predictor and the predicted variables between each other.

6.3. Existing research

The task of linking variables that are never jointly observed cannot be approached as a typical missing values problem, since most methods for inference on incomplete data require a sufficient overlap, which is completely absent in the case treated here. As a result, any approaches applying related frameworks such as multiple imputation in the context of data fusion are unsuitable for our application. For example, Carrig et al. (2015) use multiple imputation to integrate disparate datasets allowing for the absence of the overlap, but requiring a calibration dataset, where all the variables of interest must be jointly observed.

Other approaches to data fusion available in literature include factor analysis (Cudeck, 2000), statistical matching (Mitsuhiro and Hoshino, 2020), Bayesian network inference (Triantafillou et al., 2010; Tsamardinos et al., 2012) and Gaussian Markov combinations (Massa and Riccomagno, 2017). These methods are designed for linking variables that are not observed simultaneously through covariates, present for both variables of interest. This corresponds to the properties of the in vivo data described in Section 6.2. However, the covariates in these approaches are random variables,

typically continuous, and often assumed to be Gaussian, which is the case in Cudeck (2000) and Massa and Riccomagno (2017).The groups variable available through in vivo experiments may present in a continuous form, but the presence of such categories as control and sham makes it impossible to assume continuity and normality. The Bayesian network approaches introduced by Triantafillou et al. (2010) and Tsamardinos et al. (2012), aimed at inferring binary causal relationships between variables, are more suitable for large datasets with a high number of covariates. Finally, currently available research in statistical matching addresses such aspects as not-at-random missingness (Mitsuhiro and Hoshino, 2021) and high dimensionality (Mitsuhiro and Hoshino, 2020). This approach is based on the idea of comparing distances between the covariates from the datasets of interest, which cannot be done by taking the group variable as the covariate. It can be noted that the goal of the aforementioned examples in statistical matching is to group individuals before imputation, which is not necessary in our case since the groups are already known.

CHAPTER 7

ESTIMATING THE LINEAR RELATION BETWEEN VARIABLES THAT ARE NEVER JOINTLY OBSERVED

7.1. Problem and presentation of the different identification approaches

We consider a real random variable Y and a vector of d real valued random regressors $\mathbf{X} = (X_1, \ldots, X_d)$ and suppose that the following linear regression hold

$$Y = \beta_0 + \sum_{j=1}^{d} \beta_j X_j + \epsilon. \tag{7.1.1}$$

The residuals ϵ are supposed to be independent of the random covariates $X_1, \ldots, X_d$, with zero mean and variance σ_ϵ^2. With destructive in vivo experiments, we can never observe simultaneously $\mathbf{X}$ and Y, meaning that we never have at hand the pair $(\mathbf{X}, Y)$ but only $(\mathbf{X}, .)$ and $(., Y)$. This means that only the marginal moments of $\mathbf{X}$ and Y can be estimated in presence of sampled data.

In the absence of additional information and without any strong additional hypothesis, the parameters $(\beta_0, \beta_1, \ldots, \beta_d)$ and the variance of the noise σ_ϵ^2 cannot be identified. Indeed, if for example X_1 is centered with symmetric distribution, the coefficient β_1 can only be determined up to sign change since $\beta_1 X_1$ and $\beta_1(-X_1)$ have the same distribution.

To deal with this identification issue, we consider that we can perform different experiments in which the mean value of X is allowed to vary. For that, we suppose that there are K groups (corresponding to K different experiments), defined by a discrete variable G taking values in $\{1, \cdots, K\}$ observed simultaneously with Y and with X.

This means that we have now access to $(\mathbf{X}, G)$ and (Y, G) but not to $(\mathbf{X}, Y, G)$. We also suppose that ϵ is independent of G.

Given $G = k$, for $k = 1, \ldots, K$, we denote by $\mu_Y^k = \mathbb{E}(Y|G = k)$ and $\mu_{X_j}^k = \mathbb{E}(X_j|G = k)$, $j = 1, \ldots, d$ the expected values within each group. We also denote by $\pi_k = \mathbb{P}(G = k)$ the relative weight of each group in the population.

We now present different approaches developed to identify the vector $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_d)$ of regression coefficients and the noise variance σ_ϵ^2 taking account of the additional information given by the discrete variable G.

7.1.1. Moment approach. A first and simple approach is based on first moments identification. Taking the conditional expectation, given $G = k$, in (7.1.1), we have

$$\mu_Y^k = \beta_0 + \sum_{j=1}^{d} \beta_j \mu_{X_j}^k, \tag{7.1.2}$$

since the residual term ϵ is supposed to satisfy $\mathbb{E}(\epsilon|G = k) = 0$ for $k = 1, \ldots, K$. Based on (7.1.2) and introducing the probabilities of belonging to subpopulations, the following functional can be constructed:

$$\begin{aligned}\psi(\boldsymbol{\gamma}) &= \sum_{k=1}^{K} \mathbb{P}(G = k) \left(\mathbb{E}(Y|G = k) - (\gamma_0 + \sum_{j=1}^{d} \gamma_j \mathbb{E}(X_j|G = k)) \right)^2, \\ &= \sum_{k=1}^{K} \pi_k \left(\mu_Y^k - (\gamma_0 + \sum_{j=1}^{d} \gamma_j \mu_{X_j}^k) \right)^2.\end{aligned}$$

We denote by $\boldsymbol{\mu}_{1,X}$ the $K \times (d+1)$ design matrix, whose kth row is equal to $(1, \boldsymbol{\mu}_X^{k\top})$ with $\boldsymbol{\mu}_X^k = (\mu_{X_1}^k, \cdots, \mu_{X_d}^k)^\top$, by $\boldsymbol{\mu}_Y$ the K dimensional vector with elements $(\mu_Y^1, \ldots, \mu_Y^K)$, and by $\boldsymbol{\pi}$ the diagonal matrix with diagonal elements $(\pi_1, \ldots, \pi_K)$. We introduce the following assumption, guaranteeing the identifiability of the model parameters:

$$\mathbf{H}_1 \quad \text{rank}(\boldsymbol{\mu}_{1,X}) = d + 1,$$

meaning that there are at least $K \geq d + 1$ groups and that the $d + 1$ column vectors of $\boldsymbol{\mu}_{1,X}$ span a vector space of dimension $d + 1$ in $\mathbb{R}^K$.

LEMMA 7.1.1. *If assumption $\mathbf{H}_1$ is fulfilled, the unique minimizer of the functional ψ over $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \ldots, \gamma_d)^\top \in \mathbb{R}^{d+1}$ can be expressed as follows:*

$$\boldsymbol{\beta} = \left(\boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_{1,X}\right)^{-1} \boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_Y. \tag{7.1.3}$$

Proof

The functional ψ can be written in the matrix form:

$$\psi(\boldsymbol{\gamma}) = \left(\boldsymbol{\mu}_Y - \boldsymbol{\mu}_{1,X}\boldsymbol{\gamma}\right)^\top \boldsymbol{\pi} \left(\boldsymbol{\mu}_Y - \boldsymbol{\mu}_{1,X}\boldsymbol{\gamma}\right),$$

and thus its gradient:

$$\nabla\psi(\boldsymbol{\gamma}) = -2\boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \left(\boldsymbol{\mu}_Y - \boldsymbol{\mu}_{1,X}\boldsymbol{\gamma}\right).$$

Under the assumption $\mathbf{H}_1$, $\boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_{1,X}$ is invertible. It can be noticed that the gradient is equal to zero for $\boldsymbol{\beta}$ given in (7.1.3):

$$\nabla\psi(\boldsymbol{\beta}) = -2\boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_Y + 2\left(\boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_{1,X}\right)\left(\boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_{1,X}\right)^{-1} \boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_Y = 0.$$

Lastly, since ψ is strictly convex under $\mathbf{H}_1$, this minimizer is unique. □

Additionally, the expression for σ_ϵ^2 can be directly deduced from (7.1.1) by considering the variance of Y:

$$\sigma_\epsilon^2 = \sigma_Y^2 - \boldsymbol{\beta}_{-0}^\top \boldsymbol{\Gamma}_X \boldsymbol{\beta}_{-0}, \tag{7.1.4}$$

where $\boldsymbol{\Gamma}_X$ is the covariance matrix of $\mathbf{X}$, and $\boldsymbol{\beta}_{-0} = (\beta_1, \ldots, \beta_d)$.

7.1.2. Optimal transport and minimum Wasserstein distance approach. The second approach is based on optimal transport, in particular on the idea of estimating the linear transformation of the distribution of $\mathbf{X}$ that is the closest to that of Y with respect to the Wasserstein distance (see Panaretos and Zemel (2019) for a general introduction for statisticians).

For two one dimensional distributions D_1 and D_2 on $\mathbb{R}$ with finite p moments, and cumulative distribution functions F_1 and F_2, the Wasserstein distance (of order p) between the two distributions is equal to

$$W_p(D_1, D_2) = \left(\int_0^1 \left|F_1^{-1}(q) - F_2^{-1}(q)\right|^p dq\right)^{1/p}, \tag{7.1.5}$$

in particular, when $p = 1$:

$$W_1(D_1, D_2) = \int_{\mathbb{R}} |F_1(x) - F_2(x)|\, dx. \tag{7.1.6}$$

When $D_1 \sim \mathcal{N}(\mu_1, \Gamma_1)$ and $D_2 \sim \mathcal{N}(\mu_2, \Gamma_2)$,

$$W_2^2(D_1, D_2) = \|\mu_1 - \mu_2\|^2 + \text{tr}\left(\Gamma_1 + \Gamma_2 - 2\left(\Gamma_2^{1/2}\Gamma_1\Gamma_2^{1/2}\right)^{1/2}\right). \tag{7.1.7}$$

When $d = 1$ and $p = 2$, the previous equation reduces to

$$\begin{aligned} W_2^2(D_1, D_2) &= (\mu_1 - \mu_2)^2 + \sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2 \\ &= (\mu_1 - \mu_2)^2 + (\sigma_1 - \sigma_2)^2. \end{aligned} \tag{7.1.8}$$

As noted in Panaretos and Zemel (2019), the optimal transport map T between Gaussian measures on $\mathbb{R}^d$ is linear. Thus, considering a linear relation between $\mathbf{X}$ and Y is natural in a Gaussian setting and the Wasserstein distance of order 2 between Y and $\gamma_0 + \sum_{j=1}^d \gamma_j X_j + \epsilon$ is equal to

$$\varphi_0(\boldsymbol{\gamma}, \sigma^2) = \left(\mu_Y - \gamma_0 - \sum_{j=1}^d \gamma_j \mu_{X_j}\right)^2 + \left(\sigma_Y - \sqrt{\boldsymbol{\gamma}_{-0}^\top \boldsymbol{\Gamma}_X \boldsymbol{\gamma}_{-0} + \sigma^2}\right)^2, \tag{7.1.9}$$

where $\boldsymbol{\Gamma}_X$ is the covariance matrix of $\mathbf{X}$.

Taking account of the groups given by G we now assume that, for $k = 1, \ldots, K$, $\mathbf{X}|G = k \sim \mathcal{N}(\boldsymbol{\mu}_X^k, \boldsymbol{\Gamma}_X^k)$. Thus, given $G = k$, the Wasserstein distance between D_γ, the distribution of $\gamma_0 + \boldsymbol{\gamma}_{-0}^\top \mathbf{X} + \epsilon$, and D_Y, the distribution of Y, is equal to

$$W_2^2(D_\gamma, D_Y|G = k) = (\mu_Y^k - \alpha_0 - \boldsymbol{\gamma}_{-0}^\top \boldsymbol{\mu}_X^k)^2 + \left(\sigma_{Y,k} - \sqrt{\boldsymbol{\gamma}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\gamma}_{-0} + \sigma_\epsilon^2}\right)^2.$$

Considering the expectation of $W_2^2(D_\gamma, D_Y|G)$, we can define the loss criterion

$$\varphi(\boldsymbol{\gamma}, \sigma^2) = \sum_{k=1}^K \pi_k \left[(\mu_Y^k - \gamma_0 - \boldsymbol{\gamma}_{-0}^\top \boldsymbol{\mu}_X^k)^2 + \left(\sigma_{Y,k} - \sqrt{\boldsymbol{\gamma}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\gamma}_{-0} + \sigma^2}\right)^2\right]. \tag{7.1.10}$$

LEMMA 7.1.2. *If model (7.1.1) holds, and if assumption* $\mathbf{H}_1$ *is fulfilled,* $\varphi(\boldsymbol{\gamma}, \sigma_\epsilon^2)$ *has its unique minimum at* $\boldsymbol{\gamma} = \boldsymbol{\beta}$ *and* $\sigma^2 = \sigma_\epsilon^2$.

PROOF. Under model (7.1.1), $\varphi_0(\boldsymbol{\beta}, \sigma_\epsilon^2) = 0$, and thus $\varphi(\boldsymbol{\beta}, \sigma_\epsilon^2) = 0$. For all $\boldsymbol{\gamma} \in \mathbb{R}^{d+1}$ and $\sigma^2 > 0$, $\varphi(\boldsymbol{\gamma}, \sigma^2) \geq 0$, therefore $(\boldsymbol{\beta}, \sigma_\epsilon^2)$ minimizes φ. Finally, the uniqueness is guaranteed by the assumption $\mathbf{H}_1$. □

We have the following expression for the gradient $\nabla\varphi$, which is equal to zero at the minimum value of φ:

$$\nabla\varphi = \begin{pmatrix} \frac{\partial\varphi(\boldsymbol{\gamma})}{\partial\gamma_0} \\ \frac{\partial\varphi(\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}_{-0}} \\ \frac{\partial\varphi(\boldsymbol{\gamma})}{\partial\sigma^2} \end{pmatrix} = \begin{pmatrix} -2\sum_{k=1}^K \pi_k \left(\mu_Y^k - \gamma_0 - \boldsymbol{\gamma}_{-0}^\top \boldsymbol{\mu}_X^k\right) \\ -2\sum_{k=1}^K \pi_k \left[\left(\mu_Y^k - \gamma_0 - \boldsymbol{\gamma}_{-0}^\top \boldsymbol{\mu}_X^k\right) \boldsymbol{\mu}_X^k + \frac{\left(\sigma_{Y,k} - \sqrt{\boldsymbol{\gamma}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\gamma}_{-0} + \sigma^2}\right)}{\sqrt{\boldsymbol{\gamma}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\gamma}_{-0} + \sigma^2}} \boldsymbol{\Gamma}_X^k \boldsymbol{\gamma}_{-0}\right] \\ \sum_{k=1}^K \pi_k \left(1 - \frac{\sigma_{Y,k}}{\sqrt{\boldsymbol{\gamma}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\gamma}_{-0} + \sigma^2}}\right). \end{pmatrix}$$

7.2. Sampled data and estimators

We suppose that experiments are made for $K \geq 2$ different groups and that for each group k, with $k = 1, \ldots, K$, we have two independent samples $(Y_1^k, \ldots, Y_{n_y^k}^k)$ and $(X_{j,1}^k, \ldots, X_{j,n_x^k}^k)_{j=1,\ldots,d}$, with sizes n_y^k and n_x^k. For each unit $i = 1, \ldots, n_x^k$, the vector of covariates is denoted by $\mathbf{X}_i^k = (X_{1,i}, \ldots, X_{d,i})$. We also define $N_x = \sum_{k=1}^K n_x^k$ and $N_y = \sum_{k=1}^K n_y^k$, the total number of observations of the response Y and the covariates $X_1, \ldots, X_d$.

We can build estimates of μ_Y^k and μ_X^k, as well as the within variance matrices, by considering their empirical counterparts. For $k = 1, \ldots, K$ and $j = 1, \ldots, d$, we define

$$\begin{aligned}
\widehat{\mu}_Y^k &= \frac{1}{n_y^k} \sum_{i=1}^{n_y^k} Y_i^k \\
\widehat{\mu}_{X_j}^k &= \frac{1}{n_x^k} \sum_{i=1}^{n_x^k} X_{j,i}^k \\
\widehat{\sigma}_{Y,k}^2 &= \frac{1}{n_y^k} \sum_{i=1}^{n_y^k} \left(Y_i^k\right)^2 - \left(\widehat{\mu}_Y^k\right)^2 \\
\widehat{\mathbf{\Gamma}}_X^k &= \frac{1}{n_x^k} \sum_{i=1}^{n_x^k} \mathbf{X}_i^k (\mathbf{X}_i^k)^\top - \widehat{\boldsymbol{\mu}}_X^k (\widehat{\boldsymbol{\mu}}_X^k)^\top,
\end{aligned}$$

where $\widehat{\boldsymbol{\mu}}_X^k = (\widehat{\mu}_{X_1}^k, \ldots, \widehat{\mu}_{X_d}^k)$. We also consider the overall empirical mean and variance

$$\begin{aligned}
\widehat{\mu}_Y &= \frac{1}{N_y} \sum_{k=1}^K n_y^k \widehat{\mu}_Y^k \\
\widehat{\boldsymbol{\mu}}_X &= \frac{1}{N_x} \sum_{k=1}^K n_x^k \widehat{\boldsymbol{\mu}}_X^k \\
\widehat{\sigma}_Y^2 &= \frac{1}{N_y} \sum_{k=1}^K \sum_{i=1}^{n_y^k} \left(Y_i^k\right)^2 - (\widehat{\mu}_Y)^2 \\
\widehat{\mathbf{\Gamma}}_X &= \frac{1}{N_x} \sum_{k=1}^K \sum_{i=1}^{n_x^k} \mathbf{X}_i^k (\mathbf{X}_i^k)^\top - \widehat{\boldsymbol{\mu}}_X \widehat{\boldsymbol{\mu}}_X^\top.
\end{aligned}$$

Moment estimators of $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_d)$ and σ_ϵ^2 can be built by considering the empirical versions of (7.1.3) and (7.1.4):

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}^M &= \left(\widehat{\boldsymbol{\mu}}_{1,X}^\top \boldsymbol{\pi} \widehat{\boldsymbol{\mu}}_{1,X}\right)^{-1} \widehat{\boldsymbol{\mu}}_{1,X}^\top \boldsymbol{\pi} \widehat{\boldsymbol{\mu}}_Y \\
\widehat{\sigma}_\epsilon^{2,M} &= \widehat{\sigma}_Y^2 - (\widehat{\boldsymbol{\beta}}_{-0}^M)^\top \widehat{\mathbf{\Gamma}}_X \widehat{\boldsymbol{\beta}}_{-0}^M,
\end{aligned} \tag{7.2.1}$$

where $\widehat{\boldsymbol{\mu}}_Y = (\widehat{\mu}_Y^1, \ldots, \widehat{\mu}_Y^K)$, and $\widehat{\boldsymbol{\mu}}_{1,X}$ a $K \times (d+1)$ matrix, with the first column consisting of ones, and the rest equal to $\widehat{\boldsymbol{\mu}}_X$. Parameters π_k for $k = 1, \ldots, K$ are considered to be known, they can be set for instance as n_x^k/N_x.

Estimators of $\boldsymbol{\beta}$ and σ_ϵ^2 based on an optimal transport criterion are derived by minimizing the empirical version $\varphi_n(\boldsymbol{\gamma}, \sigma^2)$ of functional $\varphi(\boldsymbol{\gamma}, \sigma^2)$ defined by

$$\varphi_n(\boldsymbol{\gamma}, \sigma^2) = \sum_{k=1}^{K} \pi_k \left[(\widehat{\mu}_Y^k - \gamma_0 - \boldsymbol{\gamma}_{-0}^\top \widehat{\boldsymbol{\mu}}_X^k)^2 + \left(\widehat{\sigma}_{Y,k} - \sqrt{\boldsymbol{\gamma}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\gamma}_{-0} + \sigma^2} \right)^2 \right]. \tag{7.2.2}$$

We denote by $(\widehat{\boldsymbol{\beta}}^W, \widehat{\sigma}^{2,W})$ minimizers of $\varphi_n(\boldsymbol{\gamma}, \sigma^2)$ which are obtained with iterative optimization algorithms based on gradient descent (see (7.1.11)). The algorithm can be initialized randomly, or with $(\widehat{\boldsymbol{\beta}}^M, \widehat{\sigma}^{2,M})$.

7.3. Consistency and asymptotic distribution

To study the asymptotic behavior of the estimators of $\boldsymbol{\beta}$ defined in previous Section, we suppose that for all groups K and all variables $\mathbf{X}$ and Y, the number of observations tends to infinity. We denote by $n_{\min} = \min(n_y^1, \cdots, n_y^K, n_x^1, \cdots, n_x^K)$ the smallest sample size among all experiments. We suppose that $n_x^k/n_x \to \pi_k > 0$ as $n_{\min}$ tends to infinity.

7.3.1. Consistency.

LEMMA 7.3.1. *If $\mathbb{E}(Y^2) < +\infty$ and $\mathbb{E}(\|\mathbf{X}\|^2) < +\infty$, and assumption $\mathbf{H}_1$ is fulfilled, the sequence of estimators $(\widehat{\boldsymbol{\beta}}^M, \widehat{\sigma}_\epsilon^{2,M})$ converges in probability to $(\boldsymbol{\beta}, \sigma_\epsilon^2)$ when $n_{\min}$ tends to infinity.*

PROOF. *First note that the assumptions $\mathbb{E}(Y^2) < +\infty$ and $\mathbb{E}(\|\mathbf{X}\|^2) < +\infty$ ensure the existence of σ_Y^2 and $\boldsymbol{\Gamma}_X$. From the law of large numbers, we have that for all $k \in \{1, \ldots, K\}$, $\widehat{\boldsymbol{\mu}}_{1,X}^k \to \boldsymbol{\mu}_X^k$ and $\widehat{\mu}_Y^k \to \mu_Y^k$ in probability when $n_{\min}$ tends to infinity.*

We deduce from the continuous mapping theorem that $\widehat{\boldsymbol{\mu}}_{1,X}^\top \boldsymbol{\pi} \widehat{\boldsymbol{\mu}}_{1,X} \to \boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_{1,X}$ and $\widehat{\boldsymbol{\mu}}_{1,X}^\top \boldsymbol{\pi} \widehat{\boldsymbol{\mu}}_Y \to \boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_Y$ in probability. Under hypothesis $\mathbf{H}_1$, the inverse being continuous in a neighborhood of $\boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_{1,X}$ another application of the continuous mapping theorem gives that $\left(\widehat{\boldsymbol{\mu}}_{1,X}^\top \boldsymbol{\pi} \widehat{\boldsymbol{\mu}}_{1,X}\right)^{-1} \to \left(\boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_{1,X}\right)^{-1}$ and $\widehat{\boldsymbol{\beta}}^M = \left(\widehat{\boldsymbol{\mu}}_{1,X}^\top \boldsymbol{\pi} \widehat{\boldsymbol{\mu}}_{1,X}\right)^{-1} \widehat{\boldsymbol{\mu}}_{1,X}^\top \boldsymbol{\pi} \widehat{\boldsymbol{\mu}}_Y \to \left(\boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_{1,X}\right)^{-1} \boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_Y = \boldsymbol{\beta}$ in probability as $n_{\min} \to +\infty$.

The law of large numbers gives that $\widehat{\boldsymbol{\Gamma}}_X^2 \to \boldsymbol{\Gamma}_X^2$ and $\widehat{\sigma}_Y^2 \to \sigma_Y^2$ in probability and we deduce, with another application of the continuous mapping theorem, that $\widehat{\sigma}_Y^2 - \widehat{\boldsymbol{\beta}}^\top \widehat{\boldsymbol{\Gamma}}_X^2 \widehat{\boldsymbol{\beta}} \to \sigma_Y^2 - \boldsymbol{\beta}^\top \boldsymbol{\Gamma}_X^2 \boldsymbol{\beta} = \sigma_\epsilon^2$ in probability as $n_{\min} \to +\infty$. □

LEMMA 7.3.2. *If $\mathbb{E}(Y^2) < +\infty$ and $\mathbb{E}(\|\mathbf{X}\|^2) < +\infty$, $(\boldsymbol{\beta}, \sigma_\epsilon^2) \in \Theta$ and Θ is a compact set that does not contain 0, suppose that model (7.1.1) holds and hypothesis $\mathbf{H}_1$ is fulfilled, then the sequence of estimators $(\widehat{\boldsymbol{\beta}}^W, \widehat{\sigma}_\epsilon^{2,W})$ that minimize (7.2.2) converges in probability to $(\beta, \sigma_\epsilon^2)$.*

PROOF. *The proof is based on Lemma 2.9 in Newey and McFadden (1994) and Theorem 2.1 in Newey and McFadden (1994), which are recalled in Appendix D. Due to the law of large numbers and the continuous mapping theorem, for all $(\boldsymbol{\gamma}, \sigma_\gamma^2) \in \Theta$, $\varphi_n(\boldsymbol{\gamma}, \sigma_\gamma^2) \to \varphi(\boldsymbol{\gamma}, \sigma_\gamma^2)$ in probability, when $n_{\min}$ tends to infinity.*

Consider now $(\boldsymbol{\alpha}, \sigma_\alpha^2) \in \Theta$. We have,

$$\left|(\widehat{\mu}_Y^k - \gamma_0 - \boldsymbol{\gamma}_{-0}^\top \widehat{\boldsymbol{\mu}}_X^k)^2 - (\widehat{\mu}_Y^k - \alpha_0 - \boldsymbol{\alpha}_{-0}^\top \widehat{\boldsymbol{\mu}}_X^k)^2\right| = \left|(\boldsymbol{\alpha} - \boldsymbol{\gamma})^T \begin{pmatrix} 1 \\ \widehat{\boldsymbol{\mu}}_X^k \end{pmatrix} \left(2\widehat{\mu}_Y^k - (\boldsymbol{\alpha} + \boldsymbol{\gamma})^T \begin{pmatrix} 1 \\ \widehat{\boldsymbol{\mu}}_X^k \end{pmatrix}\right)\right|$$
$$\leq \|\boldsymbol{\alpha} - \boldsymbol{\gamma}\| \, A_n^k,$$

with Cauchy-Schwarz inequality and $A_{n,k} = \mathcal{O}_p(1)$ because $\|\widehat{\boldsymbol{\mu}}_X^k\| = O_p(1)$, $\widehat{\mu}_Y^k = O_p(1)$ and for some constant C_1 that does not depend on $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, $\|\boldsymbol{\alpha} + \boldsymbol{\gamma}\| \leq C_1 < \infty$ because Θ is supposed to be compact.

On the other hand, we have

$$\left|\left(\widehat{\sigma}_{Y,k} - \sqrt{\boldsymbol{\gamma}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\gamma}_{-0} + \sigma_\gamma^2}\right)^2 - \left(\widehat{\sigma}_{Y,k} - \sqrt{\boldsymbol{\alpha}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\alpha}_{-0} + \sigma_\alpha^2}\right)^2\right|$$
$$= \left|\sqrt{\boldsymbol{\alpha}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\alpha}_{-0} + \sigma_\alpha^2} - \sqrt{\boldsymbol{\gamma}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\gamma}_{-0} + \sigma_\gamma^2}\right| \left(2\widehat{\sigma}_{Y,k} + \sqrt{\boldsymbol{\alpha}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\alpha}_{-0} + \sigma_\alpha^2} + \sqrt{\boldsymbol{\gamma}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\gamma}_{-0} + \sigma_\gamma^2}\right)$$
$$= \left|\sqrt{\boldsymbol{\alpha}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\alpha}_{-0} + \sigma_\alpha^2} - \sqrt{\boldsymbol{\gamma}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\gamma}_{-0} + \sigma_\gamma^2}\right| O_p(1)$$

since Θ is compact and $\|\widehat{\boldsymbol{\Gamma}}_X^k\|_{sp} = O_p(1)$, where $\|.\|_{sp}$ denotes the spectral norm. Because $\boldsymbol{\alpha}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\alpha}_{-0} - \boldsymbol{\gamma}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\gamma}_{-0} = \boldsymbol{\alpha}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \left(\boldsymbol{\alpha}_{-0} - \boldsymbol{\gamma}_{-0}\right) + \left(\boldsymbol{\alpha}_{-0} - \boldsymbol{\gamma}_{-0}\right)^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\gamma}_{-0}$ we have, for some constant $C_{2,k} > 0$,

$$\left|\boldsymbol{\alpha}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\alpha}_{-0} - \boldsymbol{\gamma}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\gamma}_{-0}\right| \leq C_{2,k} \left\|\widehat{\boldsymbol{\Gamma}}_X^k\right\|_{sp} \|\boldsymbol{\alpha} - \boldsymbol{\gamma}\|. \tag{7.3.1}$$

Using now the fact that function $x \mapsto \sqrt{x}$ is concave and differentiable, we have for $x > 0$ and $y > 0$ that $\sqrt{y} \leq \sqrt{x} + \frac{y-x}{2\sqrt{x}}$. Thus, if $y > x > 0$ then $0 < \sqrt{y} - \sqrt{x} \leq \frac{y-x}{2\sqrt{x}}$ and if $x > y > 0$, then $0 < \sqrt{x} - \sqrt{y} \leq \frac{x-y}{2\sqrt{y}}$. Consequently, we have $|\sqrt{y} - \sqrt{x}| \leq \frac{|x-y|}{2\min(\sqrt{x},\sqrt{y})}$ and we deduce that,

$$\left|\sqrt{\boldsymbol{\alpha}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\alpha}_{-0} + \sigma_\alpha^2} - \sqrt{\boldsymbol{\gamma}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\gamma}_{-0} + \sigma_\gamma^2}\right| \leq B_n^k \left(\|\boldsymbol{\alpha} - \boldsymbol{\gamma}\| + |\sigma_\alpha^2 - \sigma_\gamma^2|\right) \tag{7.3.2}$$

where $B_n^k = O_p(1)$.

Combining previous inequalities, we get

$$\left|\varphi_n(\boldsymbol{\gamma},\sigma_\gamma^2)-\varphi_n(\boldsymbol{\alpha},\sigma_\alpha^2)\right| \leq \left(\|\boldsymbol{\alpha}-\boldsymbol{\gamma}\|+|\sigma_\alpha^2-\sigma_\gamma^2|\right)\sum_{k=1}^{K}\pi_k\left(B_n^k+A_n^k\right), \tag{7.3.3}$$

with $\sum_{k=1}^{K}\pi_k\left(B_n^k+A_n^k\right)=O_p(1)$. *As a result, it can be deduced from Lemma 2.9 in Newey and McFadden (1994) that*

$$\sup_{(\boldsymbol{\gamma},\sigma_\gamma^2)\in\Theta}\left|\varphi_n(\boldsymbol{\gamma},\sigma_\gamma^2)-\varphi(\boldsymbol{\gamma},\sigma_\gamma^2)\right|\to 0 \quad \textit{in probability.}$$

We conclude the proof by recalling that $\varphi(\boldsymbol{\gamma},\sigma_\gamma^2)$ *attains its unique minimum at* $(\boldsymbol{\beta},\sigma_\epsilon^2)\in\Theta$ *if assumption* $\mathbf{H}_1$ *is fulfilled, so that* $(\widehat{\boldsymbol{\beta}}^W,\widehat{\sigma}^{2,W})\to(\boldsymbol{\beta},\sigma_\epsilon^2)$ *in probability in view of Theorem 2.1 in Newey and McFadden (1994).* □

7.3.2. Asymptotic normality. As far as the asymptotic distribution of the estimators is concerned, and for sake of simplicity and lighter notations, we suppose now that the number of experiments is the same for all groups and all variables, that is to say $n=n_y^1=\cdots n_y^K=n_x^1\cdots=n_x^K$ and $\pi_k=1/K$, for $k=1,\ldots,K$.

PROPOSITION 7.3.1. *Assume that the assumptions of Lemma 7.3.1 are fulfilled. Then as* n *tends to infinity,*

$$\sqrt{n}\left(\widehat{\boldsymbol{\beta}}^M-\boldsymbol{\beta}\right)\rightsquigarrow\mathcal{N}\left(0,\boldsymbol{\Gamma}_{\beta_M}\right)$$

where the expression of the asymptotic covariance matrix $\boldsymbol{\Gamma}_{\beta_M}$ *is given in the proof.*

PROOF. *The central limit theorem applies directly to the independent sequences of independent random variables* $(\mathbf{X}_1^1,\cdots,\mathbf{X}_n^1),\ldots,(\mathbf{X}_1^K,\cdots,\mathbf{X}_n^K)$ *and* $(Y_1^1,\cdots,Y_n^1),,\ldots,(Y_1^K,\cdots,Y_n^K)$ *so that, as* n *tends to infinity*

$$\sqrt{n}\begin{pmatrix}\widehat{\boldsymbol{\mu}}_X^1-\boldsymbol{\mu}_X^1\\ \vdots\\ \widehat{\boldsymbol{\mu}}_X^K-\boldsymbol{\mu}_X^K\\ \widehat{\mu}_Y^1-\mu_Y^1\\ \vdots\\ \widehat{\mu}_Y^K-\mu_Y^K\end{pmatrix}\rightsquigarrow\mathcal{N}\left(0,\boldsymbol{\Gamma}_\mu\right) \tag{7.3.4}$$

where $\boldsymbol{\Gamma}_\mu$ *is a diagonal matrix, with diagonal elements* $(\boldsymbol{\Gamma}_X^1,\ldots,\boldsymbol{\Gamma}_X^K,\sigma_{Y,1}^2,\ldots,\sigma_{Y,K}^2)$, *with* $\boldsymbol{\Gamma}_X^k=Var(\mathbf{X}|G=k)=\mathbb{E}\left(\mathbf{X}^k(\mathbf{X}^k)^\top\right)-\boldsymbol{\mu}_X^k(\boldsymbol{\mu}_X^k)^\top$ *and* $\sigma_{Y,k}^2=\boldsymbol{\beta}^\top\boldsymbol{\Gamma}_X^k\boldsymbol{\beta}^\top+\sigma_\epsilon^2$. *Consider the*

application $g : \mathbb{R}^{dK+K} \to \mathbb{R}^{d+1}$ *defined by*

$$g(\boldsymbol{\mu}_X^1, \ldots, \boldsymbol{\mu}_X^K, \mu_Y^1, \ldots, \mu_Y^K) = \left(\boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_{1,X}\right)^{-1} \boldsymbol{\mu}_{1,X}^\top \boldsymbol{\pi} \boldsymbol{\mu}_Y.$$

Application g *is differentiable at* $\boldsymbol{\theta} = (\boldsymbol{\mu}_X^1, \ldots, \boldsymbol{\mu}_X^K, \mu_Y^1, \ldots, \mu_Y^K)$*, with non null Jacobian matrix (see Chapter 8 and more particularly Theorem 8.3 in Magnus and Neudecker (2019)) denoted by* $\mathbf{J}_\theta$*. The application of the Delta method (see Theorem 3.1 in van der Vaart (1998)) permits to get*

$$\sqrt{n}\left(\widehat{\boldsymbol{\beta}}^M - \boldsymbol{\beta}\right) \rightsquigarrow \mathcal{N}\left(0, \boldsymbol{\Gamma}_{\beta_M}\right),$$

where $\boldsymbol{\Gamma}_{\beta_M} = \mathbf{J}_\theta \boldsymbol{\Gamma}_\mu \mathbf{J}_\theta^\top$. □

REMARK 7.3.1. *The weak convergence toward a Gaussian result presented in Lemma 7.3.1 would remain true, at the expense of heavier notations, provided that there exist two constants,* $0 < c \leq C$ *such that*

$$0 < c \leq \frac{\max(n_y^1, \cdots, n_y^K, n_x^1, \cdots, n_x^K)}{\min(n_y^1, \cdots, n_y^K, n_x^1, \cdots, n_x^K)} \leq C < +\infty, \tag{7.3.5}$$

and $\min(n_y^1, \cdots, n_y^K, n_x^1, \cdots, n_x^K) \to \infty$.

It can be noted that the expression of $\boldsymbol{\Gamma}_{\beta_M}$ is complicated and thus difficult to compute manually when $d > 1$.

7.3.2.1. *Optimal transport estimators.* The asymptotic normality of $\widehat{\boldsymbol{\beta}}^W$ relies on classical results for M-estimators recalled in the Appendix (see Theorem D.0.4).

PROPOSITION 7.3.2. *If model (7.1.1) holds and hypothesis* $\mathbf{H}_1$ *is fulfilled,* $\mathbb{E}(Y^2) < +\infty$ *and* $\mathbb{E}(\|\mathbf{X}\|^4) < +\infty$*,* $(\boldsymbol{\beta}, \sigma_\epsilon^2) \in \Theta$ *and* Θ *is a compact set that does not contain* $(0, 0)$*, then, as* n *tends to infinity,*

$$\sqrt{n}\left(\begin{pmatrix}\widehat{\boldsymbol{\beta}}^W \\ \widehat{\sigma}_\epsilon^{2,W}\end{pmatrix} - \begin{pmatrix}\boldsymbol{\beta} \\ \sigma_\epsilon^2\end{pmatrix}\right) \rightsquigarrow \mathcal{N}\left(0, \boldsymbol{\Gamma}_W\right),$$

for some covariance matrix $\boldsymbol{\Gamma}_W$.

PROOF. *The proof consists in checking the different points of Theorem D.0.4. Point (i) is satisfied by the hypotheses, and the point (ii) follows directly from the fact* $\varphi_n(\boldsymbol{\gamma}, \sigma^2)$ *is twice-differentiable in a neighborhood of* $(\boldsymbol{\beta}, \sigma_\epsilon^2)$*. To show that (iii) is fulfilled, we consider the*

following expansion, based on the empirical version of the gradient of φ given in (7.1.11),

(7.3.6)

$$\nabla\varphi_n = \begin{pmatrix} -2\sum_{k=1}^K \pi_k \left(\widehat{\mu}_Y^k - \beta_0 - \boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\mu}}_X^k\right) \\ -2\sum_{k=1}^K \pi_k \left[\left(\widehat{\mu}_Y^k - \beta_0 - \boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\mu}}_X^k\right)\widehat{\boldsymbol{\mu}}_X^k + \left(\frac{\widehat{\sigma}_{Y,k}}{\sqrt{\boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2}} - 1\right)\widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0}\right] \\ \sum_{k=1}^K \pi_k \left(1 - \frac{\widehat{\sigma}_{Y,k}}{\sqrt{\boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2}}\right) \end{pmatrix}$$

Since model (7.1.1) holds, $\nabla\varphi = 0$ and $\widehat{\mu}_Y^k - \beta_0 - \boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\mu}}_X^k = (\widehat{\mu}_Y^k - \mu_Y^k) - \boldsymbol{\beta}_{-0}^\top \left(\widehat{\boldsymbol{\mu}}_X^k - \boldsymbol{\mu}_X^k\right)$, we thus deduce with (7.3.4) the asymptotic normality of the first component of the gradient $\nabla\varphi_n$, that is to say $\sqrt{n}\left(-2\sum_{k=1}^K \pi_k \left(\widehat{\mu}_Y^k - \beta_0 - \boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\mu}}_X^k\right)\right)$ converges in distribution to a centered Gaussian distribution. As far as the second component is concerned, it can be noted that $\widehat{\boldsymbol{\Gamma}}_X^k$ converges in probability to $\boldsymbol{\Gamma}_X^k$ and by the continuous mapping theorem, $\sqrt{\boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2} \to \sigma_{Y,k}$ in probability. It can also be noted that, under the moment condition $\mathbb{E}\left[\|\mathbf{X}\|^4 | G = k\right] < \infty$, the central limit theorem gives that $\sqrt{n}\left(\widehat{\boldsymbol{\Gamma}}_X^k - \boldsymbol{\Gamma}_X^k\right)$ converges in distribution to a centered Gaussian multivariate distribution, and we deduce with the Cramer-Wold device, the continuous mapping theorem and Slutsky's theorem that the second component of $\nabla\varphi_n$ multiplied by $\sqrt{n}$ also in distribution to a centered Gaussian random vector. It is immediate to deduce that the same convergence result holds for the third component, which is to say that $\sqrt{n}\left(\sum_{k=1}^K \pi_k \left(1 - \frac{\widehat{\sigma}_{Y,k}}{\sqrt{\boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2}}\right)\right)$ converges in distribution to a centered Gaussian random variable. We finally deduce, with the Cramer-Wold device, that (iii) is fulfilled.

To prove that (iv) also holds, consider the Hessian matrix of functional φ_n, evaluated at $(\boldsymbol{\beta}, \sigma_\epsilon^2)$:

$$\nabla_{00}\varphi_n = \begin{pmatrix} 2 & 2\left(\sum_{k=1}^K \pi_k \widehat{\boldsymbol{\mu}}_X^k\right)^\top & 0 \\ 2\sum_{k=1}^K \pi_k \widehat{\boldsymbol{\mu}}_X^k & \widehat{\mathbf{H}}(\boldsymbol{\beta}_{-0}) & \sum_{k=1}^K \pi_k \widehat{\sigma}_{Y,k}\left(\boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)^{-3/2} \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0} \\ 0 & \sum_{k=1}^K \pi_k \widehat{\sigma}_{Y,k}\left(\boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)^{-3/2}\left(\widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0}\right)^\top & \frac{1}{2}\sum_{k=1}^K \pi_k \widehat{\sigma}_{Y,k}\left(\boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)^{-3/2} \end{pmatrix},$$

where

$$\widehat{\mathbf{H}}(\boldsymbol{\beta}_{-0}) = 2\sum_{k=1}^K \pi_k \left[\widehat{\sigma}_{Y,k}\left(\boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)^{-3/2}\left[\left(\widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0}\right)\left(\widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0}\right)^\top - \left(\boldsymbol{\beta}_{-0}^\top \widehat{\boldsymbol{\Gamma}}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)\widehat{\boldsymbol{\Gamma}}_X^k\right] + \widehat{\boldsymbol{\mu}}_X^k \left(\widehat{\boldsymbol{\mu}}_X^k\right)^\top + \widehat{\boldsymbol{\Gamma}}_X^k\right].$$

By similar arguments as those used to show that $\varphi_n(\boldsymbol{\beta}, \sigma_\epsilon^2)$ converges in probability to $\varphi(\boldsymbol{\beta}, \sigma_\epsilon^2)$, we deduce that $\nabla_{00}\varphi_n$ converges in probability to some matrix $\mathbf{H}(\boldsymbol{\beta}, \sigma_\epsilon^2)$, defined as follows

$$\mathbf{H}(\boldsymbol{\beta}, \sigma_\epsilon^2) = \begin{pmatrix} 2 & 2\left(\sum_{k=1}^K \pi_k \boldsymbol{\mu}_X^k\right)^\top & 0 \\ 2\sum_{k=1}^K \pi_k \boldsymbol{\mu}_X^k & \mathbf{H}(\boldsymbol{\beta}_{-0}) & \sum_{k=1}^K \pi_k \sigma_{Y,k} \left(\boldsymbol{\beta}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)^{-3/2} \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} \\ 0 & \sum_{k=1}^K \pi_k \sigma_{Y,k} \left(\boldsymbol{\beta}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)^{-3/2} \left(\boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)^\top & \frac{1}{2}\sum_{k=1}^K \pi_k \sigma_{Y,k} \left(\boldsymbol{\beta}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)^{-3/2} \end{pmatrix}$$

where

$$\begin{aligned}\mathbf{H}(\boldsymbol{\beta}_{-0}) =& 2\sum_{k=1}^K \pi_k \Bigg(\sigma_{Y,k} \left(\boldsymbol{\beta}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)^{-3/2} \left[\left(\boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right) \left(\boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)^\top - \left(\boldsymbol{\beta}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right) \boldsymbol{\Gamma}_X^k \right] \\ &+ \boldsymbol{\mu}_X^k \left(\boldsymbol{\mu}_X^k\right)^\top + \boldsymbol{\Gamma}_X^k \Bigg).\end{aligned}$$

We now must check that $\mathbf{H}(\boldsymbol{\beta}, \sigma_\epsilon^2)$ is a positive definite matrix. For that we show that at the minimizer value $(\boldsymbol{\beta}, \sigma_\epsilon^2)$ its determinant is strictly positive. We first note that $\sigma_{Y,k} = \left(\boldsymbol{\beta}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)^{1/2}$ so that $\sigma_{Y,k} \left(\boldsymbol{\beta}_{-0}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} + \sigma_\epsilon^2\right)^{-3/2} = \frac{1}{\sigma_{Y,k}^2}$ and $\mathbf{H}(\boldsymbol{\beta}_{-0})$ can be written in a simpler form,

$$\mathbf{H}(\boldsymbol{\beta}_{-0}) = 2\sum_{k=1}^K \pi_k \left[\boldsymbol{\mu}_X^k \left(\boldsymbol{\mu}_X^k\right)^\top + \frac{1}{\sigma_{Y,k}^2} \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} \left(\boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)^\top \right], \tag{7.3.7}$$

which is a definite positive matrix under hypothesis $\mathbf{H}_1$. Using a block matrix determinant formula, we have

$$\left|\mathbf{H}(\boldsymbol{\beta}, \sigma_\epsilon^2)\right| = \begin{vmatrix} 2 & 0 \\ 0 & \frac{1}{2}\sum_{k=1}^K \frac{\pi_k}{\sigma_{Y,k}^2} \end{vmatrix} \left| \mathbf{H}(\boldsymbol{\beta}_{-0}) - \mathbf{C} \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{2}{\sum_k \frac{\pi_k}{\sigma_{Y,k}^2}} \end{pmatrix} \mathbf{C}^\top \right| \tag{7.3.8}$$

where $\mathbf{C} = \begin{pmatrix} 2\sum_{k=1}^K \pi_k \boldsymbol{\mu}_X^k & \sum_{k=1}^K \frac{\pi_k}{\sigma_{Y,k}^2} \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} \end{pmatrix}$ and it only has to be verified that the second determinant at the righthand side of (7.3.8) is strictly positive. We now have to show that

$$\begin{aligned}\mathbf{H}(\boldsymbol{\beta}_{-0}) - \mathbf{C} \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{2}{\sum_k \frac{\pi_k}{\sigma_{Y,k}^2}} \end{pmatrix} \mathbf{C}^\top =& 2\sum_{k=1}^K \pi_k \boldsymbol{\mu}_X^k \left(\boldsymbol{\mu}_X^k\right)^\top - 2\left(\sum_{k=1}^K \pi_k \boldsymbol{\mu}_X^k\right)\left(\sum_{k=1}^K \pi_k \boldsymbol{\mu}_X^k\right)^\top \\ &+ 2\sum_{k=1}^K \frac{\pi_k}{\sigma_{Y,k}^2} \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} \left(\boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)^\top - \frac{2}{\sum_k \frac{\pi_k}{\sigma_{Y,k}^2}} \left(\sum_{k=1}^K \frac{\pi_k}{\sigma_{Y,k}^2} \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)\left(\sum_{k=1}^K \frac{\pi_k}{\sigma_{Y,k}^2} \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)^\top\end{aligned} \tag{7.3.9}$$

is a positive matrix. We can remark that by Cauchy Schwarz inequality, for $\mathbf{u} \in \mathbb{R}^d$,

$$\begin{aligned}\mathbf{u}^\top \left(\sum_{k=1}^K \pi_k \boldsymbol{\mu}_X^k\right)\left(\sum_{k=1}^K \pi_k \boldsymbol{\mu}_X^k\right)^\top \mathbf{u} &= \left(\sum_{k=1}^K \pi_k \mathbf{u}^\top \boldsymbol{\mu}_X^k\right)^2 \\ &\leq \sum_{k=1}^K \pi_k \left(\mathbf{u}^\top \boldsymbol{\mu}_X^k\right)^2 \\ &= \mathbf{u}^\top \left(\sum_{k=1}^K \pi_k \boldsymbol{\mu}_X^k \left(\boldsymbol{\mu}_X^k\right)^\top\right) \mathbf{u}\end{aligned}$$

using the fact that $\sum_k (\sqrt{\pi_k})^2 = 1$. *It can be noted that if* $\mathbf{u} \neq 0$, *previous inequality is strict unless* $\mathbf{u}^\top \boldsymbol{\mu}_X^1 = \cdots = \mathbf{u}^\top \boldsymbol{\mu}_X^K$, *which can not happen under hypothesis* $\mathbf{H}_1$. *The second part at the righthand side of (7.3.9) is handled the same way. We have*

$$\begin{aligned}\mathbf{u}^\top \left(\sum_{k=1}^K \frac{\pi_k}{\sigma_{Y,k}^2} \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)\left(\sum_{k=1}^K \frac{\pi_k}{\sigma_{Y,k}^2} \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)^\top \mathbf{u} &= \left(\mathbf{u}^\top \left(\sum_{k=1}^K \frac{\pi_k}{\sigma_{Y,k}^2} \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)\right)^2 \\ &\leq \sum_{k=1}^K \sqrt{\frac{\pi_k}{\sigma_{Y,k}^2}}^2 \sum_{k=1}^K \left(\sqrt{\frac{\pi_k}{\sigma_{Y,k}^2}}^2 \mathbf{u}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)^2 \\ &= \sum_{k=1}^K \frac{\pi_k}{\sigma_{Y,k}^2} \sum_{k=1}^K \frac{\pi_k}{\sigma_{Y,k}^2} \mathbf{u}^\top \boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0} \left(\boldsymbol{\Gamma}_X^k \boldsymbol{\beta}_{-0}\right)^\top \mathbf{u},\end{aligned}$$

and consequently the determinant of $\mathbf{H}(\boldsymbol{\beta}, \sigma_\epsilon^2)$ *is strictly positive.*

To finish the proof, it remains to check that in a neighborhood $\mathcal{N}$ *of* $(\boldsymbol{\beta}, \sigma_\epsilon^2)$, *we have*

$$\sup_{(\boldsymbol{\gamma}, \sigma_\gamma^2) \in \mathcal{N}} \|\nabla_{00} \varphi_n(\boldsymbol{\gamma}, \sigma_\gamma^2) - \mathbf{H}(\boldsymbol{\gamma}, \sigma_\gamma^2)\| \to 0 \ \textit{in probability.}$$

This is a direct consequence of the continuous mapping theorem, which gives us that for all $(\boldsymbol{\gamma}, \sigma_\gamma^2) \in \mathcal{N}$, $\|\nabla_{00} \varphi_n(\boldsymbol{\gamma}, \sigma_\gamma^2) - \mathbf{H}(\boldsymbol{\gamma}, \sigma_\gamma^2)\| \to 0$ *in probability, and the fact that third order partial derivatives of* $\varphi_n(\boldsymbol{\gamma}, \sigma_\gamma^2)$ *are bounded in probability for* $(\boldsymbol{\gamma}, \sigma_\gamma^2)$ *so that Theorem D.0.2 can apply.*

□

7.4. Bootstrapping for confidence intervals

Since, as noted in the previous section, it is complicated to compute explicitly the asymptotic variance matrix of $\widehat{\boldsymbol{\beta}}^M$ and $\widehat{\boldsymbol{\beta}}^W$, we consider stratified bootstrap approaches in order to build confidence sets for $\boldsymbol{\beta}$.

Our bootstrap procedure is based on the sampling scheme for all considered estimators of $\boldsymbol{\beta}$ and takes account of the independence between the different groups $k = 1, \ldots, K$, as well as the independence of inputs $(X_1^k, \ldots, X_d^k)$ and output Y^k within

each group, meaning more formally that, given $G = k$, the joint probability measure $\mathbb{P}^k$ of Y and $\mathbf{X}$ is a product measure of the marginal measures $\mathbb{P}^k = \mathbb{P}^k_Y \otimes \mathbb{P}^k_{\mathbf{X}}$.

For each group k, we draw, with equal probability and with replacement, n_y^k observations among $Y_1^k, \ldots, Y_{n_y^k}^k$, and denote by μ_Y^{k*} the empirical mean of this bootstrap sample. Then we draw, with equal probability and with replacement, n_x^k observations among $\mathbf{X}_1^k, \ldots, \mathbf{X}_{n_x^k}^k$ and denote by $\boldsymbol{\mu}_X^{k*}$ the empirical mean of this bootstrap sample. Bootstrapped estimators $\boldsymbol{\beta}^{M,*}$ and $\boldsymbol{\beta}^{W,*}$ of $\boldsymbol{\beta}$ can now be computed.

To build confidence sets for each component of $\boldsymbol{\beta}$ based on previous bootstrap procedure, the bootstrap percentile technique, described in Chapter 4 of Shao and Tu (1995), can be considered.

It can be noted that our estimators are smooth functions of sample means so that classical bootstrap theory applies (see for example Shao and Tu (1995), Chapter 3). For simplicity, as in Proposition 7.3.1, we suppose that $n = n_y^1 = \ldots = n_y^K = n_x^1 = \ldots = n_x^K$. Because of the considered experimental design, our global "empirical distribution" is made of products of marginal empirical distributions, the bootstrap for means is almost surely consistent for the Kolmogorov metric, and with Theorem 3.1 in Shao and Tu (1995) the same result holds for the estimators of $\boldsymbol{\beta}$ considered in this work. Then, the application of Theorem 4.1 in Shao and Tu (1995) allows to conclude that bootstrap percentile method gives consistent confidence sets for each component of $\boldsymbol{\beta}$.

PROPOSITION 7.4.1. *Suppose that $\mathbb{E}(Y^2) < \infty$ and $\mathbb{E}\|\mathbf{X}\|^2 < \infty$ and hypothesis $\mathbf{H}_1$ is fulfilled. Then as $n \to +\infty$, the bootstrap estimator $\boldsymbol{\beta}^{M,*}$ is strongly consistent for $\boldsymbol{\beta}$ in the Kolmogorov metric. Furthermore, for each component of $\boldsymbol{\beta}$, the bootstrap percentile approach provides, for a given nominal level $1-\alpha$, a consistent confidence set.*

PROOF. The fact that the bootstrap estimator $\boldsymbol{\beta}^{M,*}$ is strongly consistent for $\boldsymbol{\beta}$ is a direct consequence of Theorem 3.1 in Shao and Tu (1995), noting that

$$\widehat{\boldsymbol{\beta}}^M = g(\widehat{\boldsymbol{\mu}}_X^1, \ldots, \widehat{\boldsymbol{\mu}}_X^K, \widehat{\mu}_Y^1, \ldots, \widehat{\mu}_Y^K)$$

is a continuously differentiable function of means at $(\boldsymbol{\mu}_X^1, \ldots, \boldsymbol{\mu}_X^K, \mu_Y^1, \ldots, \mu_Y^K)$. The fact that confidence sets based on the percentile approach are consistent is proved by checking the assumptions in Theorem 4.1 (iii) Shao and Tu (1995), namely the bootstrap estimator $\boldsymbol{\beta}^{M,*}$ is consistent, $\widehat{\boldsymbol{\beta}}^M$ is consistent (Lemma 7.3.1), with asymptotic Gaussian distribution (Proposition 7.3.1). □

PROPOSITION 7.4.2. *Suppose that $\mathbb{E}(Y^2) < \infty$ and $\mathbb{E}\|\mathbf{X}\|^4 < \infty$ and hypothesis $\mathbf{H}_1$ is fulfilled. Then as $n \to +\infty$, the bootstrap estimator $(\boldsymbol{\beta}^{W,*}, \sigma_\epsilon^{2,W,*})$ is strongly consistent*

for $(\boldsymbol{\beta}, \sigma_\epsilon^2)$ in the Kolmogorov metric. Furthermore, for each component of $\boldsymbol{\beta}$, the bootstrap percentile approach provides, for a given nominal level $1-\alpha$, a consistent confidence set.

PROOF. We denote by $\boldsymbol{\theta}_0 = (\boldsymbol{\beta}, \sigma_\epsilon^{2,W})$ the vector of true parameters, by $\widehat{\boldsymbol{\theta}} = (\boldsymbol{\beta}^W, \sigma_\epsilon^2)$ the sequence of minimum Wasserstein distance estimators and by $\boldsymbol{\theta}^* = (\boldsymbol{\beta}^{W,*}, \sigma_\epsilon^{2,W,*})$ bootstrap estimators of $\boldsymbol{\theta}_0$. The vector of parameters $\boldsymbol{\theta}^*$ is the minimizer of functional φ_n^* defined as follows,

$$\varphi_n^*(\boldsymbol{\gamma}, \sigma^2) = \sum_{k=1}^{K} \pi_k \left[(\mu_Y^{k,*} - \gamma_0 - \boldsymbol{\gamma}_{-0}^\top \boldsymbol{\mu}_X^{k,*})^2 + \left(\sigma_{Y,k}^* - \sqrt{\boldsymbol{\gamma}_{-0}^\top \boldsymbol{\Gamma}_X^{k,*} \boldsymbol{\gamma}_{-0} + \sigma^2} \right)^2 \right]. \tag{7.4.1}$$

We first show with arguments similar to those employed in the proof of Lemma 7.3.2, that $\boldsymbol{\theta}^*$ is a consistent estimator for $\boldsymbol{\theta}_0$, based on the fact that φ_n^* is a smooth function converging to φ and the sample mean theorem for bootstrap (see for example Theorem 23.4 in van der Vaart (1998)). Indeed, we first recall that for all $(\boldsymbol{\gamma}, \sigma_\gamma^2) \in \Theta$, $\varphi_n(\boldsymbol{\gamma}, \sigma_\gamma^2) \to \varphi(\boldsymbol{\gamma}, \sigma_\gamma^2)$ in probability, when $n_{\min}$ tends to infinity and

$$\left|\varphi_n^*(\boldsymbol{\gamma}, \sigma_\gamma^2) - \varphi(\boldsymbol{\gamma}, \sigma_\gamma^2)\right| \leq \left|\varphi_n^*(\boldsymbol{\gamma}, \sigma_\gamma^2) - \varphi_n(\boldsymbol{\gamma}, \sigma_\gamma^2)\right| + \left|\varphi_n(\boldsymbol{\gamma}, \sigma_\gamma^2) - \varphi(\boldsymbol{\gamma}, \sigma_\gamma^2)\right|. \tag{7.4.2}$$

Since the bootstrap means converge to the empirical ones we deduce with the continuous mapping theorem that $\varphi_n^*(\boldsymbol{\gamma}, \sigma_\gamma^2) \to \varphi_n(\boldsymbol{\gamma}, \sigma_\gamma^2)$ in probability, when $n_{\min}$ tends to infinity, so that $\varphi_n^*(\boldsymbol{\gamma}, \sigma_\gamma^2) \to \varphi(\boldsymbol{\gamma}, \sigma_\gamma^2)$. We also have, as in (7.3.3), where empirical means are replaced by bootstrap means,

$$\left|\varphi_n^*(\boldsymbol{\gamma}, \sigma_\gamma^2) - \varphi_n^*(\boldsymbol{\alpha}, \sigma_\alpha^2)\right| \leq \left(\|\boldsymbol{\alpha} - \boldsymbol{\gamma}\| + |\sigma_\alpha^2 - \sigma_\gamma^2|\right) \sum_{k=1}^{K} \pi_k \left(B_n^{k,*} + A_n^{k,*}\right), \tag{7.4.3}$$

for any $(\boldsymbol{\alpha}, \sigma_\alpha^2) \in \Theta$, with $\sum_{k=1}^K \pi_k \left(B_n^{k,*} + A_n^{k,*}\right) = O_p(1)$. As a result, we deduce from Lemma 7.3.2, inequality (7.4.2) and Lemma 2.9 in Newey and McFadden (1994) that

$$\sup_{(\boldsymbol{\gamma}, \sigma_\gamma^2) \in \Theta} \left|\varphi_n^*(\boldsymbol{\gamma}, \sigma_\gamma^2) - \varphi(\boldsymbol{\gamma}, \sigma_\gamma^2)\right| \to 0 \quad \text{in probability.}$$

We conclude that $\boldsymbol{\theta}^* \to \boldsymbol{\theta}_0$ in probability in view of Theorem 2.1 in Newey and McFadden (1994).

We now prove that $\sqrt{n}\left(\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}\right)$ and $\sqrt{n}\left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\right)$ have the same asymptotic distribution. By definition of $\widehat{\boldsymbol{\theta}}$ and Taylor expansion we have

$$\nabla \varphi_n(\widehat{\boldsymbol{\theta}}) = \nabla \varphi_n(\boldsymbol{\theta}_0) + \nabla_{00} \varphi_n(\overline{\boldsymbol{\theta}}) \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\right) = 0 \tag{7.4.4}$$

where $\overline{\boldsymbol{\theta}}$ belongs, componentwise, to the segment between $\boldsymbol{\theta}_0$ and $\widehat{\boldsymbol{\theta}}$. We have a similar expansion for boostrap estimators, as well as

$$\nabla\varphi_n^*(\boldsymbol{\theta}^*) = \nabla\varphi_n^*(\boldsymbol{\theta}_0) + \nabla_{00}^*\varphi_n(\overline{\boldsymbol{\theta}}^*)\,(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0) = 0 \tag{7.4.5}$$

where $\overline{\boldsymbol{\theta}}^*$ belongs, componentwise, to the segment between $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}^*$. Combining (7.4.4) and (7.4.5), we deduce

$$\begin{aligned}\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}} &= \left(\nabla_{00}^*\varphi_n(\overline{\boldsymbol{\theta}}^*)\right)^{-1}\nabla\varphi_n^*(\boldsymbol{\theta}_0) - \left(\nabla_{00}\varphi_n(\overline{\boldsymbol{\theta}})\right)^{-1}\nabla\varphi_n(\boldsymbol{\theta}_0) \\ &= \left(\left(\nabla_{00}^*\varphi_n(\overline{\boldsymbol{\theta}}^*)\right)^{-1} - \left(\nabla_{00}\varphi_n(\overline{\boldsymbol{\theta}})\right)^{-1}\right)\nabla\varphi_n^*(\boldsymbol{\theta}_0) + \left(\nabla_{00}\varphi_n(\overline{\boldsymbol{\theta}})\right)^{-1}\left(\nabla\varphi_n^*(\boldsymbol{\theta}_0) - \nabla\varphi_n(\boldsymbol{\theta}_0)\right)\end{aligned} \tag{7.4.6}$$

Noticing that $\nabla_{00}^*\varphi_n(\overline{\boldsymbol{\theta}}^*)$ and $\nabla_{00}\varphi_n(\overline{\boldsymbol{\theta}})$ both tend in probability to the same limit $\mathbf{H}(\boldsymbol{\beta}, \sigma_\epsilon^2)$ and we have, with similar arguments as those used in the proof of Proposition 7.3.2, that $\nabla\varphi_n^*(\boldsymbol{\theta}_0)$ is $O_p(n^{-1/2})$, it can be deduce that

$$\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}} = \left(\nabla_{00}\varphi_n(\overline{\boldsymbol{\theta}})\right)^{-1}\left(\nabla\varphi_n^*(\boldsymbol{\theta}_0) - \nabla\varphi_n(\boldsymbol{\theta}_0)\right) + o_P(n^{-1/2}). \tag{7.4.7}$$

Using arguments similar to those employed in the expansion of $\nabla\varphi_n$ in the proof of Proposition 7.3.2, we make appear the difference between bootstrap means and empirical means or a differentiable functional of these quantities:

$$\nabla\varphi_n^*(\boldsymbol{\theta}_0) - \nabla\varphi_n(\boldsymbol{\theta}_0) = \begin{pmatrix} 2\sum_{k=1}^K \pi_k \left((\widehat{\mu}_Y^k - \mu_Y^{k,*}) - \beta_0 - \boldsymbol{\beta}_{-0}^\top\left(\widehat{\boldsymbol{\mu}}_X - \boldsymbol{\mu}_X^{k,*}\right)\right) \\ \begin{pmatrix} 2\sum_{k=1}^K \pi_k \left[\left(\widehat{\mu}_Y^k - \mu_Y^k - \beta_0 - \boldsymbol{\beta}_{-0}^\top\widehat{\boldsymbol{\mu}}_X^k\right)\widehat{\boldsymbol{\mu}}_X^k + \left(\frac{\widehat{\sigma}_{Y,k}}{\sqrt{\boldsymbol{\beta}_{-0}^\top\widehat{\boldsymbol{\Gamma}}_X^k\boldsymbol{\beta}_{-0}+\sigma_\epsilon^2}} - 1\right)\widehat{\boldsymbol{\Gamma}}_X^k\boldsymbol{\beta}_{-0}\right] \\ -2\sum_{k=1}^K \pi_k \left[\left(\mu_Y^{k,*} - \mu_Y^k - \beta_0 - \boldsymbol{\beta}_{-0}^\top\boldsymbol{\mu}_X^{k,*}\right)\boldsymbol{\mu}_X^{k,*} + \left(\frac{\sigma_{Y,k}^*}{\sqrt{\boldsymbol{\beta}_{-0}^\top\boldsymbol{\Gamma}_X^{k,*}\boldsymbol{\beta}_{-0}+\sigma_\epsilon^2}} - 1\right)\boldsymbol{\Gamma}_X^{k,*}\boldsymbol{\beta}_{-0}\right] \end{pmatrix} \\ \sum_{k=1}^K \left(\frac{\widehat{\sigma}_{Y,k}}{\sqrt{\boldsymbol{\beta}_{-0}^\top\widehat{\boldsymbol{\Gamma}}_X^k\boldsymbol{\beta}_{-0}+\sigma_\epsilon^2}} - \frac{\sigma_{Y,k}^*}{\sqrt{\boldsymbol{\beta}_{-0}^\top\boldsymbol{\Gamma}_X^{k,*}\boldsymbol{\beta}_{-0}+\sigma_\epsilon^2}}\right) \end{pmatrix}, \tag{7.4.8}$$

which satisfies the central limit theorem for bootstrap means, or the Delta method for bootstrap estimators (see the Appendix as well as Theorem 23.4 and Theorem 23.5 in van der Vaart (1998)). Consequently, $\nabla\varphi_n^*(\boldsymbol{\theta}_0) - \nabla\varphi_n(\boldsymbol{\theta}_0)$ and $\nabla\varphi_n(\boldsymbol{\theta}_0) - \nabla\varphi(\boldsymbol{\theta}_0)$ have the same asymptotic distribution. By Slustky's theorem, the asymptotic distribution of $\sqrt{n}\left(\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}\right)$ is thus the same as the asymptotic distribution of $\mathbf{H}(\boldsymbol{\beta}, \sigma_\epsilon^2)\sqrt{n}\nabla\varphi_n(\boldsymbol{\theta}_0)$, and we can conclude that $\sqrt{n}\left(\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}\right)$ and $\sqrt{n}\left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\right)$ have also the same asymptotic Gaussian distribution.

□

CHAPTER 8

A SIMULATION STUDY

8.1. Simulation design

8.1.1. General setting. Some simulations have been performed in order to evaluate the finite sample performances of the proposed approaches on data that resemble *in vivo* data originating from real experiments on mice. Let $K \in \mathbb{N}$ be the number of subpopulations, $g_k = \{0, 1, \ldots, k-1\}$ be the level of the grouping variable for subpopulation $k \in \{1, \ldots, K\}$, and the number of animals observed per group was choose for simplicity $n = n_y^1 = \cdots = n_y^K = n_x^1 \cdots = n_x^K$. For every animal $i \in \{1, \ldots, n\}$ and every subpopulation $k \in \{1, \ldots, K\}$ we choose the predictor variable X_i^k in the Gaussian univariate setting:

$$X_i^k \sim \mathcal{N}(\mu_X^k, \sigma_{X,k}^2), \text{ where } \mu_X^k = \Delta_\mu g_k + C_\mu,$$

where $C_\mu \in \mathbb{R}$ is the global mean scale parameter, and $\Delta_\mu \in \mathbb{R}$ is the parameter that determines the difference between groups in terms of the mean. We simulate the predicted variable independently as follows, with regression parameters β_0 and β_1:

$$Y_i^k = \beta_0 + \beta_1 X_i'^k + \epsilon_i^k, \text{ where } X_i'^k \sim X_i^k \text{ and } \epsilon_i^k \sim \mathcal{N}(0, \sigma_{\epsilon,k}^2).$$

It should be noted that X_i^k and $X_i'^k$ are independent, we thus recreate the situation of the predictor and predicted variables not being simultaneously observed.

The variable sets $X = (X_i^k)_{1 \leq i \leq n, 1 \leq k \leq K}$ and $Y = (Y_i^k)_{1 \leq i \leq n, 1 \leq k \leq K}$ are simulated $N_{sim} \in \mathbb{N}$ times. For each simulation, $N_{boot} \in \mathbb{N}$ bootstrap samples of the size n are generated from $X^k = (X_i^k)_{1 \leq k \leq K}$ and $Y^k = (Y_i^k)_{1 \leq k \leq K}$ for each subpopulation $k \in \{1, \ldots, K\}$ independently, then moment estimators μ_X^{k*} and μ_Y^{k*} are calculated. Finally,

the bootstrap sample-based estimators $\boldsymbol{\beta}^{M,*} = (\beta_0^{M,*}, \beta_1^{M,*})$ and $\boldsymbol{\beta}^{W,*} = (\beta_0^{W,*}, \beta_1^{W,*})$ are calculated. Based on N_{boot} estimates, we calculate 95% confidence intervals using the function *quantile* of the Python library NumPy. As a result, we obtain N_{sim} confidence intervals for each regression parameter, that we use to calculate the following quantities of interest:

- Coverage rate, i.e. the proportion of intervals including the true value.
- Average amplitude of the intervals.
- Power, i.e. the proportion of intervals not including 0.

Along with the bootstrap procedure, we estimate the confidence intervals for the parameter estimators of the regression on the means by group with a naive method, assuming that the deviation of the parameter estimator from the true value divided by the estimator's standard error follows a Student's t-distribution:

$$\frac{\widehat{\beta}_j - \beta_j}{SE(\widehat{\beta}_j)} \rightsquigarrow t_{K-(d+1)} \text{ for } j \in \{0, \dots, d\}.$$

8.1.2. Parameter levels. We fix the following parameters throughout all simulations: the regression parameters $\beta_0 = 1$ and $\beta_1 = 2$, the number of simulations $N_{sim} = 500$, the number of bootstrap samples $N_{boot} = 500$, the location parameters of the predictor variable $\Delta_\mu = 1$ and $C_\mu = 10$. We vary the following parameters to study their effect:

- **The number of animals.** We take $n \in \{10, 30\}$, in particular to test whether the inference is significantly impaired in case of a small number of animals, which is often the case for real experimental data.
- **The number of groups.** We consider $K \in \{4, 10\}$, 4 being the number of groups that is often observed in real data, and 10 being a higher number that may produce sufficiently good results with the naive approach to approximating confidence intervals with Student distribution.
- **The group dispersion.** The parameter $\sigma^2_{X,k}$ can be adjusted to control the extent to which the observations per group can be easily distinguish one from another. We set $\sigma^2_{X,1} = \cdots = \sigma^2_{X,K}$, and $\sigma^2_{X,k} \in \{0.75, 2\}$, the first value corresponding to lower overlap between groups, and the second to higher overlap. The difference between the two cases and the effect on simulated data are illustrated on Figure 8.1.1.

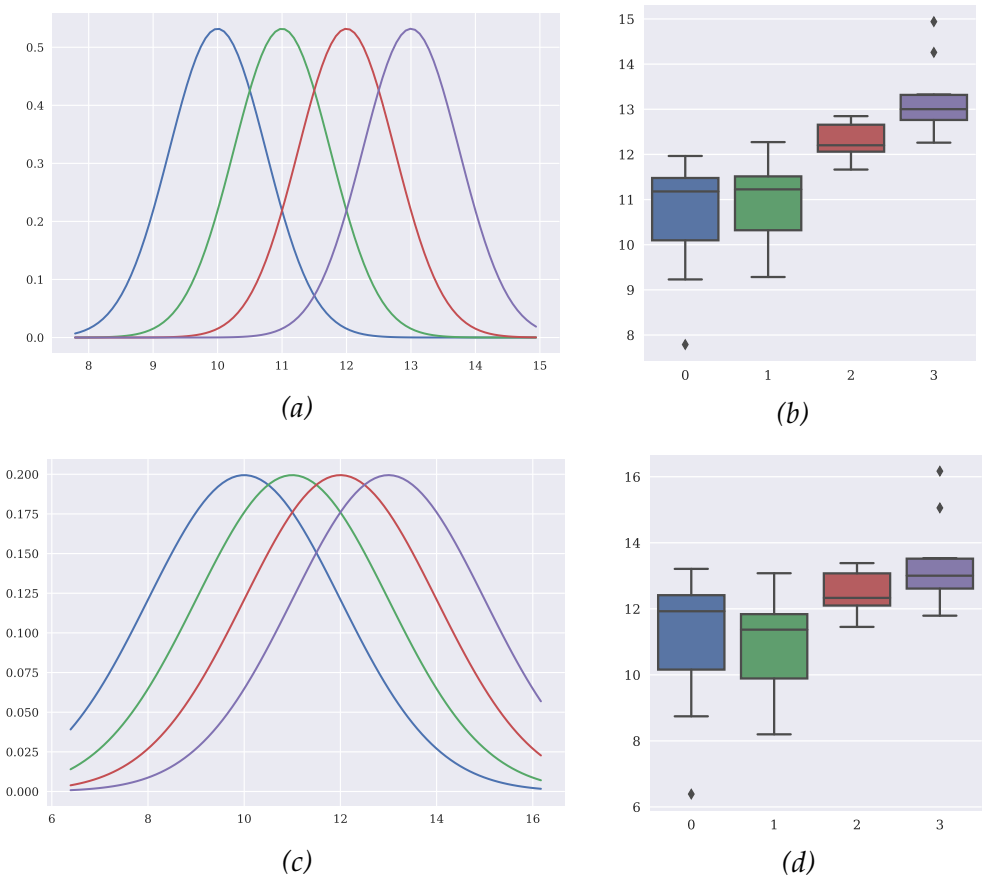
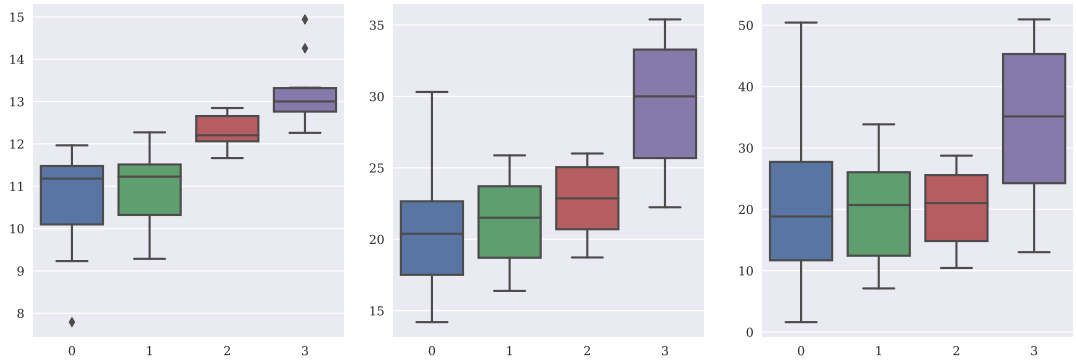


Figure (8.1.1) The effect of different values of $\sigma^2_{X,k}$ on the data, with $\sigma^2_{X,1} = \cdots = \sigma^2_{X,K}$, $K = 4$ and $\rho = 1.1$. a,b) $\sigma^2_{X,k} = 0.75$, or low overlap. c,d) $\sigma^2_{X,k} = 2$, or high overlap. a,c) True distributions of X^k for $k \in \{1, \ldots, 4\}$. b,d) Boxplots constructed from the simulated values of X^k_i.

- **The level of noise.** We introduce an additional parameter $\rho \in \mathbb{R}^+$ controlling the variance of the response to the variance of noise ratio, i.e. $\rho = \dfrac{\sigma_{Y,k}}{\sigma_{\epsilon,k}}$. The choice of adjusting the noise to signal ratio instead of the quantity of noise itself through $\sigma^2_{\epsilon,k}$ is motivated by the fact that $\sigma_{Y,k}$ depends on $\sigma_{X,k}$, hence the same level of $\sigma_{\epsilon,k}$ cannot be interpreted the same way for different values of $\sigma_{X,k}$. The variance of the noise can be expressed as follows: $\sigma^2_{\epsilon,k} = \dfrac{\beta_1^2 \sigma^2_{X,k}}{\rho^2 - 1}$. The values of ρ are chosen to correspond to the realistic situation, namely the very noisy case and a slightly less noisy one: $\rho \in \{1.01, 1.1\}$. The effect of different values of ρ on the simulated response variable is illustrated on Figure 8.1.2.

8.2. Results

The results of the simulation study are presented in Figure 8.2.1, namely the obtained coverage rates for the linear regression slope confidence intervals on Figure



(a) *(b)* *(c)*

Figure (8.1.2) The effect of different values of ρ on the data, with $K = 4$ and $\sigma^2_{X,1} = \cdots = \sigma^2_{X,4} = 0.75$. a) Boxplots constructed from the simulated values of X_i^k. b) Boxplots constructed from the simulated values of Y_i^k with lower relative noise level, i.e. $\rho = 1.1$. c) Boxplots constructed from the simulated values of Y_i^k with higher relative noise level, i.e. $\rho = 1.01$.

8.2.1a, the amplitudes on Figure 8.2.1b, and the associated test powers on Figure 8.2.1c. Every figure contains four tables, each corresponding to a combination of parameters with respect to the number of animals N_{anim} and the number of groups K. The columns of the tables represent combinations of parameters with respect to the group dispersion parameter $\sigma^2_{X,k}$ and the noise to signal ratio ρ. Finally, the lines of each table indicate the method used to estimate the confidence intervals: the proposed method of moments and optimal transport based bootstrap estimators, and the classical linear regression procedure based on estimated means within each group.

In general, it can be observed that the bootstrap estimators produce confidence intervals with smaller amplitudes and with higher power, but with lower coverage rate. Whereas the coverage rates in all cases remain within 91-96%, the extent to which the amplitudes are smaller and the powers are bigger is significantly more important for the bootstrap estimators in almost all cases. This implies that the naive approach based on the Student's distribution is more likely to produce false negatives in terms of significance.This trend of further amplified by the number of groups parameter: whereas the results are overall worsened with the decrease in either the number of animals or groups, it is the case with the small number of groups that demonstrates the biggest difference in the approaches. Indeed, in almost all cases within the tables with $K = 4$ we observe the amplitudes approximately twice as important for the naive approach, and a similar trend in terms of lower powers. The latter result is important since lower power implies bigger probability of not detecting a significant relationship between the predictor and the predicted variables, when it is actually present. Overall,

Figure (8.2.1) a) Coverage rates, b) amplitudes, and c) powers of the confidence intervals for the estimators of β_1 obtained from N_{sim} simulations for each parameter combination. The method labels on the left: "mm (boot)" stands for the method of moments with bootstrap, "ot (boot)" for the optimal transport method with bootstrap, and "mm (student)" for the naive linear regression on means approach based on Student's distribution.

these results mean that the proposed bootstrap estimators are more effective if the experimental design entails a small number of groups.

Concerning the remaining two parameters, as expected, in general the best results are obtained with lower $\sigma^2_{X,k}$ and higher ρ. In most case, the results for the naive and the bootstrap estimators are either both good or both bad in terms of power, with the latter being slightly better. A particularly complicated case can be distinguished, with high dispersion and high noise level, with few groups and few animals, where all estimators drastically fail: we observe almost equally bad powers (0.1 for both bootstrap estimators and 0.09 for the naive estimator), despite the significant difference in amplitudes. On the other hand, we can also distinguish two case where the powers of the bootstrap estimators are more than 90%, whereas those of the naive estimator are under 50%: in both cases there are 4 groups and high noise, in the first case there are only 10 animals but lower dispersion, in the second case high dispersion level is compensated but a higher number of animals. This implies that if the underlying distributions per group are characterized by a reasonable amount of overlap, or a significant overlap is compensated by having more observations, the bootstrap estimators manage to detect the significant relationship in most cases, unlike the naive estimator.

Lastly, it can be observed that the estimator based on optimal transport produces confidence intervals with slightly smaller amplitudes compared to the method of moments estimator. The difference appears to be relatively more important in the cases with higher group overlap $\sigma^2_{X,k} = 2$. However, the powers are not affected by this difference. This may be explained by a more important bias associated with the optimal transport estimator. The estimator will likely produce better results in terms of the power than the method of moments estimator if the bias is corrected.

CHAPTER 9

APPLICATION TO REAL DATA

In order to illustrate the proposed estimators on real data, we studied the data from experiments conducted on mice in order to assess the adverse effects induced in the context of different irradiated volume. In the course of these experiments, mice were exposed to either stereotactic body radiation therapy (SBRT) with different beam sizes at 90 Gy on the left lung, or whole-thorax irradiation (WTI) at 19 Gy. For one cohort, the expressions of three pro-inflammatory genes (IL1 α and β, IL6 and TNF) were measured, for the other cohort the measurements of the thickness of the alveolar septas were made due to its role as a macroscopic biomarker of radio-induced pulmonary lesions. In the case of the SBRT, the measurements were taken in multiple locations: the irradiated patch (inside the irradiation field), the remaining part of the left lung referred to as ipsilateral lung, and the right lung (contralateral lung). The goal of this statistical analysis is to determine whether there is a statistical association between the gene expression as predictors and the septal thickening as as outcome. Our approach is applied since the variables are measured on different animals, but within each irradiation condition there are shared groups in terms of the measurement time points.

The linear regression parameters were estimated with three estimators the same way it was done in the simulation study in Section 8.2. The results are presented in Table 9.0.1. The focus is placed on estimating the slope parameter β_1 in particular. The table contains the estimations of β_1 as well as the estimated confidence intervals for the slope estimator, and the corresponding test result on the significance of the estimated relationship.

			Method of moments (bootstrap)			Optimal transport (bootstrap)			Lin. Reg. on means (Student)		
Loc.	Vol.	Gene	$\widehat{\beta}_1$	95% C.I.	Signif.	$\widehat{\beta}_1$	95% C.I.	Signif.	$\widehat{\beta}_1$	95% C.I.	Signif.
Ipsilateral lung	1 mm	IL1b	-0.35	(-1.03, 0.4)	✗	-0.19	(-0.44, 0.23)	✗	-0.35	(-1.99, 1.28)	✗
		IL6	0.43	(-0.23, 1.2)	✗	0.2	(-0.2, 0.84)	✗	0.43	(-1.32, 2.17)	✗
		TNF	0.2	(-0.23, 0.84)	✗	0.16	(-0.18, 0.84)	✗	0.2	(-1.26, 1.66)	✗
	3 mm	IL1b	0.9	(0.0, 2.11)	✔	1.05	(0.01, 1.44)	✔	0.9	(-0.43, 2.24)	✗
		IL6	0.05	(-0.34, 0.46)	✗	0.06	(-0.33, 0.49)	✗	0.05	(-1.65, 1.76)	✗
		TNF	0.65	(-0.12, 1.6)	✗	0.63	(-0.12, 1.46)	✗	0.65	(-1.57, 2.87)	✗
Irradiated patch	1 mm	IL1b	2.31	(-1.73, 3.87)	✗	0.82	(-0.48, 0.94)	✗	2.31	(-0.73, 5.35)	✗
		IL6	0.85	(-0.7, 2.38)	✗	0.61	(-0.56, 1.6)	✗	0.85	(-3.75, 5.45)	✗
		TNF	0.85	(-0.6, 2.3)	✗	0.69	(-0.54, 1.53)	✗	0.85	(-3.96, 5.66)	✗
	3 mm	IL1b	2.85	(0.84, 5.14)	✔	2.59	(0.73, 3.58)	✔	2.85	(-1.07, 6.78)	✗
		IL6	1.35	(0.22, 2.47)	✔	1.3	(0.21, 2.26)	✔	1.35	(-1.8, 4.5)	✗
		TNF	3.81	(1.01, 6.37)	✔	3.37	(0.99, 5.33)	✔	3.81	(-1.86, 9.48)	✗
Right lung	0 mm	IL1b	-0.92	(-2.2, 1.06)	✗	-0.42	(-0.89, 0.51)	✗	-0.92	(-3.97, 2.14)	✗
		IL6	2.23	(-1.99, 2.28)	✗	0.83	(-0.83, 0.97)	✗	2.23	(-2.65, 7.12)	✗
		TNF	2	(-2.0, 2.82)	✗	0.88	(-1.0, 1.11)	✗	2	(-1.72, 5.72)	✗
	1 mm	IL1b	0.63	(-0.82, 2.32)	✗	0.23	(-0.33, 0.56)	✗	0.63	(-2.95, 4.21)	✗
		IL6	1.03	(-0.57, 2.19)	✗	0.46	(-0.3, 1.0)	✗	1.03	(-0.88, 2.94)	✗
		TNF	1.05	(-0.47, 2.43)	✗	0.66	(-0.31, 1.26)	✗	1.05	(-1.01, 3.11)	✗
	3 mm	IL1b	1.07	(0.17, 1.97)	✔	0.92	(0.17, 1.13)	✔	1.07	(-1.96, 4.1)	✗
		IL6	0.3	(-1.45, 1.12)	✗	0.37	(-0.74, 0.86)	✗	0.3	(-4.94, 5.53)	✗
		TNF	2.02	(0.27, 4.1)	✔	1.05	(0.04, 1.44)	✔	2.02	(0.03, 4.02)	✔
	7 mm	IL1b	6.41	(-11.6, 15.4)	✗	1.62	(-2.28, 3.1)	✗	6.41	(-16.7, 29.52)	✗
		IL6	-0.58	(-1.26, 0.09)	✗	-0.6	(-1.23, 0.1)	✗	-0.58	(-6.23, 5.08)	✗
		TNF	0.51	(-1.7, 3.86)	✗	0.43	(-1.45, 1.89)	✗	0.51	(-23.21, 24.22)	✗
Whole thorax irradiation		IL1a	2.93	(-1.18, 6.63)	✗	2.47	(-1.49, 3.02)	✗	2.93	(-11.08, 16.94)	✗
		IL6	3.7	(1.53, 5.99)	✔	2.51	(1.39, 3.43)	✔	3.7	(1.11, 6.29)	✔
		TNF	2.35	(0.57, 4.73)	✔	1.67	(0.53, 1.88)	✔	2.35	(-3.86, 8.57)	✗

Table (9.0.1) Results of estimation of the linear regression slope predicting septal thickening with the pro-inflammatory genes expression, with three methods, for WTI and SBRT with different beam sizes, with measurements taken in different parts of lungs.

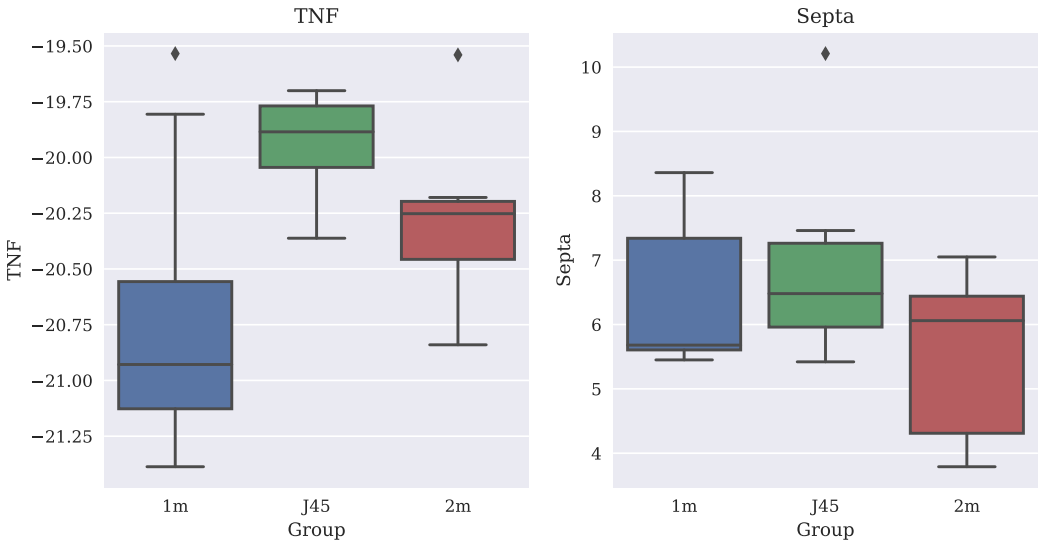


Figure (9.0.1) Distribution of the data, collected from the irradiated patch under SBRT with 7 mm beam size: the expression of the gene TNF on the left, and septal thickness on the right. The measurements were made 30, 45 and 60 days after irradiation.

Major differences between the results for the naive and the bootstrap estimators can be observed in terms of the detected significance. On the one hand, the relationship between the pro-inflammatory genes and septal thickening has been identified by all methods in the case of whole-thorax irradiation (two out of three genes for the bootstrap estimators, and only one for the naive estimator), which is an expected result. On the other hand, we also expect to identify a strong correlation in the case of the measurements taken directly from the irradiated patch. This is only the case for the bootstrap estimators, but not for the naive one. This results is in accordance with the results we obtain with simulated data: the confidence intervals are often overestimated with the naive approach, which may result in false negatives in terms of significance.

For the irradiated patch, as well as in all other cases with identified significance, it is only the case for the beam size of 3 mm, the results that is consistent with literature, indicating it as the beam size starting with which the long-term lesions start appearing (Bertho et al., 2020). Multiple significant associations have been identified with the bootstrap estimators in the ipsilateral lung and in the right lung for the beam size of 3 mm. However, none of the genes has been identified as significantly linked to lesions in the case of 7 mm. This may be caused by the relationship being of non-linear nature, as appears to be the case for the gene TNF, presented in Figure 9.0.1.

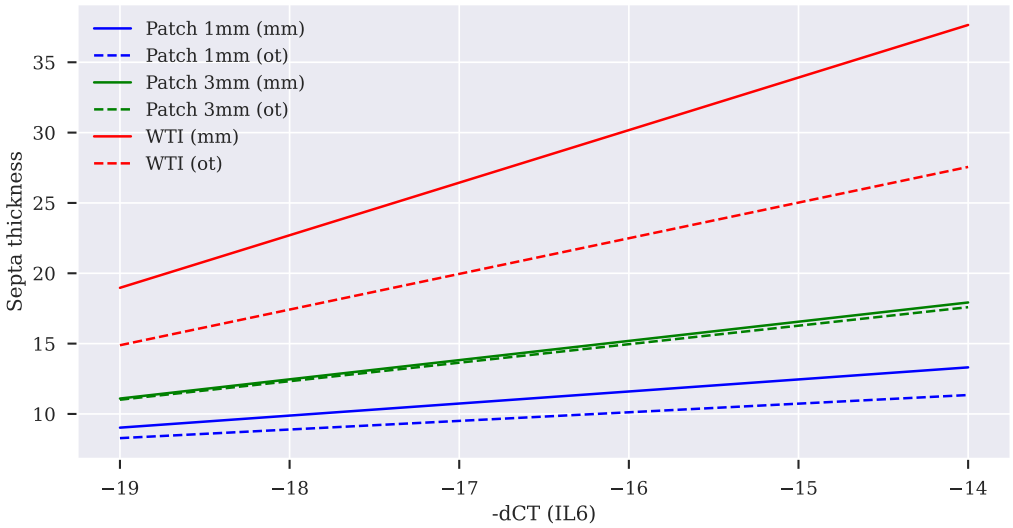


Figure (9.0.2) Linear model prediction of septal thickness based on IL6 expression, plotted for different locations and beam sizes, with the results from two bootstrap estimators.

Finally, among the cases where a significant relationship has been detected, the estimated values of the slope are always positive, which indicates a general radio-induced up-regulation trend. These values are in general bigger in case of the whole-thorax irradiation and within the patch than for the ipsilateral or right lung for SBRT, which is also in line with biological knowledge, since the genes are expected to be more strongly up-regulated in the areas of injury than further away. This effect is illustrated in Figure 9.0.2 on the example of the linear model prediction made for the gene IL6 .

DISCUSSION

Limitations and further work

In this work, we introduce modeling frameworks for specific types of data obtained with in vitro and in vivo experiments, with a goal of exploring the radio-induced adverse effects on cellular and organ/organism levels.

Part 1 of this thesis presents a framework for extracting key features from complex in vitro data, consisting of modeling and estimating time-dependent fold changes, constructing a dissimilarity measure for the fold changes that takes account of uncertainties and correlations, and proposing a time-efficient algorithm performing fold changes clustering jointly with alignment. Having shown promising results on both simulations and real data, a number of limitations can be distinguished in the proposed approach.

Limitations induced by modeling choices. Being in the context of data-driven modeling, the choices that were made were systematically motivated by either the properties of the data in the general context, or the radiobiology-specific knowledge in the context of studying radio-induced adverse effects. For example, the alignment was integrated in the clustering procedure in order to exploit the cascade-like nature of omic entities, namely to be able to detect potential predictive relationships. As mentioned in Remark 2.3.3, the way the alignment (here referred to as time warping) is define requires cutting off some parts of the fold changes that move outside of the considered temporal domain, which is a modeling choice that was made to avoid introducing unobserved information. However, as a result, some of the information is inevitably lost in the process. In particular, when comparing two fold changes under warping, the parts that are removed are often those where they are more significantly different than elsewhere, making them appear more similar than they are.

Another questionable aspect associated with alignment arises from the multivariate design and the data with unequally spaced time points, which lead to the appearance of warps of different lengths, i.e. 2-days and 1-week warps. In the application context treated here the fact of comparing these warps is justified by the same argument that led to choosing such unequally spaced time points in the first place. According to the biologists, most changes take place in the first week after irradiation, hence the necessity of having multiple time points during that period, and much less during the second and third weeks, at which point the measurements are only taken once a week. In this case, predictive relationships indicated by a 2-days warp during the first week are not less informative than those indicated by a 1-week warp during the last two weeks. Nonetheless, this may not be easy to justify in the general case, and introducing a penalty for longer warps is a potential solution.

An example of a feature motivated by radiobiology-specific knowledge is sign penalty, presented in Section 5.1.2. The penalty integrates naturally into the framework and reinforces the separation between the fold changes of different signs, which is of importance in the context of radiobiology. Nonetheless, certain aspects that are expected from the point of view application are by construction incompatible with the chosen approach. This is the case of integrating inhibition relationships between the entities as a clustering principle. For the moment, anti-correlated fold changes are considered as different in terms of the chosen dissimilarity measure, and introducing the possibility of relating anti-correlated entities between each other would imply significantly altering the framework and may not produce the desired results in combination with the already chosen features. Such relationships can only be detected empirically in the post-clustering phase, by comparing cluster templates.

Limitations in simulation design. A number of limitations can be named with respect to the simulations presented in Chapter 3. The design was intended to make the simulations as independent as possible from the inference approach in order to minimize confirmation bias, and as closer as possible to the real experimental data. More precisely, a natural way to simulate fold changes clusters that would promote our approach would be to simulate medoids for each cluster and then simulate the rest of fold changes by adding pointwise noise to the corresponding medoid. Instead, inspired by the functional data approach, for every cluster we established a group of functions, corresponding to polynomials of different degrees depending on the cluster

with random coefficients, and simulated all fold changes according to these distributions. Such simulation design prioritizes correspondence to real data and what is expected in terms of results, rather than giving an advantage to the proposed approach.

Nevertheless, the downside of the chosen approach is that we had to make a choice of simulating the fold changes directly instead of the gene expression data, which makes the simulations dependent on the assumption that the fold changes are random variables defined by the location and scale parameters. A potential improvement of the simulation design can be simulating the original data, and then vary the number of replicates, since the results are likely to be poor with a small number of replicates. Another potential extension could be modifying the simulation study with alignment by making the warps more drastic, which is the case where such methods as UMAP and SBM are likely to fail, further justifying the choice not to opt for either of the two as the main clustering approach.

Hyperparameters selection. There are three main hyperparameters, whose selection has been addressed to a different extent in this work:

- **Number of clusters K.** Such tools as cost function or silhouette score visualizations are available in the package *ScanOFC* to guide the user in number of clusters selection. Despite this fact, as discussed in Section 5.2.1, in practice we chose the number of clusters different to that suggested by the aforementioned criteria, for the sake of better interpretability. A potential research direction may be constructing an alternative selection criterion providing optimal interpretability of clusters.
- **The penalization parameter λ.** There is technically no formal way to select this parameter provided by our framework. In the application to real data, the choice was done as follows: first, the penalization parameter for the LINAC dataset was calibrated to obtain the most interpretable clusters, then the one for the SARRP dataset was chosen to maximize the pairwise cluster correspondence. It is possible for the package users to perform a simple cross-validation procedure, potentially including the number of clusters parameter, without it being too computationally costly since the computations can be done in the matter of seconds.
- **The network sparsity p.** Introduced in Section 4.1, this parameter has not been given a lot of attention, and was chosen almost arbitrarily while inferring real fold changes networks, due to the fact that in this work the networks

are mainly used for visualization purposes, and in this context the results are not very sensitive to the choice of the sparsity level. Indeed, while interpreting either microscopic or mesoscopic network representations, the interest lies in identifying general tendencies with respect to cluster connectivity distributions, which would not be drastically affected by a slight change in sparsity level. It should be noted, however, that community detection with stochastic block model, presented in Section 2.5, can be sensitive to the choice of the sparsity parameter, which in its turn has not been addressed in detail since it is not used as the main approach to clustering.

Part 2 of the thesis focuses on a statistical framework designed for extracting dependencies form in vivo experiments, specifically introducing linear regression estimators in the context where the predictor and the predicted variables are never jointly observed. In this work we chose the basic linear multivariate setting, prioritizing simplicity and computational feasibility. Particularly, the estimator based on the method of moments makes no hypotheses on data distribution and can be calculated explicitly. The estimator based on optimal transport includes a simple optimization problem, and is based on the Gaussian form of the Wasserstein distance but does not technically require the data to be Gaussian, seeking to approximate them with Gaussian variables in whatever case.

However, these approaches are inapplicable in the cases where linear relationship hypothesis cannot be satisfied. For instance, it is the case with predicting survival data with some continuous biomarker, which is of particular interest in research into radio-induced adverse effects. To be able to consider such scenarios, our model can be extended to a more general case, namely with generalized linear model. The optimal transport estimator appears promising in this context given the fact that Wasserstein distance allows to compare probability distributions of different nature (for example, continuous and discrete).

Another further research direction lies in investigating alternative methods based on integrated likelihood and Bayesian approaches, which are likely to produce better results in many cases but require putting priors on distributions. Concerning a likelihood-based approach, the joint likelihood to be maximized can be written in a factorized form due to the independence of X and Y: $L(x, y; \beta_0, \beta_1, \mu_x^k, \sigma_\epsilon, \sigma_x) = L_X(x; \mu_x, \sigma_x) L_Y(y; \beta_0, \beta_1, \mu_x, \sigma_\epsilon, \sigma_x)$ for a group k in the univariate setting. The marginal likelihoods can be expressed explicitly, assuming that the model (7.1.1) is true,

and that $X|(G = k) \sim \mathcal{N}(\mu_x^k, \sigma_{x,k}^2)$, and thus $Y|(G = k) \sim \mathcal{N}(\beta_0 + \beta_1\mu_x^k, \sigma_\epsilon^2 + \beta_1^2\sigma_{x,k}^2)$, with the noise $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ independent of X.

Finally, it would be of interest to work on improving theoretical properties of our estimators, namely correcting the negative bias that appears for both estimators. The latter is particularly important in the case of the optimal transport estimator, which appears to arise naturally with the Wasserstein distance according to Larry Wasserman (Wasserman, 2023). Correcting this bias would considerably improve the estimator, making it competitive with the approaches mentioned earlier that make numerous assumptions on the data.

Globally, in this work we addressed the problem of modeling the adverse effects in the frameworks of axes 2 and 3 of the ROSIRIS project separately. The results presented here constitute the foundation for building the predictive link between the adverse effects appearing on the two levels, which has not been addressed in this work and remains a subject for future research. A preliminary idea could be noticing that the pro-inflammatory genes, studied in the in vivo experiments, appear as key actors in the analyses conducted on the in vitro data, and thus comparing the expression levels, with the goal of potentially using the in vitro expression directly to predict the in vivo adverse effects. While this is a promising idea, it has to be approached with caution, in the view of cultured cells behaving differently with respect to their normal environment within the body, which has to be taken into account while comparing the two. The latter has been shown in particular to be the case with gene expression in endothelial cells (Afshar et al., 2023; Liu and Bouman Chen, 2023). Nevertheless, the connection is worth exploring, given its potential utility within the radiobiological community and beyond.

Talks and publications

Part 1: various parts of this work were presented at multiple seminars, national and international conferences, including the 52nd Journées de statistique (JDS 2021) of the SFdS, the 31st International Biometrics Conference (IBC 2022), the 24th International Conference on Computational Statistics (COMPSTAT 2022), the 21st European Conference on Computational Biology (ECCB2022, poster is available here: https://parsenteva.github.io/files/poster_v2.pdf), and the 7th Journée "Probabilités et Statistique Besançon-Dijon". Papers are in the process of submission.

Part 2: parts of this work were presented at the 54th Journées de statistique (JDS 2023) of the SFdS, and the 25th International Conference on Computational Statistics (COMPSTAT 2023). A paper is currently being prepared.

Funding

This work is supported by the European Union through the PO FEDER-FSE Bourgogne 2014/2020 programs as part of the project ModBioCan2020, and by Institut de Radioprotection et de Sûreté nucléaire as part of the project ROSIRIS.

Conflict of Interest: none declared.

APPENDIX A

SCANOFC : PACKAGE DOCUMENTATION

Classes

class **Clustering**(FoldChanges)

```
Clustering(data=None, means=None, cov=None, var_names=None, time_points=None,
           dist='d2hat', time_warp=False, max_warp_step=1, sign_pen=False,
           pen_param=1, random_gen=None)

A class containing tools for clustering fold changes, inherits from
FoldChanges class.

Attributes
----------
dist : str
    Distance chosen for clustering, 'd2hat' by default (L2 distance between
    random estimators), can also be 'wasserstein' (Wasserstein distance)
    and 'hellinger' (Hellinger distance).
sign_pen : bool
    If True, then the distance is penalized with sign penalty.
    The default is False.
pen_param : float
    Parameter determining the weight of sign penalty. The default is 1.
time_warp : bool
    If True, then the clustering procedure is coupled with the alignment.
    The default is False.
max_warp_step : int
    If max_warp_step=i>0, then the set of all considered warps is the
    set of all integers between -i and i.
index_pairs : ndarray
    2D array of shape (number of pairs, 2) containing pairs
    of indices in the same order as the pairwise distances array
    'distances'.
distances : ndarray
    1D array of length equal to the number of pairs containing
    pairs of distances between the fold changes in the same order as
    'index_pairs'.
dist_mat : ndarray
    Distance matrix, 2D array of shape with both dimensions
    equal to the number of fold changes (entities). If time_warp is True,
    then the distance matrix used for clustering is OWD
    (Optimal Warping Distance) matrix containing distances that minimize
    the pairwise distance over the set of all considered warps.
optimal_warp_mat : ndarray
    Optimal Warp matrix, 2D array with both dimensions equal to the
    number of fold changes (entities). The values of the upper
    triangular part of the matrix correspond to the warps minimizing
    'warped_distances' (since for every entity pair the one earlier on
    the list and with a smaller index has been warped to get
    'warped_distances', while the other entity remains static),
    whereas those of the lower triangular part have the opposite sign
    (due to the antisymmetric nature of pairwise warping).
    Defined if time_warp is True.
random_gen : RandomState instance or None
    Random number generator, used to reproduce results. If None (default),
    the generator is the RandomState instance used by `np.random`.
    If RandomState instance, random_gen is the actual random
    number generator.
```

```
Methods
-------
init_centroids(k)
    Initializes centroids (medoids pr barycenters) for k clusters.
assign_clusters(centroids, method='k-medoids', wass_dist_mat=None)
    Assigns all fold changes to one of the k clusters based on their
    distances to centroids (medoids or barycenters).
update_centroids(k, clusters, old_centroids, algorithm='k-means-like')
    Recalculates centroids based on the current cluster configuration.
compute_barycenter(k, clusters, cov0, precision=1e-5)
    Calculates barycenters with respect to the Wasserstein distance for
    k clusters by solving a fixed point problem iteratively until the
    stopping criterion is satisfied.
hierarchical_centroids(k, clusters)
    Chooses centroids among the fold changes in clusters after clustering.
    Used for non-centroid based clustering methods, such as hierarchical
    clustering.
calculate_total_cost(centroids, clusters)
    Calculates total cost for all clusters, defined as the sum of distances
    between the fold changes and their centroids with respect to the
    distance matrix. Used in k-medoids clustering as a selection criterion.
choose_k_clusters(k, method='k-medoids', algorithm='k-means-like',
                  verbose=0, plot_umap=True, nb_rep_umap=1,
                  umap_color_labels=None, plot_umap_labels=False)
    Performs clustering in k clusters of a set of random fold changes
    estimators based on one random clusters' initialization.
fc_clustering(k, nb_rep=100, method='k-medoids', verbose=0,
              disp_plot=False, algorithm='k-means-like', nb_best=1,
              tree_cutoff=5, silhouette=False, umap_color_labels=None,
              plot_umap_labels=False)
    Performs a series of clustering attempts of a set of random fold
    changes estimators for different numbers of clusters by trying
    multiple random clusters' initializations and choosing the attempt
    producing the best outcome (in the cases where random initializations
    are applicable).
plot_clusters(k, clusters, centroids, centroid_type='medoid', warps=None,
              nb_cols=4, nb_rows=None, figsize=None)
    Produces a figure with k subplots (or 2 figures if warps are provided),
    each containing plots of the fold changes' means in the corresponding
    cluster. In the case with time warping, produces a figure with
    unaligned (original) and a figure with aligned (with respect to their
    centroids) fold changes.

Inherited from FoldChanges.
```

Methods defined here:

__init__(self, data=None, means=None, cov=None, var_names=None, time_points=None, dist='d2hat', time_warp=False, max_warp_step=1, sign_pen=False, pen_param=1, random_gen=None)

```
Parameters
----------
data : ndarray or None
    If not None, 4D array with the dimensions corresponding to:
    1) nb of time points, 2) two experimental conditions
    (dim 0: control, dim 1: case)), 3) replicates, 4) nb of entities.
    If None (by default), then the fold changes are constructed based
    on 'means' and 'cov'. Either 'data' or 'means' and 'cov' have to be
    non-None, with 'data' having priority for the fold changes
    construction.
means : ndarray or None
    If not None, 2D array of shape (nb_time_pts, nb_var)
    containing data with `float` type, representing fold changes' means
    for each entity and each time point. If 'data' is None, used to
    construct fold changes. Either 'data' or 'means' and 'cov' have to
    be non-None.
cov : ndarray or None
    If not None, 3D array of shape (nb_time_pts, nb_var, nb_var)
    containing data with `float` type, representing fold changes'
    nb_var x nb_var shaped covariance matrices for each time point.
```

```
    Time-wise cross-covariances are assumes to be 0 due to experimental
    design. In case of Hellinger distance, can also be 4-dimensional
    (natural form): (nb_time_pts, nb_time_pts, nb_var, nb_var).
    If 'data' is None, used to construct fold changes. Either 'data' or
    'means' and 'cov' have to be non-None.
var_names : array-like or None
    1D array-like containing data with `string` type, representing
    names of the measured entities (ex. genes). The default is None.
time_points : array-like or None
    1D array-like containing data with `float` type, representing time
    points at which fold changes were measured. The default is None.
dist : str
    Distance chosen for clustering, 'd2hat' by default (L2 distance
    between random estimators), can also be 'wasserstein'
    (Wasserstein distance) and 'hellinger' (Hellinger distance).
time_warp : bool
    If True, then the clustering procedure is coupled with the
    alignment. The default is False.
max_warp_step : int
    If max_warp_step=i>0, then the set of all considered warps is the
    set of all integers between -i and i.
sign_pen : bool
    If True, then the distance is penalized with sign penalty.
    The default is False.
pen_param : float
    Parameter determining the weight of sign penalty. The default is 1.
random_gen : RandomState instance or None
    Random number generator, used to reproduce results. If None
    (default), the generator is the RandomState instance used by
    `np.random`. If RandomState instance, random_gen is the actual
    random number generator.
```

assign_clusters(self, centroids, method='k-medoids', wass_dist_mat=None)

```
Assigns all fold changes to one of the k clusters based on their
distances to centroids (medoids or barycenters).

Parameters
----------
centroids : ndarray
    1D array of length k containing indices in range (0, nb_var) of
    the fold changes that act as current centroids (medoids). Used for
    clusters assignment only if method=='k-medoids'.
method : str, optional
    Main approach to clustering, either 'k-medoids' (default, coupled
    with d2hat distance or Hellinger distance) or 'wass k-means'
    (Wasserstein k-means).
wass_dist_mat : ndarray, optional
    2D array of shape (k, nb_var) containing distances between the
    fold changes and the barycenters for all clusters. Used for
    clusters assignment only if method=='wass k-means', otherwise
    None (by default).
Returns
-------
clusters : ndarray
    1D array of length nb_var containing integers in range (0, k)
    indicating clusters to which the fold changes are assigned.
```

calculate_comparable_cost(self, k, clusters)

```
Calculates total comparable cost for all clusters, defined as the sum
of distances between all fold change pairs in each cluster with respect
to the distance matrix. Used to compare clustering performed with
different methods (distance matrix should be the same).

Parameters
----------
k : int
    Number of clusters.
clusters : ndarray
    1D array of length nb_var containing integers in range (0, k)
    indicating clusters to which the fold changes are assigned.

Returns
```

```
-------
float
    Value of the total comparable cost.
```
calculate_total_cost(self, centroids, clusters)
```
Calculates total cost for all clusters, defined as the sum of distances
between the fold changes and their centroids with respect to the
distance matrix. Used in k-medoids clustering as a selection criterion.

Parameters
----------
centroids : ndarray
    1D array of length k containing indices in range (0, nb_var) of
    the fold changes that act as current centroids.
clusters : ndarray
    1D array of length nb_var containing integers in range (0, k)
    indicating clusters to which the fold changes are assigned.

Returns
-------
float
    Value of the total cost.
```
choose_k_clusters(self, k, method='k-medoids', algorithm='k-means-like', verbose=0, plot_umap=True, nb_rep_umap=1, umap_color_labels=None, plot_umap_labels=False)
```
Performs clustering in k clusters of a set of random fold changes
estimators based on one random clusters' initialization.

Parameters
----------
k : int
    Number of clusters.
method : str, optional
    Main approach to clustering, options include:
        - 'k-medoids' (default, coupled with d2hat distance or
        Hellinger distance),
        - 'wass k-means' (Wasserstein k-means),
        - 'hierarchical' (hierarchical clustering based on
        d2hat distance),
        - 'umap' (UMAP projection of the d2hat distance matrix with
        subsequent k-means clustering of the projection coordinates).
algorithm : str, optional
    Indicates a choice of one of the two common variations of k-medoids
    clustering. The default is 'k-means-like' (Park, 2006), can also
    be 'PAM' (Partitioning Around Medoids; Schubert, Rousseeuw, 2019).
verbose : int, optional
    Controls the verbosity, if 1 (or larger) then informs on the
    advancement of clustering.
plot_umap : bool, optional
    If True (default) and method is 'umap', then plots the UMAP
    projection of the distance matrix.
nb_rep_umap : int, optional
    Number of k-means clustering initializations performed on the
    UMAP projection, relevant if method is 'umap'. The default is 1.
umap_color_labels : None or array-like, optional
    Relevant if method is 'umap'. If None (default), then the data
    points on the UMAP projection are colored with respect to the
    cluster labels assigned by k-means. Alternatively, can be a 1D
    array-like of length equal to nb_var, containing integers
    indicating cluster labels assigned to the fold changes.  In this
    case, colors are chosen corresponding to these labels. This option
    is intended for use in the framework of simulation studies.
plot_umap_labels : bool, optional
    Relevant if method is 'umap'. If True, then labels the data points
    on the UMAP projection with corresponding fold changes' indices.
    The default is False (no labels).

Returns
-------
List containing the following elements:
    clusters : ndarray
        1D array of length nb_var containing integers in range (0, k)
```

```
        indicating clusters to which the fold changes are assigned.
        Returned as the first element of the list in all cases.
    centroids : ndarray
        1D array of length k containing indices in range (0, nb_var) of
        the fold changes that have been chosen as centroids.
        Returned as the second element of the list in all
        cases except if method=='wass k-means'.
    bary_means : ndarray
        2D array of shape (nb_time_pts, k) representing final
        barycenter means for all clusters. Returned as the second
        element of the list if method=='wass k-means'.
    bary_cov : ndarray
        3D array of shape (nb_time_pts, nb_time_pts, k) representing
        final barycenter covariance matrices for all clusters. Returned
        as the third element of the list if method=='wass k-means'.
    total_cost : float
        Value of the final total clustering cost with respect to the
        metric associated with the chosen clustering method.
        Returned as the last element of the list if method=='k-medoids'
        or method=='wass k-means' (in other cases absent since the cost
        isn't assessed during clustering and should be calculated
        separately if needed).
```

compute_barycenter(self, k, clusters, cov0, precision=1e-05)

```
Calculates barycenters with respect to the Wasserstein distance for
k clusters by solving a fixed point problem iteratively until the
stopping criterion is satisfied.

Parameters
----------
k : int
    Number of clusters.
clusters : ndarray
    1D array of length equal to 'nb_var' with values of type 'int'
    between 0 and k-1 indicating which cluster every fold change
    belongs to.
cov0 : ndarray
    2D array of shape (nb_time_pts, nb_time_pts), a symmetric positive
    definite matrix that initializes the barycenters' covariance
    matrices.
precision : float, optional
    Stopping criterion, the fixed point equation iterations stop when
    the difference between the old and the new total costs for the
    considered cluster becomes smaller or equal to this value.
    The default is 1e-5.

Returns
-------
bary_means : ndarray
    2D array of shape (nb_time_pts, k) representing final barycenter
    means for all clusters.
bary_cov : ndarray
    3D array of shape (nb_time_pts, nb_time_pts, k) representing
    final barycenter covariance matrices for all clusters.
all_costs : ndarray
    1D array of length k containing final total costs per cluster.
```

fc_clustering(self, k, nb_rep=100, method='k-medoids', verbose=0, disp_plot=False, algorithm='k-means-like', nb_best=1, tree_cutoff=5, silhouette=False, umap_color_labels=None, plot_umap_labels=False)

```
Performs a series of clustering attempts of a set of random fold
changes estimators for different numbers of clusters by trying
multiple random clusters' initializations and choosing the attempt
producing the best outcome (in the cases where random initializations
are applicable).

Parameters
----------
k : int
    Number of clusters.
nb_rep : int, optional
    Number of random initialization attempts (k-means clustering
    initializations performed on the UMAP projection if
```

```
    method=='umap'). The default is 100.
method : str, optional
    Main approach to clustering, options include:
        - 'k-medoids' (default, coupled with d2hat distance or
        Hellinger distance),
        - 'wass k-means' (Wasserstein k-means),
        - 'hierarchical' (hierarchical clustering based on
        d2hat distance),
        - 'umap' (UMAP projection of the d2hat distance matrix with
        subsequent k-means clustering of the projection coordinates).
algorithm : str, optional
    Indicates a choice of one of the two common variations of k-medoids
    clustering. The default is 'k-means-like' (Park, 2006), can also
    be 'PAM' (Partitioning Around Medoids; Schubert, Rousseeuw, 2019).
verbose : int, optional
    Controls the verbosity, if 1 (or larger) then informs on the
    advancement of clustering.
disp_plot : bool, optional
    False by default, if True then plots the mean total cost curve with
    standard deviations or the UMAP projection of the distance matrix
    depending on the method.
nb_best : int, optional
    Number of the best random initialization attempts to be taken into
    account for the total cost plot. The default is 1.
tree_cutoff : int, optional
    Relevant if method is 'hierarchical', the number of dendrogram tree
    levels that are displayed. The default if 5.
silhouette : bool, optional
    The default is False, if True then the mean silhouette score curve
    with standard deviations is displayed along with the total costs.
umap_color_labels : None or array-like, optional
    Relevant if method is 'umap'. If None (default), then the data
    points on the UMAP projection are colored with respect to the
    cluster labels assigned by k-means. Alternatively, can be a 1D
    array-like of length equal to nb_var, containing integers
    indicating cluster labels assigned to the fold changes.  In this
    case, colors are chosen corresponding to these labels. This option
    is intended for use in the framework of simulation studies.
plot_umap_labels : bool, optional
    Relevant if method is 'umap'. If True, then labels the data points
    on the UMAP projection with corresponding fold changes' indices.
    The default is False (no labels).

Returns
-------
If k is an integer, returns same as choose_k_clusters. If k is a
container with integers, then returns a list of dictionaries, with
keys corresponding to the considered numbers of clusters, and the
values are the same as returned by choose_k_clusters.
If time_warp is True, an new element warps (or all_warps if
if different numbers of clusters are considered) is added to the list
for all distance matrix-based methods (i.e. all except 'wass k-means').
For a fixed number of clusters it is a 1D array of length nb_var
containing integers in range (-max_warp_step, max_warp_step + 1)
indicating fold changes' warps with respect to their
corresponding centroids.
hierarchical_centroids(self, k, clusters)
Chooses centroids among the fold changes in clusters after clustering.
Used for non-centroid based clustering methods, such as hierarchical
clustering.

Parameters
----------
k : int
    Number of clusters.
clusters : ndarray
    1D array of length equal to 'nb_var' with values of type 'int'
    between 0 and k-1 indicating which cluster every fold change
    belongs to.
```

```
Returns
-------
ndarray
    1D array of length k containing indices in range (0, nb_var) of
    the fold changes that represent cluster centroids.
```
init_centroids(self, k)
```
Produces a set of k random centroids to initialize clustering according
to the algorithm k-means++.

Parameters
----------
k : int
    Number of clusters.

Returns
-------
centroids : ndarray
    1D array of length k containing indices in range (0, nb_var) of
    the fold changes that have been chosen as initial centroids.
```
plot_clusters(self, k, clusters, centroids, centroid_type='medoid', warps=None, nb_cols=4, nb_rows=None, figsize=None)
```
Produces a figure with k subplots (or 2 figures if warps are provided),
each containing plots of the fold changes' means in the corresponding
cluster. In the case with time warping, produces a figure with
unaligned (original) and a figure with aligned (with respect to their
centroids) fold changes.

Parameters
----------
k : int
    Number of clusters.
clusters : ndarray or dictionary
    If ndarray, 1D array of length nb_var containing integers in range
    (0, k) indicating clusters to which the fold changes are assigned.
    If a dictionary, the keys are numbers of clusters considered, and
    for each such number the value is the latter array.
centroids : ndarray or dictionary
    If centroid_type=='medoid':
        If ndarray, 1D array of length k containing indices in range
        (0, nb_var) of the fold changes that act as centroids.
        If a dictionary, the keys are numbers of clusters considered,
        and for each such number the value is the latter array.
    If centroid_type=='barycenter':
        If ndarray, an array of barycenter means: 2D array of shape
        (nb_time_pts, k) representing final barycenter means for all
        clusters. If a dictionary, the keys are numbers of clusters
        considered, and for each such number the value is the latter
        array.
centroid_type : str, optional
    The default is 'medoid', in which case the centroids are selected
    among the fold changes (see centroids). Another option is
    'barycenter', in this case the barycenters are plotted based on
    their means.
warps : ndarray or dictionary, optional
    If ndarray, 1D array of length nb_var containing integers in range
    (-max_warp_step, max_warp_step + 1) indicating fold changes' warps
    with respect to their corresponding centroids. If a dictionary,
    the keys are numbers of clusters considered, and for each such
    number the value is the latter array. The default is None,
    otherwise the versions with and without time warping are plotted.
nb_cols : int, optional
    Number of columns of the subplot grid. The default is 4.
nb_rows : TYPE, optional
    Number of rows of the subplot grid The default is None, in which
    case nb_rows=int(np.ceil(k/nb_cols)).
figsize : (float, float), optional
    Width and height of the figure(s). The default is None, in which
    case figsize=(15, 6*nb_rows).

Returns
-------
```

```
    None.
```
update_centroids(self, k, clusters, old_centroids, algorithm='k-means-like')
```
    Recalculates centroids based on the current cluster configuration.

    Parameters
    ----------
    k : int
        Number of clusters.
    clusters : ndarray
        1D array of length nb_var containing integers in range (0, k)
        indicating clusters to which the fold changes are assigned.
    old_centroids : ndarray
        1D array of length k containing indices in range (0, nb_var) of
        the fold changes that act as current centroids (medoids)
        before the update.
    algorithm : str, optional
        Indicates a choice of one of the two common variations of k-medoids
        clustering. The default is 'k-means-like' (Park, 2006), can also
        be 'PAM' (Partitioning Around Medoids; Schubert, Rousseeuw, 2019).

    Returns
    -------
    centroids : ndarray
        1D array of length k containing indices in range (0, nb_var) of
        the fold changes that act as current centroids (medoids)
        after the update.
```

class **FoldChanges**(builtins.object)

```
    FoldChanges(data=None, means=None, cov=None, var_names=None, time_points=None)

    A class representing a set of fold changes (a measure of difference between
    the two experimental conditions over time).

    Attributes
    ----------
    means : ndarray
        2D array of shape (nb_time_points, nb_var) containing data
        with `float` type, representing fold changes' means for each entity
        and each time point.
    cov : ndarray
        3D array of shape (nb_time_pts, nb_var, nb_var) containing
        data with `float` type, representing fold changes' nb_var x nb_var
        shaped covariance matrices for each time point. Time-wise
        cross-covariances are assumed to be 0 due to experimental design.
        In case of Hellinger distance, can also be 4-dimensional (natural form):
        (nb_time_pts, nb_time_pts, nb_var, nb_var).
    sd : ndarray
        2D array of shape (nb_time_pts, nb_var) containing data
        with `float` type, representing fold changes' standard deviation for
        each entity and each time point.
    time_points : array-like
        1D array-like containing data with `float` type, representing time
        points at which fold changes were measured. If not given then range of
        indices of the corresponding dimension.
    nb_time_pts : int
        number of time points measured, i.e. len(time_points) or the size of
        the corresponding dimension.
    var_names : array-like or None
        1D array-like containing data with `string` type, representing names
        of the measured entities (ex. genes). If not given then range of
        indices of the corresponding dimension of means.
    nb_var : int
        number of entities considered, i.e. len(var_names) or the size of
        the corresponding dimension of means.

    Methods
    -------
    compute_distance_pairs(dist='d2hat', sign_pen=False, pen_param=10)
        Computes fold changes' pairwise distances.
```

```
compute_fc_norms(dist='d2hat')
    Computes fold changes' norms with respect to the chosen distance.
compute_dist_mat(index_pairs, distances)
    Transforms the set of pairwise distances into a distance matrix.
compute_cross_distances(bary_means, bary_cov, cluster=None)
    Calculates the Wasserstein distance in different configurations.
compute_warped_distance_pairs(max_warp_step=1, sign_pen=False,
                              pen_param=0.01)
    Computes fold changes' pairwise distances for all considered warps.
compute_warped_dist_mat(index_pairs, warped_distances)
    Calculates an optimal warping distance matrix and an optimal warp
    matrix from the set of pairwise warped distances.
```

Methods defined here:

__init__(self, data=None, means=None, cov=None, var_names=None, time_points=None)

```
Parameters
----------
data : ndarray or None
    If not None, 4D array with the dimensions corresponding to:
    1) nb of time points, 2) two experimental conditions
    (dim 0: control, dim 1: case)), 3) replicates, 4) nb of entities.
    If None (by default), then the fold changes are constructed based
    on 'means' and 'cov'. Either 'data' or 'means' and 'cov' have to be
    non-None, with 'data' having priority for the fold changes
    construction.
means : ndarray or None
    If not None, 2D array of shape (nb_time_pts, nb_var)
    containing data with `float` type, representing fold changes' means
    for each entity and each time point. If 'data' is None, used to
    construct fold changes. Either 'data' or 'means' and 'cov' have to
    be non-None.
cov : ndarray or None
    If not None, 3D array of shape (nb_time_pts, nb_var, nb_var)
    containing data with `float` type, representing fold changes'
    nb_var x nb_var shaped covariance matrices for each time point.
    Time-wise cross-covariances are assumes to be 0 due to experimental
    design. In case of Hellinger distance, can also be 4-dimensional
    (natural form): (nb_time_pts, nb_time_pts, nb_var, nb_var).
    If 'data' is None, used to construct fold changes. Either 'data' or
    'means' and 'cov' have to be non-None.
var_names : array-like or None
    If not None, 1D array-like containing data with `string` type,
    representing names of the measured entities (ex. genes).
    The default is None.
time_points : array-like or None
    If not None, 1D array-like containing data with `float` type,
    representing time points at which fold changes were measured.
    The default is None.
```

compute_cross_distances(self, bary_means, bary_cov, cluster=None)

```
Designed for the vectorized version of the Wasserstein k-means,
in particular:
    - case 1: to compute distances between the fold changes in one
cluster and the barycenter to update barycenter in function
'compute_barycenter' of the Clustering class.
    - case 2: to compute distances between the fold changes
and their barycenters in all clusters to assess cost in function
'choose_k_clusters'of the Clustering class.
In addition, it is used to compute pairwise
distance matrix for the Wasserstein distance when instantiating a
Clustering class (case 3, not used in k-means).

Parameters
----------
bary_means : ndarray
    Case 1: 2D array of shape (nb_time_pts, 1) containing the mean of
            the current barycenter in the fixed point iteration.
    Case 2: 2D array of shape (nb_time_pts, k), where k stands for the
            number of clusters (and hence barycenters). The array
            contains the means of all barycenters in the current
```

```
            iteration of k-means.
    Case 3: 2D array of shape (nb_time_pts, nb_var) containing
            fold changes' means.
bary_cov : ndarray
    Case 1: 2D array of shape (nb_time_pts, nb_time_pts) containing
            the covariance matrix of the current barycenter in the
            fixed point iteration.
    Case 2: 3D array of shape (nb_time_pts, nb_time_pts, k), where k
            stands for the number of clusters (and hence barycenters).
            The array contains the covariance matrices of all
            barycenters in the current iteration of k-means.
    Case 3: 3D array of shape (nb_time_pts, nb_time_pts, nb_var)
            containing marginal (diagonal) covariance matrices of the
            fold changes.

cluster : ndarray, optional
    None in cases 2 and 3, in case 1: 1D array of length equal to the
    size of the considered cluster. Contains data of type 'int'
    corresponding to the indices of the fold changes' belonging to
    this cluster.

Returns
-------
wass_dist : ndarray
    Case 1: 2D array of shape (1, cluster_size) containing distances
            between the fold changes in the cluster and the barycenter.
    Case 2: 2D array of shape (k, nb_var) containing distances between
            the fold changes and the barycenters for all clusters.
    Case 3: 2D array of shape (nb_var, nb_var) containing pairwise
            distances between the fold changes.

K : ndarray
    Case 1: 4D array of shape (1, cluster_size, nb_time_pts, nb_time_pts)
            characterizing the joint distributions of the fold changes
            in the cluster and the barycenter. Central term in the
            fixed point equation.
    Case 2: 4D array of shape (k, nb_var, nb_time_pts, nb_time_pts)
            characterizing the joint distributions of the fold changes
            and all the barycenters.
    Case 3: 4D array of shape (nb_var, nb_var, nb_time_pts, nb_time_pts)
            characterizing the joint distributions of all
            fold changes pairs.
```

compute_dist_mat(self, index_pairs, distances)

```
Transforms the set of pairwise distances into a distance matrix.

Parameters
----------
index_pairs : ndarray
    2D array of shape (number of pairs, 2) containing pairs
    of indices in the same order as the pairwise distances array
    'distances'.
distances : ndarray
    1D array of length equal to the number of pairs containing
    pairs of distances between the fold changes in the same order as
    'index_pairs'.

Returns
-------
dist_mat : ndarray
    Distance matrix, 2D array of shape with both dimensions
    equal to the number of fold changes (entities).
```

compute_distance_pairs(self, dist='d2hat', sign_pen=False, pen_param=10)

```
Computes pairwise distances for a set of fold changes encoded in
FoldChanges class instance for a chosen distance. The choice of a
distance is limited to L2 distance between random fold changes'
estimators, and Hellinger distance.

Parameters
----------
dist : str
```

```
    Can be either 'd2hat' (L2 distance between random estimators,
    default), or 'hellinger' (Hellinger distance).
sign_pen : bool
    True if sign penalty should be added to the distance, False
    otherwise (default). Sign penalty penalizes fold changes pairs that
    have different signs in one or more time points.
pen_param : float
    Sign penalty hyperparameter (weight of penalty).

Returns
-------
index_pairs : ndarray
    2D array of shape (number of pairs, 2) containing pairs
    of indices in the same order as the pairwise distances array
    'distances'.
distances : ndarray
    1D array of length equal to the number of pairs containing
    pairs of distances between the fold changes in the same order as
    'index_pairs'.
```

compute_fc_norms(self, dist='d2hat')

```
Computes norms for all fold changes in the class instance with respect
to the chosen distance (so far implemented only for the L2 distance
between random estimators).

Parameters
----------
dist : str
    So far 'd2hat' is the default and the only option.

Returns
-------
ndarray
    1D array of length equal to the number of fold changes (that is,
    the number of biological entities considered) containing the norms.
```

compute_warped_dist_mat(self, index_pairs, warped_distances)

```
Calculates an optimal warping distance matrix and an optimal warp
matrix from the set of pairwise warped distances.

Parameters
----------
index_pairs : ndarray
    2D array of shape (number of pairs, 2) containing pairs
    of indices in the same order as the pairwise optimal warping
    distances array 'warped_distances' (with respect to the second
    dimension).
warped_distances : ndarray
    2D array, the first dimension corresponds to all considered warp
    steps, the second corresponds to pairs of warped distances between
    the fold changes in the same order as 'index_pairs'.

Returns
-------
warped_dist_mat : ndarray
    Optimal Warping Distance matrix, 2D array with both dimensions
    equal to the number of fold changes (entities). Distances are such
    that minimize the pairwise distance over the set of all considered
    warps.
optimal_warp_mat : ndarray
    Optimal Warp matrix, 2D array with both dimensions equal to the
    number of fold changes (entities). The values of the upper
    triangular part of the matrix correspond to the warps minimizing
    'warped_distances' (since for every entity pair the one earlier on
    the list and with a smaller index has been warped to get
    'warped_distances', while the other entity remains static),
    whereas those of the lower triangular part have the opposite sign
    (due to the antisymmetric nature of pairwise warping).
```

compute_warped_distance_pairs(self, max_warp_step=1, sign_pen=False, pen_param=0.01)

```
Computes pairwise distances for a set of considered warps for a set of
fold changes encoded in FoldChanges class instance for a chosen
distance. Warped distances (or distances after alignment) are only
```

```
    calculated for the L2 distance between random fold changes' estimators.

    Parameters
    ----------
    max_warp_step : int
        If max_warp_step=i>0, then the set of all considered warps is the
        set of all integers between -i and i.
    sign_pen : bool
        True if sign penalty should be added to the distance, False
        otherwise (default). Sign penalty penalizes fold changes pairs that
        have different signs in one or more time points.
    pen_param : float
        Sign penalty hyperparameter (weight of penalty).

    Returns
    -------
    index_pairs : ndarray
        2D array of shape (number of pairs, 2) containing pairs
        of indices in the same order as the pairwise distances array
        'distances'.
    warped_distances : ndarray
        2D array such that the first dimension is of size
        2 * max_warp_step + 1 corresponding to all considered warps, and
        the second dimension corresponds to the number of fold changes
        pairs in the same order as 'index_pairs'.
```

class **NetworkInference**(Clustering)

```
NetworkInference(data=None, means=None, cov=None, var_names=None, time_points=None,
                 dist='d2hat', time_warp=False, max_warp_step=1, sign_pen=False,
                 pen_param=1, random_gen=None, sparsity=0.75, directed=False,
                 adj_mat=None)

A class containing tools for inference of a network of fold changes from
a dataset, inherits from Clustering and FoldChanges classes.

Attributes
----------
sparsity : float
    Sparsity of the network determining the cutoff when defining the
    binary adjacency matrix adj_mat based on the weighted one.

directed : bool
    If True, the network is directed, and undirected if False.

adj_mat : ndarray
    2D array of shape (nb_var, nb_var) indicating whether the fold changes
    are connected (i.e. similar enough) or not. If the network is
    undirected, then has 0 for connected fold changes and 1 for not
    connected (symmetric). A pair of fold changes is considered to be
    connected if their distance-based similarity is bigger then the cutoff
    value, which is equal to the empirical quantile of the similarity
    matrix corresponding to the chosen sparsity. If the network is directed,
    the matrix stops being symmetric, and the edges that exist according
    to the undirected case procedure become either 1 or 0 based on the
    corresponding warp: 1 for the edges with the corresponding warps being
    positive (predictive) or 0 (simultaneous), and 0 for
    those with negative warps (target).

Methods
-------
infer_sbm(nb_blocks, clusters, n_init=10, n_iter_early_stop=50,
          random=False, verbosity=0, pi_weight=0.8, random_gen=None)
    Performs stochastic block model inference for the fold changes' network
    based on clustering (i.e. on the constrained parameter space).
compute_network(clusters, centroids, draw_path=False, path=None,
                figtitle='Fold changes network', figsize=(25,25),
                obj_scale=1, graph_type='full', adj_mat_2=None,
                shade_intersect=False)
    Creates a NetworkX object representing the fold changes' network and
```

displays it in a block form arising from clusters. The network is
represented with a graph where nodes are the considered entities and
the edges are connections between them (i.e. ones in the adjacency
matrix). Members of every block are grouped around their centroid
(its node is bigger then other nodes), and have a color different
from other blocks.

plot_most_connected_members(clusters, centroids=None, warps=None,
nb_components=5)

Identifies the most connected components within each cluster, and
displays a plot of the corresponding fold changes' means. If warps are
given, then also displays the information on the warping groups of
the components.

compute_entity_path(path_e1_to_e2=None, entity_1=None, entity_2=None,
plot=True)

If entity_1 and entity_2 are given (and path_e1_to_e2 is not),
computes a shortest path from entity_1 to entity_2, and plots a figure
with the means of the fold changes in the path. If path_e1_to_e2 is
given, then produces a plot of the means of the fold changes'
in path_e1_to_e2.

draw_mesoscopic(clusters, centroids, obj_scale=1, node_label_size=30)

Displays a mesoscopic representation of the fold changes network, i.e.
a graph with k=len(centroids) nodes representing clusters, each labeled
by the name of the corresponding centroid, with sizes proportional to
respective cluster sizes. The edges represent connections between
clusters, their thickness is proportional to the respective number of
connections. If the network is directed, then arrow head sizes are
proportional to the percentage of connections of the corresponding
predictive type among all connections between the considered clusters.
In the latter case edges are annotated with the distribution among the
connection types (i.e. warps) in the following format: for an edge
between A and B, the annotation is of the form "% of predictive
connections from B to A - total number of connections between
A and B - % of predictive connections from A to B". In the case of
undirected graph, the edges are annotated with the corresponding
numbers of connections only.

graph_analysis(clusters, nb_top=10)

Performs a series of graph analyses of the fold changes network, in
particular: identifies among the entities nb_top top hits, authorities,
nodes with respect to pagerank, degree and betweenness centrality. It
also plots a figure displaying degree distribution of the nodes.

pathway_search(clusters)

Identifies all shortest paths between entities in the network of length
3 and bigger, and presents them along with their scores with respect to
criteria potentially relevant for hypothesis generation.

Inherited from Clustering.

Methods defined here:

__init__(self, data=None, means=None, cov=None, var_names=None, time_points=None, dist='d2hat', time_warp=False, max_warp_step=1, sign_pen=False, pen_param=1, random_gen=None, sparsity=0.75, directed=False, adj_mat=None)

Parameters
----
data : ndarray or None
If not None, 4D array with the dimensions corresponding to:
1) nb of time points, 2) two experimental conditions
(dim 0: control, dim 1: case)), 3) replicates, 4) nb of entities.
If None (by default), then the fold changes are constructed based
on 'means' and 'cov'. Either 'data' or 'means' and 'cov' have to be
non-None, with 'data' having priority for the fold changes
construction.
means : ndarray or None
If not None, 2D array of shape (nb_time_pts, nb_var)
containing data with `float` type, representing fold changes' means
for each entity and each time point. If 'data' is None, used to
construct fold changes. Either 'data' or 'means' and 'cov' have to
be non-None.
cov : ndarray or None
If not None, 3D array of shape (nb_time_pts, nb_var, nb_var)
containing data with `float` type, representing fold changes'

```
    nb_var x nb_var shaped covariance matrices for each time point.
    Time-wise cross-covariances are assumes to be 0 due to experimental
    design. In case of Hellinger distance, can also be 4-dimensional
    (natural form): (nb_time_pts, nb_time_pts, nb_var, nb_var).
    If 'data' is None, used to construct fold changes. Either 'data' or
    'means' and 'cov' have to be non-None.
var_names : array-like or None
    1D array-like containing data with `string` type, representing
    names of the measured entities (ex. genes). The default is None.
time_points : array-like or None
    1D array-like containing data with `float` type, representing time
    points at which fold changes were measured. The default is None.
dist : str
    Distance chosen for clustering, 'd2hat' by default (L2 distance
    between random estimators), can also be 'wasserstein'
    (Wasserstein distance) and 'hellinger' (Hellinger distance).
time_warp : bool
    If True, then the clustering procedure is coupled with the
    alignment. The default is False.
max_warp_step : int
    If max_warp_step=i>0, then the set of all considered warps is the
    set of all integers between -i and i.
sign_pen : bool
    If True, then the distance is penalized with sign penalty.
    The default is False.
pen_param : float
    Parameter determining the weight of sign penalty. The default is 1.
random_gen : RandomState instance or None
    Random number generator, used to reproduce results. If None
    (default), the generator is the RandomState instance used by
    `np.random`. If RandomState instance, random_gen is the actual
    random number generator.
sparsity : float, optional
    Sparsity of the network determining the cutoff when defining the
    binary adjacency matrix based on the weighted one.
    The default is 0.75.
directed : bool, optional
    If True, the network is directed, and undirected if False (default).
adj_mat : ndarray or None, optional
    If not None (default), 2D array of shape (nb_var, nb_var)
    indicating whether the fold changes are connected (i.e. similar
    enough) or not. If the network is undirected, then has 0 for
    connected fold changes and 1 for not connected (symmetric).
    A pair of fold changes is considered to be connected if their
    distance-based similarity is bigger then the cutoff value, which
    is equal to the empirical quantile of the similarity matrix
    corresponding to the chosen sparsity. If the network is directed,
    the matrix stops being symmetric, and the edges that exist
    according to the undirected case procedure become either 1 or 0
    based on the corresponding warp: 1 for the edges with the
    corresponding warps being positive (predictive) or 0
    (simultaneous), and 0 for those with negative warps (target).
    If 'adj_mat' is specified, the adjacency matrix is defined based
    its value, otherwise calculated based on the distance matrix and
    the optimal distance matrix. NB: in the former case 'optimal_warp_mat'
    is recalculated to correspond to 'adj_mat', however 'dist_mat'
    remains the same.
Returns
-------
None.
```

compute_entity_path(self, path_e1_to_e2=None, entity_1=None, entity_2=None, plot=True, figsize=(10, 7))

```
If entity_1 and entity_2 are given (and path_e1_to_e2 is not),
computes a shortest path from entity_1 to entity_2, and plots a figure
with the means of the fold changes in the path. If path_e1_to_e2 is
given, then produces a plot of the means of the fold changes'
in path_e1_to_e2.

Parameters
----------
path_e1_to_e2 : array-like or None, optional
```

```
    If not None (default), 1D container with strings (elements should
    belong to var_names) containing names of the entities as nodes
    in the path of interest (in the correct order).
    Either 'path_e1_to_e2' or 'entity_1' and 'entity_1' have to be
    non-None, with 'path_e1_to_e2' having priority for the path
    construction.
entity_1 : str or None, optional
   Starting node for path. The default is None.
   Either 'path_e1_to_e2' or 'entity_1' and 'entity_1' have to be
   non-None, with 'path_e1_to_e2' having priority for the path
   construction.
entity_2 : str or None, optional
    Ending node for path. The default is None.
    Either 'path_e1_to_e2' or 'entity_1' and 'entity_1' have to be
    non-None, with 'path_e1_to_e2' having priority for the path
    construction.
plot : bool, optional
    If True (default), displays a figure with the means of the fold
    changes in the path.
figsize : (float, float), optional
    Width and height of the figure(s). The default is (10,7).

Returns
-------
path_e1_to_e2 : array-like
    1D container with strings containing names of the entities as nodes
    in the path of interest.
path_e1_to_e2_warps : list
    Contains len(path_e1_to_e2)-1 elements, the warps between the
    consecutive nodes in the path, allows to determine the extend to
    which the path has a predictive character. Returned if the graph
    is directed.
```

compute_network(self, clusters, centroids, draw_path=False, path=None, figsize=(25, 25), obj_scale=1, graph_type='full', adj_mat_2=None, clusters_2=None, centroids_2=None, shade_intersect=False, degree_view=False)

```
Creates a NetworkX object representing the fold changes' network and
displays it in a block form arising from clusters. The network is
represented with a graph where nodes are the considered entities and
the edges are connections between them (i.e. ones in the adjacency
matrix). Members of every block are grouped around their centroid
(its node is bigger then other nodes), and have a color different
from other blocks.

Parameters
----------
clusters : ndarray
    1D array of length nb_var containing integers indicating clusters
    to which the fold changes are assigned.
centroids : ndarray
    1D array of length k containing indices in range (0, nb_var) of
    the fold changes that act as centroids.
draw_path : bool, optional
    False by default, if True and the path is given then the path is
    displayed on the graph with red nodes and thick red edges with the
    remaining edges thin and colored in light grey (the remaining
    nodes are displayed normally).
path : array-like or None, optional
    If not None (default), 1D container with strings (elements should
    belong to var_names) containing names of the entities as nodes
    in the path of interest (in the correct order).
figsize : (float, float), optional
    Width and height of the figure(s). The default is (25,25).
obj_scale : float, optional
    Parameter used to control the scale of objects in the graph, which
    zooms in if bigger than 1 and zooms out if smaller than 1.
    The default is 1.
graph_type : str, optional
    The following options are possible:
        - 'full' (default) : the whole graph is displayed, with edges
        colored in black if undirected, and grey for simultaneous
```

```
        and green for predictive connections if directed.
        - 'intersection' : if adj_mat_2 is given, displays only the
        intersection between the main network and the network defined
        by adj_mat_2.
        - 'difference' : if adj_mat_2 is given, displays the main
        network without its intersection with the network defined
        by adj_mat_2.
adj_mat_2 : ndarray or None, optional
    If not None (default), 2D array of shape (nb_var, nb_var)
    indicating whether the fold changes are connected or not (same
    to adj_mat). Represents the adjacency matrix of some other set of
    fold changes of interest. Should be based on the measurements for
    the same entities as the base network for a proper comparison.
    Used if graph_type is 'intersection' or 'difference'.
clusters_2 : ndarray, optional
    If not None (default), 1D array of length nb_var containing
    integers indicating clusters to which the fold changes are assigned.
    This alternative clustering specification serves to color the nodes
    with respect to the corresponding clustering (typically to compare
    clusters to clusters_2).
centroids_2 : ndarray, optional
    If not None (default), 1D array of length k containing indices in
    range (0, nb_var) of the fold changes that act as centroids. This
    second sets of centroids associated with an alternative clustering
    clusters_2 is used only for centroid node sizes (typically to
    compare centroids to centroids_2).
shade_intersect : bool, optional
    If True, adj_mat_2 is given, and graph_type is 'full' (makes no
    difference if 'intersection' or 'difference'), displays the entire
    graph but shades the intersection by coloring in lightgrey the
    nodes and the edges that belong entirely to the intersection with
    the network defined by adj_mat_2. The default is False.
degree_view : bool, optional
    If True, the sizes of nodes reflect their degrees (the relationship
    is increasing and non-linear). Otherwise (default), all nodes have
    the samesizes, except for the centroids that are bigger then the
    others.

Returns
-------
None.
```

draw_mesoscopic(self, clusters, centroids, obj_scale=1, node_label_size=30, figsize=(20, 20))

```
Displays a mesoscopic representation of the fold changes network, i.e.
a graph with k=len(centroids) nodes representing clusters, each labeled
by the name of the corresponding centroid, with sizes proportional to
respective cluster sizes. The edges represent connections between
clusters, their thickness is proportional to the respective number of
connections. If the network is directed, then arrow head sizes are
proportional to the percentage of connections of the corresponding
predictive type among all connections between the considered clusters.
In the latter case edges are annotated with the distribution among the
connection types (i.e. warps) in the following format: for an edge
between A and B, the annotation is of the form "% of predictive
connections from B to A - total number of connections between
A and B - % of predictive connections from A to B". In the case of
undirected graph, the edges are annotated with the corresponding
numbers of connections only.

Parameters
----------
clusters : ndarray
    1D array of length nb_var containing integers indicating clusters
    to which the fold changes are assigned.
centroids : ndarray
    1D array of length k containing indices in range (0, nb_var) of
    the fold changes that act as centroids.
obj_scale : float, optional
    Parameter used to control the scale of objects in the graph,
    which zooms in if bigger than 1 and zooms out if smaller than 1.
    The default is 1.
```

```
node_label_size : int, optional
    Font size for text labels on nodes (names of centroids).
    The default is 30.
figsize : (float, float), optional
    Width and height of the figure(s). The default is (20,20).

Returns
-------
None.
```

graph_analysis(self, clusters, nb_top=10)

```
Performs a series of graph analyses of the fold changes network, in
particular: identifies among the entities nb_top top hits, authorities,
nodes with respect to pagerank, degree and betweenness centrality. It
also plots a figure displaying degree distribution of the nodes.

Parameters
----------
clusters : ndarray
    1D array of length nb_var containing integers indicating clusters
    to which the fold changes are assigned.
nb_top : int, optional
    Number of top elements to include. The default is 10.

Returns
-------
graph_analysis : DataFrame
    2D DataFrame containing the names of entities that appeared in at
    least one of the considered tops as rows, and the following
    information as columns: cluster (number), all of the considered
    tops (1 if among the corresponding top and 0 otherwise), and total
    sum of all columns except the cluster one. Ordered so that the
    entities with the highest total score are at the top.
```

infer_sbm(self, nb_blocks, clusters, n_init=10, n_iter_early_stop=50, random=False, verbosity=0, pi_weight=0.8, random_gen=None)

```
        Performs stochastic block model inference for the fold changes' network
        based on clustering (i.e. on the constrained parameter space). This
        code is based on method 'fit' from class 'SBM' of the package SparseBM
        (https://github.com/gfrisch/sparsebm).

        Parameters
        ----------
        nb_blocks : int
            Number of blocks (communities/clusters) in the stochastic
            block model.
        clusters : ndarray or dictionary
            If ndarray, 1D array of length nb_var containing integers in range
            (0, k) indicating clusters to which the fold changes are assigned.
            If a dictionary, the keys are numbers of clusters considered, and
            for each such number the value is the latter array.
        n_init : int, optional
            Number of initializations. The default is 10.
        n_iter_early_stop : TYPE, optional
            Number of VEM iterations. The default is 50.
        random : bool, optional
            If True, stochastic block model is initialized on the parameter
            space defined by the original model. If False (default),
            stochastic block model is initialized on the constrained parameter
            space corresponding to base clustering.
        verbosity : int, optional
            Degree of verbosity. Scale from 0 (no message displayed) to 3.
            The default is 0.
        pi_weight : float, optional
            Weight parameter controlling the initialization of pi.
            pi(q,q)~Unif([pi_weight, 1)) and pi(q,q')~Unif([0,1-pi_weight))
            for q!=q'. The default is 0.8.
        random_gen : RandomState instance or None, optional
            Random number generator, used to reproduce results. If None
            (default), the generator is the RandomState instance used by
            `np.random`. If RandomState instance, random_gen is the actual
            random number generator.
```

```
        Returns
        -------
        successful_sbm : SBM instance or None
            Successfully trained stochastic block model, or None in case of
            failure.
,        sbm_centroids : ndarray or None
            1D array of length k containing indices in range (0, nb_var) of
            the fold changes that have been chosen as centroids (calculated
             after stochastic block model is inferred) if SBM is successfully
             inferred, otherwise None.
        comp_cost : float or None
            Value of the total comparable cost if SBM is successfully inferred,
            otherwise None.
```

pathway_search(self, clusters)

```
Identifies all shortest paths between entities in the network of length
3 and bigger, and presents them along with their scores with respect to
criteria potentially relevant for hypothesis generation.

Parameters
----------
clusters : ndarray
    1D array of length nb_var containing integers indicating clusters
    to which the fold changes are assigned.

Returns
-------
all_paths_dict : dict
    Dictionary with keys of type 'string' indicating the path length l,
    and the values are dataframes. Each raw of such dataframe
    corresponds to a path, the names of the nodes listed in the first
    l columns. There are three other columns: warp score (number of
    strictly positive warps in the path, i.e. number of predictive
    relationships), cluster score (number of times there is a change
    in cluster in the path), and total score (sum of the first two).
    Paths in the dataframe are ordered with respect to the total score
    (highest to lowest).
```

plot_most_connected_members(self, clusters, centroids=None, warps=None, nb_components=5, figsize=None)

```
Identifies the most connected components within each cluster, and
displays a plot of the corresponding fold changes' means. If warps are
given, then also displays the information on the warping groups of
the components.

Parameters
----------
clusters : ndarray
    1D array of length nb_var containing integers indicating clusters
    to which the fold changes are assigned.
centroids : ndarray or None, optional
    If not None (default), 1D array of length k containing indices in
    range (0, nb_var) of the fold changes that act as centroids.
warps : ndarray or None, optional
    If not None (default), 1D array of length nb_var containing
    integers in range (-max_warp_step, max_warp_step + 1)
    indicating fold changes' warps with respect to their
    corresponding centroids.
nb_components : int, optional
    Number of the most connected components to select in each cluster.
    The default is 5.
figsize : (float, float), optional
    Width and height of the figure(s). The default is None.

Returns
-------
most_connected_members_within : ndarray
    2D array of shape (nb_blocks, nb_components) containing indices
    in range (0, nb_var) of nb_components most connected components
    for each cluster (block).
```

APPENDIX B

FUNCTIONAL DATA APPROACH TO FOLD CHANGE ESTIMATION

Consider an observation from one of the studied datasets characterized by the following quantities: $i \in \{1, 2, \ldots, n_e\}$ where n_e is the number of considered biological entities, replicate $j \in \{1, 2, ..., n_r\}$, and experimental condition $k = 0$ if control and $k = 1$ if irradiated. Let $y_{ikj}(t)$ be a realization of the expression of a gene i under experimental condition k for replicate j at time point $t \in \mathbb{R}^+$. We consider the following model :

$$y_{ikj}(t) = \mu(t) + \mathbb{1}_{k=1}\alpha(t) + \beta_{ik}(t) + \epsilon_{ikj}(t). \tag{B.0.1}$$

The variables appearing in the model are as follows : $\mu(t)$ is the grand mean function, $\alpha(t)$ is the global fold change (that is, present in all genes), $\beta_{ik}(t)$ is the individual gene effect, and $\epsilon_{ikj}(t) \sim \mathcal{N}(0, \sigma^2_{\epsilon,ik})$ is the white noise. We can thus define local fold change for a gene i: $\Delta_i(t) = \beta_{i1}(t) - \beta_{i0}(t)$.

This model corresponds to a two-way functional ANOVA (f-ANOVA) model (Zhang, 2013), with the general form presented below:

$$y_{ikj}(t) = \mu(t) + \alpha_k(t) + \beta_i(t) + \theta_{ik}(t) + \epsilon_{ikj}(t). \tag{B.0.2}$$

In this model we can distinguish the main-effect of irradiation $\alpha_k(t) = \mathbb{1}_{k=1}\alpha(t)$, the main-effect of a gene $\beta_i(t) = \beta_{i0}(t)$, and the interaction-effect between irradiation and a gene: $\theta_{ik}(t) = 0$ if $k = 0$ and $\theta_{ik}(t) = \Delta_i(t)$ if $k = 1$.

The model can be rewritten in the following form allowing for direct estimation of regression parameters as presented in Ramsay and Silverman (2005):

$$Y(t) = Z\gamma(t) + \epsilon(t), \tag{B.0.3}$$

with $Y(t) = \begin{pmatrix} y_{101}(t) \\ y_{102}(t) \\ y_{103}(t) \\ y_{111}(t) \\ y_{112}(t) \\ y_{113}(t) \\ y_{201}(t) \\ \vdots \\ y_{N13}(t) \end{pmatrix}$, $\gamma(t) = \begin{pmatrix} \mu(t) \\ \alpha(t) \\ \beta_{10}(t) \\ \beta_{11}(t) \\ \vdots \\ \beta_{N0}(t) \\ \beta_{N1}(t) \end{pmatrix}$ and $Z = \begin{pmatrix} 1 & \cdots & 1 \\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1 & \cdots & 0\ 1\ 1\ 1 \\ 1\ 1\ 1\ 0 & \cdots & 0 \\ 0\ 0\ 0\ 1\ 1\ 1\ 0 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & 0\ 1\ 1\ 1\ 0\ 0\ 0 \\ 0 & \cdots & 0\ 1\ 1\ 1 \end{pmatrix}^t$.

APPENDIX C

ENRICHMENT ANALYSIS OF SARRP AND LINAC CLUSTERS AND SUBGROUPS

From the biological viewpoint, the enrichment analysis of the biological processes for each cluster after radiation at 220 kV (SARRP) has been carried out (C.0.2-C.0.6). Cluster 1 is characterized by 2 major functions mainly associated with adhesion and migration process but also with the apoptotic process or the cellular DNA damage response. Cluster 2 has fewer terms than Cluster 1 and is mainly defined by the regulation of signaling pathways such as the pi3kinase pathway which has been described in the literature as involved in the radiation response of endothelial cells (Edwards et al., 2002; Yentrapalli et al., 2013). Cluster 3 is characterized by TGFbeta and SMAD family related terms, which have been previously described in the literature on the endothelial vascular response and a glycosylation related term emerged from the enrichment analysis of cluster 3 (Milliat et al., 2006; Jaillet et al., 2017; Ladaigue et al., 2022). Cluster 4 is related to the function of the chemoattraction and cell-cell interaction with the immune system. These major features can be compared with the results for cluster 1 with respect to the term of adhesion. Moreover, in cluster 4 we see appear terms related to the control of the apoptotic process, a feature that also appears in the cluster 1. Cluster 4 is also linked to coagulation processes and fibrinolysis, described as a mark of radiation-induced endothelial response (Milliat et al., 2008). Lastly, cluster 5 is characterized by the activation of phosphorylation signaling pathways such as SMADs or the NFKB pathway. This can also be related to the enrichment of TGF feta in cluster 3. A protein glycosylation term, previously captured in cluster 3, can also be found in cluster 5, which is in accordance with the results published in the literature

showing an impact of glycosylation process in the radiation response of endothelial cells (Jaillet et al., 2017; Ladaigue et al., 2022). The results of the enrichment analysis coupled with the cluster and network analyses are illustrated in Figure C.0.1.

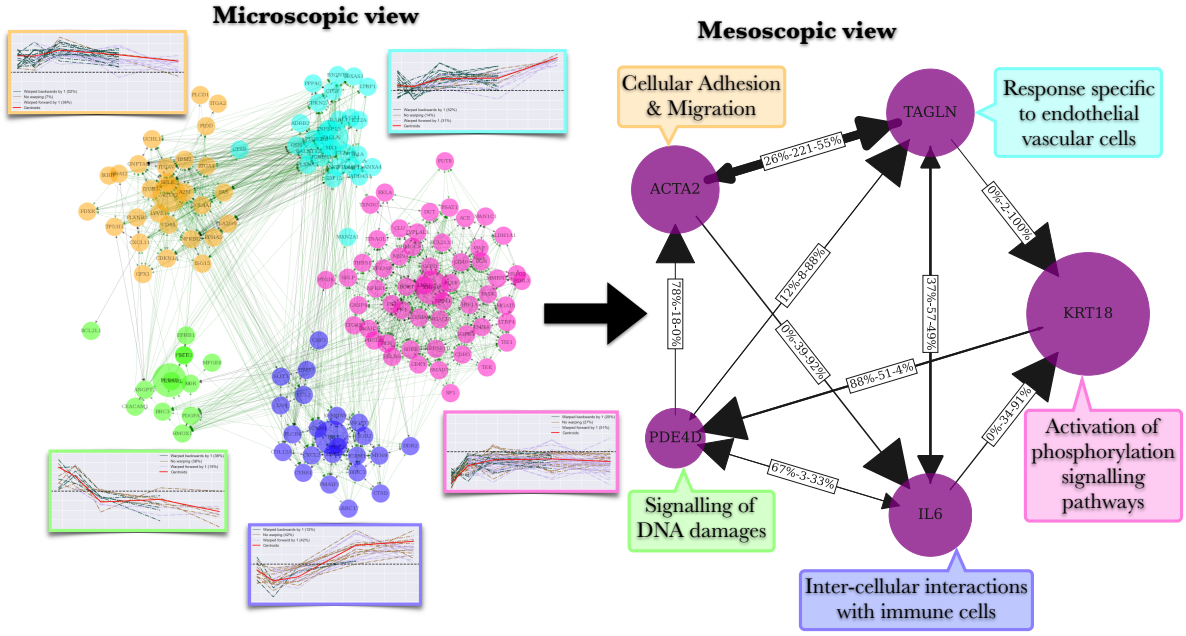


Figure (C.0.1) Summary of the cluster and network analyses performed on the SARRP dataset with the key biological functions identified as a result of the enrichment analysis.

The same analysis has been performed on the dataset obtained after irradiation at 4 MV (LINAC). The comparison of deregulated genes after radiation in the two irradiation conditions is illustrated through the Venn diagrams C.0.7-C.0.11. The enrichment analysis of biological processes (Figures C.0.12-C.0.16) reveals that irradiation at 4 MV has an impact on biological processes on all clusters. In particular, there are senescence-related terms that appear for 4 MV but are absent for 220 kV. This is consistent with the previously published results showing that cellular senescence is more important at 4 MV than at 220 kV (Paget et al., 2019), this illustrating the robustness of the analytical methodological approach implemented in this work.

The mathematical model also predicts a particular affinity for clusters 1 and 4 after irradiation at 220 kV and clusters 4 and 5 after irradiation at 4 MV suggesting that the terms appear in these various clusters may potentially explain the differences in response to the 2 energies. Combining the enrichment analyses of clusters 1 and 4 for 220 kV and clusters 4 and 5 for 4 MV, the terms that were globally identified focus mainly on cell adhesion and chemotaxis, suggesting a global energy-dependent effect on these inflammation-related parameters. It has already been shown that radiation

response of endothelial cells after 4MV compared with 220 kV is characterized by more senescence and more inflammatory induced response with upregulation of IL6 and IL8 higher at 4 MV than at 220 kV (Paget et al., 2019). These results reinforce the idea that the physical dose in Gray is not sufficient to predict a biological effect and by extrapolation a risk. Our results open biological hypotheses concerning the impact of radiation used in the medical field and in particular radiotherapy on both tumors and healthy tissues.

Figure (C.0.2) Results of the enrichment analysis performed with Pathfinder on the genes from cluster 1 of the SARRP dataset.

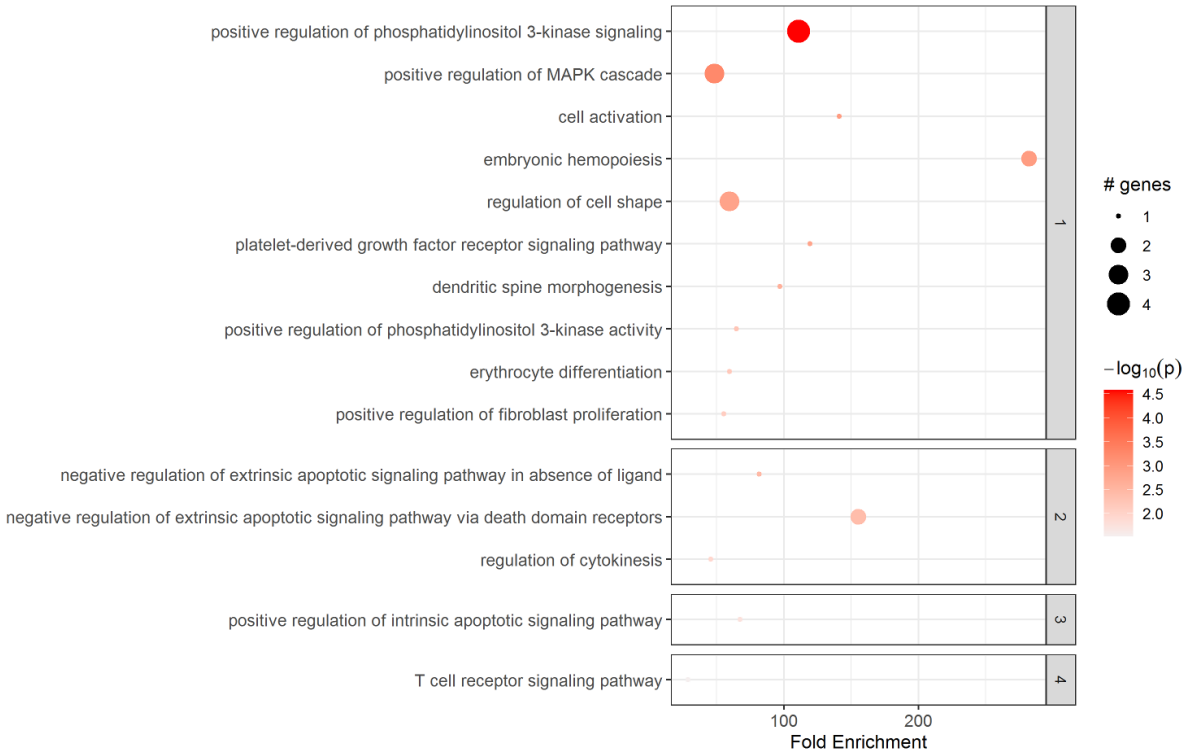


Figure (C.0.3) Results of the enrichment analysis performed with Pathfinder on the genes from cluster 2 of the SARRP dataset.

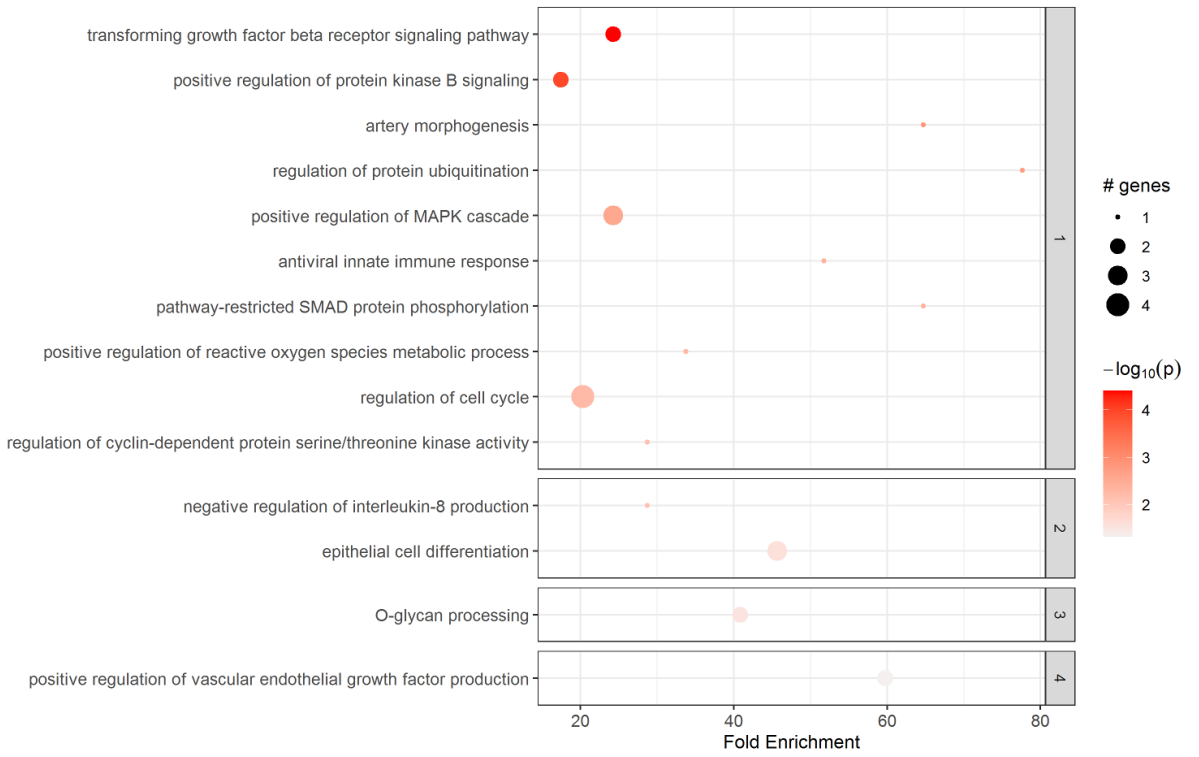


Figure (C.0.4) Results of the enrichment analysis performed with Pathfinder on the genes from cluster 3 of the SARRP dataset.

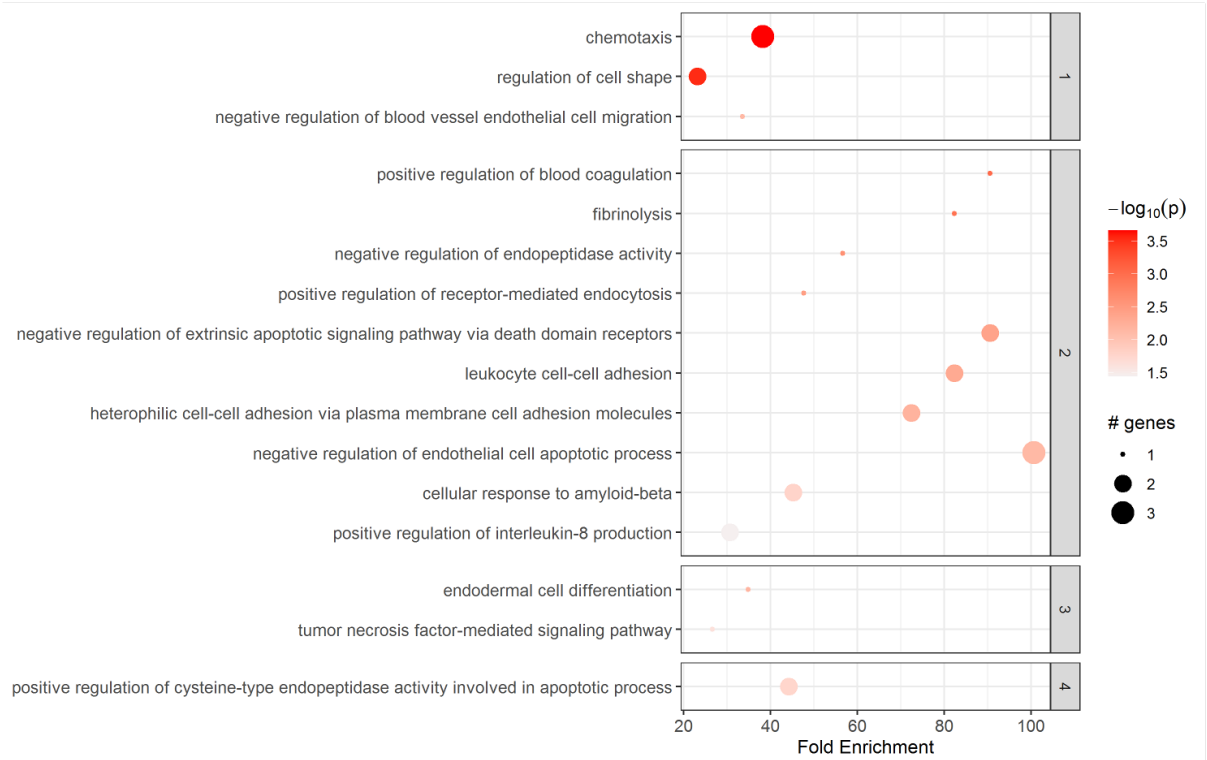


Figure (C.0.5) Results of the enrichment analysis performed with Pathfinder on the genes from cluster 4 of the SARRP dataset.

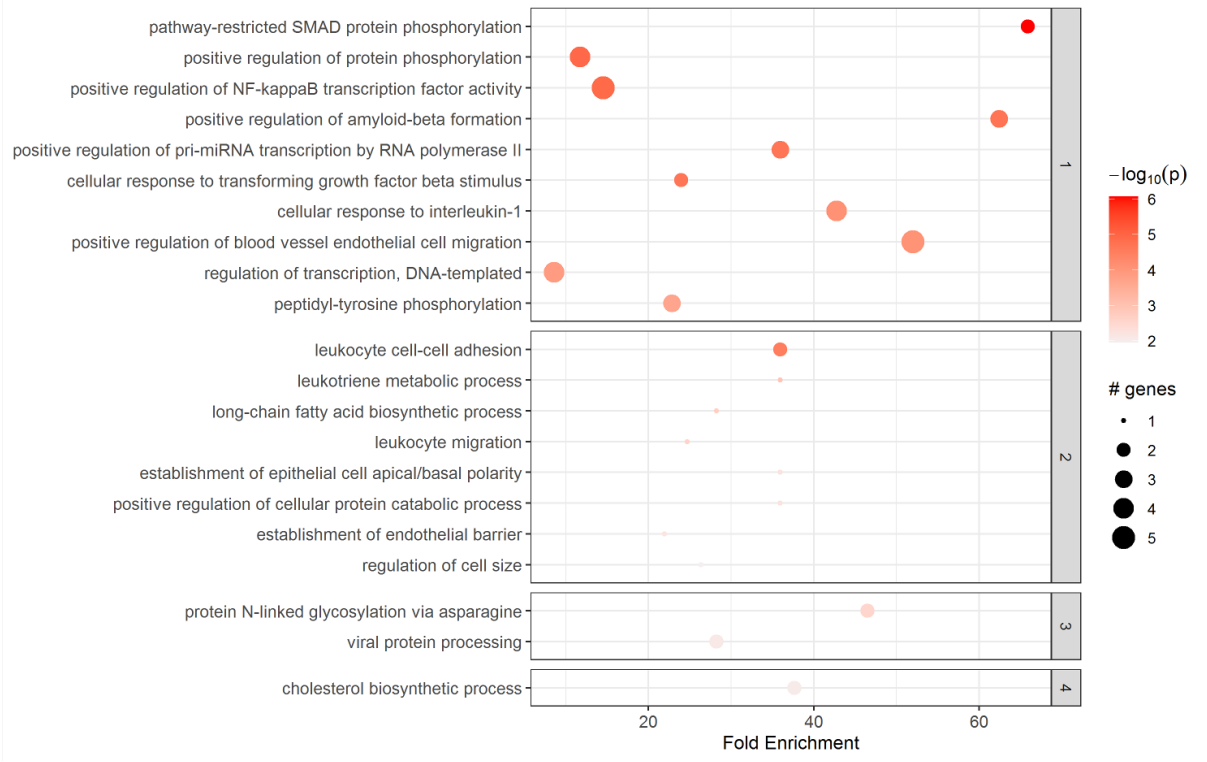


Figure (C.0.6) Results of the enrichment analysis performed with Pathfinder on the genes from cluster 5 of the SARRP dataset.

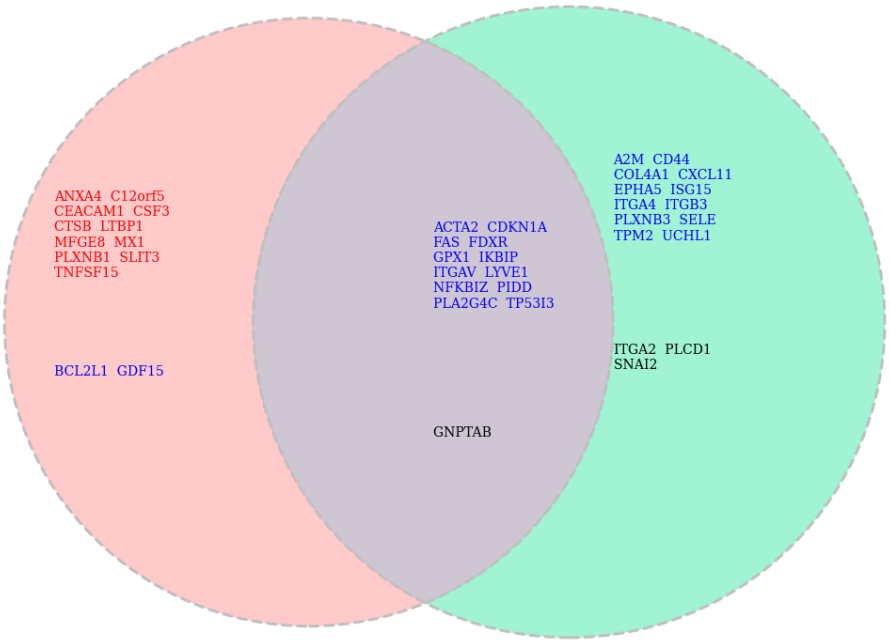


Figure (C.0.7) Venn diagrams illustrating the distribution of genes of cluster 1 between LINAC (left) and SARRP (right). Red indicates genes that have been assigned to this cluster by all methods (k-medoids, UMAP and SBM), blue indicates those that have been assigned to this cluster by two methods out of three, and black the remaining genes.

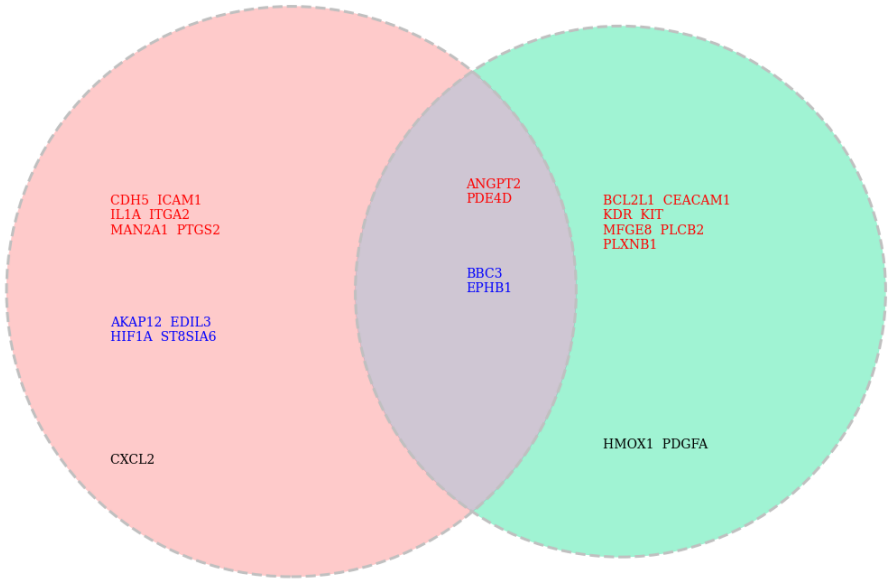


Figure (C.0.8) Venn diagrams illustrating the distribution of genes of cluster 2 between LINAC (left) and SARRP (right). Red indicates genes that have been assigned to this cluster by all methods (k-medoids, UMAP and SBM), blue indicates those that have been assigned to this cluster by two methods out of three, and black the remaining genes.

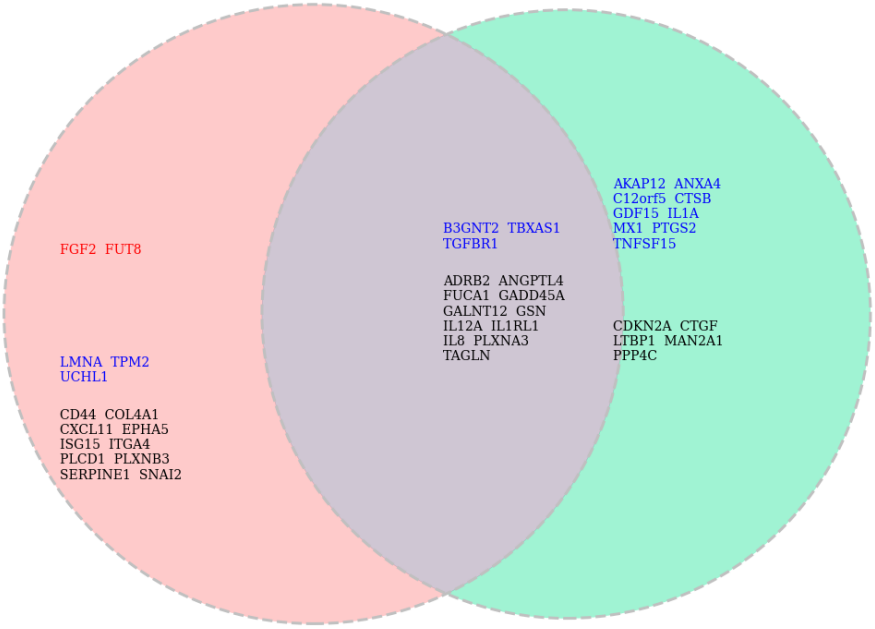


Figure (C.0.9) Venn diagrams illustrating the distribution of genes of cluster 3 between LINAC (left) and SARRP (right). Red indicates genes that have been assigned to this cluster by all methods (k-medoids, UMAP and SBM), blue indicates those that have been assigned to this cluster by two methods out of three, and black the remaining genes.

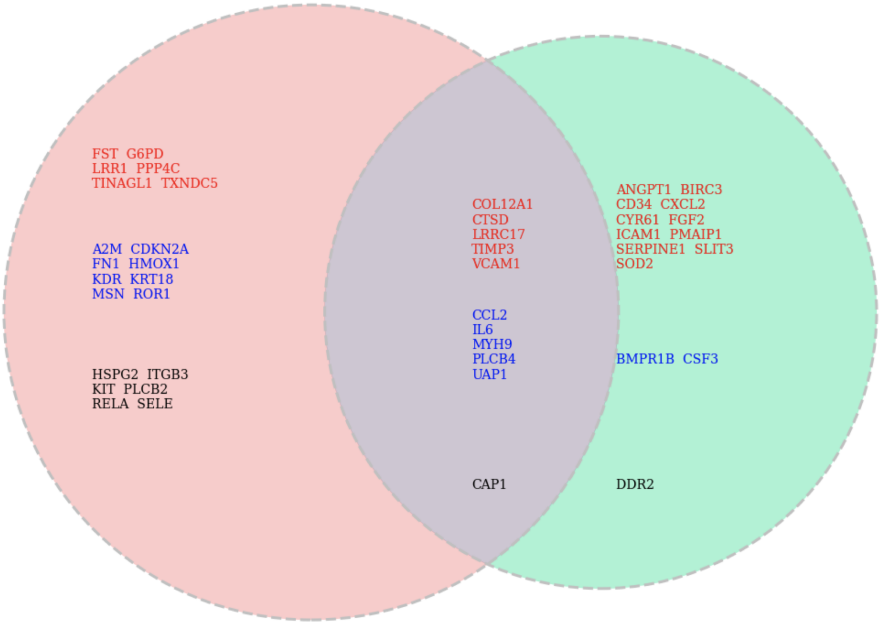


Figure (C.0.10) Venn diagrams illustrating the distribution of genes of cluster 4 between LINAC (left) and SARRP (right). Red indicates genes that have been assigned to this cluster by all methods (k-medoids, UMAP and SBM), blue indicates those that have been assigned to this cluster by two methods out of three, and black the remaining genes.

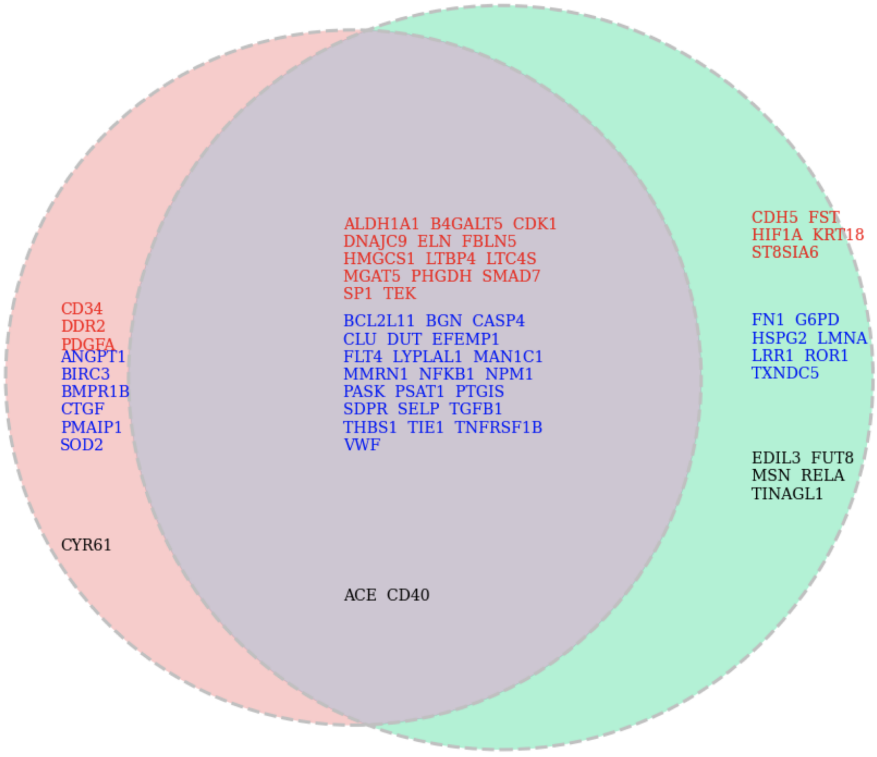


Figure (C.0.11) Venn diagrams illustrating the distribution of genes of cluster 5 between LINAC (left) and SARRP (right). Red indicates genes that have been assigned to this cluster by all methods (k-medoids, UMAP and SBM), blue indicates those that have been assigned to this cluster by two methods out of three, and black the remaining genes.

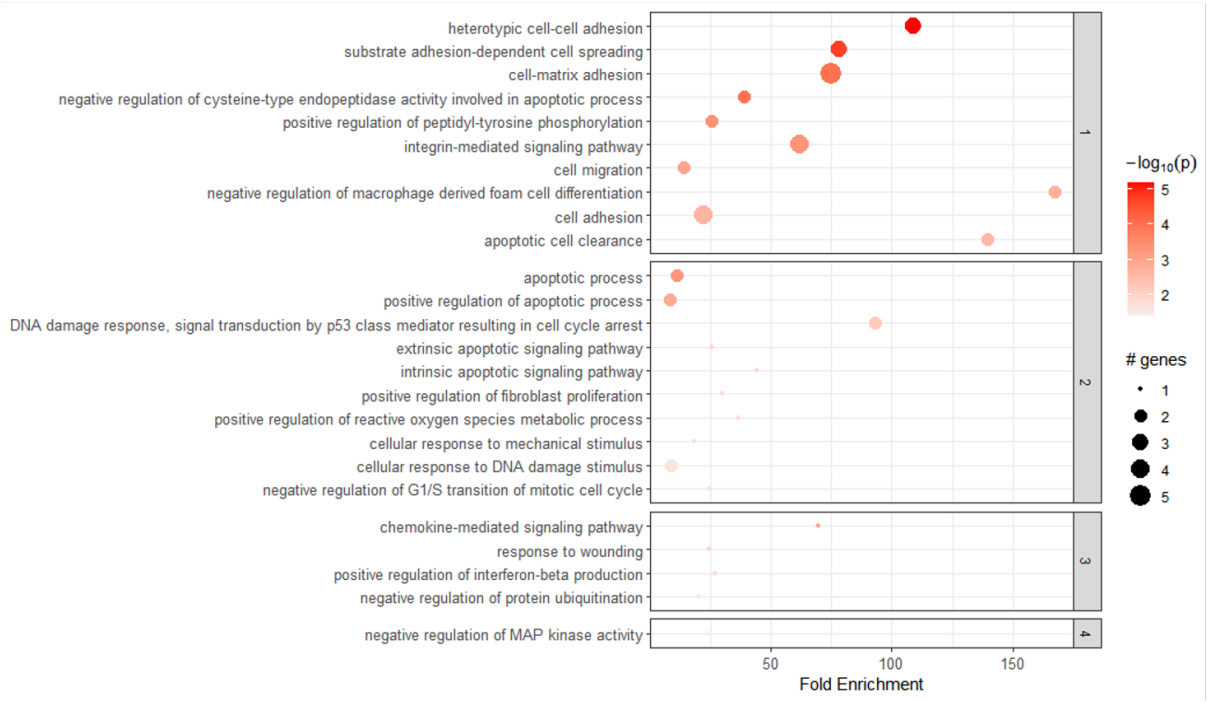


Figure (C.0.12) Results of the enrichment analysis performed with Pathfinder on the genes from cluster 1 of the LINAC dataset.

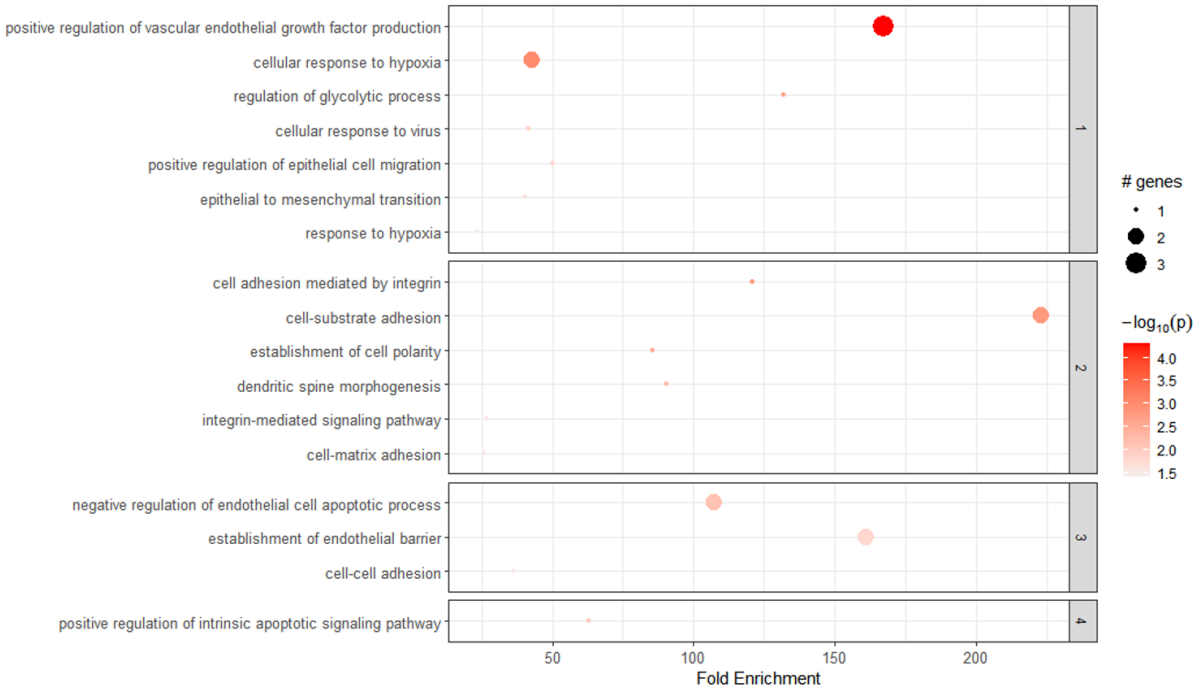


Figure (C.0.13) Results of the enrichment analysis performed with Pathfinder on the genes from cluster 2 of the LINAC dataset.

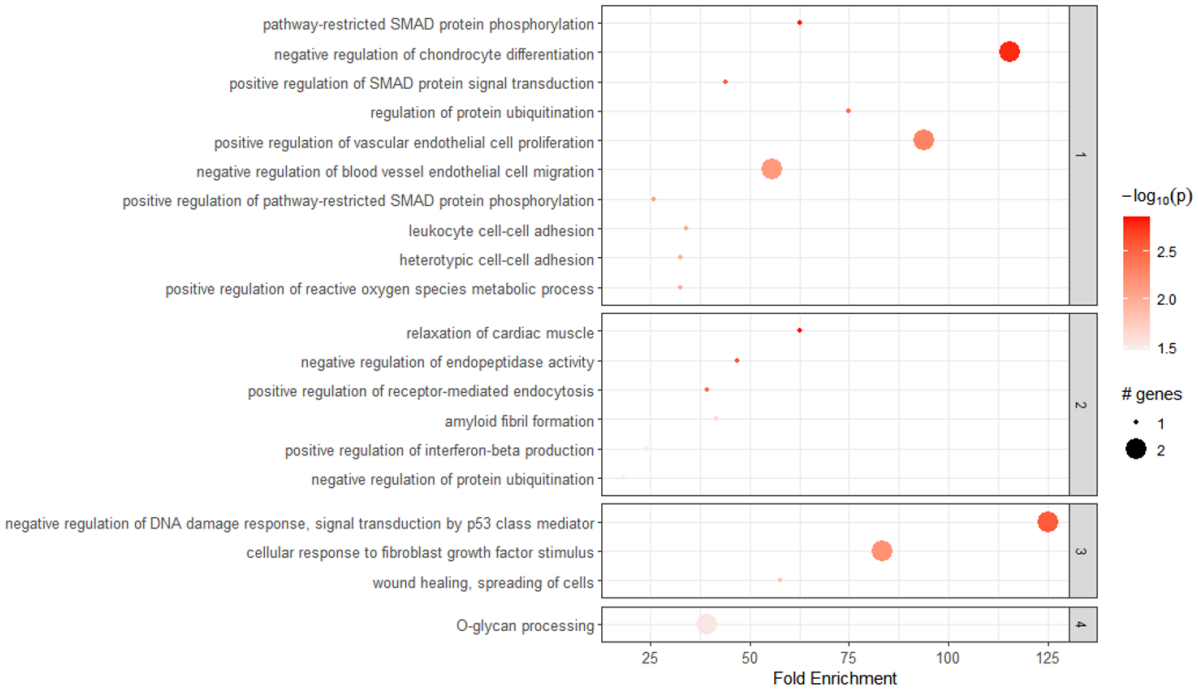


Figure (C.0.14) Results of the enrichment analysis performed with Pathfinder on the genes from cluster 3 of the LINAC dataset.

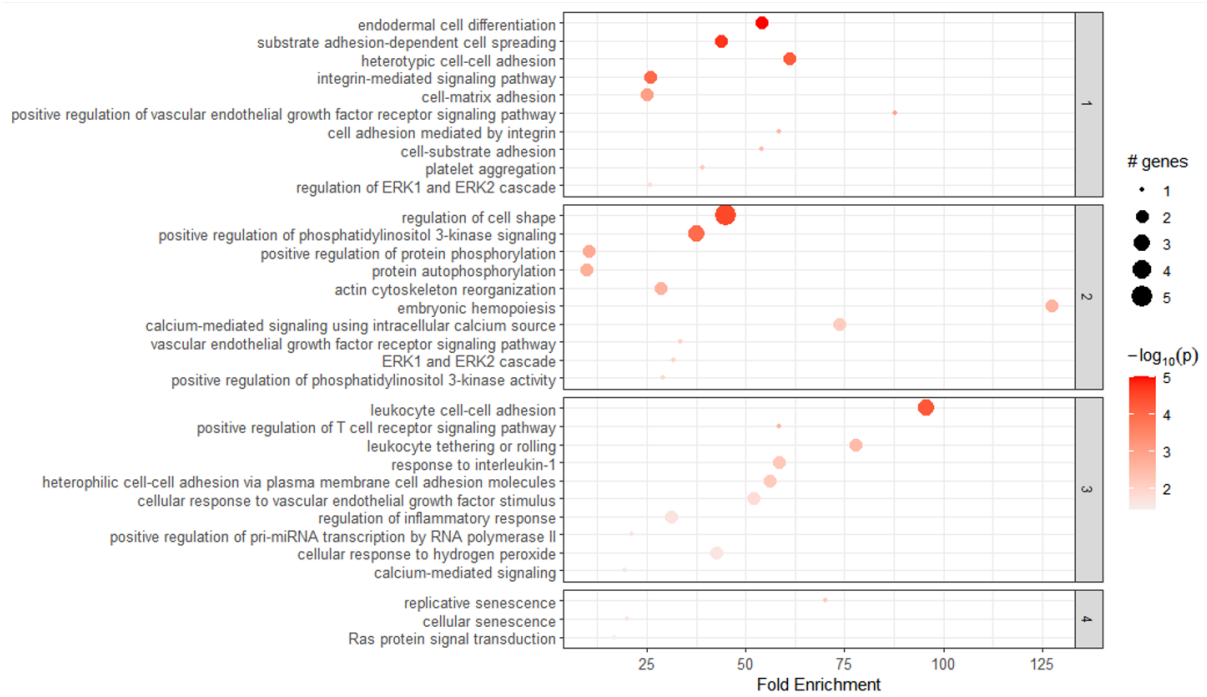


Figure (C.0.15) Results of the enrichment analysis performed with Pathfinder on the genes from cluster 4 of the LINAC dataset.

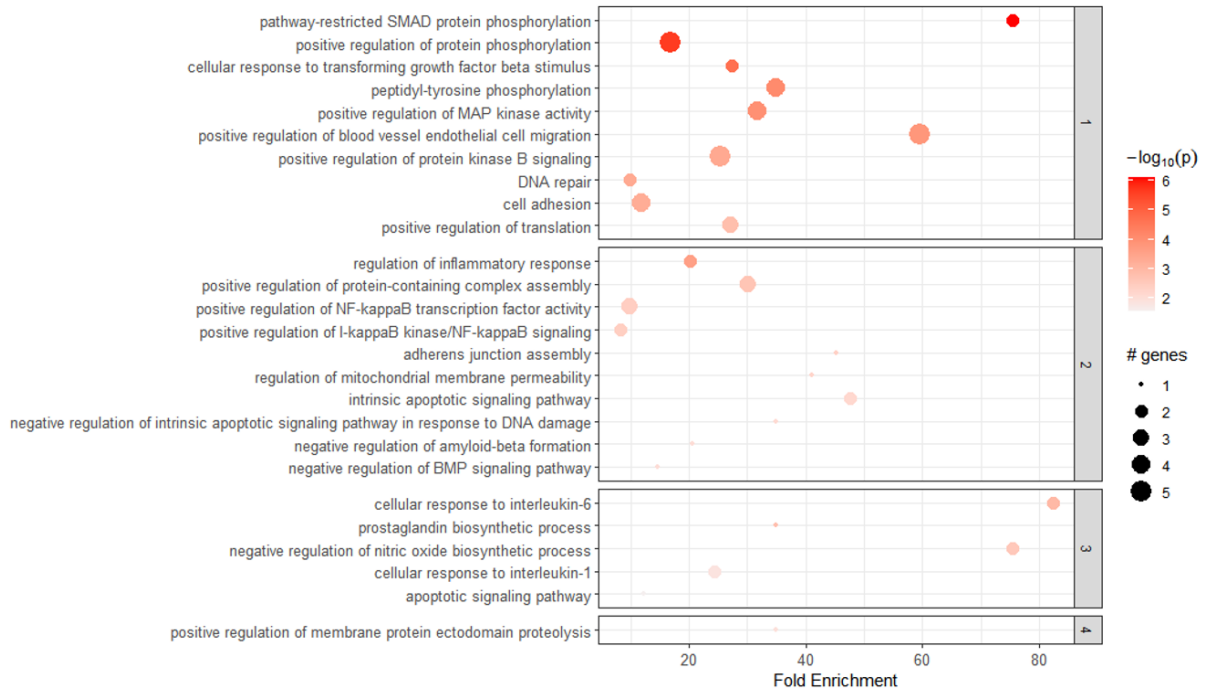


Figure (C.0.16)

Results of the enrichment analysis performed with Pathfinder on the genes from cluster 5 of the LINAC dataset.

APPENDIX D

SOME CLASSICAL THEOREMS IN ASYMPTOTIC STATISTICS

A proof of the classical continuous mapping theorem can be found in van der Vaart (1998) (Theorem 2.3).

THEOREM D.0.1. *(Continuous mapping theorem).*
Let $g : \mathbb{R}^d \to \mathbb{R}^m$ be continuous at every point of C such that $\mathbb{P}[X \in C] = 1$.
If the sequence of random variables $(X_n)_{n\geq 1}$ converges in distribution (resp. probability, resp. almost surely) to X then $(g(X_n))_{n\geq 1}$ converges in distribution (resp. probability, resp. almost surely) to $g(X)$.

We also recall some well known results that are useful to show the consistency of estimators $\widehat{\theta}_n$ defined as the minimizers of functionals $Q_n(\theta)$ which have some regularity properties at the limit.

THEOREM D.0.2. *(Lemma 2.9 in Newey and McFadden (1994))*
Suppose that $\theta \in \Theta$ and Θ is compact, $Q_0(\theta)$ is continuous and $\forall \theta \in \Theta$, $Q_n(\theta) \to Q_0(\theta)$ in probability as n tends to infinity. If there is $\alpha > 0$ and $B_n = O_p(1)$ such that

$$\forall (\widetilde{\theta}, \theta) \in \Theta \times \Theta,\ |Q_n(\widetilde{\theta}) - Q_n(\theta)| \leq B_n \|\widetilde{\theta} - \theta\|^\alpha$$

then

$$\sup_{\theta \in \Theta} |Q_n(\theta) - Q_0(\theta)| \to 0 \ \textit{in probability.}$$

THEOREM D.0.3. *(Theorem 2.1 in Newey and McFadden (1994))*
Suppose that $\theta \in \Theta$ and Θ is compact, $Q_0(\theta)$ is continuous $\forall \theta \in \Theta$. If $Q_0(\theta)$ is uniquely

maximized at θ_0 and, as n tends to infinity, $\sup_{\theta\in\Theta}|Q_n(\theta)-Q_0(\theta)|\to 0$ in probability, then $\widehat{\theta}_n\to\theta_0$ in probability.

Under additional hypotheses, we also get the asymptotic normality of the sequence of estimators $\widehat{\theta}_n$ of θ_0. we denote by $\nabla_{00}Q_n(\theta)$ the Hessian matrix of functional Q_n evaluated at θ.

THEOREM D.0.4. *(Theorem 3.1 in Newey and McFadden (1994))*
Suppose that $\widehat{\theta}_n\to\theta_0$ in probability, (i) θ_0 is an interior point of Θ, (ii) $Q_n(\theta)$ is twice differentiable in a neighborhood $\mathcal{N}$ of θ_0, (iii) $\sqrt{n}\nabla_0 Q_n(\theta_0)\rightsquigarrow\mathcal{N}(0,\mathbf{\Sigma})$, (iv) there is $\mathbf{H}(\theta)$ continuous at θ_0 and $\sup_{\theta\in\mathcal{N}}\|\nabla_{00}Q_n(\theta)-\mathbf{H}(\theta)\|\to 0$ in probability (v) $\mathbf{H}=\mathbf{H}(\theta_0)$ is non singular. Then

$$\sqrt{n}\left(\widehat{\theta}_n-\theta_0\right)\rightsquigarrow\mathcal{N}\left(0,\mathbf{H}^{-1}\mathbf{\Sigma}\mathbf{H}^{-1}\right)$$

We also recall the central limit theorem for bootstrap means (see Theorem 23.4 in van der Vaart (1998) for a proof).

THEOREM D.0.5. *(CLT for bootstrap means)*
Let $X_1,X_2,\ldots$ be i.i.d. random vectors with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Gamma}$. Then conditionally on $X_1,X_2,\ldots$, for almost every sequence $X_1,X_2,\ldots$

$$\sqrt{n}\left(\overline{X}_n^*-\overline{X}_n\right)\rightsquigarrow\mathcal{N}(0,\mathbf{\Gamma})$$

where $\overline{X}_n$ is the empirical mean and $\overline{X}_n^*$ is the empirical mean of n independent observations drawn from the empirical distribution.

BIBLIOGRAPHY

Afshar, Y., Ma, F., Quach, A., Jeong, A., Sunshine, H. L., Freitas, V., Jami-Alahmadi, Y., Helaers, R., Li, X., Pellegrini, M., Wohlschlegel, J. A., Romanoski, C. E., Vikkula, M., and Iruela-Arispe, M. L. (2023). Transcriptional drifts associated with environmental changes in endothelial cells. *eLife*, 12:e81370. Publisher: eLife Sciences Publications, Ltd.

Arthur, D. and Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 1027–1035, USA. Society for Industrial and Applied Mathematics.

Bertho, A., Santos, M. D., Buard, V., Paget, V., Guipaud, O., Tarlet, G., Milliat, F., and François, A. (2020). Preclinical model of stereotactic ablative lung irradiation using arc delivery in the mouse: Effect of beam size changes and dose effect at constant collimation. *International Journal of Radiation Oncology, Biology, Physics*, 107(3):548–562. Publisher: Elsevier.

Carrig, M. M., Manrique-Vallier, D., Ranby, K. W., Reiter, J. P., and Hoyle, R. H. (2015). A nonparametric, multiple imputation-based method for the retrospective integration of data sets. *Multivariate Behavioral Research*, 50(4):383–397.

Chacón, J. E. and Rastrojo, A. I. (2023). Minimum adjusted Rand index for two clusterings of a given size. *Advances in Data Analysis and Classification*, 17(1):125–133.

Cudeck, R. (2000). An estimate of the covariance between variables which are not jointly observed. *Psychometrika*, 65(4):539–546.

Delattre, M., Genon-Catalot, V., and Samson, A. (2016). Mixtures of stochastic differential equations with random effects: Application to data clustering. *Journal of Statistical Planning and Inference*, 173:109–124.

Donnet, S., Foulley, J.-L., and Samson, A. (2010). Bayesian analysis of growth curves using mixed models defined by stochastic differential equations. *Biometrics*, 66(3):733.

Edwards, E., Geng, L., Tan, J., Onishko, H., Donnelly, E., and Hallahan, D. E. (2002). Phosphatidylinositol 3-kinase/akt signaling in the response of vascular endothelium to ionizing radiation. *Cancer Res*, 62(16):4671–4677.

Frisch, G., Leger, J.-B., and Grandvalet, Y. (2021). SparseBM: A Python Module for Handling Sparse Graphs with Block Models. *hal-03139586*.

Genolini, C., Ecochard, R., Benghezal, M., Driss, T., Andrieu, S., and Subtil, F. (2016). kmlShape: An efficient method to cluster longitudinal data (time-series) according to their shapes. *PLOS ONE*, 11(6):e0150738. Publisher: Public Library of Science.

Givens, C. R. and Shortt, R. M. (1984). A class of Wasserstein metrics for probability distributions. *Michigan Mathematical Journal*, 31(2):231–240. Publisher: University of Michigan, Department of Mathematics.

Heerah, S., Molinari, R., Guerrier, S., and Marshall-Colon, A. (2021). Granger-causal testing for irregularly sampled time series with application to nitrogen signalling in Arabidopsis. *Bioinformatics*, 37(16):2450–2460.

Heinonen, M., Guipaud, O., Milliat, F., Buard, V., Micheau, B., Tarlet, G., Benderitter, M., Zehraoui, F., and d'Alché Buc, F. (2015). Detecting time periods of differential gene expression using Gaussian processes: an application to endothelial cells exposed to radiotherapy dose fraction. *Bioinformatics*, 31(5):728–735.

Hozumi, Y., Wang, R., Yin, C., and Wei, G.-W. (2021). UMAP-assisted K-means clustering of large-scale SARS-CoV-2 mutation datasets. *Computers in Biology and Medicine*, 131:104264.

Jaillet, C., Morelle, W., Slomianny, M.-C., Paget, V., Tarlet, G., Buard, V., Selbonne, S., Caffin, F., Rannou, E., Martinez, P., François, A., Foulquier, F., Allain, F., Milliat, F., and Guipaud, O. (2017). Radiation-induced changes in the glycome of endothelial cells with functional consequences. *Scientific Reports*, 7(1):5290.

Kaufmann, L. and Rousseeuw, P. (1987). Clustering by means of medoids. *Data Analysis based on the L1-Norm and Related Methods*, pages 405–416.

Kazlauskaite, I., Ek, C. H., and Campbell, N. (2019). Gaussian Process Latent Variable Alignment Learning. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 748–757. PMLR. ISSN: 2640-3498.

Ladaigue, S., Lefranc, A.-C., Balde, K., Quitoco, M., Bacquer, E., Busso, D., Piton, G., Dépagne, J., Déchamps, N., Yamakawa, N., Debusschere, L., Han, C., Allain, F.,

Buard, V., Tarlet, G., François, A., Paget, V., Milliat, F., and Guipaud, O. (2022). A role for endothelial alpha-mannosidase MAN1c1 in radiation-induced immune cell recruitment. *iScience*, 25(12):105482.

Lee, C. and Wilkinson, D. J. (2019). A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1):1–50. Number: 1 Publisher: SpringerOpen.

Liu, X. and Bouman Chen, Z. (2023). How their environment influences endothelial cells. *eLife*, 12:e88248. Publisher: eLife Sciences Publications, Ltd.

Luan, Y. and Li, H. (2003). Clustering of time-course gene expression data using a mixed-effects model with b-splines. *Bioinformatics*, 19(4):474–482.

Magnus, J. R. and Neudecker, H. (2019). *Matrix differential calculus with applications in statistics and econometrics*. Wiley Ser. Probab. Stat. Hoboken, NJ: John Wiley & Sons, 3rd updated edition edition.

Massa, M. S. and Riccomagno, E. (2017). Algebraic representations of Gaussian Markov combinations. *Bernoulli*, 23(1):626–644. Publisher: Bernoulli Society for Mathematical Statistics and Probability.

McInnes, L., Healy, J., and Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. Technical Report arXiv:1802.03426, arXiv.

Milliat, F., François, A., Isoir, M., Deutsch, E., Tamarat, R., Tarlet, G., Atfi, A., Validire, P., Bourhis, J., Sabourin, J.-C., and Benderitter, M. (2006). Influence of endothelial cells on vascular smooth muscle cells phenotype after irradiation. *Am J Pathol*, 169(4):1484–1495.

Milliat, F., Sabourin, J.-C., Tarlet, G., Holler, V., Deutsch, E., Buard, V., Tamarat, R., Atfi, A., Benderitter, M., and François, A. (2008). Essential role of plasminogen activator inhibitor type-1 in radiation enteropathy. *Am J Pathol*, 172(3):691–701.

Mitsuhiro, M. and Hoshino, T. (2020). Kernel canonical correlation analysis for data combination of multiple-source datasets. *Japanese Journal of Statistics and Data Science*, 3(2):651–668.

Mitsuhiro, M. and Hoshino, T. (2021). Bayesian data combination model with Gaussian process latent variable model for mixed observed variables under NMAR missingness. arXiv:2109.00462 [stat].

Munshi, A., Hobbs, M., and Meyn, R. E. (2005). Clonogenic cell survival assay. *Methods in molecular medicine*, 110:21–28.

Müller, H.-G., Chiou, J.-M., and Leng, X. (2008). Inferring gene expression dynamics via functional regression analysis. *BMC Bioinformatics*, 9(1):60.

Newey, W. K. and McFadden, D. (1994). Large sample estimation and hypothesis testing. In *Handbook of econometrics, Vol. IV*, volume 2 of *Handbooks in Econom.*, pages 2111–2245. North-Holland, Amsterdam.

Nguyen, P. and Braun, R. (2018). Semi-supervised network inference using simulated gene expression dynamics. *Bioinformatics*, 34(7):1148–1156.

Paget, V., Ben Kacem, M., Dos Santos, M., Benadjaoud, M. A., Soysouvanh, F., Buard, V., Georges, T., Vaurijoux, A., Gruel, G., François, A., Guipaud, O., and Milliat, F. (2019). Multiparametric radiobiological assays show that variation of X-ray energy strongly impacts relative biological effectiveness: comparison between 220 kv and 4 mv. *Scientific Reports*, 9(1):14328.

Panaretos, V. M. and Zemel, Y. (2019). Statistical aspects of Wasserstein distances. *Annu. Rev. Stat. Appl.*, 6:405–431.

Park, H.-S. and Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341.

Ramsay, J. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer Science & Business Media.

Riccadonna, S., Jurman, G., Visintainer, R., Filosi, M., and Furlanello, C. (2016). DTW-MIC Coexpression Networks from Time-Course Data. *PLOS ONE*, 11(3):e0152648. Publisher: Public Library of Science.

Rosenberg, A. and Hirschberg, J. (2007). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Sangalli, L. M., Secchi, P., Vantini, S., and Veneziani, A. (2009). A Case Study in Exploratory Functional Data Analysis: Geometrical Features of the Internal Carotid Artery. *Journal of the American Statistical Association*, 104(485):37–48. Publisher: Taylor & Francis.

Sangalli, L. M., Secchi, P., Vantini, S., and Vitelli, V. (2010). k-mean alignment for curve clustering. *Computational Statistics & Data Analysis*, 54(5):1219–1233.

Schaffter, T., Marbach, D., and Floreano, D. (2011). GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics (Oxford, England)*, 27(16):2263–2270.

Shao, J. and Tu, D. (1995). *The Jackknife and Bootstrap*. Springer Ser. Stat. New York, NY: Springer-Verlag.

Triantafillou, S., Tsamardinos, I., and Tollis, I. (2010). Learning causal structure from overlapping variable sets. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 860–867. JMLR Workshop and Conference Proceedings. ISSN: 1938-7228.

Tsamardinos, I., Triantafillou, S., and Lagani, V. (2012). Towards Integrative Causal Analysis of Heterogeneous Data Sets and Studies. *Journal of Machine Learning Research*, 13(39):1097–1157.

Valentin, J. (2003). Relative biological effectiveness (RBE), quality factor (q), and radiation weighting factor (wR): ICRP publication 92: Approved by the commission in january 2003. *Annals of the ICRP*, 33(4):1–121. Publisher: SAGE Publications Ltd.

van der Vaart, A. W. (1998). *Asymptotic statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge.

Verdinelli, I. and Wasserman, L. (2019). Hybrid Wasserstein distance and fast distribution clustering. *Electronic Journal of Statistics*, 13(2):5088–5119.

Wang, Y., Huang, H., Rudin, C., and Shaposhnik, Y. (2022). Understanding how dimension reduction tools work: an empirical approach to deciphering t-SNE, UMAP, TriMap, and PaCMAP for data visualization. *The Journal of Machine Learning Research*, 22(1):201:9129–201:9201.

Warren, J. L., Yabroff, K. R., Meekins, A., Topor, M., Lamont, E. B., and Brown, M. L. (2008). Evaluation of trends in the cost of initial cancer treatment. *Journal of the National Cancer Institute*, 100(12):888–897.

Wasserman, L. (2023). Optimal transport in statistics. *54th Journées de Statistique of the SFdS, Université libre de Bruxelles*.

Yao, F., Müller, H.-G., and Wang, J.-L. (2005). Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590. Publisher: Taylor & Francis _eprint: https://doi.org/10.1198/016214504000001745.

Yentrapalli, R., Azimzadeh, O., Sriharshan, A., Malinowsky, K., Merl, J., Wojcik, A., Harms-Ringdahl, M., Atkinson, M. J., Becker, K.-F., Haghdoost, S., and Tapio, S. (2013). The PI3k/akt/mTOR pathway is implicated in the premature senescence of primary human endothelial cells exposed to chronic radiation. *PLoS One*,

8(8):e70024.

Zhang, J.-T. (2013). *Analysis of Variance for Functional Data*. Chapman and Hall/CRC. Journal Abbreviation: Analysis of Variance for Functional Data Pages: 381 Publication Title: Analysis of Variance for Functional Data.