Optimal transport-based machine learning to match specific expression patterns in omics data

Thi Thanh Yen Nguyen, Olivier Bouaziz, Warith Harchaoui, Christian Neri, Antoine Chambaz

Optimal transport-based machine learning to match specific expression patterns in omics data

Thi Thanh Yen Nguyen[1], Olivier Bouaziz[1], Warith Harchaoui[1],
Christian Neri[2], Antoine Chambaz[1]

[1] MAP5 (UMR CNRS 8145), Université de Paris
[2] CNRS UMR 8256, INSERM ERL U1164, Sorbonne Université, Brain-C Lab, Paris, France

July 21, 2021

Abstract

We present two algorithms designed to learn a pattern of correspondence between two data sets in situations where it is desirable to match elements that exhibit an affine relationship. In the motivating case study, the challenge is to better understand micro-RNA (miRNA) regulation in the striatum of Huntington's disease (HD) model mice. The two data sets contain miRNA and messenger-RNA (mRNA) data, respectively, each data point consisting in a multi-dimensional profile. The biological hypothesis is that if a miRNA induces the degradation of a target mRNA or blocks its translation into proteins, or both, then the profile of the former should be similar to minus the profile of the latter (a particular form of affine relationship).

The algorithms unfold in two stages. During the first stage, an optimal transport plan P and an optimal affine transformation are learned, using the Sinkhorn-Knopp algorithm and a mini-batch gradient descent. During the second stage, P is exploited to derive either several co-clusters or several sets of matched elements.

A simulation study illustrates how the algorithms work and perform. A brief summary of the real data application in the motivating case-study further illustrates the applicability and interest of the algorithms.

Keywords. Co-clustering; omics data; Huntington's disease; matching; optimal transport; Sinkhorn algorithm; Sinkhorn loss.

1 Introduction

The analysis of numerous omics data is a challenging task in biological research [4] and disease research [11, 14]. In disease research, omics data are increasingly available for the analysis of molecular pathology. This is notably illustrated by research on Huntington's Disease (HD):

micro-RNA (miRNA), messenger-RNA (mRNA), protein data collectively quantifying several layers of molecular regulation in the brain of HD model knock-in mice [11, 12] now compose one of the largest data set available to date to understand how neurodegenerative processes may work on a systems level.

Encouraged by the promising findings of [15], our ultimate goal is to shed light on the interaction between mRNAs and miRNAs based on data collected in the striatum (a brain region) of HD model knock-in mice [11, 12]. Each data point takes the form of multi-dimensional profile. The biological hypothesis is that if a miRNA induces the degradation of a target mRNA or blocks its translation into proteins, or both, then the profile of the former should be similar to minus the profile of the latter (a particular form of affine relationship). In order to identify groups of mRNAs and miRNAs that interact, we develop a co-clustering algorithm and a matching algorithm based on optimal transport [16], spectral and block co-clustering, and a matching procedure tailored to our needs.

The present article focuses on the methodological developments. A separate article (in preparation) will show and interpret the complete results of the data analysis using the tools developed here.

Spectral co-clustering [6] and block clustering [5, 9] are two ways among many others to carry out co-clustering, an unsupervised learning task to cluster simultaneously the rows and columns of a matrix in order to obtain homogeneous blocks. There are many efficient approaches to solving the problem, often characterized as model-based or metric-based methods [18]. We derive the dissimilarity matrix to co-cluster from the data by optimal transport.

Section 2 describes the data we use. Section 3 presents a modicum of optimal transport theory. Section 4 introduces our algorithms. Section 5 evaluates the performances of the algorithms in various simulation settings. Section 6 illustrates the real data application. Section 7 closes the study on a discussion.

2 Data

The data analyzed herein cover RNA-seq data obtained in the striatum of the allelic series of HD knock-in mice (Q20, Q80, Q92, Q111, Q140, Q175) at 2-month, 6-month and 10-month of age. After preprocessing [15, Methods section], the final data set consists of $M = 13,616$ mRNA profiles, $X := \{x_1, \ldots, x_M\} \subset \mathbb{R}^d$, and in $N = 1,143$ miRNA profiles, $Y := \{y_1, \ldots, y_N\} \subset \mathbb{R}^d$ with $d = 15$.

Formally, we look for mutually disjoint $I_1, \ldots, I_R$ subsets of $[\![M]\!] := \{1, \ldots, M\}$ and mutually disjoint $J_1, \ldots, J_R$ subsets of $[\![N]\!]$ such that, for all $r \in [\![R]\!]$, each x_m with $m \in I_r$ interacts with every y_n with $n \in J_r$. Next, we describe what we mean by interacting.

It is known that the miRNAs and their target mRNAs exhibit a many-to-many mirroring relationship. We conduct our analysis under the biological hypothesis that if a miRNA induces the degradation of a target mRNA or blocks its translation into proteins, or both, then the profile y_n of the former should be similar to minus the profile x_m of the latter. However, we acknowledge that the actual mirroring relationships can be more or less acute (*e.g.*, due to threshold effects, or to multiple miRNAs targeting the same mRNA, or to a single miRNA targeting several mRNAs). Therefore, our algorithms will learn from the data a relevant transformation close to minus identity but not necessarily equal to it.

Figure 1 exhibits two profiles x_m and y_n that showcase a mirrored similarity. The corresponding miRNA and mRNA, Mir20b (which may inhibit cerebral ischemia-induced inflammation in rats [20]) and the Aryl-Hydrocarbon Receptor Repressor (Ahrr), are believed to interact in the striatum of HD model knock-in mice [15].

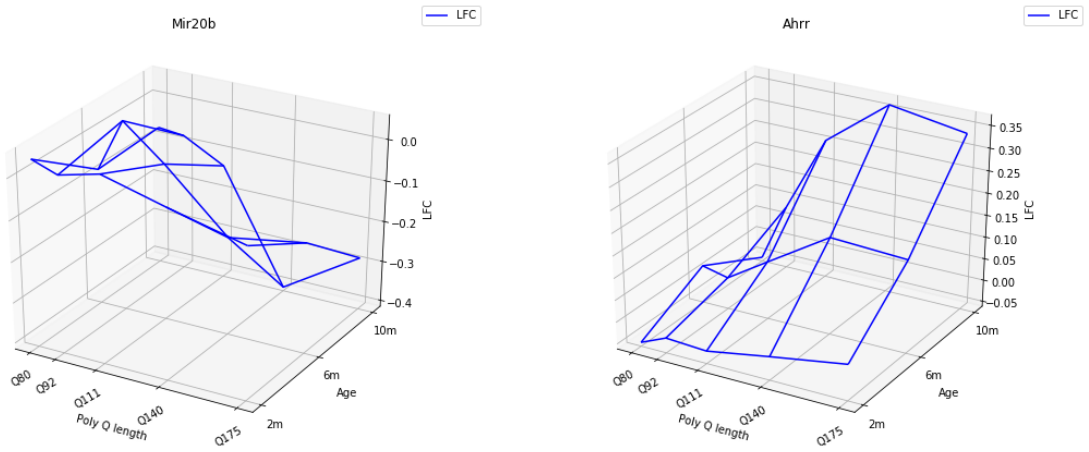

Figure 1: Left: profile y_n of a miRNA (Mir20b). Right: profile x_m of a mRNA (Ahrr). It is believed that Mir20b targets Ahrr.

3 Elements of optimal transport

Let $\Omega := \{\omega \in (\mathbb{R}_+)^M | \sum_{m\in[\![M]\!]} \omega_m = 1\}$ be the $(M-1)$-dimensional simplex and $\bar{\omega} := M^{-1}\mathbf{1}_M$, where $\mathbf{1}_M \in \mathbb{R}^M$ is the vector with all its entries equal to 1. For any $\omega \in \Omega$, define

$$\Pi(\omega) := \{P \in (\mathbb{R}_+)^{M\times N} | P\mathbf{1}_N = \omega, P^\top \mathbf{1}_M = N^{-1}\mathbf{1}_N\}$$

and let $\mu_X^\omega := \sum_{m\in[\![M]\!]} \omega_m \delta_{x_m}$, $\nu_Y := N^{-1}\sum_{n\in[\![N]\!]} \delta_{y_n}$ be the ω-weighted empirical measure attached to X and the empirical measure attached to Y. An element P of $\Pi(\omega)$ represents a joint law on $X \times Y$ with marginals μ_X^ω and ν_Y.

The celebrated Monge-Kantorovich problem [16, Chapter 2] consists in finding a joint law over $X \times Y$ with marginals $\mu_X^{\bar{\omega}}$ and ν_Y that minimizes the expected cost of transport with respect to some cost function $c : X \times Y \to \mathbb{R}_+$. We focus on c given by $c(x, y) := \|x - y\|_2^2$ (the squared Euclidean norm in $\mathbb{R}^d$). Specifically, denoting $C_{X,Y} \in \mathbb{R}^{M\times N}$ the cost matrix given by $(C_{X,Y})_{mn} := c(x_m, y_n)$ for each $(m, n) \in [\![M]\!] \times [\![N]\!]$, the problem consists in solving $\min_{P\in\Pi(\bar{\omega})} \langle C_{X,Y}, P\rangle_F$ where $\langle C_{X,Y}, P\rangle_F := \sum_{(m,n)\in[\![M]\!]\times[\![N]\!]} (C_{X,Y})_{mn} P_{mn}$ is the P-specific expected cost of transport from X to Y.

It is well known that it is very rewarding from a computational viewpoint to consider a

regularized version of the above problem [16, Chapter 4]. The penalty term is proportional to the discretized entropy of P, that is, to $E(P) := -\sum_{(m,n)\in[\![M]\!]\times[\![N]\!]} P_{mn}(\log P_{mn} - 1)$. The regularized problem (presented here for any $\omega \in \Omega$ beyond the case $\omega = \bar{\omega}$) consists, for some user-supplied $\gamma > 0$, in finding P_γ that solves

$$\mathcal{W}_\gamma\left(\mu_X^\omega, \nu_Y\right) := \min_{P\in\Pi(\omega)} \left\{\langle C_{X,Y}, P\rangle_F - \gamma E(P)\right\}. \tag{1}$$

One of the advantages of entropic regularization is that one can solve (1) efficiently using the Sinkhorn-Knopp matrix scaling algorithm.

Finally, following [8], we use $\mathcal{W}_\gamma$ to define the so called Sinkhorn loss between μ_X^ω (any $\omega \in \Omega$) and ν_Y as

$$\bar{\mathcal{W}}_\gamma\left(\mu_X^\omega, \nu_Y\right) := 2\mathcal{W}_\gamma\left(\mu_X^\omega, \nu_Y\right) - \mathcal{W}_\gamma\left(\mu_X^\omega, \mu_X^\omega\right) - \mathcal{W}_\gamma\left(\nu_Y, \nu_Y\right).$$

This loss interpolates between $\mathcal{W}_0\left(\mu_X^\omega, \nu_Y\right)$ and the maximum mean discrepancy of μ_X^ω relative to ν_Y [8, Theorem 1]. Paraphrasing the abstract of [8], the interpolation allows to find "a sweet spot" leveraging the geometry of optimal transport and the favorable high-dimensional sample complexity of maximum mean discrepancy, which comes with unbiased gradient estimates.

4 Optimal transport-based machine learning

4.1 Description of the algorithm

We introduce a parametric model Θ consisting of affine mappings $\theta : \mathbb{R}^d \to \mathbb{R}^d$ of the form $x \mapsto \theta(x) = \theta_1 x + \theta_2$, where $\theta_1 \in \mathbb{R}^{d\times d}$ and $\theta_2 \in \mathbb{R}^d$. The formal definition of Θ is given in Appendix A. Each $\theta \in \Theta$ is a candidate to formalize the aforementioned mirroring relationship. The set Θ imposes constraints on the matrices θ_1, in particular that their diagonals are made of negative values. Of course, minus identity belongs to Θ. The parametrization is identifiable, in the sense that $\theta = \theta'$ implies $(\theta_1, \theta_2) = (\theta_1', \theta_2')$. It is noteworthy that *any* identifiable, regular model Θ could be used. We focus on Θ as defined in Appendix A because of the application

that we consider in Section 6 (and in Section 5).

By analogy with Section 3 we introduce, for any $\theta \in \Theta$, $\omega \in \Omega$ and $\gamma > 0$, $\theta(X) := \{\theta(x_1), \ldots, \theta(x_M)\}$ the image of X by θ; the ω-weighted empirical measure attached to $\theta(X)$, $\mu^{\omega}_{\theta(X)} := \sum_{m \in [\![M]\!]} \omega_m \delta_{\theta(x_m)}$; the cost matrix $C_{\theta(X),Y}$ given by $(C_{\theta(X),Y})_{mn} := c(\theta(x_m), y_n)$ for each $(m,n) \in [\![M]\!] \times [\![N]\!]$; and

$$\mathcal{W}_\gamma \left(\mu^{\omega}_{\theta(X)}, \nu_Y \right) = \min_{P \in \Pi(\omega)} \left\{ \langle C_{\theta(X),Y}, P \rangle_F - \gamma E(P) \right\} \tag{2}$$

where $\langle C_{\theta(X),Y}, P \rangle_F := \sum_{(m,n) \in [\![M]\!] \times [\![N]\!]} (C_{\theta(X),Y})_{mn} P_{mn}$ is the P-specific expected cost of transport from $\theta(X)$ to Y.

Fix arbitrarily $\omega \in \Omega$. The first program that we introduce is the ω-specific program

$$\min_{\theta \in \Theta} \bar{\mathcal{W}}_\gamma \left(\mu^{\omega}_{\theta(X)}, \nu_Y \right), \tag{3}$$

where we are interested in the minimizer $\hat{\theta}$ that solves (3) *and* in the optimal joint matrix $\hat{P} \in \Pi(\omega)$ that solves

$$\min_{P \in \Pi(\omega)} \left\{ \langle C_{\hat{\theta}(X),Y}, P \rangle_F - \gamma E(P) \right\}.$$

In words, we look for an ω-specific optimal mirroring function $\hat{\theta}$ and its ω-specific optimal transport plan $\hat{P}$.

How to choose ω? We decide to optimize with respect to ω as well. This additional optimization is relevant because we do not expect to associate a y_n to every x_m eventually at the co-clustering stage. So, our main program is

$$\min_{\omega \in \Omega} \min_{\theta \in \Theta} \bar{\mathcal{W}}_\gamma \left(\mu^{\omega}_{\theta(X)}, \nu_Y \right), \tag{4}$$

where we are interested in the minimizer $(\hat{\omega}, \hat{\theta})$ *and* in the optimal matrix $\hat{P} \in \Pi(\hat{\omega})$ that solves

$$\min_{P \in \Pi(\hat{\omega})} \left\{ \langle C_{\hat{\theta}(X),Y}, P \rangle_F - \gamma E(P) \right\}.$$

We propose to solve (4) iteratively by updating ω and then θ. At round t, given ω_t, we make one step of mini-batch gradient descent to derive θ_{t+1} from θ_t (here, we notably rely on the Sinkhorn-Knopp algorithm). Given θ_{t+1}, ω_{t+1} is chosen proportional to the vector in $(\mathbb{R}_+)^M$ whose mth component equals $h^{-1}\sum_{n\in[\![N]\!]}\varphi((y_n-\theta_{t+1}(x_m))/h)$ where φ is the standard normal density and h is the arithmetic mean of the $c(y_n,y_{n'})$ for all $n\neq n'\in[\![N]\!]$. Eventually, once the final round T is completed, we compute $\tilde{P}\in\Pi(\omega_T)$ that solves

$$\min_{P\in\Pi(\omega_T)}\left\{\langle C_{\theta_T(X),Y},P\rangle_F-\gamma E(P)\right\}.$$

(again, we rely on the Sinkhorn-Knopp algorithm).

The algorithm is summarized in Procedure 1. In light of [2, Section 1.3, page 25], we inject problem-specific knowledge onto two of the three main components of the transportation problem: the representation spaces (via the mapping θ) and the marginal constraints (via the weight ω), leaving aside the cost function. Furthermore, we resort to mini-batch gradient descent because the algorithmic complexity prevents the direct computation using the whole data set. A theoretical analysis of this practice is proposed in [7].

We can now exploit $\tilde{P}$ so as to derive relevant associations between mRNAs and miRNAs. We propose two approaches. On the one hand, the first approach outputs *bona fide* co-clusters. We expect that the co-clusters can associate many mRNAs with many miRNAs, thus making it difficult to interpret and analyze the results. On the other hand, the second approach rather *matches* each mRNA with at most k miRNAs and each miRNA with at most k' mRNAs (k and k' are user-supplied integers). Details follow.

4.1.1 Co-clustering.

To carry out the co-clustering task once $\tilde{P}$ has been derived, we propose to rely either on spectral co-clustering (we will use the acronym SCC) [6], applying it once or twice, or co-clustering based on latent block models [9]. Of course, any other co-clustering algorithm could be used as well. Specifically, we develop the following algorithms (the acronym WTOT stands for weighted

transformation optimal transport).

WTOT-SCC1. Algorithm WTOT-SCC1 applies SCC *once* to build *bona fide* co-clusters based on $\tilde{P}$. It is required to provide a number of clusters. We rely on a criterion involving graph modularity to learn from the data a relevant number of clusters [1, Sections 2 and 4].

In our simulation study, we also consider algorithm WTOT-SCC1*, an oracular version of WTOT-SCC1 that benefits from relying on the *true* number of clusters. This allows to assess how relevant is the learned number of clusters in WTOT-SCC1.

WTOT-SCC2. Algorithm WTOT-SCC2 applies SCC *twice* to build *bona fide* co-clusters based on $\tilde{P}$. It proceeds in three successive steps.

- In step 1, WTOT-SCC2 applies SCC a first time to derive an initial co-clustering. A relevant number of co-clusters is learnt as in WTOT-SCC1.
- In step 2, WTOT-SCC2 selects and removes some rows and columns corresponding to mRNAs and miRNAs that are deemed irrelevant. The selection is based on a numerical criterion computed from $\tilde{P}$. In our simulation study (Section 5), all rows and columns that correspond to diagonal blocks with a variance larger than two times the overall variance of $\tilde{P}$ are selected and removed. In the real data application (Section 6), we implement and use a different procedure.
- In step 3, WTOT-SCC2 applies SCC a second time, the relevant number of co-clusters being learnt as in WTOT-SCC1.

In our simulation study, we also consider algorithm WTOT-SCC2*, an oracular version of WTOT-SCC2 that is provided the *true* number of clusters for its third step. This allows to assess how relevant is the sub-procedure to learn the numbers of clusters in WTOT-SCC2.

WTOT-BC. Algorithm WTOT-BC applies the so called block clustering algorithm to build *bona fide* co-clusters based on $\tilde{P}$. It is required to provide the row- and column-specific

numbers of clusters. We rely on an integrated completed likelihood criterion [5] to learn relevant values from the data.

The co-clusters obtained *via* WTOT-SCC1, WTOT-SCC2 or WTOT-BC should reveal the interplay between the (remaining, as far as WTOT-SCC2 is concerned) mRNAs and miRNAs in HD.

4.1.2 Matching.

The larger $\tilde{P}_{mn}$ is, the more we are encouraged to believe that the profiles x_m and y_n reveal a strong relationship between the mth mRNA and the nth miRNA. This simple rule prompts the following matching procedure applied once $\tilde{P}$ has been derived.

WTOT-matching. Fix two integers $k, k' \geq 1$ and let $\tilde{\tau}$ be the quantile of order q of all the entries of $\tilde{P}$. For every $m \in [\![M]\!]$ and $n \in [\![N]\!]$, we introduce

$$\mathcal{N}_m^0 := \Big\{ n \in [\![N]\!] : \tilde{P}_{mn} \in \{\tilde{P}_{m(1)}, \ldots, \tilde{P}_{m(k)}\} \text{ and } \tilde{P}_{mn} \geq \tilde{\tau} \Big\},$$
$$\mathcal{M}_n^0 := \Big\{ m \in [\![M]\!] : \tilde{P}_{mn} \in \{\tilde{P}_{(1)n}, \ldots, \tilde{P}_{(k')n}\} \text{ and } \tilde{P}_{mn} \geq \tilde{\tau} \Big\}$$

where $\tilde{P}_{m(1)}, \ldots, \tilde{P}_{m(k)}$ are the k largest values among $\tilde{P}_{m1}, \ldots, \tilde{P}_{mN}$ and $\tilde{P}_{(1)n}, \ldots, \tilde{P}_{(k')m}$ are the k' largest values among $\tilde{P}_{1n}, \ldots, \tilde{P}_{Mn}$. For instance, $\mathcal{N}_m^0$ identifies the miRNAs that are the k more likely to have a strong relationship with the mth mRNA. However, this does not qualify them as relevant matches yet. In order to keep only matches that are really relevant, we also introduce, for each $m \in [\![M]\!]$ and $n \in [\![N]\!]$,

$$\mathcal{N}_m := \mathcal{N}_m^0 \cap \{n \in [\![N]\!] : m \in \mathcal{M}_n^0\},$$
$$\mathcal{M}_n := \mathcal{M}_n^0 \cap \{m \in [\![M]\!] : n \in \mathcal{N}_m^0\}.$$

Algorithm WTOT-matching outputs the collections $\{\mathcal{N}_m : m \in [\![M]\!]\}$ and $\{\mathcal{M}_n : n \in [\![N]\!]\}$.

Now if, for instance, $n \in \mathcal{N}_m$ then y_n is among the k miRNA profiles upon which $\tilde{P}$ puts more mass when it "transports" x_m onto Y *and* x_m is among the k' mRNA profiles upon which $\tilde{P}$ puts more mass when it "transports" y_n onto X.

Note that we expect that some $\mathcal{N}_m$ and $\mathcal{M}_n$ will be empty, depending on k and k'. The mRNAs and miRNAs worthy of interest are those for which $\mathcal{N}_m$ and $\mathcal{M}_n$ are not empty. The integers k and k' should be chosen relatively small, to make their interpretation and analysis feasible, but not too small because otherwise few matchings will be made.

In the simulation study, we use $k = k'$ between 2 and 200, depending on the simulation scheme. Moreover, we choose $q = 50\%$ so that $\tilde{\tau}$ is the median of the entries of $\tilde{P}$.

4.2 Implementation of the method

Our code is written in `python` and will be made available soon. We adapt the Sinkhorn algorithm implemented by Aude Genevay and available here. The stochastic gradient descents relies on the machine learning framework `pytorch`. We use the implementation of SCC available in the `sklearn` `python` module. To learn a relevant number of clusters, we rely on the `coclust` `python` module. Finally, we rely on the `blockcluster` `R` package to carry out block clustering.

Our algorithm bears a similarity to the one developed in [10]. The main differences are *(i)* our use of the parametric model Θ and weights ω, *(ii)* the fact that we apply SCC or block clustering to the approximation of the optimal transport matrix $\tilde{P}$. Our algorithm also bears a similarity to [19], a fast and certifiable point cloud registration algorithm. We plan to study the similarities and differences closely.

5 Simulation study

To assess the performances of the algorithm described in Section 4.1, we conduct a simulation study in three parts. As we go on, the task gets more difficult. In all cases, the laws of the synthetic observations are mixtures of Gaussian laws. In Section 5.4, the weights of the mixtures and parameters of the Gaussian laws are chosen by us. Moreover, the two mixtures (to simulate

X and Y) share the same weights and induce a perfect mirroring relationship (details below), thus making the co-clustering task less difficult. In Section 5.5, the weights of the mixtures and parameters of the Gaussian laws are randomly generated. Moreover, the two mixtures do not share the same weights and do not induce a perfect mirroring relationship anymore, so that the co-clustering task is much more difficult. Finally, in Section 5.6, we use plus or minus real, randomly chosen miRNA profiles *and* $\mathbf{0}_d$ as means of the Gaussian laws to simulate X and Y, in such a way that there is no perfect mirroring relationship. We think that the corresponding co-clustering task is the most difficult of the three.

Section 5.1 briefly introduces two competing algorithms to identify matchings [10]. Section 5.2 lists all the algorithms that compete in the simulation study and Section 5.3 presents the measure of discrepancy between two co-clusterings and the matching criteria that we rely on to assess how well the algorithms perform. Sections 5.4, 5.5 and 5.6 present in turn the data-generating mechanisms and report the results in terms of co-clustering and matching performances.

5.1 Two "Gromov-Wasserstein co-clustering" algorithms

We compare our algorithms with two co-clustering algorithms adapted from [10]. For self-containedness, we summarize here how these algorithms work.

The first step of both algorithms consists in computing the similarity matrices $K_X \in (\mathbb{R}_+)^{M\times M}$ and $K_Y \in (\mathbb{R}_+)^{N\times N}$ given by

$$(K_X)_{mm'} := \exp\left\{-\frac{\|x_m - x_{m'}\|_2^2}{2\ell_X^2}\right\} \quad (m, m' \in [\![M]\!]),$$
$$(K_Y)_{nn'} := \exp\left\{-\frac{\|y_n - y_{n'}\|_2^2}{2\ell_Y^2}\right\} \quad (n, n' \in [\![N]\!])$$

where ℓ_X (respectively, ℓ_Y) is the mean of all pairwise Euclidean distances between elements of X (respectively, of Y). The similarity matrices K_X and K_Y now represent X and Y through the lense of the so called radial basis function kernel.

For any integers $a, b \geq 1$ and pair of matrices $A \in \mathbb{R}^{a\times a}$ and $B \in \mathbb{R}^{b\times b}$, define

$$\begin{aligned}
\Pi_{a,b} &:= \left\{ P \in (\mathbb{R}_+)^{a\times b} | P\mathbf{1}_b = a^{-1}\mathbf{1}_a, P^\top \mathbf{1}_a = b^1 \mathbf{1}_b \right\}, \\
\langle [A,B],[P,P] \rangle_F &:= \sum_{i,k\in[\![a]\!],j,\ell\in[\![b]\!]} (A_{ik} - B_{j\ell})^2 P_{ij} P_{k\ell} \quad (P \in \Pi_{a,b}), \\
\mathcal{GW}_\gamma(A,B) &:= \min_{P\in\Pi_{a,b}} \{ \langle [A,B],[P,P] \rangle_F - \gamma E(P) \qquad (5)
\end{aligned}$$

where $E(P) := -\sum_{(i,j)\in[\![a]\!]\times[\![b]\!]} P_{ij}(\log P_{ij} - 1)$. The quantity $\mathcal{GW}_\gamma(A,B)$ is known in the literature as an entropic Gromov-Wasserstein discrepancy between A and B. It can be used to define an entropic Gromov-Wasserstein barycenter of A and B and its barycenter transport matrices. Specifically, setting $s = \lfloor \frac{1}{2}(a+b) \rfloor$ (one choice among many), $(\hat{\Gamma}, \hat{P}_A, \hat{P}_B) \in (\mathbb{R}_+)^{s\times s} \times \Pi_{s,a} \times \Pi_{s,b}$ that solves

$$\min_{\Gamma, P_A, P_B} \frac{1}{2} \left\{ \Big(\langle [\Gamma, A], [P_A, P_A] \rangle_F - \gamma E(P_A) \Big) + \Big(\langle [\Gamma, B], [P_B, P_B] \rangle_F - \gamma E(P_B) \Big) \right\} \qquad (6)$$

(where (Γ, P_A, P_B) ranges over $(\mathbb{R}_+)^{s\times s} \times \Pi_{s,a} \times \Pi_{s,b}$) can be interpreted as a barycenter between A and B ($\hat{\Gamma}$) and the optimal transport matrices between $\hat{\Gamma}$ and A ($\hat{P}_A$) and between $\hat{\Gamma}$ and B ($\hat{P}_B$).

The second step of the algorithms consists either in solving numerically (5) with $(A,B) = (K_X, K_Y)$, yielding $\tilde{Q}$, or in solving numerically (6) with $(A,B) = (K_X, K_Y)$, yielding in particular the transport matrices $\tilde{Q}_X$ and $\tilde{Q}_Y$. We call CCOT-GWD and CCOT-GWB the corresponding algorithms. In both cases, the Sinkhorn-Knopp algorithm is used and provides solutions that decompose as

$$\begin{aligned}
\tilde{Q} &= \operatorname{diag}(\rho) \xi \operatorname{diag}(\rho'), \\
\tilde{Q}_X &= \operatorname{diag}(\rho_X) \xi_X \operatorname{diag}(\rho'_X), \\
\tilde{Q}_Y &= \operatorname{diag}(\rho_Y) \xi_Y \operatorname{diag}(\rho'_Y),
\end{aligned}$$

for some $\rho, \rho_X \in \mathbb{R}^M$, $\rho', \rho'_Y \in \mathbb{R}^N$, $\rho_X, \rho_Y \in \mathbb{R}^s$ and $\xi \in \mathbb{R}^{M\times N}, \xi_X \in \mathbb{R}^{s\times M}, \xi_Y \in \mathbb{R}^{s\times N}$ [17].

The third and last step builds upon either (ρ, ρ') or (ρ'_X, ρ'_Y) to derive partitions of X and Y, by detecting "jumps" along the vectors. The two partitions finally yield a co-clustering.

5.2 Listing all competing algorithms

We run and compare algorithms WTOT-SCC1, WTOT-SCC2 (and their oracular counterparts WTOT-SCC1*, WTOT-SCC2*), WTOT-BC on the one hand (see Sections 4.1.1) and CCOT-GWD and CCOT-GWB on the other hand (see Section 5.1). In addition, we also run algorithm WTOT-matching (see Section 4.1.2).

In view of Algorithm 1, $\widetilde{M}$ and $\widetilde{N}$ equal approximately $M/2$ and $N/2$ respectively, $(\eta, \gamma_0) = (1, 0)$ (no decay), $T = 500$, and the initial mapping θ_0 is drawn randomly.

5.3 Assessing performances

A measure of discrepancy between two co-clusterings. In order to assess the quality of the co-clusterings that we derive, and to compare performances, we propose to rely on a commonly used measure of discrepancy between two co-clusterings. Its definition extends that of a measure of discrepancy between partitions that we first present.

Let z and z' be two partitions of the set $[\![M]\!]$ into K components, taking the form of matrices $z = (z_{mk})_{m\in[\![M]\!],k\in[\![K]\!]}$ and $z' = (z'_{mk})_{m\in[\![M]\!],k\in[\![K]\!]}$ with convention $z_{mk} = 1$ (respectively, $z'_{mk} = 1$) if m belongs to component k of z (respectively, z') and 0 otherwise. The corresponding confusion matrix $C(z, z') = (c_{k\ell})_{k,\ell\in[\![K]\!]}$ is given by $c_{k\ell} := \sum_{m\in[\![M]\!]} z_{mk} z'_{m\ell}$ (every $k, \ell \in [\![K]\!]$). Suppose that the labels of the partitions z and z' are such that

$$\mathrm{Tr}(C(z, z')) = \max_{\sigma\in\Sigma_K} \mathrm{Tr}(C(z, (z'_{m\sigma(k)})_{m\in[\![M]\!],k\in[\![K]\!]})),$$

where Σ_K is the set of permutations of the elements of $[\![K]\!]$. Then the proportion

$$\delta(z, z') := 1 - \frac{1}{M} \sum_{m \in [\![M]\!], k \in [\![K]\!]} z_{mk} z'_{mk} \tag{7}$$

is a natural measure of discrepancy between z and z'. As suggested earlier, the measure can be extended to compare pairs of partitions.

Consider now (z, w) and (z', w') two pairs of partitions, z and z' partitioning $[\![M]\!]$ into K components, w and w' partitioning $[\![N]\!]$ into L components. We represent (z, w) and (z', w') with

$$u = (u_{mnk\ell})_{m \in [\![M]\!], n \in [\![N]\!], k \in [\![K]\!], \ell \in [\![L]\!]}$$

and

$$u' = (u'_{mnk\ell})_{m \in [\![M]\!], n \in [\![N]\!], k \in [\![K]\!], \ell \in [\![L]\!]}$$

where $u_{mnk\ell} := z_{mk} \times w_{n\ell}$ and $u'_{mnk\ell} := z'_{mk} \times w'_{n\ell}$ (for every $m \in [\![M]\!], n \in [\![N]\!], k \in [\![K]\!], \ell \in [\![L]\!]$), supposing again that the labels of the partitions z, z' on the one hand and w, w' on the other hand maximize the traces of the confusion matrices $C(z, z')$ and $C(w, w')$ as above (then two pairs of partitions define without ambiguity a co-clustering). By analogy with (7), the proportion

$$\Delta((z, w), (z', w')) := 1 - \frac{1}{KL} \sum_{m \in [\![M]\!], n \in [\![N]\!], k \in [\![K]\!], \ell \in [\![L]\!]} u_{mnk\ell} u'_{mnk\ell} \tag{8}$$

is a measure of discrepancy between (z, w) and (z', w'). It can be shown that

$$\Delta((z, w), (z', w')) = \delta(z, z') + \delta(w, w') - \delta(z, z') \times \delta(w, w'). \tag{9}$$

In the rest of this section we report means and standard deviations, computed across 30 independent replications of each analysis, of the above measure of discrepancy between the derived partition/co-clustering and the true one.

Matching criteria. Set arbitrarily $m \in [\![M]\!]$ and suppose that we have derived the subset $\mathcal{N}_m \subset [\![N]\!]$ that matches x_m to $\{y_n : n \in \mathcal{N}_m\}$. Suppose moreover that in reality x_m is matched to $\{y_n : n \in \mathcal{N}_m^\star\}$ for some $\mathcal{N}_m^\star \subset [\![N]\!]$. We propose to use three real-valued criteria to compare $\mathcal{N}_m$ with $\mathcal{N}_m^\star$.

Let $\mathrm{TP}_m := \mathrm{card}(\mathcal{N}_m \cap \mathcal{N}_m^\star)$, $\mathrm{FP}_m := \mathrm{card}(\mathcal{N}_m \cap (\mathcal{N}_m^\star)^c)$, $\mathrm{TN}_m := \mathrm{card}((\mathcal{N}_m)^c \cap (\mathcal{N}_m^\star)^c)$, $\mathrm{FN}_m := \mathrm{card}((\mathcal{N}_m)^c \cap \mathcal{N}_m^\star)$ be the numbers of true positives, false positives, true negatives and false negatives, respectively. The so called m-specific

- precision: $\mathrm{TP}_m/(\mathrm{TP}_m + \mathrm{FP}_m)$,
- sensitivity: $\mathrm{TP}_m/(\mathrm{TP}_m + \mathrm{FN}_m)$,
- specificity: $\mathrm{TN}_m/(\mathrm{TN}_m + \mathrm{FP}_m)$

quantify how similar are $\mathcal{N}_m$ and $\mathcal{N}_m^\star$, larger values indicating better concordance.

In the rest of this section we report means and standard deviations, computed across 30 independent replications of each analysis, of the average of the m-specific precision, sensitivity and specificity. We also report means and standard deviations, computed across the same 30 independent replications of each analysis, of

$$\tilde{k}_r := \frac{\sum_{m \in [\![M]\!]} \mathrm{card}(\mathcal{N}_m)}{\mathrm{card}(\{m \in [\![M]\!] : \mathcal{N}_m \neq \emptyset\})},$$
$$\tilde{k}_c := \frac{\sum_{n \in [\![N]\!]} \mathrm{card}(\mathcal{M}_n)}{\mathrm{card}(\{n \in [\![N]\!] : \mathcal{M}_n \neq \emptyset\})}$$

the row- and column-specific averages of the cardinalities of the sets $\mathcal{N}_m$ and $\mathcal{M}_n$ that are not empty.

5.4 First simulation study

Simulation scheme. For four different choices of the hyperparameters $M \geq 200, N \geq 200, K \geq 2, d \geq 2$, $\mu_1, \ldots, \mu_K \in \mathbb{R}^d$, $\sigma \in \mathbb{R}_+^*$, $\alpha \in (\mathbb{R}_+)^K$ such that $\sum_{k \in [\![K]\!]} \alpha_k = 1$, we sample indepen-

dently $x_1, \ldots, x_M$ from the mixture of Gaussian laws

$$\sum_{k \in [\![K]\!]} \alpha_k N(\mu_k, \sigma^2 \mathrm{Id}_d) \tag{10}$$

and $y_1, \ldots, y_N$ from

$$\sum_{k \in [\![K]\!]} \alpha_k N(-\mu_k, \sigma^2 \mathrm{Id}_d). \tag{11}$$

One way to sample x from the mixture (10) consists in sampling a latent label u in $[\![K]\!]$ from the multinomial law with parameter $(1; \alpha_1, \ldots, \alpha_K)$ then in sampling x from the Gaussian law $N(\mu_u, \sigma^2 \mathrm{Id}_d)$. Similarly, sampling y from the mixture (11) can be carried out by sampling a latent label v in $[\![K]\!]$ from the multinomial law with parameter $(1; \alpha_1, \ldots, \alpha_K)$ then by sampling y from the Gaussian law $N(-\mu_v, \sigma^2 \mathrm{Id}_d)$. We think of x and y as having a mirrored relationship if $u = v$. In this light, the challenge that we tackle consists in finding such relationships without having access to the latent labels.

Table 1 describes the four configurations that we investigate. Note that configuration A2 is more difficult to deal with than A1 because *(i)* the weights in α are balanced in the latter and unbalanced in the former, and *(ii)* because the variance σ^2 is smaller in A1 than in A2. Moreover, configurations A3 and A4 are more challenging than A2 because there is $K = 4$ components in the Gaussian mixture under A3 and A4 and $K = 3$ components under A2.

configuration	(M, N)	K	$\mu_1, \ldots, \mu_K$	σ^2	α
A1	$(200, 200)$	3	$\begin{pmatrix}4.0\\0.5\\1.5\end{pmatrix}, \begin{pmatrix}1.8\\4.5\\1.1\end{pmatrix}, \begin{pmatrix}1.5\\1.5\\5.5\end{pmatrix}$	0.10	$(1/3, 1/3, 1/3)$
A2	$(300, 300)$	3	$\begin{pmatrix}4.0\\0.5\\1.5\end{pmatrix}, \begin{pmatrix}1.8\\4.5\\5.1\end{pmatrix}, \begin{pmatrix}3.5\\1.5\\5.5\end{pmatrix}$	0.15	$(0.2, 0.3, 0.5)$
A3	$(400, 300)$	4	$\begin{pmatrix}4.0\\0.5\end{pmatrix}, \begin{pmatrix}0.5\\3.5\end{pmatrix}, \begin{pmatrix}7.5\\7.8\end{pmatrix}, \begin{pmatrix}0.5\\0.5\end{pmatrix}$	0.20	$(0.4, 0.2, 0.2, 0.2)$
A4	$(300, 300)$	4	$\begin{pmatrix}4.0\\0.5\end{pmatrix}, \begin{pmatrix}0.5\\3.5\end{pmatrix}, \begin{pmatrix}7.5\\7.8\end{pmatrix}, \begin{pmatrix}0.5\\0.5\end{pmatrix}$	0.10	$(0.5, 0.2, 0.1, 0.2)$

Table 1: Four different configurations for the first simulation scheme. Configuration A1 is less challenging than A2 which is itself less challenging than A3 and A4.

Results. Thirty times, independently, we simulated synthetic data sets X and Y under the simulation scheme described above, then we applied the various algorithms as presented in Section 5.2. We summarize the results in Tables 4, 5, and 6. Table 4 summarizes the results of the seven algorithms listed in Section 5.2 that rely on *bona fide* co-clustering algorithms (see Section 4.1.1), that is, of our algorithms WTOT-SCC1*, WTOT-SCC1, WTOT-SCC2*, WTOT-SCC2, WTOT-BC* and of algorithms CCOT-GWD and CCOT-GWB. As for Tables 5 and 6, they summarize the results of our algorithm that relies on matching (see Section 4.1.2).

Table 4. Except in configuration A1, where they perform equally well, our algorithms WTOT-SCC1, WTOT-SCC2 outperform their competitors CCOT-GWD and CCOT-GWB.

Recall that WTOT-SCC1 and WTOT-SCC2 learn the number of co-clusters. When they underestimate it, they pay a high price, partly explaining why the standard deviations are rather large. In order to assess how well they work relative to their counterparts which benefit from knowing in advance the true number of co-clusters, we can compare their measures of performance to those of algorithms WTOT-SCC1* and WTOT-SCC2*. In configurations A1 and A2, algorithms WTOT-SCC1, WTOT-SCC2 perform almost as well as WTOT-SCC1* and WTOT-SCC2*, respectively. In configuration A3, they are clearly outperformed. In configuration A4, algorithm WTOT-SCC1 performs better in average but not in standard deviation.

Finally, we note that algorithm WTOT-BC* outperforms all our other algorithms. Unfortunately, its counterpart that learns the number of co-clusters performs poorly (results not shown).

Tables 5 and 6. Table 5 illustrates the influence of $k = k'$ on the performances of algorithm WTOT-matching. In configuration A1, specificity is not impacted much by the value of $k = k'$, whereas precision decreases and sensitivity increases as $k = k'$ grows. More specifically, precision does not change much when one goes from $k = k' = 10$ to $k = k' = 75$ but it drops for larger values of $k = k'$. As for sensitivity, it increases dramatically when one

goes from $k = k' = 10$ to $k = k' = 75$ and slightly for higher values of $k = k'$. Furthermore we note that, in configuration A1, when $k = k'$ equal either 65 or 75 and are thus closest to $N\alpha_\ell = M\alpha_\ell \approx 67$, $\tilde{k}_r$ is close to 67 and precision, sensitivity and specificity are quite satisfying. In configuration A4 (as in configuration A1), specificity is not impacted much by the value of $k = k'$; on the contrary, precision decreases and sensitivity increases steadily as $k = k'$ grows. The best performances are achieved for $k = k' = 95$ and $k = k' = 150$, that is, when $k = k'$ get closer to $M \max_{i \leq 4}\{\alpha_i\} = N \max_{i \leq 4}\{\alpha_i\}$. As emphasized earlier, deriving relevant matchings is more difficult in configuration A4 than in configuration A1 because the weights given in parameter α are unbalanced in the former and balanced in the latter.

Table 6 summarizes the results of WTOT-matching in all configurations for a specific choice of $k = k'$ in terms of the row- and column-specific averages $\tilde{k}_r$ and $\tilde{k}_c$, precision, sensitivity and specificity. In each configuration, we chose the value of $k = k'$ among many retrospectively, so that the overall performance (in terms of precision, sensitivity and specificity) is good. The left-hand-side (m-specific) and right-hand-side (n-specific) tables in Table 6 are very similar. This does not come as a surprise because the first simulation scheme imposes symmetry.

5.5 Second simulation study

Simulation scheme. The second simulation scheme also relies on mixtures of Gaussian laws, but the means and weights are generated randomly from a Gaussian determinantal point process (DPP) for the former and from a Dirichlet law for the latter. More specifically, given the hyperparameters $M \geq 200, N \geq 200, K \geq L \geq 3$, $\sigma \in \mathbb{R}_+^*$,

1. we sample $\mu_1, \ldots, \mu_K$ from a Gaussian DPP on $[0,1]^2$ with a kernel proportional to $x \mapsto \exp(-\|x/0.05\|_2^2)$ conditionally on obtaining exactly K points [13, 3];

2. independently, we sample $\alpha \in (\mathbb{R}_+)^K$ and $\beta \in (\mathbb{R}_+)^L$ from the Dirichlet laws with param-

eters $7\,\mathbf{1}_K$ and $7\,\mathbf{1}_L$;

3. we sample independently $x_1, \ldots, x_M$ from the mixture of Gaussian laws

$$\sum_{k \in [\![K]\!]} \alpha_k N(\mu_k, \sigma^2 \mathrm{Id}_2)$$

and $y_1, \ldots, y_N$ from

$$\sum_{k \in [\![L]\!]} \beta_k N(-\mu_k, \sigma^2 \mathrm{Id}_2).$$

We use a DPP to generate $\mu_1, \ldots, \mu_K$ to avoid the arbitrary choice of the mean parameters in such a way that the randomly picked $\mu_1, \ldots, \mu_K$ are dispersed in $[0,1]^2$ (because the DPP is a repulsive point process).

Table 2 describes the four configurations that we investigate. The larger L is the more challenging the configuration is. In configurations B2, B3, B4, it holds that $K = L + 1$, hence the data points from the Kth cluster should not be matched. Moreover, for given (K, L) and (M, N), a configuration gets more challenging as its σ^2 parameter increases. It is noteworthy that the values of σ^2 as reported in Table 2 cannot be compared straightforwardly to those reported in Table 1, because $\mu_1, \ldots, \mu_K$ live in $[0,1]^2$ in the present simulation study whereas they do not in the simulation study of Section 5.4.

configuration	(M, N)	(K, L)	σ^2
B1	$(200, 200)$	$(3, 3)$	5×10^{-4}
B2	$(300, 300)$	$(7, 6)$	10^{-4}
B3	$(300, 300)$	$(16, 15)$	10^{-5}
B4	$(300, 300)$	$(16, 15)$	10^{-4}

Table 2: Four different configurations for the second simulation scheme. The larger $\ell \in [4]$ is the more challenging configuration Bℓ is.

Results. Thirty times, independently, we simulated synthetic data sets X and Y under the simulation scheme described above, then we applied the various algorithms as presented in Section 5.2. Table 7 summarizes the results of the seven algorithms listed in Section 5.2 that

rely on *bona fide* co-clustering algorithms (see Section 4.1.1). Tables 8 and 9 summarize the results of our algorithm that relies on matching (see Section 4.1.2).

Table 7. We first note that WTOT-SCC1, WTOT-SCC2 and CCOT-GWD perform similarly in configurations B1 and B2, much better than CCOT-GWB, but less well than the oracular algorithms WTOT-SCC1*, WTOT-SCC2* and WTOT-BC*. More generally, across configurations B1, B2, B3, B4, the oracular algorithms WTOT-SCC1* and WTOT-SCC2* perform much better than the other algorithms (and WTOT-BC* fails to find a partition with the given number of co-clusters in B3 and B4). Moreover, WTOT-SCC1 and WTOT-SCC2 perform poorly in configurations B2, B3 and B4 though not as poorly as CCOT-GWD and CCOT-GWB in configurations B3 and B4. It seems that WTOT-SCC1 and WTOT-SCC2 fail to learn a "practical" number of co-clusters from $\tilde{P}$, in part because of those among $x_1, \ldots, x_M$ that are drawn from the Gaussian law $N(\mu_K, \sigma^2 \mathrm{Id}_2)$ when $K = L + 1$ (these data points should not be matched at all). The fact that WTOT-SCC1 and WTOT-SCC2 perform similarly in configurations B3 and B4 although σ^2 is 10 times larger in B4 than in B3 gives credit to the previous interpretation.

Tables 8 and 9. Table 8 illustrates the influence of $k = k'$ on the performances of algorithm WTOT-matching in configurations B1 and B4. In each configuration, the values of $k = k'$ are chosen in the vicinity of M/K (67 in configuration B1, 11 in configuration B4). We observe the same patterns in configurations B1 and B4: precision decreases (gradually) and specificity decreases (slightly) as $k = k'$ grows, while sensitivity increases (strongly in B1 and dramatically in B4).

Table 9 summarizes the results of WTOT-matching in configurations B1, B2, B3, B4 for a specific choice of $k = k'$ in terms of the row- and column-specific averages $\tilde{k}_r$ and $\tilde{k}_c$, precision, sensitivity and specificity. In each configuration, we chose the value of $k = k'$ among many retrospectively so that the overall performance (in terms of precision, sensitivity and specificity) is good. The left-hand-side (m-specific) and right-hand-side

(n-specific) tables in Table 9 are very similar although $K > L$ in configuration B3 and B4. Interestingly, the fact that σ^2 is 10 times larger in configuration B4 than in B3 does not affect much the performance of the matching algorithm.

5.6 Third simulation study

Simulation scheme. The third simulation scheme aspires to generate synthetic data sets X and Y that are more similar to the real data sets than those generated in the two first simulation studies. Once again, we rely on mixtures of Gaussian laws. This time, however, the various means are neither chosen arbitrarily (unlike in the first simulation study) nor drawn randomly (unlike in the second simulation study) but are sampled in the real collection of miRNAs. Moreover, the weights of the mixtures are random.

Specifically, given the hyperparameters $K \geq 3$, $\lambda_x, \lambda'_x \geq 0$, $\lambda_y, \lambda'_y \geq 0$ and $\sigma, \sigma' \in \mathbb{R}_+^*$ (with σ' much larger than σ),

1. we sample $\mu_1, \ldots, \mu_K$ uniformly without replacement from the collection of observed miRNA profiles conditionally on $\min_{k \neq k'} \|\mu_k - \mu_{k'}\|_2 \geq 2$;

2. independently, we sample independently $(m_1 - 1), \ldots, (m_K - 1)$ from the Poisson law with parameter λ_x, $(n_1 - 1), \ldots, (n_K - 1)$ from the Poisson law with parameter λ_y, $(m_{K+1} - 1)$ and $(n_{K+1} - 1)$ from the Poisson laws with parameter λ'_x and λ'_y;

3. for each $1 \leq k \leq K$, we sample independently $x_{k,1}, \ldots, x_{k,m_k}$ from the Gaussian law $N(\mu_k, \sigma^2 \mathrm{Id}_{18})$ and $y_{k,1}, \ldots, y_{k,n_k}$ from the Gaussian law $N(-\mu_k, \sigma^2 \mathrm{Id}_{18})$. Moreover, we also sample independently $x_{K+1,1}, \ldots, x_{K+1,m_{K+1}}$ and $y_{K+1,1}, \ldots, y_{K+1,n_{K+1}}$ from the Gaussian law $N(\mathbf{0}_{18}, (\sigma')^2 \mathrm{Id}_{18})$.

Here, we think of x and y as having a mirrored relationship if there exists $k \in [\![K]\!]$ such that x and y are drawn from the laws $N(\mu_k, \sigma^2 \mathrm{Id}_{18})$ and $N(-\mu_k, \sigma^2 \mathrm{Id}_{18})$. Furthermore, we view x and y drawn from the law $N(\mathbf{0}_{18}, (\sigma')^2 \mathrm{Id}_{18})$ as noise.

Table 3 describes the four configurations that we investigate. The larger K is the more challenging the configuration is.

configuration	(λ_x, λ_y)	(λ'_x, λ'_y)	K	(σ, σ')
C1	$(50, 50)$	$(50, 10)$	3	$(0.1, 5)$
C2	$(15, 15)$	$(0, 0)$	15	$(0.01, 5)$
C3	$(15, 15)$	$(30, 30)$	15	$(0.01, 5)$
C4	$(15, 15)$	$(30, 30)$	15	$(0.1, 5)$

Table 3: Four different configurations for the third simulation scheme. The larger $\ell \in [4]$ is the more challenging configuration Cℓ is.

Results. Thirty times, independently, we simulated synthetic data sets X and Y under the simulation scheme described above, then we applied the various algorithms as presented in Section 5.2. Table 10 summarizes the results of the seven algorithms listed in Section 5.2 that rely on *bona fide* co-clustering algorithms (see Section 4.1.1). Tables 11 and 12 summarize the results of our algorithm that relies on matching (see Section 4.1.2).

Table 10. We first focus on configuration C1. We note that WTOT-SCC1 and WTOT-SCC2 perform similarly, much better than CCOT-GWD and CCOT-GWB, better than the oracular algorithm WTOT-BC*, but not as well as the oracular algorithms WTOT-SCC1* and WTOT-SCC2*.

We now turn to configurations C2, C3 and C4. Configuration C3 is more challenging than configuration C2 because it shares the same hyperparameters as C2 except for (λ'_x, λ'_y) (which drives the number of noisy data points), set to $(0, 0)$ in C2 and to $(30, 30)$ in C3. Similarly, configuration C4 is more challenging than configuration C3 because it shares the same hyperparameters as C3 except for σ (the standard deviation of the Gaussian variations around the mean profiles), set to 0.01 in C3 and to 0.1 in C4. The comparisons will not concern algorithms WTOT-BC* (which never converges in these simulations), CCOT-GWD and CCOT-GWB (which perform very poorly).

In configuration C2, in the absence of noisy data points, algorithm WTOT-SCC1 performs

slightly better than WTOT-SCC2, as well as the oracular algorithm WTOT-SCC2*, and almost as well as the oracular algorithm WTOT-SCC1* (in average). In configurations C3 and C4, the introduction of noisy data points then the increase in variability strongly degrade the performances of WTOT-SCC1, WTOT-SCC1* and, to a lesser extent, those of WTOT-SCC2 and WTOT-SCC2*. Algorithm WTOT-SCC2 outperforms WTOT-SCC1 and the oracular algorithm WTOT-SCC1* too.

Tables 11 and 12. Table 11 illustrates the influence of $k = k'$ on the performances of algorithm WTOT-matching in configurations C1 and C4. In each configuration, the values $k = k'$ are chosen in the vicinity of λ_x or λ_y (50 in configuration C1, 15 in configuration C4). For specificity and sensitivity, we observe the same patterns in configurations C1 and C4: specificity is not impacted much as $k = k'$ grows whereas sensitivity increases dramatically. Precision remains high in configuration C1 for all choices of $k = k'$. In configuration C4, precision remains high for $k = k'$ ranging between 5 and 20, then it decreases when $k = k'$ grows from 25 to 30.

Table 12 summarizes the results of WTOT-matching in configurations C1, C2, C3, C4 for a specific choice of $k = k'$ in terms of the row- and column-specific averages $\tilde{k}_r$ and $\tilde{k}_c$, precision, sensitivity and specificity. In each configuration, we chose the value of $k = k'$ among many retrospectively, so that the overall performance (in terms of precision, sensitivity and specificity) is good. The left-hand-side (m-specific) and right-hand-side (n-specific) tables in Table 12 are very similar. In configurations C1 and C2, all precision, sensitivity and specificity are quite satisfying. In configurations C3, C4, sensitivity and specificity are quite satisfying as well while precision falls bellow 0.86.

6 Illustration on real data

Next, we apply algorithms WTOT-SCC2 and WTOT-matching to discover patterns hidden in RNA-seq data obtained in the striatum of HD model mice. As explained in Section 1,

multidimentional miRNA and mRNA sequencing data were obtained in the striatum of these mice [11, 12] and an earlier analysis of these data using shape analysis concepts [15] has demonstrated their value. We briefly illustrate the results we obtain. A separate article (in preparation) will show the complete results and their careful biological interpretation.

6.1 Tuning

Specifically, in view of Algorithm 1, we choose $\widetilde{M} = 1,024$, $\widetilde{N} = 512$, $T = 500$. The entries of the 3×5 matrices $\tilde{\theta}_1^a, \tilde{\theta}_1^b, \tilde{\theta}_1^c$ are constrained to take their values in $]-10, 0[$ (for WTOT-SCC2) or $]-2, 0[$ (for WTOT-matching), $]-0.2, 0.2[$ and $]-0.2, 0.2[$ respectively. We also choose $(\eta, \gamma_0) = (0.95, 3)$. Finally, the initial mapping θ_0 is drawn randomly.

Furthermore, regarding step 2 of algorithm WTOT-SCC2, we remove rows and columns based on the following loop: 100 times successively, *(i)* we compute the Kullback-Leibler divergence between each row (renormalized) and the uniform distribution then remove the 100 rows with the smallest divergences, then *(ii)* we compute the Kullback-Leibler divergence between each column (renormalized) and the uniform distribution then remove the 5 columns with the smallest divergences. By doing so, we successively get rid of the rows and columns which, viewed as distributions, are too uniform and therefore deemed irrelevant. Finally, we remove all rows for which the (columnwise) sum of the remaining entries of $\tilde{P}$ is smaller than one tenth of the maximal (columnwise) sum, and all columns for which the (rowwise) sum of the remaining entries of $\tilde{P}$ is smaller than one tenth of the maximal (rowwise) sum.

6.2 Results

Co-clustering. The selection procedure (step 2 of WTOT-SCC2) keeps 3,409 mRNA profiles (among the 13,616 available in the data set) and 602 miRNA (among the 1,143 available in the data set). Eventually, algorithm WTOT-SCC2 outputs 8 co-clusters. The co-clusters's sizes (numbers of mRNA and miRNA gathered in each co-cluster) are $(321, 86)$, $(333, 30)$, $(261, 6)$, $(498, 125)$, $(127, 5)$, $(708, 203)$, $(703, 119)$, $(458, 28)$. Figure 2 represents the averages, computed

across all blocks, of the entries of the matrix derived from the optimal transport matrix $\tilde{P}$ during step 2 of algorithm WTOT-SCC2 and after its rearrangement. Squares located on the diagonal tend to be slightly darker than the other squares. This reveals that, in average, a pair (x_m, y_n) of mRNA and miRNA gathered in a diagonal co-cluster tends to exhibit a mirrored relationship that is slightly stronger than those of the form $(x_m, y_{n'})$ or $(x_{m'}, y_n)$ which do not fall in the same co-cluster. However, few of the off-diagonal averages are small in comparison to the on-diagonal averages, a disappointing observation that comes on top of the fact that the co-clusters' sizes are so large that it is difficult to interpret the results. This makes it even more relevant to focus on algorithm WTOT-matching.

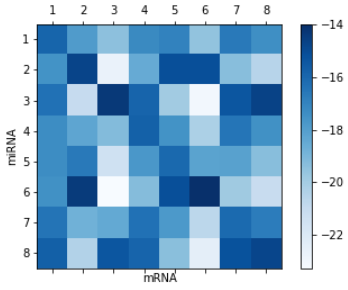

Figure 2: Logarithms of the averages, computed across all blocks, of the entries of the matrix derived from the optimal transport matrix $\tilde{P}$ during step 2 of algorithm WTOT-SCC2 and after its rearrangement.

Matching. We run the WTOT-matching algorithm with $k = k' = 10$ and $q = 90\%$. For the anecdote, we observe $(\tilde{k}_r, \tilde{k}_c) \approx (1.82, 6.04)$ (recall that $\tilde{k}_r, \tilde{k}_c$ are the row- and column-specific averages of the cardinalities of the sets $\mathcal{N}_m$ and $\mathcal{M}_n$ that are not empty). We report the parameters that characterize the mapping $\widehat{\theta}$ in Appendix A.

As an illustration, the mirrored profile (the opposite value of y_n) of the Mir20b miRNA is displayed in Figure 3 along with its three matched mRNAs (Ahrr, Cnih3 and Relb) obtained by running algorithm WTOT-matching algorithm with $k = k' = 10$. Recall that the original profile of Mir20b can be found in Figure 1.

To illustrate the matchings that we obtain, we propose to proceed as follows. We first

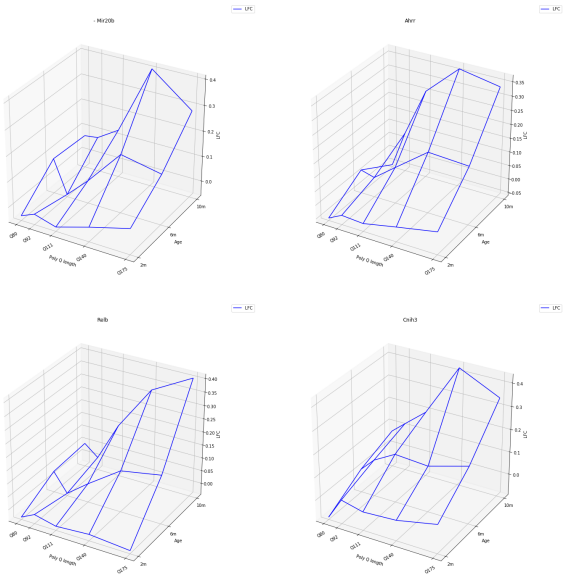

Figure 3: Minus the profile $-y_n$ of the Mir20b miRNA (top left), and profiles x_m of its matched mRNAs, Ahrr (top right), Relb (bottom left) and Cnih3 (bottom right).

identify miRNAs that are particularly susceptible to play a distinct role in HD in mice. To do so, we evaluate two simple criteria on the mRNAs associated to each miRNA (the miRNAs with no matched mRNAs are obviously less interesting in our study). The criteria assess to what extent a mRNA profile is "monotonous" and, on the contrary, to what extent it is "peaked", accounting for the amplitude of log-fold change. Formally, rewriting each profile $x \in \mathbb{R}^{15}$ as a matrix $(\tilde{x}_{tq})_{t\in[\![3]\!],q\in[\![5]\!]}$, the first criterion is the minimum (relative to time t) of the absolute values of the slopes of the regression lines of the sets $\{(q, \tilde{x}_{tq}) : q \in [\![5]\!]\}$ and the second criterion is $\max_{q\in[\![5]\!]}(\tilde{x}_{1q} - \tilde{x}_{2q}) \times (\tilde{x}_{2q} - \tilde{x}_{3q})$. We retain only the miRNAs for which at least one of its associated mRNAs is such that either its first criterion is larger than 95% of the similar criteria or its second criterion is smaller than 99% of the similar criteria. Second, we perform an enrichment analysis of the mRNAs associated to the miRNAs that are retained at the previous stage.

The first criterion yields 212 mRNAs and 122 matched miRNAs. The second criterion yields 43 mRNAs and 68 matched miRNAs. We retrieve the Ensembl gene stable identifiers of the mRNAs then request an overrepresentation analysis from reactome (converting the identifiers to their human equivalents; only 97 of the 255 identifiers have a human equivalent in the reactome data base). Four pathways obtain a False Discovery Rate (FDR) smaller than 5%. The pathways are labelled "neuronal system" (FDR 2.56%), "cardiac conduction" (FDR 2.67%), "presynaptic depolarization and calcium channel opening" (FDR 2.67%) and "muscle contraction" (FDR 3.34%). A comprehensive biological content analysis of the resulting miRNA-mRNA networks will be reported in a separate article (in preparation).

7 Discussion

We have developped two co-clustering algorithms (WTOT-SCC1 and WTOT-SCC2) and a matching algorithm (WTOT-matching) for the purpose of identifying groups of mRNAs and miRNAs that interact. The algorithms apply in any situation where it is of interest to cluster or match the elements of two data sets based on a parametric model Θ expressing what it means to interact for any two pair of elements from the two data sets. The algorithms rely on optimal transport, spectral co-clustering and a matching procedure. In light of [2, Section 1.3, page 25], problem-specific knowledge is injected onto two of the three main components of the transportation problem: the representation spaces (via Θ) and the marginal constraints, leaving aside the cost function.

During the first stage, an optimal optimal transport plan P and mapping in Θ are learned from the data using the Sinkhorn-Knopp algorithm and a mini-batch gradient descent. During the second stage, P is exploited to derive either co-clusters or several sets of matched elements.

As in [15], the motivation of our study is to shed light on the interaction between mRNAs and miRNAs based on data collected in the striatum of HD model knock-in mice [11, 12]. Each data point takes the form of multi-dimensional profile. The biological hypothesis is that if a miRNA induces the degradation of a target mRNA or blocks its translation into proteins, or both, then

the profile of the former should be similar to minus the profile of the latter — this particular form of affine relationship drives the definition of model Θ. The fact that the algorithm learns from the data a best element in Θ provides more flexibility than in [15].

The simulation study reveals on the one hand that WTOT-SCC2 works overall better than WTOT-SCC1, but that the co-clustering task can be very challenging in the presence of many irrelevant data points (data points that do not interact). On the other hand, it shows that the performances of WTOT-matching are satisfying. A brief illustration on real data is given. The complete data analysis will be presented in a separate article (in preparation).

References

[1] M. Ailem, F. Role, and M. Nadif. Graph modularity maximization as an effective method for co-clustering text data. *Knowledge-Based Systems*, 109:160–173, 2016.

[2] D. Alvarez-Melis. *Optimal Transport in Structured Domains: Algorithms and Applications*. PhD thesis, Massachusetts Institute of Technology, 2019.

[3] A. Baddeley and R. Turner. spatstat: An R package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6):1–42, 2005.

[4] B. Benayoun, E. Pollina, P. Singh, S. Mahmoudi, I. Harel, K. Casey, B. Dulken, A. Kundaje, and A. Brunet. Remodeling of epigenome and transcriptome landscapes with aging in mice reveals widespread induction of inflammatory responses. *Genome Research*, 29(4):697–709, 2019.

[5] V. Brault, C. Keribin, G. Celeux, and G. Govaert. Estimation and selection for the latent block model on categorical data. *Statistics and Computing*, 25:1201–1216, 2014.

[6] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge*

Discovery and Data Mining, KDD '01, page 269–274, New York, NY, USA, 2001. Association for Computing Machinery.

[7] K. Fatras, Y. Zine, R. Flamary, R. Gribonval, and N. Courty. Learning with minibatch Wasserstein : asymptotic and gradient properties. In *The 23nd International Conference on Artificial Intelligence and Statistics*, volume volume 108 of *PMLR*, Palermo, Italy, 2020.

[8] A. Genevay, G. Peyré, and M. Cuturi. Learning generative models with sinkhorn divergences. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1608–1617, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[9] G. Govaert and M. Nadif. Model-based co-clustering for continuous data. In *Machine Learning and Applications, Fourth International Conference on*, pages 175–180, Los Alamitos, CA, USA, dec 2010. IEEE Computer Society.

[10] C. Laclau, I. Redko, B. Matei, Y. Bennani, and V. Brault. Co-clustering through Optimal Transport. In *34th International Conference on Machine Learning*, volume 70, pages 1955–1964, Sydney, Australia, August 2017.

[11] P. Langfelder, J. Cantle, D. Chatzopoulou, N. Wang, F. Gao, I. Al-Ramahi, X. Lu, E. Ramos, K. Merz, Y. Zhao, S. Deverasetty, A. Tebbe, C. Schaab, D. Lavery, D. Howland, S. Kwak, J. Botas, J. Aaronson, J. Rosinski, and X. Yang. Integrated genomics and proteomics define Huntingtin CAGlength–dependent networks in mice. *Nature Neuroscience*, 19, 02 2016.

[12] P. Langfelder, F. Gao, N. Wang, D. Howland, S. Kwak, T. Vogt, J. Aaronson, J. Rosinski, G. Coppola, S. Horvath, and X. Yang. MicroRNA signatures of endogenous Huntingtin CAG repeat expansion in mice. *PloS one*, 13(1), 2018.

[13] F. Lavancier, J. Møller, and E. Rubak. Determinantal point process models and statistical inference. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 77(4):853–877, 2015.

[14] S. Maniatis, T. Äijö, S. Vickovic, C. Braine, K. Kang, A. Mollbrink, D. Fagegaltier, Ž. Andrusivová, S. Saarenpää, G. Saiz-Castro, M. Cuevas, A. Watters, J. Lundeberg, R. Bonneau, and H. Phatnani. Spatiotemporal dynamics of molecular pathology in amyotrophic lateral sclerosis. *Science*, 364(6435):89–93, 2019.

[15] L. Mégret, S. Sasidharan Nair, J. Dancourt, J. Aaronson, J. Rosinski, and C. Neri. Combining feature selection and shape analysis uncovers precise rules for miRNA regulation in Huntington's disease mice. *BMC Bioinformatics*, 21(1):75, 2020.

[16] G. Peyre and M. Cuturi. *Computational Optimal Transport: With Applications to Data Science*. Foundations and Trends in Machine Learning Series. Now Publishers, 2019.

[17] G. Peyré, M. Cuturi, and J. Solomon. Gromov-Wasserstein Averaging of Kernel and Distance Matrices. In *ICML 2016*, Proc. 33rd International Conference on Machine Learning, New-York, United States, June 2016.

[18] B. Pontes, R. Giráldez, and J. S. Aguilar-Ruiz. Biclustering on expression data: A review. *Journal of Biomedical Informatics*, 57:163–180, 2015.

[19] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2021. doi: 10.1109/TRO.2020.3033695.

[20] J. Zhao, H. Wang, L. Dong, S. Sun, and L. Li. miRNA-20b inhibits cerebral ischemia-induced inflammation through targeting NLRP3. *Int J Mol Med*, 43(3):1167–1178, 2019.

A Supplementary material

Parametric model Θ. Introduced in Section 4.1, the parametric model Θ consists of affine mappings $\theta : \mathbb{R}^d \to \mathbb{R}^d$ of the form $x \mapsto \theta_1 x + \theta_2$, where θ_1 takes its values in a subset T_1 of $\mathbb{R}^{d\times d}$ and θ_2 takes its values in $\mathbb{R}^d$ (without any constraint). It is easier to describe the set of linear mappings $\{x \mapsto \theta_1 x : \theta_1 \in T_1\}$ after a reparametrization.

In the rest of this section only, we rewrite the mRNA and miRNA profiles $x, y \in \mathbb{R}^d$ under the form of $d_1 \times d_2$ matrices $\tilde{x} = (\tilde{x}_{tq})_{t\in[\![d_1]\!],q\in[\![d_2]\!]}$ and $\tilde{y} = (\tilde{y}_{tq})_{t\in[\![d_1]\!],q\in[\![d_2]\!]}$. For each $t \in [\![d_1]\!]$ and $q \in [\![d_2]\!]$, $\tilde{x}_{t\bullet}$ and $\tilde{x}_{\bullet q}$ are the tth row and qth column of $\tilde{x}$. Here, indices t and q correspond to the age and CAG lengths of the mice whose RNA sequencing yielded $\tilde{x}_{tq}$ and $\tilde{y}_{tq}$.

The definition of T_1 should formalize what we consider to be a (plausible) mirroring relationship. The simplest mirroring relationship is $y = -x$ or, equivalently, $\tilde{y} = -\tilde{x}$. The equality is of course too stringent/rigid, and the definition of T_1 is driven by our wish to relax it.

Biological arguments encourage us to consider that y and x exhibit a (plausible) mirroring relationship if, for each (t, q) ($t \in [\![d_1]\!]$, $q \in [\![d_2]\!]$), $\tilde{y}_{tq}$ is strongly negatively correlated with $\tilde{x}_{tq}$, mainly, and (positively or negatively) correlated with $\tilde{x}_{(t-1)q}$ (if $t > 1$) and/or with $\tilde{x}_{t(q-1)}$ (if $q > 1$), secondarily. We thus formalize $\{x \mapsto \theta_1 x : \theta_1 \in T_1\}$ as the set of all linear mappings of the form

$$x \mapsto \tilde{\theta}_1^a \odot \tilde{x} + \tilde{\theta}_1^b \odot \begin{pmatrix} \mathbf{0}_{d_2}^\top \\ \tilde{x}_{1\bullet} \\ \vdots \\ \tilde{x}_{(d_1-1)\bullet} \end{pmatrix} + \tilde{\theta}_1^c \odot \left(\mathbf{0}_{d_1} \tilde{x}_{\bullet 1} \ \cdots \tilde{x}_{\bullet(d_2-1)}\right)$$

where $\tilde{\theta}_1^a$ and $\tilde{\theta}_1^b, \tilde{\theta}_1^c$ are $d_1 \times d_2$ matrices (here, $\odot$ is the componentwise multiplication). In the simulation study presented in Section 5, the entries of $\tilde{\theta}_1^a$ are constrained to take their values in the interval $]-5, 0[$ while those of $\tilde{\theta}_1^b, \tilde{\theta}_1^c$ are constrained to take their values in $]-1/2, 1/2[$.

In the illustration of the WTOT-matching algorithm presented in Section 6.2, the mapping

$\widehat{\theta}$ is parametrized by

$$\tilde{\theta}_1^a = \begin{pmatrix} -0.88 & -1.47 & -0.73 \\ -0.59 & -0.90 & -0.89 \\ -0.62 & -0.70 & -1.17 \\ -0.97 & -1.30 & -0.95 \\ -0.56 & -1.16 & -1.24 \end{pmatrix}, \quad \tilde{\theta}_1^b = \begin{pmatrix} 0.00 & 0.00 & 0.00 \\ 0.13 & -0.19 & 0.13 \\ 0.17 & 0.09 & 0.13 \\ 0.19 & 0.09 & -0.00 \\ 0.18 & 0.15 & 0.08 \end{pmatrix},$$

$$\tilde{\theta}_1^c = \begin{pmatrix} 0.00 & 0.18 & -0.18 \\ 0.00 & 0.19 & 0.17 \\ 0.00 & 0.04 & 0.15 \\ 0.00 & 0.05 & 0.11 \\ 0.00 & 0.18 & 0.14 \end{pmatrix}, \quad \theta_2 = \begin{pmatrix} -0.01 & 0.01 & -0.00 \\ 0.00 & 0.01 & 0.01 \\ 0.00 & 0.01 & 0.00 \\ 0.01 & 0.01 & 0.01 \\ -0.01 & 0.01 & 0.01 \end{pmatrix}$$

(the numbers are rounded to two decimal places).

Procedure 1 *Main optimal transport algorithm.*

Input: X, Y, minibatch sizes $\widetilde{M}, \widetilde{N}$, decay rate $\eta \in]0,1]$, initial regularization parameter γ_0, initial mapping $\theta_0 \in \Theta$, maximal number of iterations T

Output: Transport coupling $\tilde{P}_T \in (\mathbb{R}_+)^{M \times N}$, mapping $\theta_T \in \Theta$, weight ω_T

Compute:

- $\underline{\gamma} = \text{mean}\{\|x - x'\|_2 : x, x' \in X\}$ {for entropy regularization}
- $h = \text{mean}\{\|y - y'\|_2 : y, y' \in Y\}$ {for window calibration}

Set $t \leftarrow 0$

Set stop $\leftarrow$ FALSE

while $\neg$ stop or $t < T$ **do**

$\gamma_t \leftarrow \max(\gamma_0 \times \eta^t, \underline{\gamma})$

Sample uniformly a minibatch of $\widetilde{M}$ observations $\tilde{x}_{1:\widetilde{M}} := (\tilde{x}_1, \ldots, \tilde{x}_{\widetilde{M}})$ from X

Sample uniformly a minibatch of $\widetilde{N}$ observations $\tilde{y}_{1:\widetilde{N}} := (\tilde{y}_1, \ldots, \tilde{y}_{\widetilde{N}})$ from Y

Define and compute $\theta_t(\tilde{x}_{1:\widetilde{M}}) := \big(\theta_t(\tilde{x}_1), \ldots, \theta_t(\tilde{x}_{\widetilde{M}})\big)$

Define and compute $\omega_t \in (\mathbb{R}_+)^{\widetilde{M}}$ such that $\sum_{m \in [\![\widetilde{M}]\!]} (\omega_t)_m = 1$ by setting

$$(\omega_t)_m \propto \sum_{n \in [\![\widetilde{N}]\!]} \varphi\left(\frac{\tilde{y}_n - \theta_t(\tilde{x}_m)}{h}\right) \quad (\text{all } m \in [\![\widetilde{M}]\!])$$

where φ is the standard normal density

Define $\mu^{\omega_t}_{\theta_t(\tilde{x}_{1:\widetilde{M}})}$, the ω_t-weighted empirical measure attached to $\theta_t(\tilde{x}_{1:\widetilde{M}})$, and $\nu_{\tilde{y}_{1:\widetilde{N}}}$, the empirical measure attached to $\tilde{y}_{1:\widetilde{N}}$

Compute $\text{Loss}_t = \bar{\mathcal{W}}_{\gamma_t}\left(\mu^{\omega_t}_{\theta_t(\tilde{x}_{1:\widetilde{M}})}, \nu_{\tilde{y}_{1:\widetilde{N}}}\right)$ and ∇Loss_t, the gradient of Loss_t relative to the parameter defining θ_t {relies on the Sinkhorn-Knopp algorithm}

Update the parameter defining θ_t by performing one step of stochastic gradient descent, yielding θ_{t+1}

Check stopping criterion and update stop variable accordingly

$t \leftarrow t + 1$

end while

Set $\theta_T \leftarrow \theta_{t-1}$

Set $\gamma_T \leftarrow \gamma_{t-1}$

Define and compute $\omega_T \in (\mathbb{R}_+)^M$ such that $\sum_{m \in [\![M]\!]} (\omega_T)_m = 1$ by setting

$$(\omega_T)_m \propto \sum_{n \in [\![N]\!]} \varphi\left(\frac{y_n - \theta_T(x_m)}{h}\right) \quad (\text{all } m \in [\![M]\!])$$

Compute $\tilde{P}_T \in \Pi(\omega_T)$ solving $\min_{P \in \Pi(\omega_T)} \mathcal{W}_{\gamma_T}\left(\mu^{\omega_T}_{\theta_T(X)}, \nu_Y\right)$

	the WTOT(...) algorithms					the CCOT(...) algorithms	
	WTOT-SCC1*	WTOT-SCC1	WTOT-SCC2*	WTOT-SCC2	WTOT-BC*	CCOT-GWD	CCOT-GWB
A1	0	0.068 ± 0.126	0	0.068 ± 0.126	0	0.054 ± 0.14	0.092 ± 0.15
A2	0 ± 0.001	0.014 ± 0.029	0 ± 0.001	0.016 ± 0.035	0.033 ± 0.125	0.105 ± 0.13	0.121 ± 0.146
A3	0.005 ± 0.005	0.189 ± 0.175	0.0182 ± 0.033	0.233 ± 0.179	0.029 ± 0.087	0.612 ± 0.03	0.532 ± 0.068
A4	0.326 ± 0.064	0.282 ± 0.232	0.257 ± 0.256	0.393 ± 0.164	0.05 ± 0.093	0.507 ± 0.123	0.522 ± 0.116

Table 4: Mean ($\pm$ standard deviation) computed across the 30 independent replications of the co-clustering discrepancy obtained for configurations A1, A2, A3, A4.

	$k = k'$	$\tilde{k}_r$	precision	sensitivity	specificity
A1	10	7.825 ± 0.091	1.0 ± 0.0	0.118 ± 0.001	1.0 ± 0.0
A1	35	29.373 ± 0.261	1.0 ± 0.0	0.442 ± 0.003	1.0 ± 0.0
A1	65	60.649 ± 0.998	0.999 ± 0.002	0.913 ± 0.014	1.0 ± 0.0
A1	75	67.418 ± 0.9	0.981 ± 0.006	0.991 ± 0.013	0.994 ± 0.002
A1	95	76.335 ± 1.282	0.888 ± 0.014	1.0 ± 0.0	0.957 ± 0.005
A1	150	97.049 ± 1.182	0.727 ± 0.012	1.0 ± 0.0	0.879 ± 0.005

	$k = k'$	$\tilde{k}_r$	precision	sensitivity	specificity
A4	10	6.964 ± 0.161	0.998 ± 0.003	0.089 ± 0.003	1.0 ± 0.0
A4	35	28.632 ± 0.668	0.995 ± 0.009	0.374 ± 0.01	1.0 ± 0.0
A4	65	54.653 ± 0.927	0.986 ± 0.011	0.668 ± 0.018	0.998 ± 0.002
A4	75	61.193 ± 0.724	0.963 ± 0.016	0.709 ± 0.022	0.993 ± 0.003
A4	95	75.856 ± 0.749	0.893 ± 0.017	0.768 ± 0.022	0.975 ± 0.003
A4	150	121.273 ± 3.63	0.783 ± 0.025	0.976 ± 0.023	0.936 ± 0.011

Table 5: Mean ($\pm$ standard deviation) computed across the 30 independent replications of $\tilde{k}_r$, precision, sensitivity and specificity of the m-specific matchings averaged across all mRNAs for configuration A1 (left) and A4 (right).

	$k = k'$	$\tilde{k}_r$	precision	sensitivity	specificity
A1	75	67.418 ± 0.9	0.981 ± 0.006	0.991 ± 0.013	0.994 ± 0.002
A2	130	100.217 ± 2.127	0.976 ± 0.017	0.894 ± 0.027	0.995 ± 0.004
A3	120	82.764 ± 1.105	0.881 ± 0.015	0.902 ± 0.025	0.968 ± 0.004
A4	120	97.561 ± 1.836	0.821 ± 0.015	0.853 ± 0.025	0.95 ± 0.005

	$k = k'$	$\tilde{k}_c$	precision	sensitivity	specificity
A1	75	67.418 ± 0.9	0.982 ± 0.006	0.991 ± 0.015	0.994 ± 0.002
A2	130	100.217 ± 2.127	0.984 ± 0.012	0.894 ± 0.028	0.995 ± 0.004
A3	120	110.352 ± 1.473	0.878 ± 0.017	0.9 ± 0.024	0.967 ± 0.004
A4	120	97.561 ± 1.836	0.84 ± 0.018	0.853 ± 0.026	0.951 ± 0.004

Table 6: Mean ($\pm$ standard deviation) computed across the 30 independent replications of $\tilde{k}_r$ or $\tilde{k}_c$, precision, sensitivity and specificity of the m-specific matchings (left) and n-specific matchings (right) averaged across all mRNAs (left) and all miRNAs (right).

	the WTOT(...) algorithms					the CCOT(...) algorithms	
	WTOT-SCC1*	WTOT-SCC1	WTOT-SCC2*	WTOT-SCC2	WTOT-BC*	CCOT-GWD	CCOT-GWB
B1	0.062 ± 0.151	0.204 ± 0.221	0.082 ± 0.161	0.204 ± 0.221	0.049 ± 0.125	0.276 ± 0.204	0.53 ± 0.168
B2	0.114 ± 0.108	0.418 ± 0.265	0.178 ± 0.207	0.455 ± 0.258	0.382 ± 0.121	0.477 ± 0.14	0.523 ± 0.115
B3	0.175 ± 0.086	0.724 ± 0.236	0.163 ± 0.082	0.775 ± 0.176	–	0.858 ± 0.042	0.867 ± 0.044
B4	0.174 ± 0.092	0.747 ± 0.196	0.171 ± 0.112	0.782 ± 0.159	–	0.882 ± 0.041	0.883 ± 0.04

Table 7: Mean ($\pm$ standard deviation) computed across the 30 independent replications of the co-clustering discrepancy obtained for configurations B1, B2, B3, B4.

	$k = k'$	$\tilde{k}_r$	precision	sensitivity	specificity
B1	60	48.578 ± 5.201	0.885 ± 0.209	0.658 ± 0.191	0.985 ± 0.025
B1	80	63.96 ± 6.126	0.851 ± 0.199	0.816 ± 0.222	0.968 ± 0.03
B1	85	67.537 ± 6.193	0.837 ± 0.193	0.842 ± 0.222	0.961 ± 0.031
B1	90	71.214 ± 6.208	0.823 ± 0.186	0.864 ± 0.219	0.953 ± 0.031
B1	110	85.833 ± 6.358	0.753 ± 0.156	0.918 ± 0.202	0.913 ± 0.029

	$k = k'$	$\tilde{k}_r$	precision	sensitivity	specificity
B4	10	6.78 ± 0.259	0.926 ± 0.102	0.321 ± 0.046	0.999 ± 0.001
B4	20	15.163 ± 0.619	0.873 ± 0.091	0.72 ± 0.087	0.996 ± 0.003
B4	25	19.033 ± 0.784	0.817 ± 0.084	0.837 ± 0.084	0.991 ± 0.004
B4	30	22.889 ± 0.997	0.754 ± 0.076	0.907 ± 0.077	0.984 ± 0.005
B4	40	31.118 ± 1.086	0.618 ± 0.053	0.969 ± 0.049	0.963 ± 0.005

Table 8: Mean ($\pm$ standard deviation) computed across the 30 independent replications of $\tilde{k}_r$, precision, sensitivity and specificity of the m-specific matchings averaged across all mRNAs for configurations B1 (left) and B4 (right).

	$k = k'$	$\tilde{k}_r$	precision	sensitivity	specificity
B1	85	67.537 ± 6.193	0.837 ± 0.193	0.842 ± 0.222	0.961 ± 0.031
B2	60	48.282 ± 3.449	0.751 ± 0.194	0.838 ± 0.2	0.979 ± 0.022
B3	25	19.546 ± 1.151	0.833 ± 0.136	0.837 ± 0.152	0.992 ± 0.006
B4	25	19.033 ± 0.784	0.817 ± 0.084	0.837 ± 0.084	0.991 ± 0.004

	$k = k'$	$\tilde{k}_c$	precision	sensitivity	specificity
B1	85	63.732 ± 8.642	0.844 ± 0.175	0.836 ± 0.229	0.96 ± 0.033
B2	60	44.349 ± 2.495	0.792 ± 0.218	0.819 ± 0.227	0.971 ± 0.024
B3	25	18.766 ± 0.97	0.847 ± 0.125	0.833 ± 0.152	0.991 ± 0.005
B4	25	18.833 ± 0.793	0.834 ± 0.087	0.827 ± 0.099	0.99 ± 0.005

Table 9: Mean ($\pm$ standard deviation) computed across the 30 independent replications of $\tilde{k}_r$ or $\tilde{k}_c$, precision, sensitivity and specificity of the m-specific matchings (left) and n-specific matchings (right) averaged across all mRNAs (left) and all miRNAs (right).

	the WTOT(...) algorithms					the CCOT-(...) algorithms	
	WTOT-SCC1*	WTOT-SCC1	WTOT-SCC2*	WTOT-SCC2	WTOT-BC*	CCOT-GWD	CCOT-GWB
C1	0.106 ± 0.1	0.203 ± 0.135	0.101 ± 0.056	0.194 ± 0.116	0.265 ± 0.255	0.496 ± 0.16	0.902 ± 0.007
C2	0.209 ± 0.131	0.252 ± 0.182	0.262 ± 0.141	0.345 ± 0.205	–	0.938 ± 0.023	0.971 ± 0.026
C3	0.609 ± 0.113	0.693 ± 0.154	0.385 ± 0.151	0.521 ± 0.198	–	0.926 ± 0.027	0.987 ± 0.002
C4	0.63 ± 0.141	0.751 ± 0.145	0.435 ± 0.197	0.6 ± 0.233	–	0.939 ± 0.027	0.987 ± 0.002

Table 10: Mean ($\pm$ standard deviation) computed across the 30 independent replications of the co-clustering discrepancy obtained for configurations C1, C2, C3, C4.

	$k = k'$	$\tilde{k}_r$	precision	sensitivity	specificity
C1	10	7.748 ± 0.446	0.973 ± 0.03	0.156 ± 0.01	1.0 ± 0.0
C1	30	25.888 ± 1.418	0.972 ± 0.029	0.526 ± 0.032	1.0 ± 0.0
C1	50	45.521 ± 2.441	0.944 ± 0.025	0.916 ± 0.04	1.0 ± 0.001
C1	55	49.108 ± 3.018	0.93 ± 0.021	0.972 ± 0.025	0.999 ± 0.002
C1	60	51.365 ± 3.335	0.919 ± 0.024	0.993 ± 0.011	0.997 ± 0.004
C1	70	55.296 ± 3.312	0.881 ± 0.034	1.0 ± 0.0	0.985 ± 0.01

	$k = k'$	$\tilde{k}_r$	precision	sensitivity	specificity
C4	5	3.293 ± 0.096	0.895 ± 0.023	0.185 ± 0.012	1.0 ± 0.0
C4	10	7.278 ± 0.303	0.899 ± 0.022	0.474 ± 0.029	1.0 ± 0.0
C4	15	11.982 ± 0.578	0.888 ± 0.02	0.787 ± 0.04	1.0 ± 0.0
C4	20	15.935 ± 0.864	0.843 ± 0.022	0.96 ± 0.023	0.997 ± 0.001
C4	25	19.138 ± 0.89	0.762 ± 0.032	0.997 ± 0.005	0.989 ± 0.003
C4	30	22.578 ± 1.11	0.671 ± 0.04	1.0 ± 0.0	0.978 ± 0.004

Table 11: Mean ($\pm$ standard deviation) computed across the 30 independent replications of $\tilde{k}_r$, precision, sensitivity and specificity of the m-specific matchings averaged across all mRNAs for configurations C1 (left) and C4 (right).

	$k = k'$	$\tilde{k}_r$	precision	sensitivity	specificity
C1	55	49.108 ± 3.018	0.93 ± 0.021	0.972 ± 0.025	0.999 ± 0.002
C2	20	16.203 ± 0.956	0.955 ± 0.016	0.965 ± 0.021	0.997 ± 0.001
C3	20	15.552 ± 0.877	0.854 ± 0.024	0.968 ± 0.019	0.997 ± 0.001
C4	20	15.935 ± 0.864	0.843 ± 0.022	0.96 ± 0.023	0.997 ± 0.001

	$k = k'$	$\tilde{k}_c$	precision	sensitivity	specificity
C1	55	49.056 ± 3.461	0.898 ± 0.06	0.971 ± 0.026	0.981 ± 0.009
C2	20	16.371 ± 0.812	0.953 ± 0.018	0.963 ± 0.023	0.997 ± 0.001
C3	20	15.879 ± 0.691	0.804 ± 0.025	0.969 ± 0.018	0.993 ± 0.001
C4	20	15.867 ± 0.635	0.812 ± 0.032	0.961 ± 0.021	0.993 ± 0.002

Table 12: Mean ($\pm$ standard deviation) computed across the 30 independent replications of $\tilde{k}_r$ or $\tilde{k}_c$, precision, sensitivity and specificity of the m-specific matchings (left) and n-specific matchings (right) averaged across all mRNAs (left) and all miRNAs (right).