A. Proofs

A.1. Proof of Proposition 2.1

We denote by $P^{\varepsilon}_{\ell_+}$ the matrix $\mathrm{diag}(\mathbf{u}_\ell)K\mathrm{diag}(\mathbf{v}_\ell)$ and $P^{\varepsilon}_{\ell_-} = \mathrm{diag}(\mathbf{u}_{\ell-1})K\mathrm{diag}(\mathbf{v}_\ell)$. Recall that $P^{\varepsilon}_{\ell_+}\mathbf{1}_m = \mathbf{a}$ whereas $(P^{\varepsilon}_{\ell_-})^T\mathbf{1}_n = \mathbf{b}$ For convenience, we can assume that the array $\mathbf{x}$ is sorted in non-decreasing order and that the entries of $\mathbf{x}$ are distinct. The first assumption is without loss of generality, since applying a permutation to the entries of $\mathbf{x}$ and $\mathbf{a}$ has the effect of applying the same permutation to the vectors $\widetilde{R}_\varepsilon$ and $\widetilde{T}_\varepsilon$. The latter assumption can be accomplished by infinitesimally perturbing the entries of $\mathbf{x}$ and using the fact that $\widetilde{R}_\varepsilon$, $\widetilde{S}_\varepsilon$, and $\widetilde{T}_\varepsilon$ are all continuous functions of $\mathbf{x}$.

Under these assumptions, it suffices to prove that the vectors $\widetilde{S}_\varepsilon(\mathbf{a},\mathbf{x};\mathbf{b},\mathbf{y})$, $\widetilde{R}_\varepsilon(\mathbf{a},\mathbf{x};\mathbf{b},\mathbf{y})$, and $\widetilde{T}_{\varepsilon,\mathbf{b},\mathbf{q}}(\mathbf{a},\mathbf{x};\mathbf{y})$ are non-decreasing. These three claims follow from the following monotonicity property of $P^{\varepsilon}_{\ell_+}$ and $P^{\varepsilon}_{\ell_-}$.

Lemma A.1. *For any $0 \leqslant k \leqslant m$, the sum of the last k columns of $\mathrm{diag}(\mathbf{a})^{-1}P^{\varepsilon}_{\ell_+}$ is a vector whose entries are non-decreasing. Similarly, for any $0 \leqslant k \leqslant n$, the sum of the last k rows of $\mathrm{diag}(\mathbf{b})^{-1}P^{\varepsilon}_{\ell_-}$ is a vector whose entries are non-decreasing.*

Let us first see how this implies the proposition. Let M be any matrix each of whose rows sums to 1 and such that the sum of its last k columns is a non-decreasing vector. Under these conditions, if $\mathbf{w}$ is a non-decreasing vector, then $M\mathbf{w}$ is non-decreasing. Indeed, if we denote by M_j the jth column of M, we can write

$$\begin{aligned} M\mathbf{w} &= \sum_j M_j w_j = \sum_j M_j \left(w_1 + \sum_{1\leqslant J<j} w_{J+1} - w_J \right) \\ &= w_1 \sum_j M_j + \sum_J (w_{J+1} - w_J) \sum_{j>J} M_j \,. \end{aligned}$$

By assumption, $\sum_j M_j = \mathbf{1}$, the all-ones vector, and $\sum_{j>J} M_j$ is a non-decreasing vector. Since $\mathbf{w}$ is non-decreasing, $w_{J+1} - w_J$ is non-negative for each J. We obtain that $M\mathbf{w}$ is the sum of a constant vector and a non-negative linear combination of non-decreasing vectors, and is therefore non-decreasing.

Applying this argument to $\mathrm{diag}(\mathbf{b})^{-1}(P^{\varepsilon}_{\ell_-})^T$ and the non-decreasing vector $\mathbf{x}$ gives the first claim on the vector of sorted values, whereas applying the same argument to $\mathrm{diag}(\mathbf{a})^{-1}P^{\varepsilon}_{\ell_+}$ and the non-decreasing vectors $\overline{\mathbf{b}}$ and $\mathbf{q}$ gives the second and third claims.

All that remains is to prove the lemma.

Proof of Lemma A.1. We prove only the the first claim, since the second follows upon taking transposes and interchanging $(\mathbf{a},\mathbf{x})$ and $(\mathbf{b},\mathbf{y})$. Write $M = \mathrm{diag}(\mathbf{a})^{-1}P^{\varepsilon}_{\ell_+}$. Writing M_j for the jth column of M, our goal is to show that $\sum_{j>J} M_j$ is a non-decreasing vector for any J. Fix $i < i'$. We first note that $j \mapsto r(j) := \frac{M_{ij}}{M_{i'j}}$ is non-increasing. Indeed, for $j < j'$, we have $r(j)/r(j') = \frac{M_{ij}M_{i'j'}}{M_{i'j}M_{ij'}} \geqslant 1$ by Lemma A.2. Therefore, for any $i < i'$, we have

$$\begin{aligned} \sum_{j\leqslant J} M_{i'j} \sum_{j>J} M_{ij} &= \sum_{j\leqslant J} r(j)^{-1} M_{ij} \sum_{j>J} r(j) M_{i'j} \\ &\geqslant (r(m-k)^{-1} \sum_{j\leqslant J} M_{ij})(r(m-k) \sum_{j>J} M_{i'j}) \\ &= \sum_{j\leqslant J} M_{ij} \sum_{j>J} M_{i'j} \,. \end{aligned}$$

Recall that each row of M sums to 1. Adding $\sum_{j>J} M_{ij} \sum_{j>J} M_{i'j}$ to both sides of the above inequality therefore yields

$$\sum_{j>J} M_{ij} \leqslant \sum_{j>J} M_{i'j} \,.$$

Since this argument holds for any $i < i'$, the vector $\sum_{j>J} M_j$ is non-decreasing, as claimed. □

Lemma A.2. *If c is submodular and $\mathbf{x}$ and $\mathbf{y}$ are non-decreasing, then for any $\ell \geqslant 0$ the matrix $M := \mathrm{diag}(\mathbf{a})^{-1}P^{\varepsilon}_{\ell_+}$ satisfies $M_{ij}M_{i'j'}/M_{i'j}M_{ij'} \geqslant 1$ for all $i \leqslant i'$, $j \leqslant j'$.*

Proof. By the definition of $P^{\varepsilon}_{\ell_+}$, we can write $M = \mathrm{diag}(\mathbf{a})^{-1}\mathrm{diag}(\mathbf{u}_\ell)K\mathrm{diag}(\mathbf{v}_\ell)$, so

$$\begin{aligned} \frac{M_{ij}M_{i'j'}}{M_{i'j}M_{ij'}} &= \frac{a_i^{-1}a_{i'}^{-1}(u_\ell)_i(u_\ell)_{i'}(v_\ell)_j(v_\ell)_{j'}K_{ij}K_{i'j'}}{a_i^{-1}a_{i'}^{-1}(u_\ell)_i(u_\ell)_{i'}(v_\ell)_j(v_\ell)_{j'}K_{i'j}K_{ij'}} \\ &= \frac{K_{ij}K_{i'j'}}{K_{i'j}K_{ij'}} \\ &= e^{\frac{1}{\varepsilon}\left(c(x_{i'},y_j)+c(x_i,y_{j'})-c(x_i,y_j)-c(x_{i'},y_{j'})\right)} \\ &= \exp\left(-\frac{1}{\varepsilon} \int_{x_i}^{x_{i'}} \int_{y_j}^{y_{j'}} \frac{\partial^2 c}{\partial x \partial y} \,\mathrm{d}y\mathrm{d}x \right) \\ &\geqslant 1 \,, \end{aligned}$$

where the last inequality follows from the assumption that c is submodular. □

A.2. Computing the Jacobians

For $\mathbf{z} \in \mathbb{R}^{n+m}$ we write $\mathbf{z}_f \in \mathbb{R}^n$ (resp. $\mathbf{z}_g \in \mathbb{R}^m$) for the subvector of $\mathbf{z}$ with the first n (resp. the last m) entries of $\mathbf{z}$, i.e., $\mathbf{z} = (\mathbf{z}_f^T, \mathbf{z}_g^T)^T$. Let $\Pi : \mathbb{R}^{n+m} \to \mathbb{R}^{n\times m}$ be the linear mapping defined for any $\mathbf{z} \in \mathbb{R}^{n+m}$ by $\Pi\mathbf{z} = -(\mathbf{z}_f\mathbf{1}_m^T + \mathbf{1}_n\mathbf{z}_g^T)$. For any vector $u \in \mathbb{R}^d$, we denote by $\mathrm{diag}(u)$ the $d \times d$ diagonal matrix with diagonal equal to u.

We define for $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^{n+m}$ the function

$$\tau : (\mathbf{x}, \mathbf{z}) \mapsto \begin{bmatrix} \min_\varepsilon(C(\mathbf{x}) + \Pi\mathbf{z}) + \varepsilon \log \mathbf{a} \\ \min_\varepsilon(C(\mathbf{x})^T + (\Pi\mathbf{z})^T) + \varepsilon \log \mathbf{b} \end{bmatrix},$$

where $C(\mathbf{x}) = [c(x_i, y_j)]_{ij} \in \mathbb{R}^{n \times m}$, and for any $A \in \mathbb{R}^{n \times m}$, $\min_\varepsilon(A) = -\varepsilon \log(e^{-A/\varepsilon}\mathbf{1}_m)$.

If we denote by $\mathbf{z}(\mathbf{x}) = (\mathbf{f}(\mathbf{x})^T, \mathbf{g}(\mathbf{x})^T)^T$ the output of the Sinkhorn iterations upon convergence, then it holds that $\tau(\mathbf{x}, \mathbf{z}(\mathbf{x})) = 0$. τ being continuously differentiable, the implicit function theorem tells us that if the Jacobian $J_\mathbf{z}\tau(\mathbf{x}, \mathbf{z}(\mathbf{x}))$ is invertible, then there exists an open neighborhood of $\mathbf{x}$ where $\mathbf{x} \mapsto \mathbf{z}(\mathbf{x})$ is invertible and its Jacobian satisfies $J_\mathbf{x}\mathbf{z}(\mathbf{x}) = -J_\mathbf{z}\tau(\mathbf{x}, \mathbf{z}(\mathbf{x}))^{-1} J_\mathbf{x}\tau(\mathbf{x}, \mathbf{z}(\mathbf{x}))$. Let us therefore compute these terms.

In order to compute $-J_\mathbf{z}\tau(\mathbf{x}, \mathbf{z})^{-1}$, we first observe that for any $H \in \mathbb{R}^{n \times m}$,

$$[J_A \min_\varepsilon(A)](H) = \frac{(e^{-A/\varepsilon} \circ H)\mathbf{1}_m}{e^{-A/\varepsilon}\mathbf{1}_m},$$

therefore, for any $\delta \in \mathbb{R}^{n+m}$,

$$[J_\mathbf{z} \min_\varepsilon(C(\mathbf{x}) + \Pi\mathbf{z})](\delta) = \frac{(M \circ \Pi\delta)\mathbf{1}_m}{M\mathbf{1}_m},$$

where we write for convenience

$$M = e^{-\frac{C(\mathbf{x}) + \Pi\mathbf{z}}{\varepsilon}}.$$

Notice now that

$$M \circ \Pi\delta = -M \circ (\delta_f \mathbf{1}_m^T + \mathbf{1}_n \delta_g^T) = -\mathrm{diag}(\delta_f) M - M \mathrm{diag}(\delta_g),$$

therefore

$$(M \circ \Pi\delta)\mathbf{1}_m = -\delta_f \circ (M\mathbf{1}_m) - M\delta_g,$$

from which we obtain

$$[J_\mathbf{z} \min_\varepsilon(C(\mathbf{x}) + \Pi\mathbf{z})](\delta) = -\frac{\delta_f \circ (M\mathbf{1}_m) + M\delta_g}{M\mathbf{1}_m} = -\delta_f - \frac{M\delta_g}{M\mathbf{1}_m}.$$

Similarly, we obtain

$$[J_\mathbf{z} \min_\varepsilon(C^T(\mathbf{x}) + (\Pi\mathbf{z})^T)](\delta) = -\delta_g - \frac{M^T\delta_f}{M^T\mathbf{1}_n}.$$

Wrapping up, we finally obtain that

$$[J_\mathbf{z}\tau(\mathbf{x}, \mathbf{z})](\delta) = -\begin{bmatrix} \delta_f + \frac{M\delta_g}{M\mathbf{1}_m} \\ \frac{M^T\delta_f}{M^T\mathbf{1}_n} + \delta_g \end{bmatrix},$$

and therefore, writing $M_1 = \mathrm{diag}(1/M\mathbf{1}_m)M$ and $M_2 = \mathrm{diag}(1/M^T\mathbf{1}_n)M^T$:

$$-J_\mathbf{z}\tau(\mathbf{x}, \mathbf{z}) = \begin{bmatrix} I_n & M_1 \\ M_2 & I_m \end{bmatrix}.$$

Using matrix inversion with the Schur complement, we finally get

$$-J_\mathbf{z}\tau(\mathbf{x}, \mathbf{z})^{-1} = \begin{bmatrix} I_n + M_1 S^{-1} M_2 & -M_1 S^{-1} \\ -S^{-1} M_2 & S^{-1} \end{bmatrix}, \quad (2)$$

where $S = I_m - M_2 M_1$.

To compute $J_\mathbf{x}\tau(\mathbf{x}, \mathbf{z})$, we first observe that for any $\delta \in \mathbb{R}^n$,

$$[J_\mathbf{x}\tau(\mathbf{x}, \mathbf{z})](\delta) = \begin{bmatrix} [J_A \min_\varepsilon(C(\mathbf{x}) + \Pi\mathbf{z})]\,([J_\mathbf{x}C(\mathbf{x})](\delta)) \\ [J_A \min_\varepsilon(C^T(\mathbf{x}) + (\Pi\mathbf{z})^T)]\,\left([J_\mathbf{x}C^T(\mathbf{x})](\delta)\right) \end{bmatrix}.$$

Here, $[J_\mathbf{x}C(\mathbf{x})](\delta) = \mathrm{diag}(\delta)\Delta$ and $[J_\mathbf{x}C^T(\mathbf{x})](\delta) = \Delta^T \mathrm{diag}(\delta)$, where $\Delta = [c'(x_i, y_j)]_{i,j}$. Therefore, using again the notation M_1 and M_2 , one has

$$[J_\mathbf{x}\tau(\mathbf{x}, \mathbf{z})](\delta) = \begin{bmatrix} (M_1 \circ \mathrm{diag}(\delta)\Delta)\mathbf{1}_m \\ (M_2 \circ \Delta^T \mathrm{diag}(\delta))\mathbf{1}_n \end{bmatrix} = \begin{bmatrix} \delta \circ (M_1 \circ \Delta)\mathbf{1}_m \\ (M_2 \circ \Delta^T)\delta \end{bmatrix}. \quad (3)$$

Combining (2) and (3), we finally get from the implicit function theorem that $[J_\mathbf{x}\mathbf{z}(\mathbf{x})](\delta)$ is equal to:

$$\begin{bmatrix} \left(I_n + M_1 S^{-1} M_2\right)(\delta \circ (M_1 \circ \Delta)\mathbf{1}_m) - M_1 S^{-1}(M_2 \circ \Delta^T)\delta \\ S^{-1}\left(-M_2(\delta \circ (M_1 \circ \Delta)\mathbf{1}_m) + (M_2 \circ \Delta^T)\delta\right) \end{bmatrix}.$$

At this point, we should notice that the above derivation is only valid is the Jacobian $J_\mathbf{z}\tau(\mathbf{x}, \mathbf{z}(\mathbf{x}))$ is invertible. However, on easily see that for any $(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^n \times \mathbb{R}^{n+m}$, $\tau(\mathbf{x}, \mathbf{z}) = \tau(\mathbf{x}, \mathbf{z} + \lambda\mathbf{z}_0)$ with $\mathbf{z}_0 = (\mathbf{1}_n^T, \mathbf{1}_m^T)^T$ and $\lambda > 0$; and simultaneously, the $n + m$ equality in $\tau(\mathbf{x}, \mathbf{z})$ are redundant, since as soon as $n + m - 1$ of them are satisfied then they are all satisfied. This implies that $J_\mathbf{z}\tau$ is nowhere invertible. In order to make it invertible, we can just remove the first dimension in the definition of $\tau(\mathbf{x}, \mathbf{z})$, and simultaneously constrain the first coordinate of z to be 0. One can easily check that in that case, all the computations above remain valid after removing the first row/column of each matrix vector of dimension n.

B. Additional experiments

B.1. Simulations

In this section we provide more experimental results for the "larger experiment" simulated problem described in the main text, where we factorize a matrix with dimensions $d = 500$, $n = 256$, $k = 10$, modified by a ground truth quantile normalization and corrupted by truncated Gaussian noise. Figure 4 showed the performance during training of NMF, QMQF and QMF with different batch size for a learning rate equal to 0.01, and $m = 16$ quantiles.

We first assess the influence of the learning rate. In Figure 8, we plot the performance during training of NMF and QMF with various batch size with learning rate 0.01 (left, identical

to Figure 4), and a larger learning rate 0.1 (right). While NMF does not seem to be influenced by the learning rate in this case, we see that the performance of QMF degrades when the learning rate is too large, particularly for small batch sizes, as expected. Overall, this confirms that taking 0.01 allows QMF to converge to a good solution, at least when the batch size is at least 64.

Second, we discuss the impact of m, the number of quantile levels. Figure 9 shows the training error of QMF when the learning rate is fixed to 0.01, and we vary m among 4, 8 and 16. We see that $m = 4$ leads to a suboptimal approximation compared to $m = 8$ or $m = 16$, suggesting that m should be large enough to model the quantile transformation. On the other hand, the fact that $m = 16$ is not better than $m = 8$ (while the ground truth quantile transformation is obtained with $m = 256$ quantile levels) suggests that a relatively small number of quantile levels is enough to approximate a complex transform, in that case.

Figure 10 illustrates the different behaviors of NMF, QMF and QMFQ on a simple matrix X (simulated according to the "toy illustration", with $d = 160$, $n = 80$, $k = 8$, see main text for details), where we see strong row-wise patterns due to different quantile transformations applied rowwise. We see in particular the that residuals after matrix approximation by NMF have still strong rowwise patterns, and overall larger values than those after QMF and QMFQ approximation.

In Figures 11 and 12, finally, we compare the quantile transforms inferred by QMF and QMFQ, respectively, on the "larger experiment" with the parameters of Figure 4. Each figure shows the quantile functions inferred for the first 20 features (out of a total of $d = 500$ features). While the reconstructed quantiles are generally very good approximations of the ground truth (in blue), we see a few cases where QMFQ (a more costly option) recovers slightly better the ground truth quantile function than QMF. In particular, it seems that QMF sometimes allocates its budget of quantile values not optimally (e.g. lower left plot of Figure 11) whereas QMFQ does a better job in Figure 12. It would be interesting to better understand why we see this behavior.

B.2. Genomics

In this section we provide additional experimental results regarding the use of QMF for cancer genomics data integration. In particular, to assess the influence of the number of quantile values m, we show in Figure 13 the decrease in KL loss during optimization, on the 9 cancer data sets, for NMF and for QMF with $m = 8$ or 16 target quantiles. The loss tends to decrease initially faster with NMF, but after about 100 iterations QMF reaches lower loss values than NMF consistently across all cancers and converges to lower values. We do not see any important difference between $m = 8$ and $m = 16$.

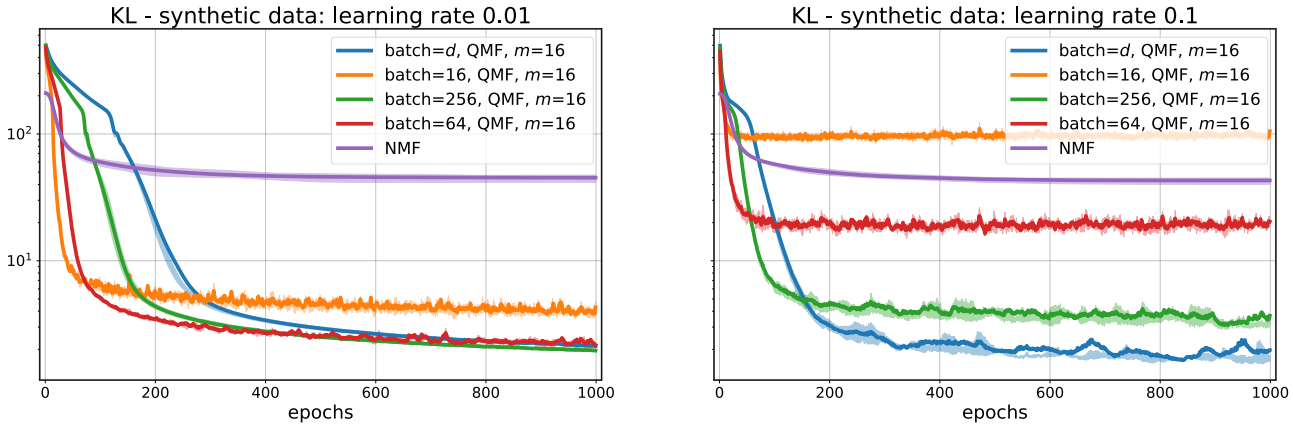

Figure 8. Sensitivity of QMF to learning rate. The setup here is identical to that of Figure 4 in the paper: we consider a synthetic model with additive censored Gaussian noise. We show the results of different methods for a learning rate equal to 0.01 *(left)* or 0.1 *(right)*.

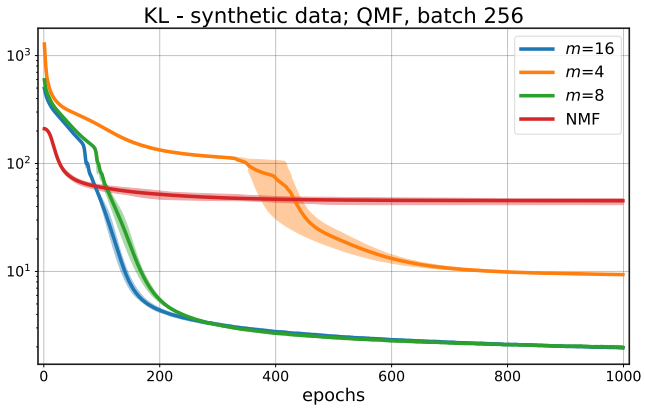

Figure 9. Sensitivity of QMF to the number of target quantiles: we have observed that setting m to a number larger than 8 is usually sufficient to obtain good results. Here again the setup is identical to that of Figure 4, with a learning rate set to 0.01

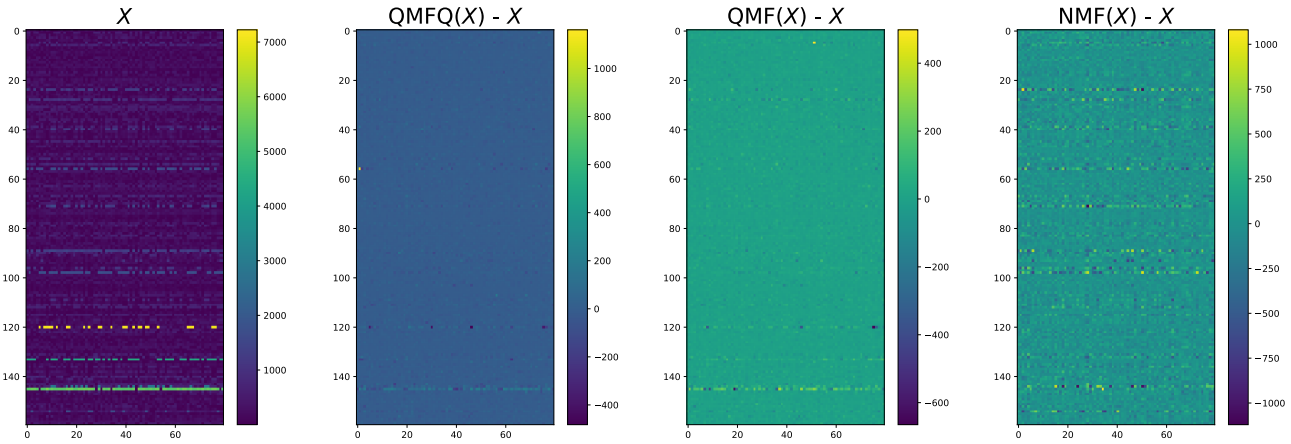

Figure 10. Example of data matrix on the left, along reconstruction errors of all 3 approaches considered here, QMFQ, QMF and NMF.

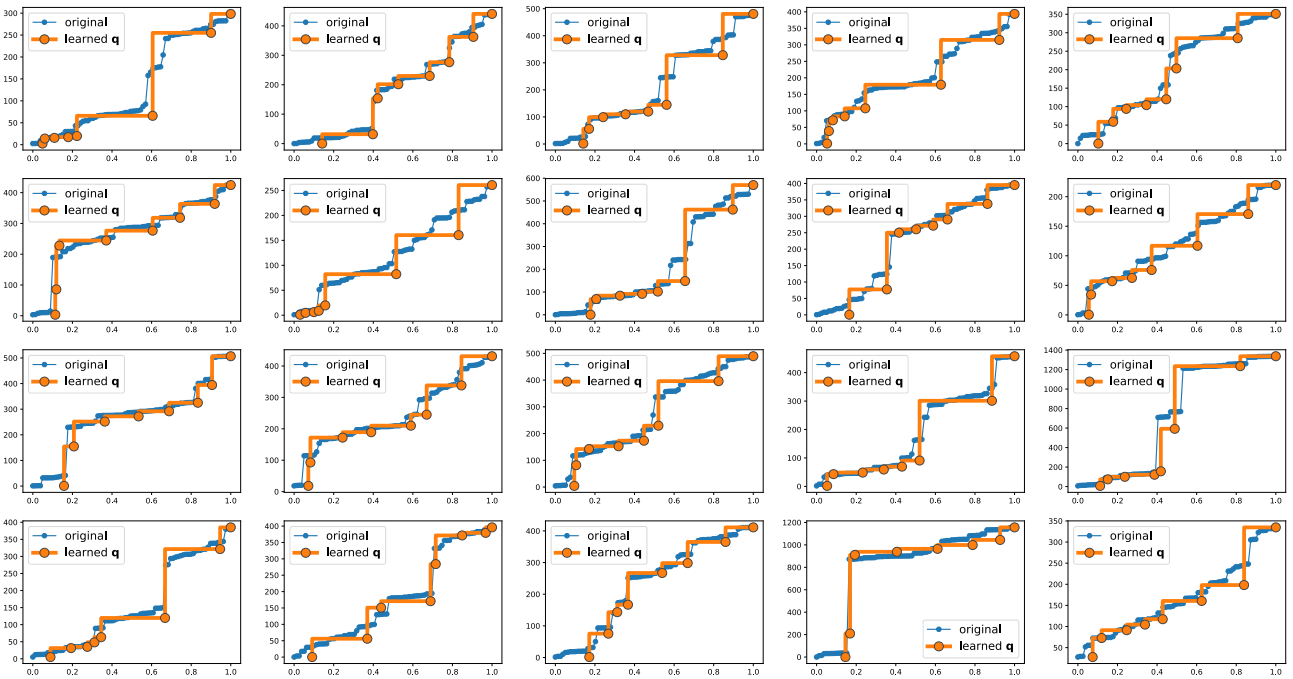

Figure 11. Reconstruction of quantile distributions for QMF in the synthetic + noise setting of Fig. 4 in the paper for the first 20 features.

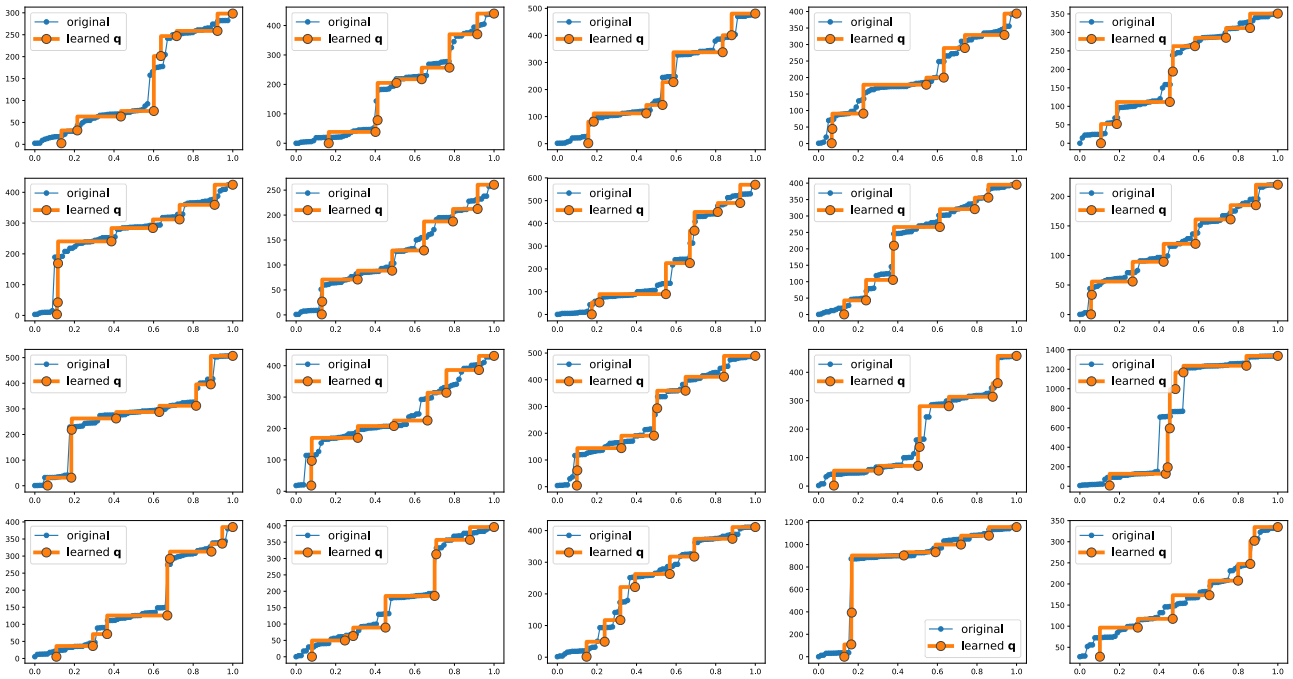

Figure 12. Reconstruction of quantile distributions for QMFQ in the synthetic + noise setting of Fig. 4 in the paper for the first 20 features.

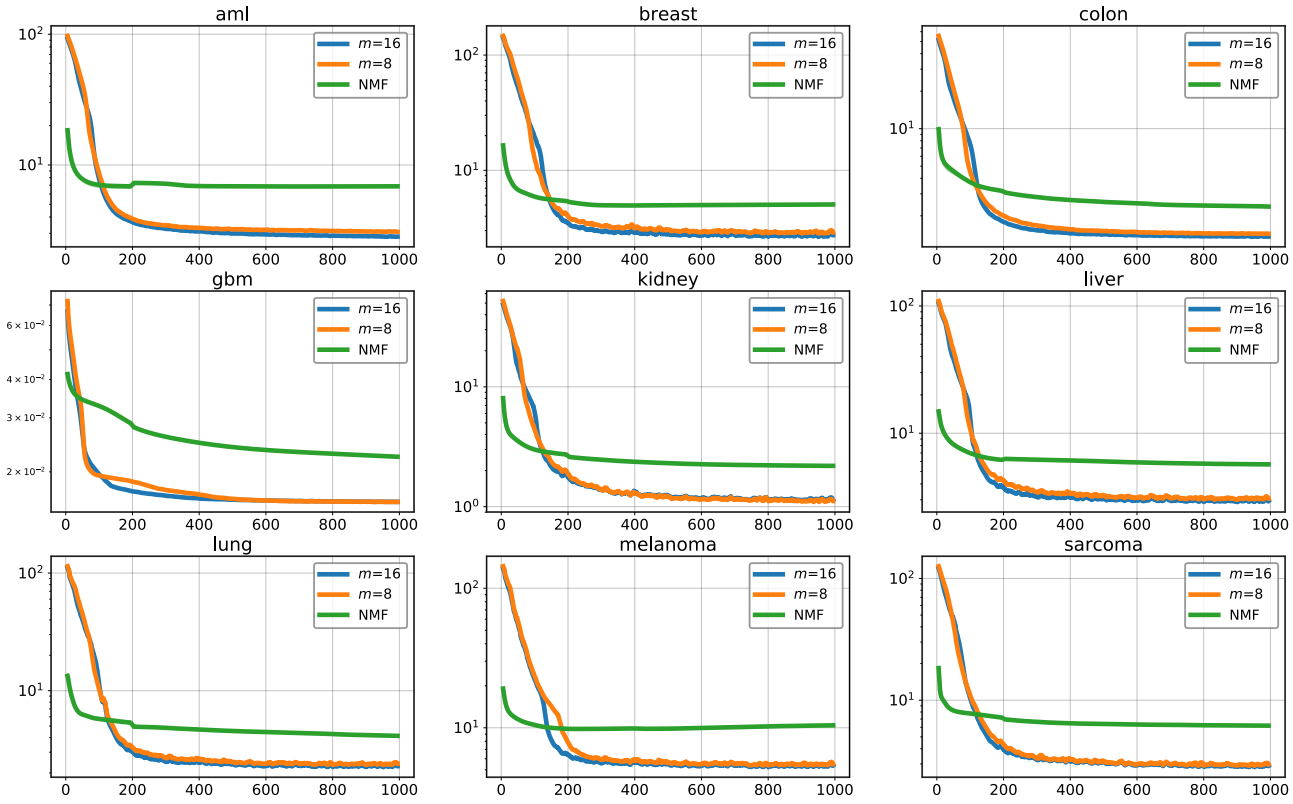

Figure 13. Decrease of Kullback-Leibler divergence on the 9 genomics datasets, using QMF with a batch size of 64 and a learning rate of 0.001 with different number of quantiles m.