Online Nonnegative and Sparse Canonical Polyadic Decomposition of Fluorescence Tensors

Isaac Sanou, Roland Redon, Xavier Luciani, Stéphane Mounier

Online Nonnegative and Sparse Canonical Polyadic Decomposition of Fluorescence Tensors

Isaac Wilfried Sanou[a,b,*], Roland Redon[a], Xavier Luciani[b], Stéphane Mounier[a,*]

[a] *Université de Toulon, Aix Marseille Université, CNRS/INSU, IRD, MIO UM 110, CS 83041 La Garde, France*
[b] *Université de Toulon, Aix Marseille Université, CNRS, LIS, UMR 7020, F-83957 La Garde, France*

Abstract

The NonNegative Canonical Polyadic Decomposition (NN-CPD) is used in many fields such as in chemistry, biology and medicine. The data coming from these fields can be dynamic which lead to use real-time or "online" decomposition. Even though there are a variety of online tensor decomposition algorithms, the main assumption of all these algorithms is that the rank of the decomposition is known and/or does not vary over time. However this should not be the case in experimental conditions. In this work, we propose three algorithms to compute the online NN-CPD based on sparse dictionary learning for tracking chemical components in water by using a set of Emission and Excitation Matrices (EEMs) of fluorescence. The methods developed in this work is not limited to this application field and it addresses the major challenges posed by the variation of the CPD rank in real-time. First, the algorithms take into account the unknown factors and the variation of tensor rank. Second, previous extracted information are used to decompose upcoming new tensors. In addition to the development of these algorithms, one of the contributions of this paper is the real-time acquisition of fluorescence data in a semi-controlled environment. These algorithms were applied on these real datasets and compared to state of the art algorithms.

Keywords: Tensor decomposition; PARAFAC; CPD; Online CPD; Dictionary learning; Sparsity; Rank variation; Recursive computations; Chemometrics; Fluorescence spectroscopy, EEM

[*]We gratefully acknowledge Region Sud PACA (France) and BIOCEANOR for their support.

Email addresses: isaacwilfried@gmail.com (Isaac Wilfried Sanou), roland.redon@univ-tln.fr (Roland Redon), xavier.luciani@univ-tln.fr (Xavier Luciani), stephane.mounier@univ-tln.fr (Stéphane Mounier)

1. Introduction

In the last decade, the Canonical Polyadic Decomposition (CPD) also known as PARAFAC (PARAllel FActor analysis) has been successfully applied in many fields, including psychometric [14], neuroscience [8], chemometric [3], computer vision [46], data mining [50], biomedical image processing [19], hyper-spectral imaging [54], telecommunications [30], etc. The data from these fields can be put in a multidimensional data array (or tensor). Then, the CPD allows to decompose this array into factors that can be interpreted by the user. Indeed one of the main advantages of this approach is that the CPD is essentially unique under some mild assumptions [20]. In environmental sciences the CPD of three-way fluorescence data tensors can be used in order to characterize fluorescent dissolved organic matters (FDOM) in natural water samples [41, 29, 38, 28]. Fluorescence data tensors are built by concatenation of 2D signals called Emission and Excitation Matrices (EEMs) of fluorescence measured from a set of samples. Each entry of an EEM corresponds to the fluorescence intensity of one sample at a given couple of excitation and emission wavelengths. At low concentrations, the non-linear model based on the Beer-Lambert law can be linearized and it then follows the CPD model [21]. The CPD of the fluorescence data tensors allows to recover the individual emission and excitation spectra of the fluorophores present in the different samples along with their respective contributions.

We can notably cite some of algorithms to compute the CPD: Direct TriLinear Decomposition (DTLD) [39], SEmi-algebraic framework for approximate CP decompositions via SImultaneous matrix diagonalization (SECSI) [36], DIrect AlGorithm for canonical polyadic decomposition (DIAG) [24], Alternating Least Square (ALS) [14], Hierarchical Alternating Least Square (HALS) [5, 9, 34], Alternating Direction Method of Multipliers (ADMM) [23] and also some traditional descent algorithms adapted to the CPD problem [51, 17, 43].

In several applications such as fluorescence spectroscopy the decomposition factors have physical meaning and are known to be nonnegative. Indeed, in most algorithms nonnegative constraints can be easily implemented using penalty terms or a projection approach [15, 51, 49, 37,

49, 23]. Most of these algorithm suppose that the rank of the decomposition, that is, the pertinent number of factors is known. However, in many practical situations, this value is unknown and we can only assume an order of magnitude while an incorrect choice of this value can severely affects factor estimations [2]. Two opposite approaches are traditionally used to deal with this issue, namely rank estimation methods and overfactoring. Rank estimation consists in estimating the appropriate CPD rank before the decomposition. Among these methods, we can cite the CORe CONsistency DIAgnostic (CORCONDIA) [4], split half validation [41] or AutoTen [32]. Though these approaches do not always allow to clearly decide between several possible rank values. In the opposite, overfactoring can be seen as *a posteriori* rank estimation method because the appropriate rank value is deduced from the CPD computation. In this approach an overestimated value is chosen (that is supposed to be greater that the actual rank) and CPD algorithms are designed to produce extra factors with null contributions [2]. Such algorithms usually resorts to sparsity constraint upon the factors [48, 6, 13].

At this stage, we have talked about offline CPD. In the particular context of fluorescences spectroscopy and environmental sciences, this situation can correspond to a sampling campaign and the following scheme: obtain K samples, measure the K corresponding EEM, correct eventual non-linear effects, build a 3 way array with the K EEMs on the last mode and lastly compute the CPD as in Figure 1.

However, in this work we are interested in an other practical situation, called online CPD where the data tensor grows with time and its decomposition have to be updated regularly [31]. We now precise the concept of online CPD in the particular context of fluorescence data and environmental sciences but it also applies in other scientific fields [1, 12]. First, we assume that the data tensor is a 3-way array that gathers on its last mode a collection of EEMs and that new collections of EEMs called sub-tensors are recorded regularly as illustrated in Figure 2. A sub-tensor is thus defined as a collection of consecutive EEMs measured during a given time interval. Of course, in practice some fluorophores can appear and/or disappear from one sub-tensor to another. As a consequence the number of fluorophores in the sub-tensors varies with time so that the CPDs of two successive sub-tensors do not necessary share the same rank and factors. More-

over, these variations are unknown. The problem of rank variation has already been addressed in the context of online Nonnegative Matrix Factorization (NMF) in [25, 16]. In this context, NMF would allow to treat each new EEM acquisition as 2D signal. However, doing so, we would also loose the uniqueness properly of the CPD (NMF factors are not necessary unique).

On one hand, the goal of online CPD is to update at each new time interval the CPD factors previously estimated without performing the CPD of the whole tensor and taking into account possible disappearance and/or appearance of some factors. Indeed, real-time performances are required *i.e* the data processing time must be lower than the data acquisition time. On the other hand, we can reasonably assume that consecutive EEMs share several common factors. Thus we may not want to decompose each new sub-tensor independently in order to improve the factor estimation. To the best of our knowledge a few approaches have been proposed to compute online CPD. In [31], the authors introduced two adaptive algorithms: Simultaneous Diagonalization Tracking (SDT) and Recursive Least Squares Tracking (RLST): SDT incrementally tracks the SVD of the unfolded tensor. RLST recursively updates the decomposition factors by minimizing the mean squared error. We can also cite the grid-based tensor factorization algorithm (GridTF) [35] whose the principle is to partition the large tensor into a number of small grids and to use ALS algorithm in parallel to decompose the grids. More recently in [55] authors proposed OnlineCP which is a recursive computation of the factor matrices based on ALS. In [45], the authors proposed a non-linear Least Squares Updating of the CPD based on a non-linear least squares algorithm to update the last factor matrix ($\mathbf{C}$ matrix) in an alternate way. In [11], the authors introduced SamBaTen and they assume that they receive slices on the third mode, which in turn have to add new rows to the $\mathbf{C}$ matrix and update this matrix with a diagonal scaling matrix and a permutation matrix. Lastly, in [12], the authors developed an approach name OCTen based on parallel and compressing incremental decomposition method. However, all theses approaches assume that the rank is known and does not vary with time. In [33], the authors introduced an approach called SeekAndDestroy where the sub-tensors are treated independently using AutoTen for rank estimation before each CPD. Eventually, in [44] authors introduced an online Tucker decomposition with a fixed rank. In this paper, we propose an approach based on

sparse dictionary learning to compute the NonNegative Online CPD (NN-OCPD) and to be able of handling unknown rank variation over time. We consider the most general case and we make no assumption about these variations. The main idea of the proposed approach is thus to combine dictionary learning with LASSO [42] in a simple but appropriate way for NonNegative Online CPD. The factors of the two fixed modes are selected within two dictionaries. Both dictionaries are learned and updated throughout the online process. The appropriate number of selected vectors in the dictionaries (i.e. the CPD rank) is deduced from a sparsity constraint on two atoms matrices. A second idea is then to exploit the recursive relationship between the gradients of the cost-function throughout the online process, as proposed in [44] in an other context, in order to speed up the optimization process since we are looking for a fast algorithm. We have derived from these two ideas three different algorithms. These mainly differ in the way the dictionaries are updated from one sub-tensor to another. We then propose a laboratory experiment in order to obtain real online fluorescence data with rank variations in a semi-controlled environment. This experiment allowed us to evaluate our approach and provide comparison results with state of the art techniques.

The paper is organized as follow. In section 2, the notations are introduced, some basic definitions are recalled and the problem of online CPD in the particular context of fluorescence spectroscopy and environmental sciences is formally described. In section 3, the new proposed approach is depicted in the form of three algorithms to compute the nonnegative online CPD of a third order tensor. Two of these algorithms were briefly introduced in a previous conference paper [40]. In section 4, an experimental design is proposed to evaluate the approaches. Results and comparisons with reference approaches are provided and discussed in section 5. Section 6 concludes.

2. Problem Formulation

2.1. Fluorescence tensors

The main notations used in this paper are presented in Table 1.

In all the following, fluorescence tensors and sub-tensors are thus denoted with letter $\mathcal{T}$ and are

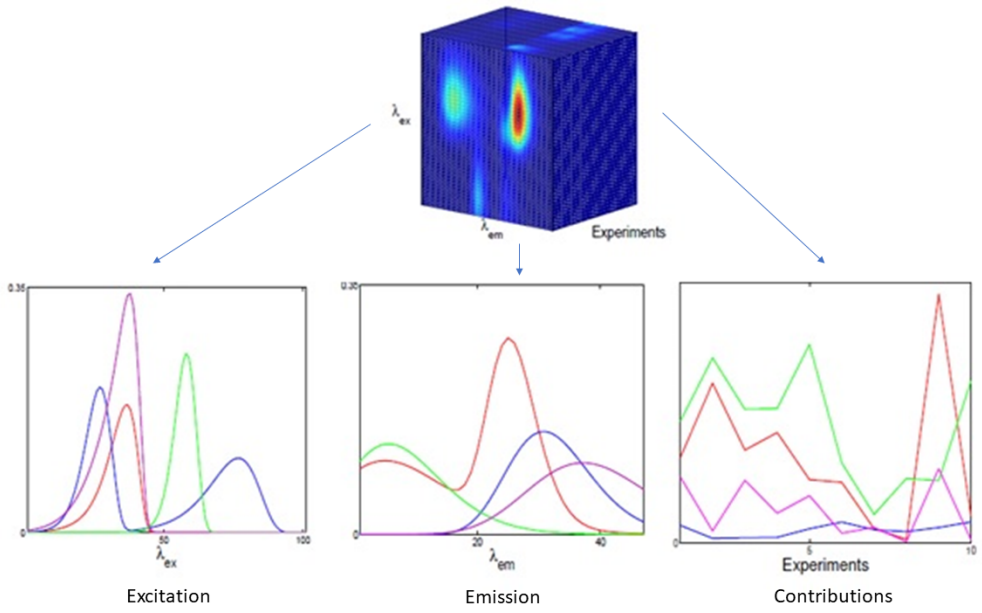

Figure 1: Example of the rank 4 CPD of the fluorescence tensor.

Symbols	**Definition**
$\boldsymbol{\mathcal{T}}$, $\boldsymbol{\mathcal{T}}_n$	Tensor, Sub-tensor at time t_n
$\mathbf{A}$, $\mathbf{A}^\intercal$, $\mathbf{a}$, a	Matrix, Transposed matrix, Column vector, Scalar
$\mathbf{1}_R$	A matrix (R, R) of 1
$\mathbb{R}$	Set of Real Numbers
$\|\|.\|\|_F$, $\|.\|_{1,1}$	Frobenius norm, $L_{1,1}$ norm
$\mathbf{A} \geq 0$	Means that all the elements of matrix $\mathbf{A}$ are nonnegative
$\widetilde{R}_n$,R_n	Physical rank of the sub-tensor $\boldsymbol{\mathcal{T}}_n$, CPD rank of the sub-tensor
$\odot$	Khatri-Rao product

Table 1: Main notations used in the paper.

of size $(I \times J \times K)$ meaning that $\boldsymbol{\mathcal{T}}$ gather K EEM of size $(I \times J)$ on its last mode. All non-linear effects such as diffusion scatters or inner filter effects are supposed to have been corrected so that considered the fluorescence tensors can be well modeled by a low rank (nonnegative) CPD.

2.1.1. Offline Canonical Polyadic Decomposition of a fluorescence tensor

The nonnegative CPD model of a noisy fluorescence tensor $\boldsymbol{\mathcal{T}}$ is given by:

$$\forall i,j,k, \quad \boldsymbol{\mathcal{T}}_{i,j,k} \simeq \widehat{\boldsymbol{\mathcal{T}}}_{i,j,k}(\mathbf{A},\mathbf{B},\mathbf{C}) = \sum_{r=1}^{R} \mathbf{A}_{ir}\mathbf{B}_{jr}\mathbf{C}_{kr} \tag{1}$$

Where matrices $\mathbf{A} \in \mathbb{R}_+^{I \times R}$, $\mathbf{B} \in \mathbb{R}_+^{J \times R}$, $\mathbf{C} \in \mathbb{R}_+^{K \times R}$ are the so-called factor or loading matrices. In the considered application, these matrices are usually full column rank so that the Kruskal condition [20] is fulfilled and guarantees that the decomposition is unique up to trivial scaling and permutation indeterminacy. Columns of **A**, **B** and **C** define the CPD factors and R is the CPD rank. When the CPD factors have a physical interpretation as it is the case here, it is important to distinguish between two particular values of R:

- The maximal value of R for which all the factors have a physical interpretation. We call this value the physical rank $\widetilde{R}$ of $\boldsymbol{\mathcal{T}}$. In other words, $\widetilde{R}$ corresponds to the the number of fluorophores that contribute to the fluorescence signal in at least one EEM of $\boldsymbol{\mathcal{T}}$. In most real world application, this value is unknown and this is the case here. If we choose the correct value of R (that is $R = \widetilde{R}$) then the column of **A**, **B** and **C** can be interpreted as: the excitation, emission and the contributions matrices respectively.

- The minimal value of R for which the Eq. (1) is exact. This value defines the tensor rank. Due to the noise, it is usually much greater than $\widetilde{R}$ and it is useless here.

In order to compute the factor matrices of $\boldsymbol{\mathcal{T}}$ for a given R, most algorithms aim to minimizing the squared $Euclidian$ distance E of the reconstruction error. If the nonnegative constraint is taking into consideration, the offline CPD problem is thus given by:

$$\begin{aligned} &\min \left\{ E(\mathbf{A},\mathbf{B},\mathbf{C}) = \|\boldsymbol{\mathcal{T}} - \widehat{\boldsymbol{\mathcal{T}}}(\mathbf{A},\mathbf{B},\mathbf{C})\|_F^2 \right\} \\ &s.t.\ \mathbf{A} \geq 0, \mathbf{B} \geq 0, \mathbf{C} \geq 0 \end{aligned} \tag{2}$$

2.1.2. Online Canonical Polyadic Decomposition of real-time fluorescence measurements

In the online case, tensor $\boldsymbol{\mathcal{T}}$ can be seen as a collection of **sub-tensors** $\boldsymbol{\mathcal{T}}_n$ of size (I, J, K_n), recorded at various consecutive time intervals. In the simplest case (no overlapping), $\boldsymbol{\mathcal{T}}_n$ gathers on its last mode the EEMs measured between time intervals t_n and t_{n+1}.

However, one may want to build sub-tensors with an overlapping and in this case $\boldsymbol{\mathcal{T}}_n$ also contains the last η EEMs of $\boldsymbol{\mathcal{T}}_{n-1}$ as explained on Figure 2. Sub-tensor overlapping should help the factor estimations but then the online decomposition becomes slower because we find ourselves with a larger number of sub-tensors.

As explained in the introduction, some fluorophores can appear and/or disappear between two consecutive time intervals. For instance, when monitoring chemical components in a natural marine environment, fluorophore appearance and disappearance can be explained by sea currents or pollution events, natural degradation...Therefore, the physical rank of the sub-tensors can vary with time as illustrated on Figure 3. In the following, the physical rank sub-tensor $\boldsymbol{\mathcal{T}}_n$ will be denoted $\widetilde{R}_n$. The online CPD problem with rank variation has been clearly described in [33]. It thus consists in solving Eq. (2) for each sub-tensor $\boldsymbol{\mathcal{T}}_n$. However, here, we consider that $\widetilde{R}_n$ is unknown and has not been estimated previously and the factor matrices are nonnegative. We speak of: NonNegative Online Canonical Polyadic Decomposition (NN-Online CPD). In order to reduce the computational costs and improve the estimation of the factors one can take benefit of the redundant information in consecutive sub-tensors. As a consequence, the computation of the factor matrices of sub-tensor $\boldsymbol{\mathcal{T}}_n$ will consist in updating the factor matrices of sub-tensor $\boldsymbol{\mathcal{T}}_{n-1}$.

3. Three algorithms for computing the Nonnegative Online Canonical Polyadic Decomposition

We propose here 3 algorithms to solve the NN-Online CPD problem introduced in the previous section. These algorithms share the same approach in order to deal with the unknown rank variations. This approach mainly consists in looking for the two factor matrices, corresponding to the modes which do not increase over time, as a product of a dictionary matrix and a sparse

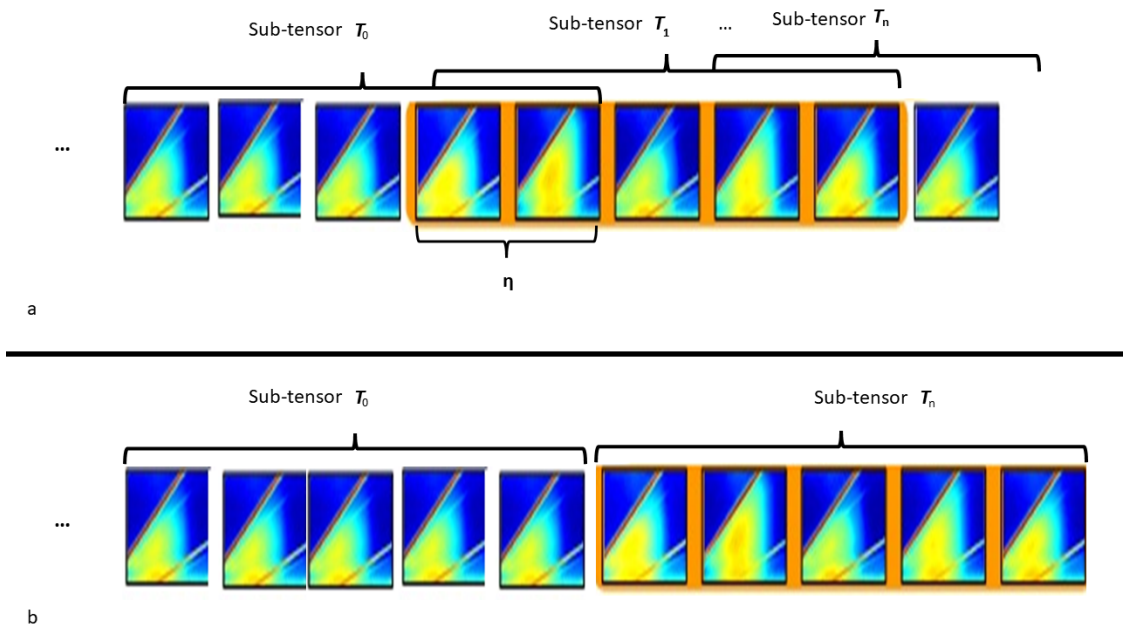

Figure 2: Example of splitting a set of observations into sub-tensors $\mathcal{T}_n$. **a-** Case with data overlapping. **b-** Case with no overlapping.

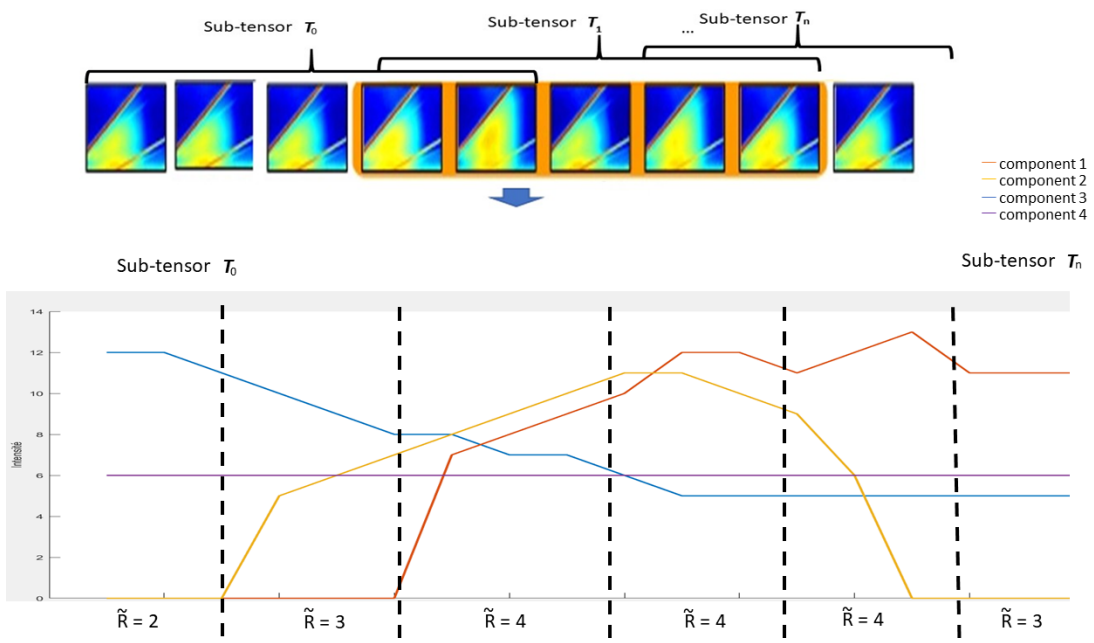

Figure 3: Example of online fluorescence data with rank variation. Plotted lines in the bottom subfigure correspond to the evolution of the contribution (arbitrary units) of the CPD factors.

atom matrix. The atom matrix allows to select the appropriate factors in the dictionary matrix while canceling the others.

Furthermore, we distinguish between an offline and an online step. The offline step can be seen as an initialization step performed at time t_0, that is, after the first acquisition. This step is common to all the three proposed algorithms and it is described in the next subsection.

3.1. Offline step: NN-CPD of $\boldsymbol{\mathcal{T}}_0$

In this step we aim at computing the NN-CPD of the first sub-tensor, acquired before time t_0 and thus denoted $\boldsymbol{\mathcal{T}}_0$. The value of $\widetilde{R}_0$ is unknown but we suppose that we know a value R so that $\forall n, \widetilde{R}_n << R$. We then want to compute the CPD of $\boldsymbol{\mathcal{T}}_0$ but we look for matrices $\mathbf{A}$ and $\mathbf{B}$ as the product of sparse dictionaries $\mathbf{D}^A$ and $\mathbf{D}^B$ by atoms $\mathbf{V}^A$ and $\mathbf{V}^B$ respectively:

$$\mathbf{A} = \mathbf{D}^A\mathbf{V}^A \text{ and } \mathbf{B} = \mathbf{D}^B\mathbf{V}^B, \tag{3}$$

where $\mathbf{D}^A$ and $\mathbf{D}^B$ are of size $(I \times R)$ and $(J \times R)$ respectively and $\mathbf{V}^A$ and $\mathbf{V}^B$ are $(R \times R)$. We expect that $\mathbf{D}^A$ and $\mathbf{D}^B$ contain $\widetilde{R}$ true factors among their columns along with $R-\widetilde{R}$ factors without physical meaning. We thus expect that $\mathbf{V}^A$ and $\mathbf{V}^B$ have $R - \widetilde{R}$ null columns and that the others columns form generalized permutation matrix. For instance, for $R = 4$ and $\widetilde{R}_0 = 2$ we could have ideally:

$$\mathbf{A} = \mathbf{D}^A\mathbf{V}^A = \begin{pmatrix} 2 & 9 & 1 & 11 \\ 5 & 7 & 4 & 13 \\ 10 & 21 & 3 & 4 \\ 2 & 1 & 6 & 3 \\ 14 & 0 & 2 & 5 \\ 2 & 7 & 5 & 9 \end{pmatrix} . \begin{pmatrix} 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 1.8 & 0 & 22 & 0 \\ 4.5 & 0 & 26 & 0 \\ 9 & 0 & 8 & 0 \\ 1.8 & 0 & 6 & 0 \\ 3.6 & 0 & 10 & 0 \\ 1.8 & 0 & 18 & 0 \end{pmatrix}$$

Thereby the number of non-null columns of $\mathbf{A}$ or $\mathbf{B}$ gives a *posteriori* estimation of $\widetilde{R_0}$. A similar problem has been studied in depth by Cohen and Gillis in [6] for the case in which the dictionary is known and the atom is a selection matrix. We thus want to promote the sparsity of matrices $\mathbf{V}^A$ and $\mathbf{V}^B$. In this purpose, we will aim at minimizing their $L_{1,1}$ norm. $L_{1,1}$ norm is classically

used to promote the sparsity [22] since it is more tractable than the $L_{0,0}$ norm. It has been used in similar contexts in [51, 37, 48, 6]. It should be noted that contrary to most CPD algorithms with sparsity constraint, we do not apply the constraint directly on the factor matrix but on the atom. This is why we are not trying to minimize the column rank but to promote the appearance of zeros in the whole matrix. We therefore favor here the $L_{1,1}$ norm instead of the mixed norms such as in [13].

In other words, we aim to solve the following problem for $\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{T}}_0$:

$$\begin{aligned} &\min \left\{ E_1(\mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C}) \right\} \ s.t. \ \mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C} \geq 0 \\ &\text{where } E_1 = \frac{1}{2} \|\boldsymbol{\mathcal{T}} - \widehat{\boldsymbol{\mathcal{T}}}(\mathbf{D}^A\mathbf{V}^A, \mathbf{D}^B\mathbf{V}^B, \mathbf{C})\|_F^2 + \alpha \|\mathbf{V}^A\|_{1,1} + \alpha \|\mathbf{V}^B\|_{1,1} \end{aligned} \tag{4}$$

with $\alpha > 0$ a penalty term.

We call this approach Sparse NonNegative CPD (SNNCPD).

In order to solve Eq. (4) we propose here to resort to a Stochastic Gradient Descent (SGD) algorithm called Nadam [7]. Nadam is a variant of Adam that has been shown to be less sensitive to the choice of the step size at each iteration. The main idea of Nadam optimizer is to rewrite the update of the matrix to be minimized in a conjugate gradient way with momentum incorporation. The goal is to avoid getting stuck in a local minimum during the optimization process. Details of the Nadam algorithm are given in supplementary information 0 (SI 0).

Here we rely on an adaptive step size described in [26]. Note that SGD algorithms had already been used to compute the CPD in [47, 10, 18], including the CPD of fluorescence tensors [38]. In order to ensure the nonnegativity of matrix entries, all the element of matrices $\mathbf{D}^A, \mathbf{D}^B, \mathbf{C}, \mathbf{V}^A$ and $\mathbf{V}^B$ are projected on $\mathbb{R}_+$ at each iteration [52]. We can thus differentiate the $L_{1,1}$ norm and

the gradients with respect to the different variables are given by:

$$\begin{aligned}
\frac{\partial E_1}{\partial \mathbf{V}^A} &= -(\mathbf{D}^A)^\top(\mathbf{T}^A - \mathbf{D}^A\mathbf{V}^A(\mathbf{L}^A)^\top)\mathbf{L}^A + \alpha\mathbf{1}_{R,R} \\
\frac{\partial E_1}{\partial \mathbf{V}^B} &= -(\mathbf{D}^B)^\top(\mathbf{T}^B - \mathbf{D}^B\mathbf{V}^B(\mathbf{L}^B)^\top)(\mathbf{L}^B) + \alpha\mathbf{1}_{R,R} \\
\frac{\partial E_1}{\partial \mathbf{D}^A} &= -(\mathbf{T}^A - \mathbf{D}^A\mathbf{V}^A(\mathbf{L}^A)^\top)\mathbf{L}^A(\mathbf{V}^A)^\top \\
\frac{\partial E_1}{\partial \mathbf{D}^B} &= -(\mathbf{T}^B - \mathbf{D}^B\mathbf{V}^B(\mathbf{L}^B)^\top)(\mathbf{L}^B)(\mathbf{V}^B)^\top \\
\frac{\partial E_1}{\partial \mathbf{C}} &= -(\mathbf{T}^C - \mathbf{C}(\mathbf{L}^C)^\top)(\mathbf{L}^C)
\end{aligned} \tag{5}$$

Matrices $\mathbf{T}^A, \mathbf{T}^B, \mathbf{T}^C$ are obtained by unfolding the tensor $\boldsymbol{\mathcal{T}}$ with respect to the first, second and third modes respectively. With $\mathbf{L}^A = \mathbf{C} \odot (\mathbf{D}^B\mathbf{V}^B)$, $\mathbf{L}^B = \mathbf{C} \odot (\mathbf{D}^A\mathbf{V}^A)$ and $\mathbf{L}^C = (\mathbf{D}^B\mathbf{V}^B) \odot (\mathbf{D}^A\mathbf{V}^A)$. At the beginning of the minimization process, $\mathbf{V}^A$ and $\mathbf{V}^B$ are initialized as the identity matrix while $\mathbf{D}^A$ and $\mathbf{D}^B$ and $\mathbf{C}$ are initialized with nonnegative random values. It should be noted that it is possible to use more sophisticated approaches to solve our constrained optimization such as ADMM [25] used for online matrix factorization or APG [51].

The different steps of SNNCPD are summarized in Algorithm 0.

Algorithm 0 Sparse NonNegative Canonical Polyadic Decomposition (SNNCPD). Offline step (initialization)

- **Input:** $\boldsymbol{\mathcal{T}}$, R overestimated

while *a convergence criterion is not reached* **do**
  Update the dictionaries $\mathbf{D}^A$ *and* $\mathbf{D}^B$ *using Nadam optimizer*
  Update the atoms $\mathbf{V}^A$ *and* $\mathbf{V}^B$ *using Nadam optimizer*
  Update the mode $\mathbf{C}$ *using Nadam optimizer*
end while

- **Output:** $\mathbf{D}_a, \mathbf{D}_b, \mathbf{C}, \mathbf{V}_a, \mathbf{V}_b$

3.2. Online step, algorithm 1

The online step consists in computing the factors of the CPD of $\boldsymbol{\mathcal{T}}_n$ from those of $\boldsymbol{\mathcal{T}}_{n-1}$. For this first algorithm, we consider that the factors of $\boldsymbol{\mathcal{T}}_{n-1}$ were correctly estimated and we only take care of eventual factors appearances and disappearances. As a consequence, we first set $\mathbf{D}_n^A = \mathbf{A}_{n-1}$ and $\mathbf{D}_n^B = \mathbf{B}_{n-1}$ as initialization before the optimization process. Then, Eq. (4) is solved for $\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{T}}_n$ in the same way than in the offline step. However, instead of updating

the entire matrices, we now keep the non-null columns of $\mathbf{D}^A$ and $\mathbf{D}^B$ unchanged throughout the optimization process. We have thus $R - \widetilde{R}_n$ null columns to optimize. For instance, if $\widetilde{R}_{n-1} = 3$ we have:

$$\mathbf{D}^A = \mathbf{A}_{n-1} = \overbrace{\begin{pmatrix} A_{11} & A_{12} & A_{13} & ? & \dots & ? \\ A_{21} & A_{22} & A_{23} & ? & \dots & ? \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{I1} & A_{I2} & A_{I3} & ? & \dots & ? \end{pmatrix}}^{R}$$

fixed entries | optimized entries

Indeed, the fixed columns correspond to the factors previously estimated. If one of these factors has disappeared at time t_n we should then obtain a zero in $\mathbf{V}^A$ and $\mathbf{V}^B$. Combined with SNNCPD for the offline step, we call this first algorithm Online Sparse NonNegative CPD 1 (OSNNCPD 1). The different steps are summarized in Algorithm 1. In order to into account the permutation indeterminacy, after each sub-tensor decomposition, the columns of matrix **B** (associated with the emission spectra) are permuted so that the positions of the maxima of each column are sorted in ascending order. This permutation is then applied to modes **A** and **C**.

Algorithm 1 Online Sparse and NonNegative CPD 1

- **STEP 1: Initialization phase**

$Input$: $\boldsymbol{\mathcal{T}}_0$, R overestimated
$Solve$ Eq. (4) with $\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{T}}_0$ using Algorithm 0 $(SNNCPD)$
$Compute\ matrices\ \mathbf{A}_0$ and $\mathbf{B}_0\ using$ Eq. (3)
$Output$: $\mathbf{A}_0\ and\ \mathbf{B}_0\ and\ \mathbf{C}_0$

- **STEP 2: Online phase**

$Input$: $\mathcal{T}_n\ and\ \mathbf{A}_{n-1}, \mathbf{B}_{n-1}$
$Initialize\ \mathbf{D}^A = \mathbf{A}_{n-1}, \mathbf{D}^B = \mathbf{B}_{n-1}$
$Fill\ the\ null\ columns\ of\ \mathbf{D}^A\ and\ \mathbf{D}^B\ by\ random\ numbers, exclude\ the\ others\ columns\ from\ the\ optimization\ problem$ Eq. (4) $and\ solve\ it\ for\ \boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{T}}_n$
$Update\ current\ factor\ matrices$: $\mathbf{A}_n = \mathbf{D}^A\mathbf{V}^A\ and\ \mathbf{B}_n = \mathbf{D}^B\mathbf{V}^B$
$Output$: $\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n\ and\ \widetilde{R}_n$

- **STEP 3: Return to Step 2 with n=n+1**

3.3. Online step, algorithm 2

For this second algorithm we consider that the factor estimated at time t_n (from $\boldsymbol{\mathcal{T}}_{n-1}$) can be improved and we thus choose to update the whole matrices $\mathbf{A}$ and $\mathbf{B}$. We first replace the null columns of $\mathbf{A}_{n-1}$ and $\mathbf{B}_{n-1}$ by columns of random numbers. Then, we compute $\mathbf{A}_n$ and $\mathbf{B}_n$ thanks to linear combination of the entries of $\mathbf{A}_{n-1}$ and $\mathbf{B}_{n-1}$ and we select the updated factors :

$$\mathbf{A}_n = \mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^A \text{ and } \mathbf{B}_n = \mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}_n^B. \tag{6}$$

$\mathbf{U}_n^A$ and $\mathbf{U}_n^B$ are two square matrices of sizes I and J respectively. With respect to the previous algorithm, $\mathbf{U}_n^A \mathbf{A}_{n-1}$ and the $\mathbf{U}_n^B \mathbf{B}_{n-1}$ play the role of the updated dictionaries. While $\mathbf{V}_n^A$ and $\mathbf{V}_n^B$ matrices are still sparse atoms. Therefore the optimization problem becomes:

$$\begin{aligned} &\min \left\{ E_2(\mathbf{U}_n^A, \mathbf{V}_n^A, \mathbf{U}_n^B, \mathbf{V}_n^B, \mathbf{C}) \right\} \; s.t \; \mathbf{U}_n^A, \mathbf{U}_n^B, \mathbf{C}, \mathbf{V}_n^B, \mathbf{V}_n^B \geq 0 \\ &\text{where } E_2 = \frac{1}{2} \| \boldsymbol{\mathcal{T}} - \widehat{\boldsymbol{\mathcal{T}}}(\mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^B, \mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}^B, \mathbf{C}) \|_F^2 + \alpha \| \mathbf{V}_n^A \|_{1,1} + \alpha \| \mathbf{V}_n^B \|_{1,1} \end{aligned} \tag{7}$$

It is solved in the same way than the offline minimization problem using the Nadam algorithm. This second algorithm, called Online Sparse and NonNegative CPD 2 (OSNNCPD 2) is more flexible but has a higher numerical complexity, since we have more parameters to optimize. The different steps are summarized in Algorithm 2. The permutation indeterminacy between the sub-tensor is solved in the same in OSNNCPD1. The expression of the gradients with respect to the different variables can be easily deduced from the expressions obtained in the offline step and are given in Algorithm 2.

3.4. Online step, algorithm 3

This third algorithm is inspired by the approaches proposed in [27, 44, 55] for accelerating various online decomposition. In [27], authors deal with online matrices decomposition. In [55] it has been extended to online tensor CPD but it does not take into account the rank variation issue. While, in [44], authors focus on the Tucker decomposition. As in [44], we now want that the factor that we are looking for, at time t_{n+1}, describe correctly all the sub-tensors measured previously and not only $\boldsymbol{\mathcal{T}}_n$. We thus have to modify our cost function and we now consider the

Algorithm 2 Online Sparse and Nonnegative CPD 2

• **STEP 1: Initialization phase**

$Identical\ to\ the\ initialization\ of\ the$ Algorithm 1 $(SNNCPD)$

• **STEP 2: Online phase**

$Input$: $\boldsymbol{\mathcal{T}}_n$, and $\mathbf{A}_{n-1}, \mathbf{B}_{n-1}$

$Fill\ null\ columns\ of$ $\mathbf{A}_{n-1}, \mathbf{B}_{n-1}$ $by\ random\ numbers\ and\ Solve$ Eq. (7) for $\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{T}}_n$ $with$:

$\frac{\partial E_2}{\partial \mathbf{V}^A} = -(\mathbf{U}^A\mathbf{A}_{n-1})^\top(\mathbf{T}_{(1)}^{I,KJ} - \mathbf{U}^A\mathbf{A}_{n-1}\mathbf{V}^A\mathbf{Z}_{(1)}^\top)\mathbf{Z}_{(1)} + \alpha\mathbf{1}_{R,R}$

$\frac{\partial E_2}{\partial \mathbf{V}^B} = -(\mathbf{U}^B\mathbf{B}_{n-1})^\top(\mathbf{T}_{(2)}^{J,KI} - \mathbf{U}^B\mathbf{B}_{n-1}\mathbf{V}^B\mathbf{Z}_{(2)}^\top)\mathbf{Z}_{(2)} + \alpha\mathbf{1}_{R,R}$

$\frac{\partial E_2}{\partial \mathbf{U}^A} = -(\mathbf{T}_{(1)}^{I,KJ} - \mathbf{U}^A\mathbf{A}_{n-1}\mathbf{V}^B\mathbf{Z}_{(1)}^\top)\mathbf{Z}_{(1)}(\mathbf{V}^A)^\top(\mathbf{A}_{n-1})^\top$

$\frac{\partial E_2}{\partial \mathbf{U}^B} = -(\mathbf{T}_{(2)}^{J,KI} - \mathbf{U}^B\mathbf{B}_{n-1}\mathbf{V}^B\mathbf{Z}_{(2)}^\top)\mathbf{Z}_{(2)}(\mathbf{V}^B)^\top(\mathbf{B}_{n-1})^\top$

$\frac{\partial E_2}{\partial \mathbf{C}} = -(\mathbf{T}_{(3)}^{K,JI} - \mathbf{C}\mathbf{Z}_{(3)}^\top)\mathbf{Z}_{(3)}$

$Where$:

$\mathbf{Z}_{(1)} = \mathbf{C} \odot (\mathbf{U}^B\mathbf{B}_{n-1}\mathbf{V}^B)$; $\mathbf{Z}_{(2)} = \mathbf{C} \odot (\mathbf{U}^A\mathbf{A}_{n-1}\mathbf{V}^A)$ and $\mathbf{Z}_{(3)} = (\mathbf{U}^B\mathbf{B}_{n-1}\mathbf{V}^B) \odot (\mathbf{U}^A\mathbf{A}_{n-1}\mathbf{V}^A)$

$Update\ current\ factors\ matrices$: $\mathbf{A}_n$ and $\mathbf{B}_n$ $using$ Eq. (6)

$Output$: $\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n$ and $\widetilde{R}_n$

• **STEP 3: Return to Step 2 with n=n+1**

following minimization problem:

$$\begin{aligned} &\min\left\{E_3(\mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C})\right\}\ s.t.\ \mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C} \geq 0 \\ &\text{where } E_3 = \frac{1}{n+1}\sum_{i=0}^{n}\frac{1}{2}\|\boldsymbol{\mathcal{T}}_i - \widehat{\boldsymbol{\mathcal{T}}}_i(\mathbf{D}^A\mathbf{V}^A, \mathbf{D}^B\mathbf{V}^B, \mathbf{C})\|_F^2 + \alpha\|\mathbf{V}^A\|_{1,1} + \alpha\|\mathbf{V}^B\|_{1,1} \end{aligned} \tag{8}$$

Note that at time t_0 ($n = 0$), the problem is equivalent to the SNNCPD problem of equation (4). One may argue that this contradictory with our objective of updating the factors without considering the whole data set. Actually, this new cost function will allow us to derive updating formulas for the gradient expressions of the dictionaries between two consecutive time intervals. This is traduced by a recursive calculation of these gradients throughout the online step that dramatically speeds up the optimization process. Furthermore, we will see that these formulas only imply the current sub-tensor $\boldsymbol{\mathcal{T}}_n$.

For the online step, let us define:

$$\begin{aligned}\mathbf{P}_n^A &= \sum_{i=0}^{n} \mathbf{T}_i^A \mathbf{L}_i^A (\mathbf{V}_i^A)^\intercal \text{ and } \mathbf{Q}_n^A = \sum_{i=0}^{n} \mathbf{V}_i^A (\mathbf{L}_i^A)^\intercal (\mathbf{L}_i^A)(\mathbf{V}_i^A)^\intercal, \\ \mathbf{P}_n^B &= \sum_{i=0}^{n} \mathbf{T}_i^B \mathbf{L}_i^B (\mathbf{V}_i^B)^\intercal \text{ and } \mathbf{Q}_n^B = \sum_{i=0}^{n} \mathbf{V}_i^B (\mathbf{L}_i^B)^\intercal (\mathbf{L}_i^B)(\mathbf{V}_i^B)^\intercal.\end{aligned} \tag{9}$$

We can then easily show that (SI 1) $\forall\, n > 0$:

$$\frac{\partial E_3}{\partial \mathbf{D}^A} = -\frac{\mathbf{P}_n^A}{n+1} + \frac{\mathbf{D}^A \mathbf{Q}_n^A}{n+1} \text{ and } \frac{\partial E_3}{\partial \mathbf{D}^B} = -\frac{\mathbf{P}_n^B}{n+1} + \frac{\mathbf{D}^B \mathbf{Q}_n^B}{n+1} \tag{10}$$

The crucial point here is that we can now use matrices $\mathbf{P}_{n-1}^A$ and $\mathbf{Q}_{n-1}^A$ (respectively $\mathbf{P}_{n-1}^B$ and $\mathbf{Q}_{n-1}^B$) to compute $\mathbf{P}_n^A$ and $\mathbf{Q}_n^A$ (respectively $\mathbf{P}_n^B$ and $\mathbf{Q}_n^B$):

$$\begin{aligned}\mathbf{P}_n^A &= \mathbf{P}_{n-1}^A + \mathbf{T}_n^A \mathbf{L}_n^A (\mathbf{V}_n^A)^\intercal; \;\; \mathbf{Q}_n^A = \mathbf{Q}_{n-1}^A + \mathbf{V}_n^A (\mathbf{L}_n^A)^\intercal (\mathbf{L}_n^A)(\mathbf{V}_n^A)^\intercal \\ \mathbf{P}_n^B &= \mathbf{P}_{n-1}^B + \mathbf{T}_n^B \mathbf{L}_n^B (\mathbf{V}_n^B)^\intercal; \;\; \mathbf{Q}_n^B = \mathbf{Q}_{n-1}^B + \mathbf{V}_n^B (\mathbf{L}_n^B)^\intercal (\mathbf{L}_n^B)(\mathbf{V}_n^B)^\intercal.\end{aligned} \tag{11}$$

During the online process, matrices $\mathbf{P}_n^A$ and $\mathbf{Q}_n^A$ are computed recursively. At iteration k of the optimization process, $\mathbf{P}_{n-1}^A$ and $\mathbf{Q}_{n-1}^A$ are updated by the matrices obtained at the iteration $k-1$. Then, the gradients with respect to $\mathbf{D}^A$ and $\mathbf{D}^B$ are computed thanks to the unfolding matrices of sub-tensor $\boldsymbol{\mathcal{T}}_n$ ($\mathbf{T}_n^A$ and $\mathbf{T}_n^B$) and the matrices $\mathbf{P}_n^A$ and $\mathbf{Q}_n^A$ obtained at iteration $k-1$ using the Eq. (11) while the gradients with respect to $\mathbf{V}^A$, $\mathbf{V}^B$ and $\mathbf{C}$ are computed in a traditional way. Regarding matrix $\mathbf{C}$, no hypothesis on a possible link between $\mathbf{C}_{n-1}$ and $\mathbf{C}_n$ is done. We call this algorithm Recursive Online Sparse NonNegative CPD (ROSNNCPD). The different steps are summarized in Algorithm 3. Again, the permutation indeterminacy is solved in the same way as in OSNNCPD1.

4. Experiment

It is quite difficult to simulate realistic noisy fluorescence data and several CPD algorithms that works well on noisy simulated data fails in real or semi-real conditions [2, 40]. We call semi-real conditions when the data tensors are built from real fluorescence measurement obtained from known laboratory mixtures. This is the case of the experimental set-up that we describe in this section and that allowed us to validate the proposed approach and compare the algorithms.

Algorithm 3 Recursive Online Sparse NonNegative CPD

• **STEP 1: Initialization phase**

Identical to initialization of the Algorithm 1 $(SNNCPD)$

Compute $\mathbf{P}_0^A, \mathbf{Q}_0^A, \mathbf{P}_0^B, \mathbf{Q}_0^B$ *using Eq.* (9)

• **STEP 2: Online phase**

Input : n, $\boldsymbol{\mathcal{T}}_n$, $\mathbf{A}_{n-1}, \mathbf{B}_{n-1}, \mathbf{P}_{n-1}^A, \mathbf{Q}_{n-1}^A, \mathbf{P}_{n-1}^B, \mathbf{Q}_{n-1}^B$

Initialize $\mathbf{D}^A = \mathbf{A}_{n-1}$, $\mathbf{D}^B = \mathbf{B}_{n-1}$,

while *iter* $<$ *iterMax or any other stopping criterion* **do**

Compute $\mathbf{P}_n^A, \mathbf{Q}_n^A, \mathbf{P}_n^B, \mathbf{Q}_n^B$ *using Eq.* (11)

Compute $\frac{\partial E_3}{\partial \mathbf{D}^A}$ and $\frac{\partial E_3}{\partial \mathbf{D}^B}$ *define in Eq.* (10)

Update the dictionaries $\mathbf{D}^A$ *and* $\mathbf{D}^B$

with $\frac{\partial E_3}{\partial \mathbf{D}^A}$ and $\frac{\partial E_3}{\partial \mathbf{D}^B}$ *using projected Nadam optimizer*

Update the atoms $\mathbf{V}^A$ *and* $\mathbf{V}^B$ *using projected Nadam optimizer*

Update the mode $\mathbf{C}$ *using projected Nadam optimizer*

end while

Update current factors matrices : $\mathbf{A}_n = \mathbf{D}^A\mathbf{V}^A$ *and* $\mathbf{B}_n = \mathbf{D}^B\mathbf{V}^B$

Output : $\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, , \mathbf{P}_n^A, \mathbf{Q}_n^A, \mathbf{P}_n^B, \mathbf{Q}_n^B$ *and* $\widetilde{R}_n$

• **STEP 3: Return to Step 2 with n=n+1**

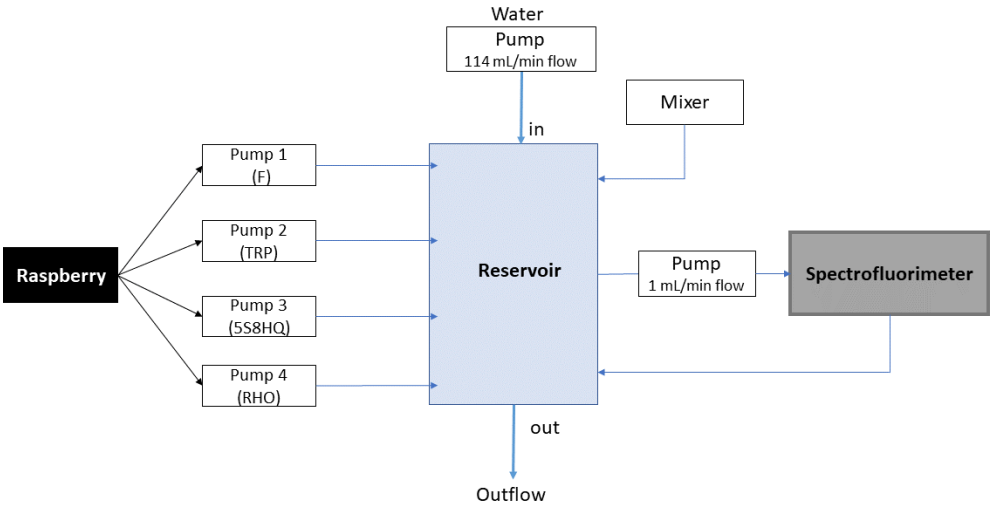

Figure 4: Experimental device for online fluorescence data acquisition.

4.1. Experimental setup

We made in laboratory series of injections of 4 well-known diluted fluorophores (Fluorescein (F), Tryptophan (TRP), 5-sulfonic-8-hydroxyquinolein (5S8HQ) and Rhodamine (RHO)) at 1 $g.L^{-1}$ for each at regular and controlled time intervals in order to simulate the appearance and disappearance of some fluorophores through the time. These fluorophores were chosen because of their peak overlap characteristics and their good fluorescing ability. Fluorophores are injected in a reservoir of water (V=26,8 L) with outflow controlled and with constant water renewal (D=114 $mL.min^{-1}$). Four pumps corresponding to the 4 fluorophores allowed to control the injections. Pumps injections (volume and time delay) were controlled by a Raspberry Pi model 4 as it is shown in Figure 4. On top of the reservoir a mixer was used to stir all the volume. Then the mixture was conveyed at 1 $mL.min^{-1}$ to the spectrofluorimeter for EEMs acquisition at regular time intervals.

The EEMs were recorded with a spectrofluorimeter *Hitachi F7000* with a quartz flow-cell with 1 cm of optical path using a pump (1 $mL.min^{-1}$ flow). The PM voltage was 700 V and the scan speed was 30000 $nm.min^{-1}$. EEMs were measured in the excitation wavelength range of 220 nm-650 nm (slit 10 nm) and in the emission wavelength range of 220 nm-700 nm (slit 10 nm). A 10 nm step size was used both in excitation and emission. Thus, EEMs size is $(49, 44)$.

4.2. Calibration

The calibration step consists in computing the CPD of tensors containing several EEMs of an unique fluorophore at different concentrations. The objectives are threefold: *i.* obtain reference excitation and emission spectra of each fluorophore; *ii.* confirm the linear relationship between the actual concentrations and the estimated profiles; *iii.* detect possible impurities and inner-filter effects. A small volume v_i of the fluorophore is injected in the water, the whole is mixed and conveyed to the spectrofluorimeter for acquisition. The process is repeated to obtain the known concentration range of each component. We should then obtain a tensor of rank $\widetilde{R} = 1$ corresponding to the injected fluorophore. The same protocol is repeated for each fluorophore. Note that after each tensor acquisition, the system is totally cleaned in order to avoid fluorophore

mixture. Furthermore, we ensure that the *pH* of the water does not vary during the acquisition ($pH = 6$). We finally obtain 4 tensors of "pure" fluorophores.

Data processing is done under the progMEEF environment in Matlab [1]. Raman and Rayleigh scattering are removed from each EEM numerically, according to the method proposed in [53] combined with a non-linear 2D-filter.

A CPD of each calibration tensor is first computed using the nonnegative ALS algorithm [4] for different rank values from 1 to 5 and we compute the CORe CONsistency DIAgnostic [4] for each rank value. Then the CPD is computed with our SNNCPD algorithm with $R = 5$.

4.2.1. Background noise

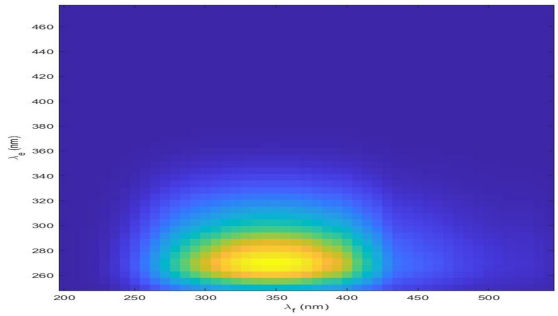

Figure 5: Fluorescent background noise after isolated from the injected components.

For each of the 4 tensors both the CORCONDIA and SNNCPD indicate a physical rank of $\widetilde{R} = 2$ instead of the expected $\widetilde{R} = 1$. Excitation and emission spectra of the extra-component do not vary from one calibration tensor to another. In addition, its contribution is quite high and constant. Moreover, excitation spectra, emission spectra and concentration profile of each fluorophore were correctly estimated by the CPD, no inner-filter effects were observed. As a consequence, the extra-component will be considered as a fifth fluorophore present in all the measured EEMs. It will be referred as to the background noise. Its EEM spectrum is plotted in Figure 5.

[1] http://woms18.univ-tln.fr/progmeef/

4.3. Online fluorescence tensor acquisition

After the calibration step, 1000 EEMs were recorded online in order to evaluate the proposed online CPD algorithms. Data acquisition parameters are the same as for the calibration step and diffusion scattering are corrected in the same way. We then obtain a data tensor $\boldsymbol{\mathcal{T}}$ of size $(49, 44, 1000)$. $\boldsymbol{\mathcal{T}}$ is then split into 10 sub-tensors of size $(49, 44, 100)$ without overlapping between the sub-tensors.

Sub-tensor $\boldsymbol{\mathcal{T}}_n$	$\boldsymbol{\mathcal{T}}_0$	$\boldsymbol{\mathcal{T}}_1$	$\boldsymbol{\mathcal{T}}_2$	$\boldsymbol{\mathcal{T}}_3$	$\boldsymbol{\mathcal{T}}_4$	$\boldsymbol{\mathcal{T}}_5$	$\boldsymbol{\mathcal{T}}_6$	$\boldsymbol{\mathcal{T}}_7$	$\boldsymbol{\mathcal{T}}_8$	$\boldsymbol{\mathcal{T}}_9$
$\widetilde{R}_n$	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
RHO	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
5S8HQ	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
TRP	×	✔	✔	×	✔	✔	✔	✔	✔	✔
F	×	×	✔	×	✔	×	✔	×	✔	✔
Background noise	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

Table 2: The different sub-tensors $\boldsymbol{\mathcal{T}}_n$ with the true rank.

Table 2 indicates the fluorophores present in each sub-tensor $\boldsymbol{\mathcal{T}}_n$ along with the physical ranks and the corresponding fluorophore. Recall that the background noise is not removed and it is considered as an extra-fluorophore present in all the EEMs.

4.4. Estimation of the theoretical concentrations

Let us denote V_0 the volume of the reservoir and C the fluorophore concentration. The number of moles N present in the constant volume V_0 of the reservoir is thus equal to $N = CV_0$. During an elementary time interval dt, an elementary volume dV of water and fluorophores have been replaced by the same volume dV of water. The corresponding variation of the number of moles present in the new mixture is therefore given by $dN = -CdV$. The volume V_0 remaining constant we also have $dN = V_0 dC$. By combining the two previous relations, we obtain:

$$dC = -\frac{1}{V_0} C dV. \tag{12}$$

The constant flow D_0 of the water discharged into the reservoir is defined by:

$$D_0 = \frac{dV}{dt}. \tag{13}$$

This yield to:

$$dC = -\alpha_0 C dt, \tag{14}$$

where $\alpha_0 = \frac{D_0}{V_0}$. Now let us denote C_0, the fluorophore concentration at the injection. We can thus express the variation of the concentration with time:

$$C(t) = C_0 H(t) e^{(-\alpha_0 t)} \tag{15}$$

where $H(t)$ is the Heaviside step function. We have then to take into account that EEM acquisition is not instantaneous. Let us denote Δt the acquisition time and t_k the beginning time of the acquisition of EEM k. We thus finally estimate the corresponding concentration C_k by averaging the variation in concentration from t_k to $t_k + \Delta t$:

$$C_k = \frac{1}{\Delta t} \int_{t_k}^{t_k + \Delta t} C_0 e^{(-\alpha_0 t)} dt = \frac{C_0}{\alpha_0 \Delta t} (1 - e^{(-\alpha_0 \Delta t)}) e^{(-\alpha_0 t_k)}) \tag{16}$$

4.5. Online CPD of the data tensor $\mathcal{T}$

The proposed algorithms are compared with state of the art, namely SeekAndDestroy [33], Block-Coordinate Variable Metric Forward-Backward (BCVMFB) proposed in [48], Nonlinear Least Squares updating of the canonical polyadic decomposition (NLS) [45] and Robust Canonical Polyadic Decomposition (R-CPD) proposed in [13]. BCVMFB and R-CPD are based on the overestimation of the rank using the $L_{1,1}$ norm and mixed norm respectively. These algorithms can be used in an online way by overestimating the rank at start and computing the CPD for each new sub-tensor. NLS was chosen here because in [45], it showed better performances than SDT and RLST. For the same reason [33], we used SeekAndDestroy rather than OnlineCP and SamBaTen.

Then, three cases of data arrangement are considered: *i.* without data overlapping between two consecutive sub-tensors, *ii.* 50% overlapping (due to the lake of space, this case is presented in SI 2), *iii.* 90% overlapping. Three performance criteria are used for the evaluation. The first criterion is the estimated values of the physical ranks. SeekAndDestroy, NLS, R-CPD include their own rank estimation procedure (a priori for SeekAndDestroy and NLS and a posteriori for

R-CPD). Regarding the others algorithms, the rank is estimated a posteriori by individuating the non-null columns in the factor matrices. We consider that a column is null if its $L_{1,1}$ norm is lower than a threshold value. Here this value is set to 10^{-5}.

Secondly, after removing null columns of $\widehat{\mathbf{A}}$, $\widehat{\mathbf{B}}$ and the corresponding columns of $\widehat{\mathbf{C}}$, the estimated factor matrices are compared with the true factors thanks to normalized root mean squared errors defined as:

$$E_A = \frac{\|\mathbf{A} - \widehat{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F},\ E_B = \frac{\|\mathbf{B} - \widehat{\mathbf{B}}\|_F}{\|\mathbf{B}\|_F} \text{ and } E_C = \frac{\|\mathbf{C} - \widehat{\mathbf{C}}\|_F}{\|\mathbf{C}\|_F} \tag{17}$$

Here, $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are the estimated matrices. The matrices $\mathbf{A}$, $\mathbf{B}$ were obtained from the calibration step and matrix $\mathbf{C}$ from the Eq. (16). Our second criterion is the thus the mean error:

$$E_M = (E_A + E_B + E_C)/3 \tag{18}$$

It is supposed that the column is close to zero, if the sum of the elements of the column is zero otherwise, the extra factors estimated are considered duplicate components. Among the duplicated components, the components which have a strong correlation with the real components and with a higher concentration are retained to compute the error. Thirdly, the relative reconstruction quality of the whole tensor defined as:

$$E_{\boldsymbol{\mathcal{T}}}(\%) = 100(1 - \frac{\|\boldsymbol{\mathcal{T}} - \widehat{\boldsymbol{\mathcal{T}}}\|_F}{\|\boldsymbol{\mathcal{T}}\|_F}) \tag{19}$$

In addition, we give in SI 2, the mean running times of each proposed. Algorithms are compared for various overestimated rank: 7, 10 and 20. These values are used as an upper bound by all algorithms.

5. Results and discussions

5.1. No overlapping case

5.1.1. Rank estimation

The case without overlapping is difficult to treat because the true rank can vary from one sub-tensor to another. Table 3 shows the estimated rank for each sub-tensor by each algorithm in comparison with the true rank $\widetilde{R}$. Algorithms SeekAndDestroy and NLS allow to determine the

Sub-tensor $\mathcal{T}_n$		$\mathcal{T}_0$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$	$\mathcal{T}_7$	$\mathcal{T}_8$	$\mathcal{T}_9$
True rank $\widetilde{R}$		**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
SeekAndDestroy	R=7	**3**	**4**	4	4	**5**	3	3	**4**	4	4
	R=10	**3**	**4**	4	4	**5**	3	3	**4**	4	4
	R=20	**3**	**4**	4	4	**5**	3	3	**4**	4	4
NLS	R=7	**3**	**4**	4	2	4	3	4	3	4	4
	R=10	**3**	**4**	4	2	4	3	4	3	4	4
	R=20	**3**	**4**	4	2	4	3	4	3	4	4
R-CPD	R=7	**3**	**4**	6	5	6	5	6	6	6	6
	R=10	**3**	**4**	6	5	6	5	6	6	6	6
	R=20	**3**	**4**	6	5	6	5	6	6	6	6
BCVMFB	R=7	**3**	**4**	4	4	6	6	6	6	6	6
	R=10	**3**	**4**	4	4	6	6	6	6	6	6
	R=20	**3**	**4**	4	4	6	8	10	8	10	10
OSNNCPD1	R=7	**3**	**4**	4	4	**5**	**4**	4	**4**	**5**	**5**
	R=10	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
	R=20	**3**	**4**	**5**	**3**	**5**	**4**	4	5	6	6
OSNNCPD2	R=7	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
	R=10	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
	R=20	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
ROSNNCPD	R=7	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
	R=10	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
	R=20	**3**	**4**	**5**	**3**	**5**	10	**5**	8	10	10

Table 3: Rank estimations in the no overlapping case.

correct rank $\widetilde{R}$ for some sub-tensors but tend to underestimated the rank value for most sub-tensors. Conversely, R-CPD and BCVMFB overestimate the rank for most sub-tensors. On the other hand OSNNCPD1, OSNNCPD2 and ROSNNCPD find the correct rank for most sub-tensors, especially OSNNCPD2 finds the correct rank $\widetilde{R}$ for every sub-tensor and whatever the CPD rank used. Using CPD rank $R = 10$, OSNNCPD1, OSNNCPD2 and ROSNNCPD give the exact rank $\widetilde{R}$ for every sub-tensors. The rank overestimation observed with R-CPD and BCVMFB can be particularly explained by the small threshold value that we used here to define the null column in the estimated factor matrices. However, this also indicate that our algorithms are less sensible to this value and that the factors with the smallest contribution can be clearly identified as extra factors.

5.1.2. Mean error on factors

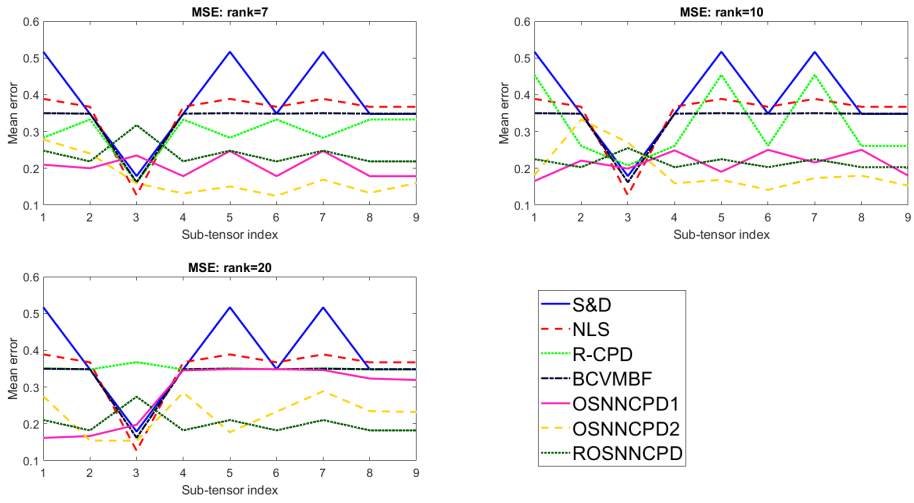

Figure 6: Mean error on the factor matrices in no overlapping case.

Figure 6 shows the evaluation of square error (Eq. (17)) on the factor matrices for the 3 different values of R. For the 3 chosen CPD ranks and for most sub-tensors, algorithms OSNNCPD1, OSNNCPD2 and ROSNNCPD allow to obtain lower mean errors than the other algorithms. This is a direct consequence of the underestimation or overestimation of the rank obtained by

these algorithms.

5.1.3. The relative reconstruction quality

Figure 7 shows the relative reconstruction quality (Eq. (19)) for the 3 different values of R. All the tested algorithms have a high relative reconstruction quality, which implies that all these algorithms have correctly converged. Concerning SeekAndDestroy, NLS and BCVMBF, the relative reconstruction quality is stable versus the rank R. This quality ratio increases with the value of R for the algorithms R-CPD, OSNNCPD1 and OSNNCPD2 while the ratio decreases in the case of ROSNNCPD. We can notice the algorithms OSNNCPD1 and OSNNCPD2 obtain the highest relative reconstruction quality whatever the CPD rank R. OSNNCPD2 gives the best results it is more time consuming (see SI 2 for time comparison).

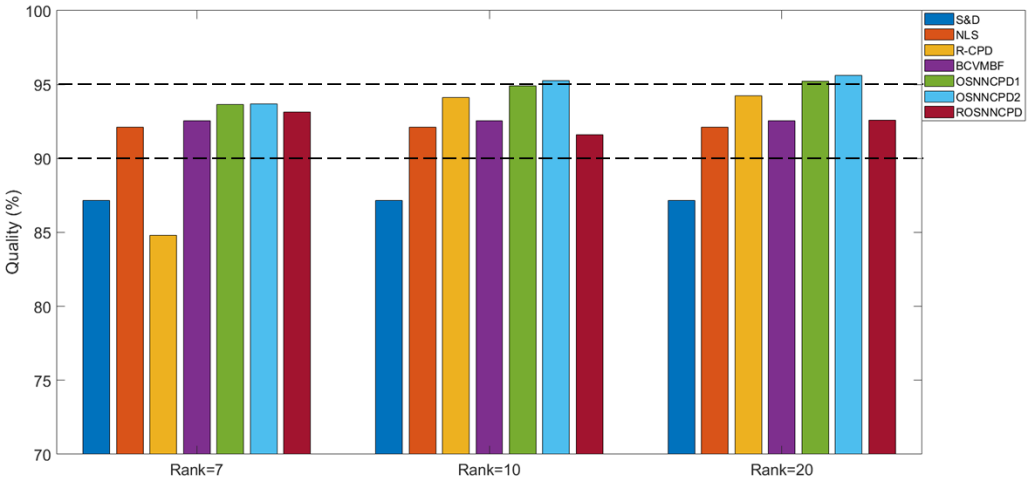

Figure 7: The relative reconstruction quality for no overlapping case.

5.2. 90% overlapping case

5.2.1. Rank estimation

In this case, we have 91 sub-tensors but due to the lake of space, we kept 10 main sub-tensors for rank estimation. These 10 main sub-tensors ($\mathcal{T}_0$, $\mathcal{T}_{11}$, $\mathcal{T}_{21}$, $\mathcal{T}_{31}$, $\mathcal{T}_{41}$, $\mathcal{T}_{51}$, $\mathcal{T}_{61}$, $\mathcal{T}_{71}$, $\mathcal{T}_{81}$,

Sub-tensor $\mathcal{T}_n$		$\mathcal{T}_0$	$\mathcal{T}_{11}$	$\mathcal{T}_{21}$	$\mathcal{T}_{31}$	$\mathcal{T}_{41}$	$\mathcal{T}_{51}$	$\mathcal{T}_{61}$	$\mathcal{T}_{71}$	$\mathcal{T}_{81}$	$\mathcal{T}_{91}$
True rank $\widetilde{R}$		**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
SeekAndDestroy	R=7	**3**	**4**	4	4	**5**	3	3	**4**	**5**	**5**
	R=10	**3**	**4**	4	4	**5**	3	3	10	10	10
	R=20	**3**	**4**	4	4	**5**	3	3	20	20	20
NLS	R=7	**3**	**4**	**5**	4	**5**	3	3	**4**	**5**	**5**
	R=10	**3**	**4**	4	4	**5**	3	3	10	10	10
	R=20	**3**	**4**	4	4	**5**	3	3	20	20	20
R-CPD	R=7	**3**	**4**	4	4	3	3	3	3	3	3
	R=10	**3**	**4**	4	4	3	3	3	3	3	3
	R=20	**3**	**4**	4	4	3	3	3	3	3	3
BCVMFB	R=7	7	7	7	7	7	7	7	7	7	7
	R=10	10	10	10	10	10	10	10	10	10	10
	R=20	20	20	20	20	20	20	20	20	20	20
OSNNCPD1	R=7	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
	R=10	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
	R=20	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
OSNNCPD2	R=7	**3**	**4**	4	4	**5**	**4**	**5**	**4**	**5**	**5**
	R=10	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
	R=20	**3**	**4**	**5**	**3**	**5**	5	4	**4**	**5**	7
ROSNNCPD	R=7	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
	R=10	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**
	R=20	**3**	**4**	**5**	**3**	**5**	**4**	**5**	**4**	**5**	**5**

Table 4: Rank estimation in the 90% overlapping case.

$\mathcal{T}_{91}$) are chosen in function of the rank variation in the case of no overlapping. Results of the different algorithms are quite similar from the previous case as shown the Table 4. State of art (SeekAndDestroy, NLS, R-CPD and BCVMBF) tend to overestimate the rank in each sub-tensor by favoring the apparition of duplicated components. Conversely ROSNNCPD and OSNNCPD1 find the correct rank for sub-tensors. This is also the case of OSNNCPD2 for $R = 10$, but some errors are observed for $R = 7$ and $R = 20$.

5.2.2. Mean error on factors

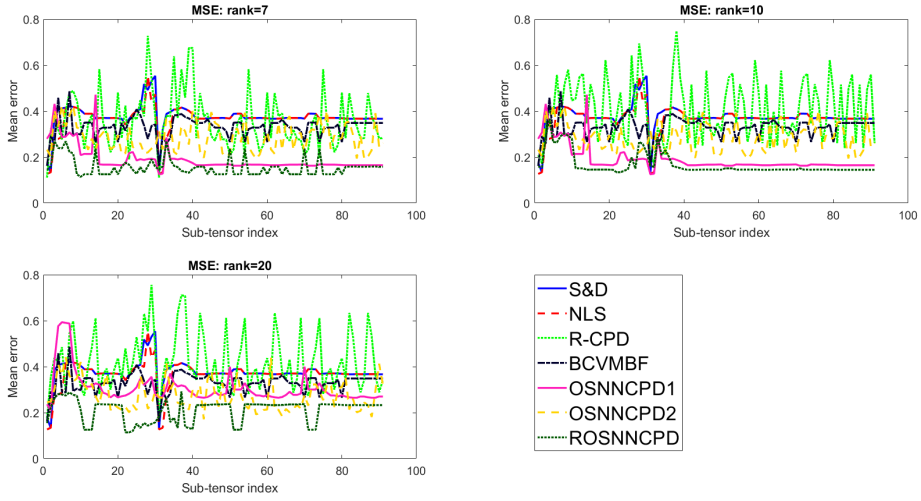

Figure 8: Mean error for the 90% overlapping case

Figure 8 shows the evolution of the mean error on the factor matrices for the three different values of R. With the $R = 7$ or $R = 10$, ROSNNCPD and OSNNCPD1 provide the lowest errors for the most sub-tensors. OSNNCPD2 is less efficient and stable but it still provides better results than state of art algorithms. With $R = 20$, ROSNNCPD clearly provides the best results.

5.2.3. The relative reconstruction quality

Results are plotted on Figure 9. We observe a slight improvement in the relative reconstruction quality with respect to no overlapping case except for R-CPD. The three proposed algorithms allow to reconstruct correctly the data especially for $R = 20$. However, there is no clear cor-

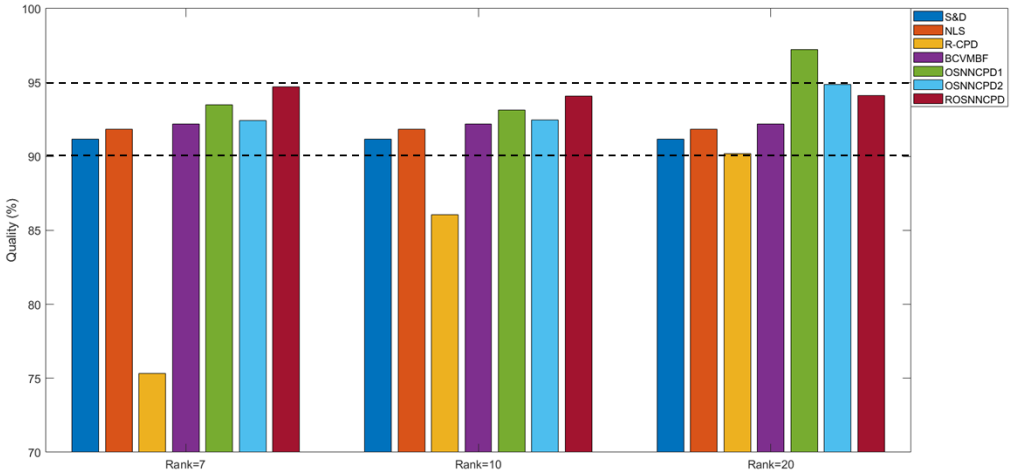

Figure 9: The relative reconstruction quality for 90% of overlapping case.

relation between the relative reconstruction quality and the mean error on the estimated factor matrices.

In order to give physical meaning to these error terms, an example of the normalized factors obtained from the initialization phase are plotted on the Figure 10 (sub-tensor $\mathcal{T}_0$). This figure corresponds to $E_M = 0.18$.

6. Conclusion

In this article, we have proposed an original approach for computing the nonnegative online canonical polyadic decomposition of sub-tensors flow that take into account possible unknown rank variations. Indeed, this approach, based on sparse dictionary learning allows to overestimate the appropriate rank of the CPD by favoring extra-factors with null contribution. We have then derived three distinct NN-Online CPD algorithms. The first algorithm (OSNNCPD1) supposes that the factors estimated from the CPD of the first sub-tensor are correct (offline step) and seeks for new factors during the analysis of the upcoming sub-tensors (online step). Of course,

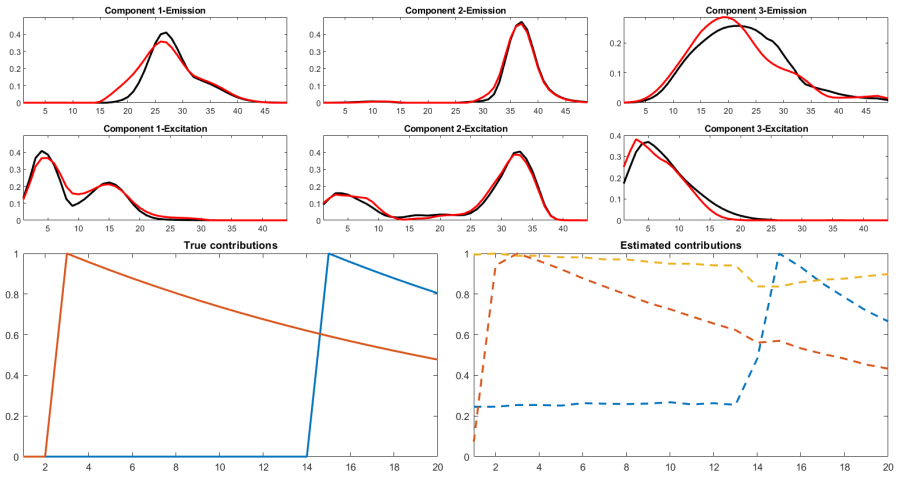

Figure 10: Emission and excitation spectra of the fluorophores in the initialization phase ($\mathcal{T}_0$). Top: Emission spectra, middle: Excitation spectra. Red lines: estimated spectra, black lines: true spectra. All spectra are normalized. Bottom: Fluorophore concentration in the initialization phase ($\mathcal{T}_0$). Full line: "true" contributions (the contribution of the background noise is unknown), dotted line: the estimated contributions. All concentration are normalized.

it also allows to cancel the factors that would become useless throughout the online step. The second algorithm (OSNNCPD2) allows to update the factors estimated previously by using linear combinations of large dictionaries. This algorithm offers more flexibility but the computationnal cost increases. The last algorithm (ROSNNCPD) rewrites the problem of Algorithm 1 in a recursive way in order to speed up the possessing time. In addition, it takes into account all the past sub-tensors in its optimization cost function.

These algorithms are presented in the particular case of the NN-Online CPD of third order fluorescence tensors but they are not limited to this application field and they can be easily extended to higher order tensors. Other constraints such as orthogonal factors could be implemented. A real online fluorescence spectroscopy experiment was conducted in laboratory in order to validate our approach and compare the three algorithms with state of the art approaches. The three proposed algorithms allow to correctly follow the rank variations in most of the considered situations contrary to reference approaches. In fluorescence spectroscopy, the mean error on the estimated factors allows to enhance the identification of the fluorophores. Regarding this criterion

too, our approach usually provide better results that the other online strategies. More precisely, when an overlapping between consecutive sub-tensors is allowed OSNNCPD1 and ROSNNCPD provide clearly the best results. In this case, the factors are also estimated with more accuracy and the algorithms are stabler. Conversely, in the no overlapping case, OSNNCPD2 appears as the best option (but it is more sensible to rank overestimation). Eventually, in most situations, ROSNNCPD provides slightly better results than OSNNCPD1. It is also the fastest algorithm. We thus recommend to combine sub-tensor overlapping and ROSNNCPD.

However, further investigations are planned to test other controlled mixtures and other overlapping rates and strategies. Since the term of the penalty coefficient plays a main role in the rank estimation, we consider to add in the future work the L-curve method to choose it automatically. An extension of the recursive approach used in ROSNNCPD to OSNNCPD2 is also to be considered. Eventually, these encouraging results allows to plan a monitoring in the natural environment in future work.

References

[1] Anaissi, A., Suleiman, B., Zandavi, S.M., 2020. Online tensor decomposition with optimized stochastic gradient descent: an application in structural damage identification, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1257–1264. doi:10.1109/SSCI47803.2020.9308310.

[2] André, R., Luciani, X., Albera, L., Moreau, E., 2020. A two-step algorithm for joint eigenvalue decomposition-application to canonical polyadic decomposition of fluorescence spectra. Chemometrics and Intelligent Laboratory Systems 206, 104065.

[3] Bro, R., 1997. PARAFAC. tutorial and applications. Chemometrics and intelligent laboratory systems 38, 149–171.

[4] Bro, R., Kiers, H.A., 2003. A new efficient method for determining the number of components in PARAFAC models. Journal of Chemometrics 17, 274–286.

[5] Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.i., 2009. Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons.

[6] Cohen, J.E., Gillis, N., 2017. Dictionary-based tensor canonical polyadic decomposition. IEEE Transactions on Signal Processing 66, 1876–1889.

[7] Dozat, T., 2016. Incorporating nesterov momentum into adam. Proceedings of 4th International Conference on Learning Representations, Workshop Track .

[8] Field, A.S., Graupe, D., 1991. Topographic component (parallel factor) analysis of multichannel evoked potentials: practical issues in trilinear spatiotemporal decomposition. Brain topography 3, 407–423.

[9] Fu, X., Huang, K., Sidiropoulos, N.D., Ma, W.K., 2019. Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. IEEE Signal Process. Mag. 36, 59–80.

[10] Ge, R., Huang, F., Jin, C., Yuan, Y., 2015. Escaping from saddle points—online stochastic gradient for tensor decomposition, in: Conference on learning theory, PMLR. pp. 797–842.

[11] Gujral, E., Pasricha, R., Papalexakis, E.E., 2018. Sambaten: Sampling-based batch incremental tensor decomposition, in: Proceedings of the 2018 SIAM International Conference on Data Mining, SIAM. pp. 387–395.

[12] Gujral, E., Pasricha, R., Yang, T., Papalexakis, E.E., 2019. Octen: Online compression-based tensor decomposition, in: 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), IEEE. pp. 455–459.

[13] Han, X., Albera, L., Kachenoura, A., Senhadji, L., Shu, H., 2017. Low rank canonical polyadic decomposition of tensors based on group sparsity, in: 2017 25th European Signal Processing Conference (EUSIPCO), IEEE. pp. 668–672.

[14] Harshman, R.A., 1970. Foundation of PARAFAC procedure: Models and conditions for an 'explanatory' multi-mode factor analysis. UCLA working papers in Phonetics , 1–84.

[15] Huang, K., Sidiropoulos, N.D., Liavas, A.P., 2016. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. IEEE Transactions on Signal Processing 64, 5052–5065.

[16] Iordache, M.D., Bioucas-Dias, J.M., Plaza, A., 2013. Collaborative sparse regression for hyperspectral unmixing. IEEE Transactions on geoscience and remote sensing 52, 341–354.

[17] Kolda, T.G., Bader, B.W., 2009. Tensor decompositions and applications. SIAM review 51, 455–500.

[18] Kolda, T.G., Hong, D., 2020. Stochastic gradients for large-scale tensor decomposition. SIAM Journal on Mathematics of Data Science 2, 1066–1095.

[19] Kopriva, I., Cichocki, A., 2009. Blind multispectral image decomposition by 3d nonnegative tensor factorization. Optics letters 34, 2210–2212.

[20] Kruskal, J.B., 1977. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. Linear algebra and its applications 18, 95–138.

[21] Lakowicz, J., 2006. Principles of Fluorescence Spectroscopy. volume 1. ISBN 978-0-387-31278-1. doi:10.1007/978-0-387-46312-4.

[22] Lee, H., Battle, A., Raina, R., Ng, A.Y., 2007. Efficient sparse coding algorithms, in: Advances in neural information processing systems, Citeseer. pp. 801–808.

[23] Liavas, A.P., Sidiropoulos, N.D., 2015. Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers. IEEE Transactions on Signal Processing 63, 5450–5463.

[24] Luciani, X., Albera, L., 2014. Canonical polyadic decomposition based on joint eigenvalue decomposition. Chemometrics and Intelligent Laboratory Systems 132, 152–167.

[25] Ludivine, N., Miron, S., Jaillais, B., Moussaoui, S., David, B., 2020. A semi-supervised rank tracking algorithm for on-line unmixing of hyperspectral images, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 1514–1518.

[26] Madsen, K., Nielsen, H.B., Tingleff, O., 2004. Methods for non-linear least squares problems .

[27] Mairal, J., Bach, F., Ponce, J., Sapiro, G., 2009. Online dictionary learning for sparse coding, in: Proceedings of the 26th annual international conference on machine learning, pp. 689–696.

[28] Mounier, S., Redon, R., 2006. The use of 3-d fluorescence and its decomposition in environmental organic matter studies. Encyclopedia of Analytical Chemistry: Applications, Theory and Instrumentation , 1–16.

[29] Murphy, K.R., Stedmon, C.A., Graeber, D., Bro, R., 2013. Fluorescence spectroscopy and multi-way techniques.PARAFAC. Analytical Methods 5, 6557–6566.

[30] Nion, D., De Lathauwer, L., 2008. An enhanced line search scheme for complex-valued tensor decompositions. application in ds-cdma. Signal Processing 88, 749–755.

[31] Nion, D., Sidiropoulos, N.D., 2009. Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor. IEEE Transactions on Signal Processing 57, 2299–2310.

[32] Papalexakis, E.E., 2016. Automatic unsupervised tensor mining with quality assessment, in: Proceedings of the 2016 SIAM International Conference on Data Mining, SIAM. pp. 711–719.

[33] Pasricha, R., Gujral, E., Papalexakis, E.E., 2018. Identifying and alleviating concept drift

in streaming tensor decomposition, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 327–343.

[34] Phan, A.H., Cichocki, A., 2008. Multi-way nonnegative tensor factorization using fast hierarchical alternating least squares algorithm (hals), in: Proc. of The 2008 International Symposium on Nonlinear Theory and its Applications.

[35] Phan, A.H., Cichocki, A., 2011. PARAFAC algorithms for large-scale problems. Neurocomputing 74, 1970–1984.

[36] Roemer, F., Haardt, M., 2013. A semi-algebraic framework for approximate cp decompositions via simultaneous matrix diagonalizations (secsi). Signal Processing 93, 2722–2738.

[37] Royer, J.P., Thirion-Moreau, N., Comon, P., 2011. Computing the polyadic decomposition of nonnegative third order tensors. Signal Processing 91, 2159–2171.

[38] Royer, J.P., Thirion-Moreau, N., Comon, P., Redon, R., Mounier, S., 2015. A regularized nonnegative canonical polyadic decomposition algorithm with preprocessing for 3D fluorescence spectroscopy. Journal of Chemometrics 29, 253–265.

[39] Sanchez, E., Kowalski, B.R., 1990. Tensorial resolution: A direct trilinear decomposition. Journal of Chemometrics 4, 29–45.

[40] Sanou, I.W., Redon, R., Luciani, X., Mounier, S., 2021. Online nonnegative canonical polyadic decomposition: Algorithms and application, in: 2021 29th European Signal Processing Conference (EUSIPCO), IEEE. pp. 1805–1809.

[41] Stedmon, C.A., Bro, R., 2008. Characterizing dissolved organic matter fluorescence with parallel factor analysis: a tutorial. Limnology and Oceanography: Methods 6, 572–579.

[42] Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58, 267–288.

[43] Tomasi, G., Bro, R., 2006. A comparison of algorithms for fitting the parafac model. Computational Statistics & Data Analysis 50, 1700–1734.

[44] Traoré, A., Berar, M., Rakotomamonjy, A., 2019. Online multimodal dictionary learning. Neurocomputing 368, 163–179.

[45] Vandecappelle, M., Vervliet, N., De Lathauwer, L., 2017. Nonlinear least squares updating of the canonical polyadic decomposition, in: 2017 25th European Signal Processing Conference (EUSIPCO), IEEE. pp. 663–667.

[46] Vasilescu, M.A.O., Terzopoulos, D., 2002. Multilinear analysis of image ensembles: Tensorfaces, in: European conference on computer vision, Springer. pp. 447–460.

[47] Vervliet, N., De Lathauwer, L., 2015. A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors. IEEE Journal of Selected Topics in Signal Processing 10, 284–295.

[48] Vu, X., Chaux, C., Thirion-Moreau, N., Maire, S., Carstea, E.M., 2017. A new penalized nonnegative third-order tensor decomposition using a block coordinate proximal gradient approach: Application to 3d fluorescence spectroscopy. Journal of Chemometrics 31, 2859.

[49] Vu, X.T., Chaux, C., Maire, S., Thirion-Moreau, N., 2014. Study of different strategies for the canonical polyadic decomposition of nonnegative third order tensors with application to the separation of spectra in 3d fluorescence spectroscopy, in: 2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), IEEE. pp. 1–6.

[50] Wang, H., Ahuja, N., 2004. Compact representation of multidimensional data using tensor rank-one decomposition. vectors 1, 5.

[51] Xu, Y., Yin, W., 2013. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM Journal on imaging sciences 6, 1758–1789.

[52] Zdunek, R., Cichocki, A., 2008. Fast nonnegative matrix factorization algorithms using projected gradient approaches for large-scale problems. Computational intelligence and neuroscience 2008.

[53] Zepp, R.G., Sheldon, W.M., Moran, M.A., 2004. Dissolved organic fluorophores in southeastern us coastal waters: correction method for eliminating rayleigh and raman scattering peaks in excitation–emission matrices. Marine chemistry 89, 15–36.

[54] Zhang, Q., Wang, H., Plemmons, R.J., Pauca, V.P., 2008. Tensor methods for hyperspectral data analysis: a space object material identification study. JOSA A 25, 3001–3012.

[55] Zhou, S., Vinh, N.X., Bailey, J., Jia, Y., Davidson, I., 2016. Accelerating online cp decompositions for higher order tensors, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM. pp. 1375–1384.