Computers and Chemical Engineering 182 (2024) 108557

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Tensorial approaches combining time series and batch data for the end-to-end batch manufacturing of monoclonal antibodies

Fabian Mohr [1,a], Moo Sun Hong [1,a,e], Chris D. Castro [b], Benjamin T. Smith [b], Jacqueline M. Wolfrum [c], Stacy L. Springs [c], Anthony J. Sinskey [c,d], Roger A. Hart [b], Tom Mistretta [b], Richard D. Braatz [*,a,c]

[a] Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA
[b] Amgen, One Amgen Center Drive, Thousand Oaks, CA, USA
[c] Center for Biomedical Innovation, Massachusetts Institute of Technology, Cambridge, MA, USA
[d] Department of Biology, Massachusetts Institute of Technology, Cambridge, MA, USA
[e] School of Chemical and Biological Engineering, Seoul National University, Seoul, Republic of Korea

ARTICLE INFO

Keywords:
Process data analytics
Tensorial analytics
Biopharmaceutical manufacturing
Bioprocess analytics
Partial least squares

ABSTRACT

The product quality for biopharmaceutical production processes is characterized in terms of critical quality attributes. Some quality attributes, however, cannot be easily measured in a timely fashion to support effective intervention and mitigation, particularly for assays that are time-consuming and/or late in the purification process. This article describes the application of predictive modeling techniques to industrial end-to-end biomanufacturing datasets for two monoclonal antibodies to predict critical quality attributes. Methods are proposed for combining time series and batch data that significantly improve the accuracy of the model predictions. These tools are able to take batch-to-batch correlations into account to construct more accurate predictive models from biomanufacturing datasets.

1. Introduction

Prescription drug sales are predicted to continue strong growth, reaching $1.4 trillion in 2026 worldwide. Biopharmaceuticals – products derived from biological organisms for the purpose of treating or preventing diseases – are expected to grow even further from the current 52% to 57% of pharmaceutical sales by 2026 (Evaluate, 2021; Hong et al., 2018). One of the fastest-growing classes of biopharmaceuticals is monoclonal antibodies (mAbs), which may be used to treat a variety of diseases including those associated with cardiovascular, respiratory, immunology, and oncology (Singh et al., 2018). Currently, mAbs account for the second largest type of drugs in the pipeline, surpassed only by small molecules (Lloyd, 2021).

A challenge of mAbs, and biopharmaceuticals more generally, is their high production cost, which is associated with the high doses and quality standards needed to achieve the desired effects (Farid, 2007). These quality requirements are achieved in part by using process analytical technology (PAT), which is a system developed to design, analyze, and control manufacturing in a timely fashion while monitoring critical quality, performance, and process parameters. Critical quality attributes (CQAs) are measured throughout the process to inform the operators about the quality of the products (FDA, 2004). Besides ensuring quality, PAT seeks to improve manufacturing efficiency and success metrics via earlier identification and mitigation of deviations in process parameters and CQAs.

For many CQAs, however, the attribute cannot be easily measured in a timely fashion to support effective intervention and mitigation, particularly for assays that are time-consuming and/or late in the purification process. Instead, CQAs may be predicted via data-driven methods based on process parameters and performance indicator measurements (Severson et al., 2015, 2018; Maruthamuthu et al., 2020). Such papers have demonstrated the application of both traditional multivariable statistical methods (e.g., partial least squares) and more modern machine learning methods to industrial data for mAb production. Some of these models include raw material properties and cell culture process variables as model inputs (Rathore et al., 2015).

* Corresponding author.
E-mail address: braatz@mit.edu (R.D. Braatz).
[1] These authors contributed equally.

https://doi.org/10.1016/j.compchemeng.2023.108557

Available online 15 December 2023

To build these models, the data are first collected into a matrix (aka, a two-dimensional tensor), with batch number along one ordinate and the process variables along the other (batch × process variable). Time can be treated as an additional dimension, to create a third-order tensor (batch × process variable × timestep). A common practice is to unfold the third-order tensor or to use different timesteps as different features to create a matrix for which standard matrix-based multivariate data analytics (e.g., partial least squares) can be applied (Stubbs et al., 2018). In biopharmaceutical manufacturing, some process variables are measured at multiple time points during the production process whereas other variables are only measured once or significantly less frequently (Rathore et al., 2015). Methods have been developed for each type of order, but methods are not available for datasets of multiple orders. This practice raises the question of how to effectively combine datasets with different tensorial orders to guarantee accurate CQA predictions for quality control of mAb production.

Tensorial analytical methods have been applied to process monitoring and CQA prediction for the fed-batch fermentation of penicillin (Hu and Yuan, 2009; Luo et al., 2013, 2014, 2015). The latter two publications consider how to address data of varying sizes in one dimension of the tensor. The most common approach is to use data alignment techniques such as dynamic time warping, but there are also generalized tensorial methods capable of handling uneven lengths in one mode (Luo et al., 2014, 2015). Various tensorial methods exist in the literature, such as multiway principal component analysis (PCA), multiway partial least squares (PLS), higher order PLS models (HOPLS), TUCKER decomposition, and parallel factor analysis (PARAFAC) (Luo et al., 2014, 2016). Simpler methods such as multiway PLS are much more computationally efficient than the more complicated methods, and have been reported to produce model predictions of similar accuracy (Luo et al., 2016). In contrast to handling varying mode lengths, this article addresses datasets consisting of tensors with different numbers of modes.

As cited above, methods are available for building models from data of a single order, such as second- *or* third-order tensorial data, but not on building models from datasets that contain multiple orders. We propose several distinct approaches for combining second- and third-order tensorial data to build models, which were motivated by our efforts to build data-driven models for predicting CQAs for mAb production processes. Additionally, the ability of tensorial approaches to improve CQA predictions by accounting for batch-to-batch correlations is demonstrated. Overall, the full utilization of datasets of different orders and have batch-to-batch correlations is shown to significantly improve the prediction accuracy compared to the widely used approach of only utilizing second-order data. To our knowledge, this article is the first to apply third-order tensorial data analytics to mAb production.

The next section describes tensorial prediction methods and the proposed ways to combine tensors of different orders as model inputs. Then the biomanufacturing case studies are introduced. Subsequently, the results of different methods are assessed, and prescriptive approaches are identified based on data characteristics.

2. Tensorial prediction methods

A common scenario for model building is that a vector of inputs is used to predict a single output. A vector is a first-order tensor. Another common scenario is where a vector of output variables is predicted based on a vector of input variables. In this scenario, the vectors are related by a matrix, which is a second-order tensor. For either scenario, a variety of methods are available for building a predictive model, including Partial Least Squares (PLS), Elastic Net, and Random Forest (Severson et al., 2015; Nikita et al., 2022). A more complicated situation occurs when the input data are also measured over multiple time instances. In this case, the data matrix forms a third-order tensor consisting of the input measurements, predicted variables, and time steps as respective dimensions (Fig. 1).

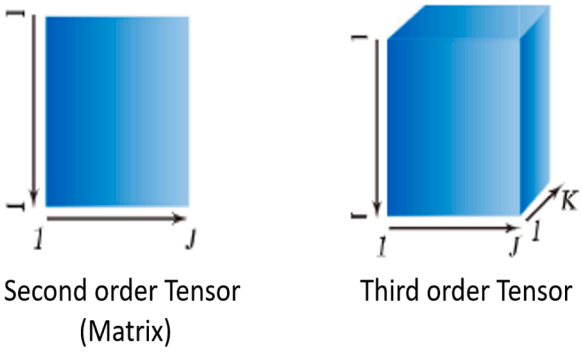

Fig. 1. Visualization of a second-order tensor (aka matrix) and a third-order tensor (3D cube). Figure adapted from Wu et al. (2020).

While industrial practice is to unwrap higher-order tensors to form second-order tensors which are amenable to PLS and other standard methods, existing data analytics literature describes techniques to directly utilize third- and higher-order tensors. Wu et al. (2020) reviews tensorial methods with respect to third- and higher order models applied to chemical processes. Tensorial methods include parallel factor analysis (PARAFAC), the Tucker3 model, bilinear decomposition-based multivariate curve resolution (MCR), and multiway extensions of latent variable methods such as multiway PLS (N-PLS). These methods have had wide application in the chemical industries including to absorbance and Raman spectra and electrochemical data (Wu et al., 2020; Bro et al., 2005). Lopez-Fornieles et al. (2022) discuss the application of N-PLS to time series images of spectral data. Similar to the mAb dataset, the third dimension is introduced by considering time series data and not just spectral data itself. As mentioned in the Introduction, Rathore et al. (2015) analyze an input dataset equivalent to a third-order tensor to predict CQAs of monoclonal antibodies. They apply the well-established method of unfolding to produce a matrix for which PLS can be applied. In this case, the third-order tensor is separated into many second- order tensors that are aligned next to each other to form a larger second-order tensor (aka matrix) (Rathore et al., 2015). While this approach allows the application of all methods capable of dealing with second-order tensors, the downside is that some dependencies between the different dimensions are lost and cannot be exploited when building the prediction models.

The remainder of this section summarizes the widely used partial least squares (PLS), the higher order tensorial PLS used in this article, and proposed strategies for dealing with tensorial datasets of multiple orders.

2.1. Partial least squares (PLS)

When dealing with second-order tensors for the prediction of CQAs, the industry-standard method is PLS. Examples for the application of PLS include the use of near- or mid-infrared spectroscopy for real-time prediction of CQAs (Rosas et al., 2012; Wasalathanthri et al., 2020). PLS is a dimensionality reduction technique that is capable of dealing with high collinearity between the variables. PLS maximizes the covariance between the predictor and predicted variables for each component of the reduced space (Jiao et al., 2015; Chiang et al., 2000). Consider a matrix of input data $\mathbf{X}$ with N measurements and m variables, and a matrix of output data $\mathbf{Y}$ with N measurements and l variables,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix} \in \mathbb{R}^{N\times m}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^\top \\ \vdots \\ \mathbf{y}_N^\top \end{bmatrix} \in \mathbb{R}^{N\times l} \tag{1}$$

$$\mathbf{x}_i \in \mathbb{R}^m, \quad \mathbf{y}_i \in \mathbb{R}^l, \quad i = 1, \cdots, N.$$

In a subsequent step, the variables are projected onto an uncorrelated latent factor space $\mathbf{T}$ with G latent factors as

$$\mathbf{T} = \mathbf{XW} = [\mathbf{t}_1 \cdots \mathbf{t}_G] \in \mathbb{R}^{N \times G} \tag{2}$$

$$\mathbf{X} = \mathbf{TP}^\top + \widetilde{\mathbf{X}} \tag{3}$$

$$\mathbf{Y} = \mathbf{TQ}^\top + \widetilde{\mathbf{Y}} \tag{4}$$

where $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ are residuals, $\mathbf{P} \in \mathbb{R}^{m \times G}$ and $\mathbf{Q} \in \mathbb{R}^{l \times G}$ are the loading matrices of $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{W} \in \mathbb{R}^{m \times G}$ is the projection matrix from the inputs to the lower dimensional subspace, and the vector $\mathbf{t}_i$ is called the *score* of the ith measurement. Maximizing the covariance between the predictor variables and the predicted variables leads to the objective function

$$\max_{\mathbf{w}_i, \mathbf{q}_i} \mathbf{w}_i^\top \mathbf{X}_i^\top \mathbf{Y}_i \mathbf{q}_i \tag{5}$$

$$\text{s.t. } \mathbf{w}_i^\top \mathbf{w}_i = \mathbf{q}_i^\top \mathbf{q}_i = 1$$

for each of the K latent factors. Once the matrices $\mathbf{Q}$ and $\mathbf{W}$ are determined by the optimization, the matrix $\mathbf{T}$ can be constructed and the predictions for $\mathbf{X}$ and $\mathbf{Y}$, $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{Y}}$, can be calculated.

2.2. Multiway partial least squares (N-PLS)

When dealing with third- or higher-order tensors as inputs for CQA prediction, the basic PLS algorithm can only be used in combination with the aforementioned unfolding methods. Instead, higher-order extensions of PLS have been developed, such as the multiway PLS algorithm introduced by Bro Bro (1996). While multiway PLS can handle arbitrarily high orders, here the method is presented only for third-order tensors due to its simplicity and sufficiency for this analysis.

Before tackling the third-order case, consider that, in the second-order case, the predictions for the data matrix $\mathbf{X}$ can be constructed from the weight vector $\mathbf{W}$ as

$$\widehat{x}_{i,j} = \mathbf{t}_i \mathbf{w}_j, \quad i \in 1, \cdots, N, \quad j \in 1, \cdots, m, \tag{6}$$

where $\mathbf{w}_j$ is the jth weight, which is the jth row of $\mathbf{W}$. This second-order case can be extended to the trilinear case by utilizing one score vector and two different weight vectors $\mathbf{w}^J$ and $\mathbf{w}^K$. The predictions for the data matrix $\mathbf{X}$ can be constructed from

$$\begin{aligned} \widehat{x}_{i,j,k} = \mathbf{t}_i \mathbf{w}_j^J \mathbf{w}_k^K, \quad & i \in 1, \cdots, N, \\ & j \in 1, \cdots, m, \\ & k \in 1, \cdots, n. \end{aligned} \tag{7}$$

The desired weights are defined by the optimization

$$\max_{\mathbf{w}_i^J, \mathbf{w}_i^K, \mathbf{q}_i^J, \mathbf{q}_i^K} \mathbf{w}_i^{J\top} \mathbf{w}_i^{K\top} \mathbf{X}_i^\top \mathbf{Y}_i \mathbf{q}_i^J \mathbf{q}_i^K \tag{8}$$

$$\text{s.t. } \mathbf{w}_i^{J\top} \mathbf{w}_i^J = \mathbf{w}_i^{K\top} \mathbf{w}_i^K = \mathbf{q}_i^{J\top} \mathbf{q}_i^J = \mathbf{q}_i^{K\top} \mathbf{q}_i^K = 1.$$

Similar to the bilinear case, the matrices $\mathbf{Q}^J$, $\mathbf{Q}^K$, $\mathbf{W}^J$, and $\mathbf{W}^K$ can be used to calculate the predictions for $\mathbf{X}$ and $\mathbf{Y}$. This article uses the Matlab implementation of N-PLS of Andersson and Bro (2000).

2.3. Proposed approaches for combined handling of second- and third-order tensorial datasets

When thinking about whether to use basic PLS or a higher-order tensorial method such as N-PLS, it is important to understand the dataset and its characteristics. For example, dynamic effects within the predicted variables should also be considered by analyzing the autocorrelation of the predicted variables. Autocorrelation is defined as the correlation between a time series and the same time series shifted by an integer number of lags. If there are batch-to-batch correlations in the CQAs, third-order tensor methods are more suitable than second-order tensor methods. Third-order tensor methods are capable of capturing the batch-to-batch effects better by also considering the timestep dimension that contains additional information across the time order of the batch (Bro, 1996).

If there is autocorrelation in the predicted variables, we will also introduce autoregressive or lagged elements. Autoregressive models consider a lagged version of the predicted variable as an additional input. The predicted variable at a time instance t with a lag order of h can be described in the linear case by Shibata (1976).

$$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_h y_{t-h} \tag{9}$$

During the production of monoclonal antibodies, it is common that some parameters are measured at several time points during the production process whereas others are only measured once. Examples of such datasets are given in Section 3.2. The quality of a whole batch is defined by the CQAs. Consequently, the input data consist of second-order tensors (batches × variables) and third-order tensors (batches × variables × timesteps). One way to deal with tensors of different order is to convert one order into the other., such as the use of the time average of a time series to reduce the dimension of the third-order tensor to second order. Alternatively, the different timesteps can be considered different features, which can be reasonable for variables measured once a day, but results in gigantic matrices if there are frequent measurements. Additionally, there will be very high multicollinearity between the variables.

Methods have been developed for each type of order, but methods are not available for datasets of multiple orders. Here several approaches are proposed to combine datasets of different orders, which are compared in the next section to provide guidance as to which strategy is more suitable. In this article, second-order tensors are denoted with a bold capital letter while third-order tensors are denoted by a bold capital letter with an underbar. For the analyzed biomanufacturing process, the input data are in the form of second- and third-order tensors. The available second-order tensor inputs are denoted by $\mathbf{X}_{\text{batch}}$, which are only measured once for each batch, whereas the available third-order tensor inputs are denoted by $\underline{\mathbf{X}}_{\text{time series}}$, which are time series of measurements for each variable and batch.

Understanding the various methods requires a more detailed description of the process of unfolding. Fig. 2 shows an overview of the two different unfolding methods, where I is the number of batches, J is the number of variables, and K is the number of timesteps. The objective is to predict the CQAs for each batch, so only unfolding methods that have I as one of the dimensions are considered. The first unfolding method aligns K matrices with the variables for each batch next to each other as shown in Fig. 2, which results in a second-order tensor with dimensions $I \times KJ$. The second method aligns J matrices with the timesteps for each batch next to each other, which results in a second-order tensor with dimensions $I \times JK$ (Westerhuis et al., 1999). The only difference between the resulting second-order tensors is the order of the column vectors. When using PLS, the order of column vectors does not influence the result. Consequently, either unfolding method can be used.

Combining these unfolding methods with the aforementioned tensorial analytics applied in the biopharmaceutical and chemical industry Hu and Yuan (2009); Luo et al. (2013, 2014, 2015) led us to propose six approaches of combining tensors of second and third order for CQA prediction:

1. Unfold the third-order tensor $\underline{\mathbf{X}}_{\text{time series}}$ to obtain the second-order tensor $\mathbf{X}_{\text{time series}}$, append to the other second-order tensor $\mathbf{X}_{\text{batch}}$ to form $\mathbf{X} = [\mathbf{X}_{\text{batch}}, \mathbf{X}_{\text{time series}}]$, and apply basic PLS.

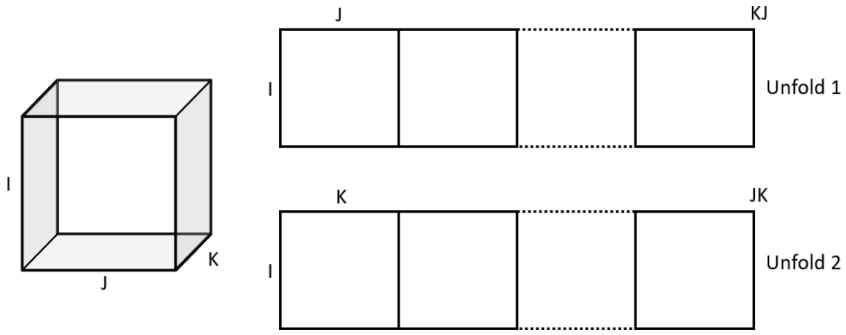

Fig. 2. Visualization of the two relevant unfolding techniques to reduce a third-order tensor to a second-order tensor.

2. Duplicate values for each of the entries in the matrix $\mathbf{X}_{\text{batch}}$ for each timestep K to create the third-order tensor $\underline{\mathbf{X}}_{\text{batch}}$, append to the other third-order tensor $\underline{\mathbf{X}}_{\text{time series}}$ to form $\underline{\mathbf{X}} = [\underline{\mathbf{X}}_{\text{time series}}, \underline{\mathbf{X}}_{\text{batch}}]$, and apply N-PLS.
3. Duplicate values for each of the entries in the matrix $\mathbf{X}_{\text{batch}}$ for each timestep K to create the third-order tensor $\underline{\mathbf{X}}_{\text{batch}}$, append to the other third-order tensor $\underline{\mathbf{X}}_{\text{time series}}$ to create the third-order tensor $\underline{\mathbf{X}} = [\underline{\mathbf{X}}_{\text{time series}}, \underline{\mathbf{X}}_{\text{batch}}]$, unfold the resulting third-order tensor $\underline{\mathbf{X}}$ to form the second-order tensor $\mathbf{X}$, and apply PLS.
4. Only use the third-order tensor and apply N-PLS to $\underline{\mathbf{X}} = \underline{\mathbf{X}}_{\text{time series}}$.
5. Only use the third-order tensor $\underline{\mathbf{X}}_{\text{time series}}$, but unfold to obtain the second-order tensor $\mathbf{X}_{\text{time series}}$ and apply basic PLS.
6. This approach employs two steps. In the first step, only the time series data $\underline{\mathbf{X}}_{\text{time series}}$ are used to predict the CQAs. In the second step, the CQA predictions, or the difference of the predictions to the real values, are fed to the Smart Data Analytics approach for predictive modeling (SPA) Sun and Braatz (2021) as an additional feature appended to the second-order batch tensor. This step results in the second-order tensor $\mathbf{X} = [\mathbf{X}_{\text{batch}}, \text{CQA prediction}]$ as input to SPA. This approach has four variations:
   - 6.1 Use N-way PLS to create CQA predictions from $\underline{\mathbf{X}}_{\text{time series}}$, and use CQA predictions directly as an additional input for SPA.
   - 6.2 Use PLS to create CQA predictions from the unfolded time series tensor $\mathbf{X}_{\text{time series}}$, and use CQA predictions directly as an additional input for SPA.
   - 6.3 Use N-way PLS to create CQA predictions from $\underline{\mathbf{X}}_{\text{time series}}$, and use the difference between the CQA predictions and the real values as an additional input for SPA.
   - 6.4 Use PLS to create CQA predictions from the unfolded time series tensor $\mathbf{X}_{\text{time series}}$, and use the difference between the CQA predictions and the real values as an additional input for SPA.

The first three approaches directly combine the second-order tensor with the third-order tensor in some way, and then data analytics developed for either a second- or third-order tensor are applied depending on the dimensions of the final tensor. Approaches 4 and 5 only use the third-order tensor, and data analytics methods are applied in the same fashion. The motivation for these approaches is the expectation that the time series and batch data contain different information that can be combined to achieve a better model overall. Approach 6 is different as it first uses the information from the third-order, time series tensor to predict the CQAs, and then this prediction or the prediction difference is incorporated as an additional feature for the SPA software. Using the difference in CQA predictions would be expected to perform better if the datasets contain similar information resulting in similar predictions. The prediction difference can be interpreted as a correction term in the second model. A similar concept is used by the dynamic matrix control algorithm (Cutler and Ramaker, 1980). All of these methods can be combined with the autoregressive approach by introducing a lagged form of the predicted variable as an additional input according to Eq. (9).

The next section describes the datasets used as case studies for the proposed approaches.

3. Biopharmaceutical manufacturing datasets

The proposed approaches have been applied to the monoclonal antibodies mAb#1 and mAb#2.

3.1. Monoclonal antibody#1 (mAb#1)

The mAb#1 production process starts with a series of scale-up and expansion steps within shake flasks, culture bags, and seed bioreactors to achieve the desired viable cell density and inoculum volume for the production bioreactor. After centrifugation and depth filtration harvest, a sequence of purification steps including column 1 chromatography, viral inactivation, column 2 chromatography, column 3 chromatography, virus filtration, and ultrafiltration/diafiltration yields the final drug substance. The predicted CQA for mAb#1 is the Column 2 basic peaks %. Basic peaks contain different types of impurities which can be influenced by both the production bioreactor and downstream process steps (Hong et al., 2023). The mAb#1 dataset consists of 169 input variables for 75 batches which form a second-order tensor $\mathbf{X}_{\text{batch}}$.

Additionally, time series data for this process includes parameters such as agitation, pH, dissolved oxygen, O_2 flow, different temperature measurements, pressures, and volumes at 15 minute timesteps. The time series data are used from the point of inoculation until the end of culture. Batch culture durations varied slightly lot-to-lot and consequently each batch has a different number of timesteps. However, third-order tensor methods require the same number of timesteps for each variable and batch, so the dynamic time warping (DTW) approach is used. This method finds a warping path $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_K]$ between two different time series and minimizes the overall distance norm while matching the ith element of the first time series to the jth element of the second time series described by the warp path elements $\mathbf{v}_k$ Salvador and Chan (2007). Given two time series $\mathbf{q}$ with its ith timestep q_i and $\mathbf{c}$ with its jth timestep c_j, the algorithm can be expressed as

$$\text{DTW}(\mathbf{q}, \mathbf{c}) = \min_{\{\mathbf{v}_1, \cdots, \mathbf{v}_k, \cdots, \mathbf{v}_K\}} \sqrt{\sum_{k=1, \mathbf{v}_k=(i,j)}^{K} (q_i - c_j)^2} \tag{10}$$

DTW determines the minimum distance by using an optimal warping path. By choosing the longest time series with K timesteps and matching all the other time series to it according to the calculated warping path, the same length can be achieved for all of the batches. Overall, this results in a third-order tensor with 75 batches, 18 variables, and 1150 timesteps.

3.2. Monoclonal antibody#2 (mAb#2)

The production process of mAb#2 is overall similar to mAb#1 with minor differences in the downstream unit operations. The predicted

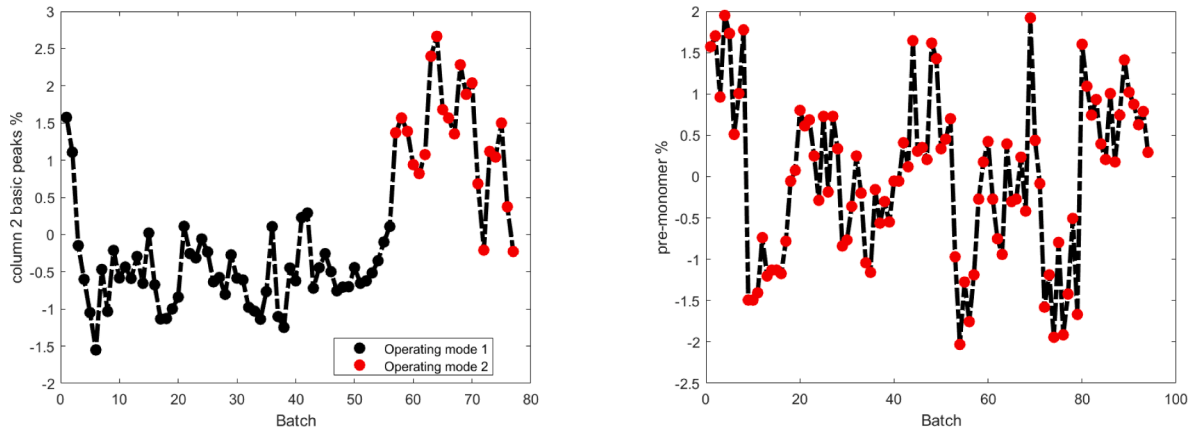

Fig. 3. Overview of the z-scored predicted variables for both monoclonal antibodies. mAb#1 on the left shows a distinct shift in the predicted variable while mAb#2 on the right does not. This results in two different regimes for mAb#1, which are a result of two different sets of operating conditions.

CQA for mAb#2 is the drug substance pre-monomer %, where the pre-monomer is an impurity composed of an intact mAb#2 bound to a mAb#2 fragment. This impurity is undesirable, and the process requires the pre-monomer % to be under a certain threshold, and accurate predictions of the CQA are crucial to assure product quality. This CQA is impacted by different process steps, including both upstream and downstream processes (Hong et al., 2023). For the mAb#2 case study, the second-order tensor $\mathbf{X}_{\text{batch}}$ consists of 118 input variables for 94 batches. mAb#2 has the same time series parameters available as for mAb#1, and DTW is similarly used to align the lengths of the time series data. Overall, this results in a third-order tensor with 94 batches, 18 variables, and 1224 timesteps. In this case, the time series data are available for the same batches as for the batch data. Consequently, a direct comparison of method performance is possible in this case.

4. Results and discussion

4.1. Predicted variable autocorrelation analysis

An important factor when building a model for prediction of CQAs from process variables is to understand the dataset and its characteristics such as the autocorrelation behavior. Additionally, it is also important to analyze how the predicted variables develop over time to detect possible shifts in operating conditions. Fig. 3 shows the predicted variables for both mAbs for each of the batches. mAb#1 has two distinct regimes with respect to the predicted CQA.

Fig. 4 shows the autocorrelation for both monoclonal antibodies considering all available batches. The blue envelope shows the confidence bounds based on a 95% confidence level. If this threshold is crossed, autocorrelation is detected at the confidence level of 95%. Significant autocorrelation can be detected for mAb#1 for the first ten lags and significant autocorrelation is detected for mAb#2 for the first three lags.

An important consideration is whether the strong autocorrelation for mAb#1 is an artefact of the apparent shift in CQA response. Autocorrelation is defined as the correlation between a time series and the same time series shifted by an integer number of lags. Thus, a shift in the mean response will result in high autocorrelations even if the autocorrelation of data is small before or after the shift, respectively. The autocorrelation for the mAb#1 dataset has also been analyzed before and after the shift in CQA response (Fig. 5). From this figure, it is apparent that the observed autocorrelation over all batches is actually just a result of having the shift in the response, so there is no statistically significant autocorrelation for mAb#1. However, given the significant autocorrelation for a lag of up to three for mAb#2, tensorial methods are expected to perform better for mAb#2 and we can also include an additional autoregressive term as shown in equation 9.

In our case, the lagged predicted variables are considered as additional inputs to the aforementioned approaches, allowing the direct consideration of autocorrelation effects. The lag order h can be determined based on the autocorrelation. In our case, only lag orders up to 3 are considered as only the first 3 lags show significant autocorrelation. There are some later lags that are slightly above the significance threshold, but the effect is weaker than for the first 3 lags. When determining the order of an autoregressive model, the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) are oftentimes considered. Both lead to worse performance if additional irrelevant variables are included and a simpler model only considering strongly statistically significant inputs is preferred Tsay (1984).

4.2. Application of approaches 1 through 6 on mAb#1

In the previous section, the autocorrelation behavior of mAb#1 is analyzed in detail. Strong autocorrelation can be detected when looking at all available batches for mAb#1; however, this result is an artefact of the shift in the response variable. When the data before and after the shift are analyzed separately, no significant autocorrelation can be detected. While tensorial methods are known to perform well when there is autocorrelation or batch-to-batch effects in the data, it is not surprising that this method does not lead to improved predictiveness for mAb#1 due to a lack of autocorrelation.

To facilitate discussion and comparisons between the approaches, the analyses were grouped within four strategies. The relationship to the relevant approaches is shown in Table 1.

The base case only using the second-order tensorial data ($\mathbf{X}_{\text{batch}}$) in combination with a smart data analytics approach Sun and Braatz (2021) has been applied to the mAb#1 dataset (Strategy A). Subsequently, Approaches 1 through 5 have been applied to the mAb#1 dataset (Strategy B). While Approaches 1 through 3 contain both batch data and time-series data, Approaches 4 and 5 only utilize time series data. Similarly a model can be built based only on batch data. For this purpose, a smart data analytics approach is used that first analyzes the characteristics of the dataset and subsequently applies the most suitable subset of algorithms Sun and Braatz (2021). In particular, the algorithms Algebraic Learning Via Elastic Net (ALVEN), Support Vector Regression (SVR), Random Forest (RF), Partial Least Squares (PLS), Sparse PLS (SPLS), and Elastic Net (EN) are applied as these algorithms handle the present multicollinearity and nonlinearity in the dataset well Sun and Braatz (2021). Additionally, a log transformation has been applied to the predicted variable. To enable a statistically robust comparison of the results, 15 outer folds, 5 inner folds, and 10 repetitions are chosen in combination with a 70-30 train-test split ratio for all different approaches. The results of the batch-only base case compared to the tensorial Approaches 1–5 are shown in Fig. 6 in the form of a violin plot.

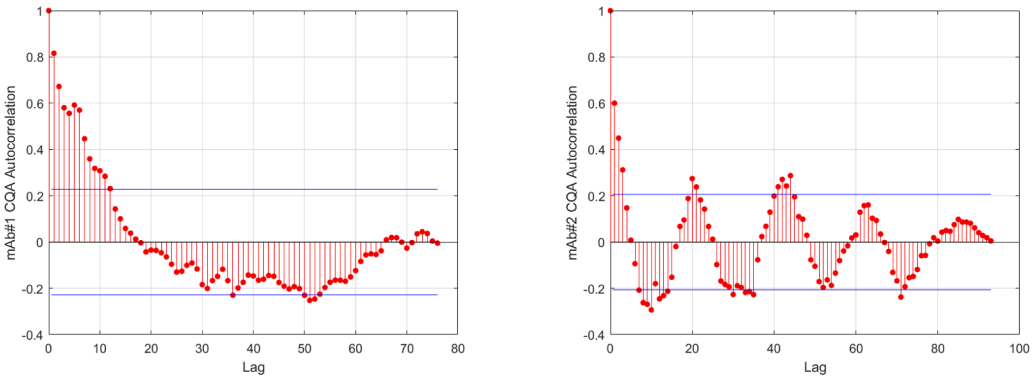

Fig. 4. Autocorrelation function for the predicted variables for mAb#1 on the left and mAb#2 on the right, when applied to the entire datasets. Both plots seem to indicate significant autocorrelation.

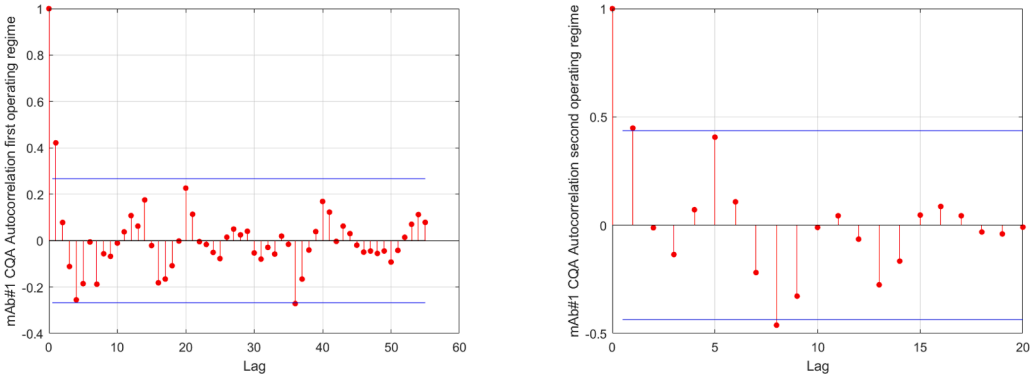

Fig. 5. Autocorrelation function for the predicted variables for mAb#1. The left plot and right plots show the autocorrelations before and after the shift in CQA response variable, respectively.

Table 1
Overview of Strategies A–D and their relationship to prior introduced approaches.

Strategy	Strategy Description	Relationship Approach
A	Base case for comparison using only second-order tensorial data ($\mathbf{X}_{\text{batch}}$)	None
B	Approaches 1–5 applied to all second- and third-order tensorial data ($\mathbf{X}_{\text{batch}}, \underline{\mathbf{X}}_{\text{time series}}$)	Approaches 1–5
C	Approaches 1–5 applied to reduced second-order tensorial data and the full third-order tensorial data. The reduced feature set is based on the most predictive features ($\mathbf{X}_{\text{batch,red}}, \underline{\mathbf{X}}_{\text{time series}}$)	Approaches 1–5
D	Approaches 6.1–6.4 applied to all second- and third-order tensorial data ($\mathbf{X}_{\text{batch}}, \underline{\mathbf{X}}_{\text{time series}}$)	Approaches 6.1–6.4

Each black bar in the violins represents one of the 15 different outer folds. The blue left side of each violin represents the training data results, and the orange right side shows the results for the testing data.

The best performing method when using only batch data (Strategy A) was identified to be PLS, since it yields the lowest values for the testing mean and variance of the RMSE, of 0.408 and 0.00612 respectively. RF achieves the next best mean RMSE for the testing data; however, the significant difference in training versus test RMSE suggests overfitting. Among the tensorial methods (Strategy B), Approach 2 is best with a mean RMSE of 0.449 and a variance of 0.00689. Whereas the mean and the variance of Approach 2 are worse for this prediction compared to just using batch data with PLS, the performance of the testing data is more similar to the performance of the training data using tensorial approaches.

Despite the tensorial Approaches 1–3 using more information via both the batch and time series data, they do not yield better results for mAb#1 than using only the batch data. One possible reason is that the multiway PLS algorithm has trouble picking the most predictive batch features as they are greatly multiplicated for some of the approaches. This hypothesis seems reasonable since Approaches 4 and 5, which use only time series data, perform almost as well as Approaches 1–3. If so, the prediction may be improved by reducing the batch feature matrix to the most predictive features right away (Strategy C). In order to determine a predictive subset of the batch features, we can use the models built when using only batch data. Of these models, two are sparse linear prediction methods utilizing only a reduced feature set: SPLS and EN. Three different reduced batch matrix models have been investigated and they are based on the most predictive features suggested by either EN or SPLS. An overview of the mean prediction errors and the variances of each of the reduced models compared to the full model is shown in Table 2. An example set of results for the reduced model, which includes only batch features 2, 4, 5, 6, 7, 18, and 45, is shown in Fig. 7a. The performance is overall very comparable between the different reduced models, and no significant improvements are noted in terms of testing RMSEs. However, the training and testing performance is more similar[2].

Finally, Approach 6 can be applied by first using the time series data to predict the CQA either by using multiway PLS or PLS on the unfolded data (Strategy D), and subsequently using either the predictions directly or the difference to the real values as an additional input to the smart process analytics software by Sun and Braatz (2021). Approach 6 results in a very similar prediction problem as the base comparison case that only uses batch data. In this case, just one additional input variable is derived from the time series data. For the mAb#1 dataset, each of the

[2] If we only use the reduced batch data, performance does not vary significantly from the base case.

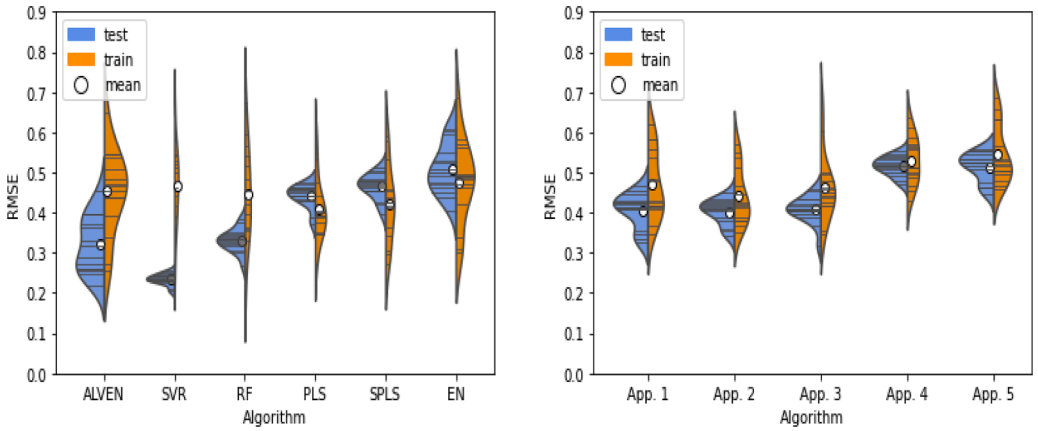

Fig. 6. For the mAb#1 dataset, RMSE results utilizing only batch data are shown in the left plot (Strategy A) as compared against tensorial Approaches 1–5 using both batch and time series datasets in the right plot (Strategy B). The results are shown for 15 outer folds in the form of a violin plot for training and testing data separately.

Table 2
Overview of the results of the model performance when using a reduced batch matrix in combination with time series data (Strategy C) compared to the model using all batch features (Strategy B) and the SPA model using only batch data without time series data (Strategy A) in terms of mean RMSE and variance (Var).

Included batch feature numbers	Approach	RMSE Train/ Test	Var Train/Test
Batch data only (Strategy A)	PLS	0.441/0.408	0.00085/ 0.00612
All batch features (Strategy B)	2	0.403/0.442	0.00099/ 0.00491
2, 4–7, 18, 45 (Strategy C1)	2	0.461/0.461	0.00204/ 0.00897
2, 4–8, 18, 21, 45, 145 (Strategy C2)	2	0.440/0.467	0.00404/ 0.00837
4, 5, 7 (Strategy C3)	2	0.461/0.461	0.0203/ 0.00897

different Approaches 6.1–6.4 perform similarly because the additional feature created from the time series data is assigned a weight that is either 0 or very close to 0. One example for Approach 6, specifically Approach 6.3, is shown in Fig. 7b.

In this case, smart process analytics software is again used to make the final prediction, and the algorithms ALVEN, SVR, RF, PLS, SPLS, and EN are applied. The best performing algorithm is PLS with an average RMSE of 0.407 and a variance of 0.00609. This result is comparable to the result achieved when only using the batch dataset but shows somewhat higher variance. An overview of the results achieved by using the different approaches is shown in Table 3.

For the mAb#1 dataset, tensorial methods combining both batch and time series data, i.e., second- and third-order tensors, do not improve the predictions significantly compared to just using the batch data. A likely explanation is that batch-to-batch correlation is negligible, as evidenced by autocorrelation assessment, resulting in a lower potential benefit of using tensorial approaches.

4.3. Application of approaches 1 through 6 on mAb#2

The autocorrelation behavior of mAb#2 is analyzed in Section 4.1. In contrast to mAb#1, significant batch-to-batch autocorrelation was detected for mAb#2. Consequently, the tensorial approaches are expected to use these batch-to-batch effects for mAb#2 to yield better predictions. As a result, we also analyze additional strategies exploring the effects of autoregressive terms and reduced batch matrices (Strategies E and F). The six strategies assessed for mAb#2 and their relationship to the approaches are described in Table 4.

Similar to mAb#1, a baseline case using only batch data with the algorithms ALVEN, SVR, RF, PLS, SPLS, and EN are analyzed (Strategy A). Approaches 1 through 5 are also applied to the mAb#2 dataset containing both batch data and time series data (Strategy B). The same cross-validation procedure with 15 outer folds, 5 inter folds, and 10 repeats and a log transformation on the output variables is applied. These results compared to the base case are shown in Fig. 8 in the form of a violin plot.

The best performing method when using only batch data was identified to be RF based on yield and variance, with an RMSE mean and variance of 0.0407 and 1.63×10^{-5}, respectively (Strategy A). However, the training performance of the RF model is significantly better than the testing performance, suggesting some overfitting. Similar to mAb#1, the tensorial methods for mAb#2 show different performances. The methods utilizing only time series data (Approaches 4 and 5) do not perform well but show similar performance for training and testing data. Approaches 1–3 show significant overfitting but achieve lower testing RMSEs. The best performing method based on testing RMSEs is Approach 3 with a mean RMSE of 0.0499 and a variance of 6.09×10^{-5} (Strategy B). Both the mean and especially the variance are worse for this prediction when compared to using only batch data, and this observation could be due to a larger number of variables in the batch matrix.

Similar to mAb#1, reduced models which consider only the most predictive batch features are evaluated (Strategy C). The most important features for the batch data are shown in Table 5 and have been determined by applying an SPLS approach to the batch data. The results of a new approach using a reduced batch matrix containing only the features shown in Table 5 are illustrated in Fig. 9.

For the mAb#2 dataset, better prediction performance is obtained using the tensorial methods with the reduced batch matrix. The best performing method is Approach 3 with a mean RMSE of 0.0356 and a variance of 4.07×10^{-5} (Strategy C). This result is significantly better than the basic tensorial approaches and even shows a lower RMSE than the SPA approach using only batch data. Approach 2 also performs well, but results in a slightly higher mean RMSE and variance. Both Approaches 2 and 3 show significantly reduced overfitting compared to the basic tensorial Approaches 1–5 without a reduced batch data matrix[3].

For mAb#2, Approach 6 can be applied by first using the time series data to predict the CQA and then using this prediction or the difference of the predictions as inputs to the smart process analytics software for predictive modeling (Strategy D). The results from Approaches 6.1 and 6.3 are shown in the left and right plots, respectively, in Fig. 10.

[3] If we only use the reduced batch data, performance does not vary significantly from the base case.

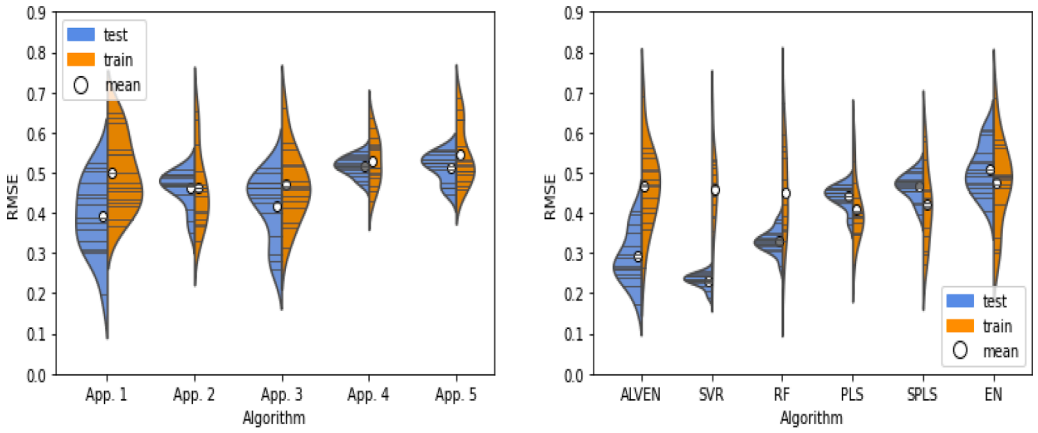

Fig. 7. Comparison of the RMSE results utilizing Approaches 1–5 when using a reduced batch data matrix (Strategy C) as an input on the left and using the prediction methods ALVEN, SVR, RF, PLS, SPLS, and EN for Approach 6.3 on the right side (Strategy D). The results are shown for 15 outer folds in the form of a violin plot for training and testing data separately.

Table 3
Summary of the prediction performance of the different models compared to the base case using only batch data for mAb#1. The Method column highlights the best approach (App.) or Algorithm.

Strategies	Method	RMSE Train/ Test	Var Train/Test
Batch only (Strategy A)	PLS	0.441/0.408	0.00085/ 0.00612
Tensorial 1–5 (Strategy B)	App. 2	0.403/0.442	0.00099/ 0.00491
Tensorial 1–5 Red. (Strategy C)	Red. based on EN	0.461/0.461	0.00204/ 0.00897
Tensorial 6 (Strategy D)	App. 6.3 PLS	0.441/0.407	0.00083/ 0.00609

Table 4
Overview of Strategies A–D and their relationship to prior introduced approaches.

Strategy	Strategy Description	Relationship Approach
A	Base case for comparison using only second-order tensorial data ($\mathbf{X}_{batch}$)	None
B	Approaches 1–5 applied to all second- and third-order tensorial data ($\mathbf{X}_{batch}, \underline{\mathbf{X}}_{time\ series}$)	Approaches 1–5
C	Approaches 1–5 applied to reduced second-order tensorial data and the full third-order tensorial data. The reduced feature set is based on the most predictive features ($\mathbf{X}_{batch,red}, \underline{\mathbf{X}}_{time\ series}$)	Approaches 1–5
D	Approaches 6.1–6.4 applied to all second- and third-order tensorial data ($\mathbf{X}_{batch}, \underline{\mathbf{X}}_{time\ series}$)	Approaches 6.1–6.4
E	Approaches 6.1–6.4 applied to reduced second-order tensorial data and the full third-order tensorial data ($\mathbf{X}_{batch,red}, \underline{\mathbf{X}}_{time\ series}$)	Approaches 6.1–6.4
F	Approaches 6.1–6.4 applied to reduced second-order tensorial data and the full third-order tensorial data; additionally, an autoregressive term is added to the second-order tensorial data to better capture batch-to-batch effects ($\mathbf{X}_{batch,red}, \underline{\mathbf{X}}_{time\ series}$)	Approaches 6.1–6.4

Approach 6.2 performs very similarly to Approach 6.1, while Approach 6.4 performs very similarly to 6.3.

Approach 6.1 performs almost identically to the baseline which only considers batch data. The best algorithm is also RF, with mean RMSE and variance of 0.0407 and 1.63×10^{-5}, respectively. The reason for the similarity is that the newly added feature is assigned a weight that is either 0 or very close to 0, so has negligible influence on the results for RF. This observation is similar for the other algorithms.

Approach 6.3 shows a different performance, particularly for the ALVEN algorithm. In this case, six folds have similar RMSEs to the other algorithms, whereas 9 out of 15 folds show a greatly improved prediction performance. Overall, a mean RMSE of 0.0194 and a variance of 2.64×10^{-4} are achieved, demonstrating that the prediction difference is a feature that can be leveraged by the ALVEN algorithm (Strategy D). This case can occur, for example, if the information contained by the batch and the time series data yield similar prediction models. Knowing the error of the first model can help improve the second model significantly. Additionally, ALVEN has a unique way of introducing nonlinearities, which explains why other algorithms cannot utilize the additional prediction difference feature as well. Referencing the violin plot, one notices that the violin for Approach 6.3 with ALVEN goes lower than 0. Referencing the violin plot, one notices that the violin for Approach 6.3 with ALVEN goes lower than 0, which is an artefact of the statistics-based violin plot applied to non-negative, high-variance RMSE values near zero. However, negative RMSE values are infeasible and all real values are larger than 0. Lastly, when comparing the similar training and the testing performance for Approach 6.3 with ALVEN, there is less overfitting when compared to the approach using only batch data.

Better results can be achieved for mAb#2 when using a reduced batch dataset, as shown in Fig. 9. This technique can also be combined with Approach 6 by only using a reduced batch data matrix in addition to the CQA predictions or prediction differences (Strategy E). Fig. 11a shows the prediction results when using a reduced batch matrix. For this method, we note a low RMSE of 0.0394 and a low variance of 1.55×10^{-5} for PLS. However, the best performing method is still ALVEN, which achieved an even lower mean RMSE of 0.0181 and a variance of 1.44×10^{-4} (Strategy E). Both PLS and ALVEN also show similar behavior for training and testing, indicating no overfitting.

Since significant autocorrelation was discovered for the predicted variable of mAb#2, a different approach can be applied that further leverages the autocorrelation for prediction. For this approach, past values of the predicted variable are appended as additional inputs to the batch matrix. This method is called an autoregressive (AR) model and allows the user to account for past realizations of the predicted variable Lewis and Reinsel (1985); Wei (2013). Since the AR model adds another input to the batch matrix and the tensorial methods did not perform well when handling too many batch variables, this feature is combined with the reduced batch data matrix (Strategy F). The results of this are shown in Fig. 11b.

For this approach, PLS does not perform as well as it did in the prior

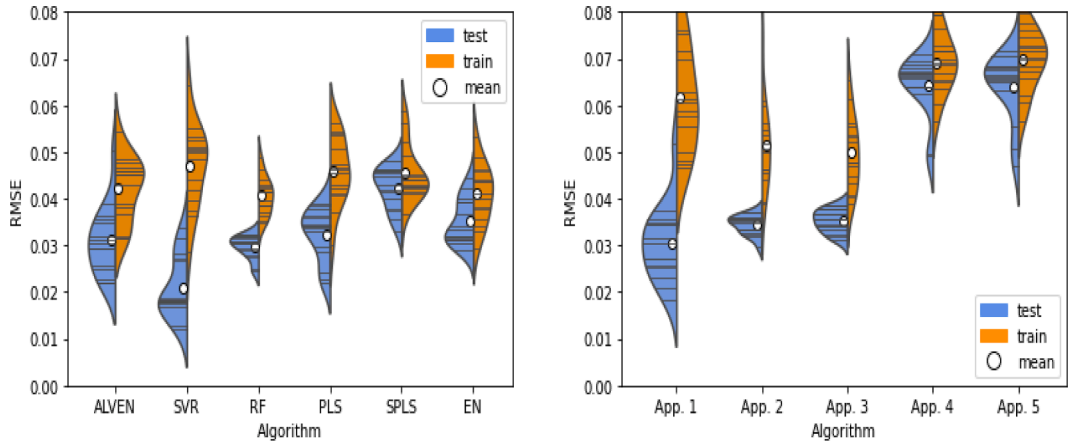

Fig. 8. For the mAb#2 dataset, RMSE results utilizing only batch data are shown in the left plot (Strategy A) as compared against tensorial Approaches 1–5 using both batch and time-series datasets in the right plot (Strategy B). The results are shown for 15 outer folds in the form of a violin plot for training and testing data separately.

Table 5
Overview of the most predictive features from the batch data for mAb#2. The feature number describes the number assigned to them in the initial feature set and is used as a reference.

Feature Number	Feature description
x_{81}	Production Bioreactor Day 4 Osmolality
x_{100}	Column 1 Product Monomer %
x_{115}	Column 2 Pool Volume
x_{117}	Column 2 Product High-Molecular Weight %

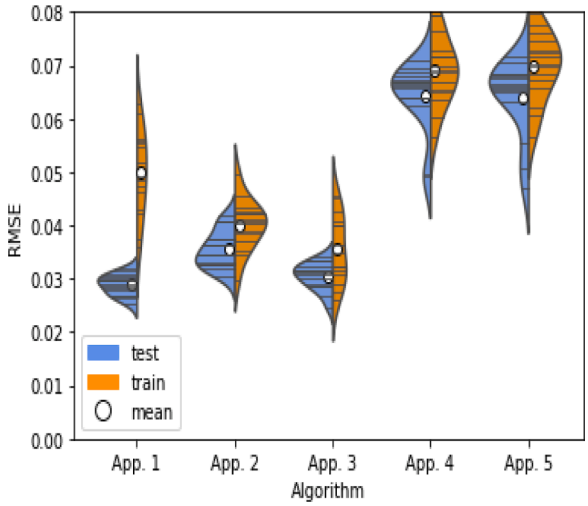

Fig. 9. RMSE results of Approaches 1–5 when using a reduced batch data matrix as an input for mAb#2 (Strategy C). The results are shown for 15 outer folds in the form of a violin plot for training and testing data separately.

model that did not have the autoregressive term, but the performance of the best method ALVEN further improves. A mean RMSE of 0.0168 and a variance of 6.43×10^{-5} can be achieved for the testing data, with similar results for the training data (Strategy F). This model performance is the best, and significantly improves upon the models using only batch or only time series data, without significant overfitting.

An overview of the different approaches used for the CQA prediction for mAb#2 and their results is shown in Table 6 and Fig. 12. The RMSEs improve significantly from Strategy A-D1 to D2-F. Additionally, the variance is also significantly reduced from D2-F, making Strategy F the best outcome.

To further understand why ALVEN shows such a good performance, we analyzed the weights assigned by the ALVEN algorithm to the different input features or nonlinear transformations of the nonlinear features. Overall, we only have 6 basic input features for this model, consisting of features x_7, x_{58}, x_{81}, x_{115} (see Table 5), the autoregressive term, and the prediction difference variable. These weights differ for all of the 15 different folds, but the average weight assigned to the features or their nonlinear transformation can be analyzed. Additionally, 4 folds yield higher testing errors (RMSEs between 0.022 and 0.040) and 11 folds yield lower testing errors (RMSEs between 0.008 and 0.018). Consequently, these folds are analyzed separately to see whether there are any obvious differences between the variables. An overview of the average weights for each of the considered features is in Table 7.

When looking at the CQADiff feature as constructed for Approaches 6.3 and 6.4, the nonlinear transformation $\sqrt{\text{CQADiff}}$ has a low average weight. However, the variable itself x_{CQADiff} is assigned an average weight of -0.836. If the tensorial prediction model and the batch prediction model were identical, a weight of -1 would be the ideal solution. Assume that the prediction of the first time series model is off by an average value of a. In this case, the variable x_{CQADiff} would average the value a. If the second model performed identically to the first model, but with the additional input term x_{CQADiff} with a weight of -1, the second model would now yield an average error of $a + (-1) \times a = 0$. As such, a weight near -1 suggests that the time series and the subsequent batch model contain similar information and consequently show similar performance.

The assigned weight between the high and the low RMSE cases differ. For the high RMSE cases, the weight is only -0.360, whereas the weight is -1.009 for the low RMSE cases. This further shows that a weight closer to -1 yields better performances because the time series and the batch data contain similar information and result in similarly performing models.

The second variable with high weights for its nonlinear transformations and its basic form is the Column 2 Volume x_{115}. Here big differences are seen between the low and the high RMSE case. In the high RMSE case, x_{115} and its nonlinear transformations $\sqrt{x_{115}}$ and $\ln x_{115}$ show lower weights, with x_{115} being the main factor with a weight of -0.891. For the low RMSE case, $\ln x_{115}$ shows an average weight of -2.520, $\sqrt{x_{115}}$ has an average weight of 0.575, and x_{115} has an average weight of 1.968. In this case, the basic feature and the nonlinear transformations seem to achieve an equilibrium with high positive and negative weights of the Column 2 Volume. This balance of linear and nonlinear capabilities can only be achieved when using ALVEN. Among the other algorithms, PLS, SPLS, and EN do not consider nonlinearities at all; meanwhile, SVR and RF have limited nonlinear capabilities and

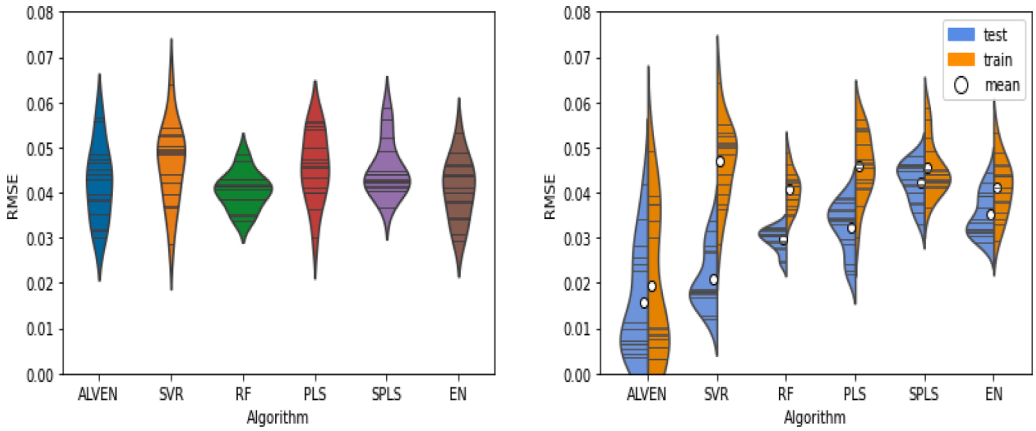

Fig. 10. RMSE results for using Approach 6.1 on the left and 6.3 on the right for mAb#2 (Strategy D). The results are shown for 15 outer folds in the form of a violin plot for training and testing data separately.

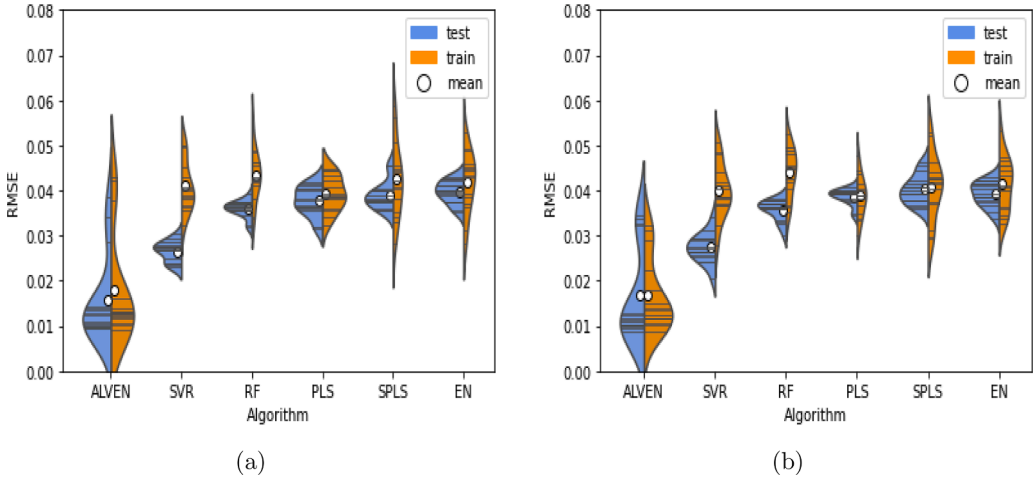

Fig. 11. Comparison of the RMSE results utilizing Approach 6.4 when using a reduced batch data matrix on the left (Strategy E) and when using an additional autoregressive term on the right (Strategy F). The results are shown for 15 outer folds in the form of a violin plot for training and testing data separately.

Table 6
Summary of the prediction performance of the different models compared to the base case only utilizing batch data for mAb#2. The Method column highlights the best approach (App.) or Algorithm.

Strategies	Method	RMSE Train/ Test	Var Train/Test
Batch only (Strategy A)	RF	0.0297/ 0.0407	$6.29 \times 10^{-6}/1.63 \times 10^{-5}$
Tensorial 1–5 (Strategy B)	App. 3	0.0352/ 0.0499	$5.89 \times 10^{-6}/6.09 \times 10^{-5}$
Tensorial 1–5 Red. (Strategy C)	App. 3	0.0305/ 0.0356	$6.62 \times 10^{-6}/4.07 \times 10^{-5}$
Tensorial 6.1 (Strategy D1)	RF	0.0296/ 0.0405	$6.67 \times 10^{-6}/1.68 \times 10^{-5}$
Tensorial 6.3 (Strategy D2)	ALVEN	0.0159/ 0.0194	$1.54 \times 10^{-4}/2.64 \times 10^{-4}$
Tensorial 6.4 Red. (Strategy E)	ALVEN	0.0159/ 0.0181	$9.16 \times 10^{-5}/1.44 \times 10^{-4}$
Tensorial 6.4 Red. Auto (Strategy F)	ALVEN	0.0168/ 0.0168	$1.08 \times 10^{-4}/6.43 \times 10^{-5}$

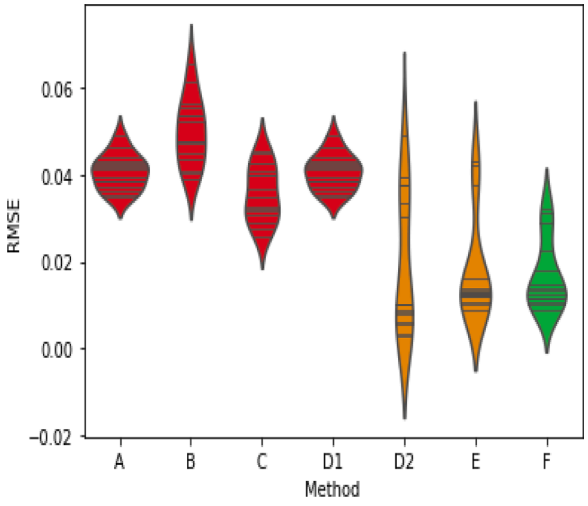

Fig. 12. Summary of the prediction performance of the different strategies compared to the base case only using batch data for mAb#2. Strategy F is the best performing strategy with a mean RMSE of 0.0168 and a variance of 6.43×10^{-5}.

cannot simultaneously consider different linear and nonlinear transformations.

The best results are achieved for mAb#2 when using Approach 6.4 in combination with a reduced batch data matrix and an additional

Table 7
Average weights assigned to each of the ALVEN features for all folds and for the low RMSE and high RMSE folds separately.

Feature Number	Weight	Weight Low RMSE	Weight High RMSE
$1/x_{117}$	0.017	− 0.045	0.188
$1/x_{\text{auto}}$	− 0.057	− 0.060	− 0.048
$\ln x_{115}$	− 1.719	− 2.520	0.484
$\ln x_{\text{auto}}$	− 0.050	− 0.044	− 0.069
$\sqrt{x_{81}}$	− 0.0222	− 0.016	− 0.040
$\sqrt{x_{100}}$	− 0.004	− 0.005	0
$\sqrt{x_{115}}$	0.421	0.575	0
$\sqrt{x_{117}}$	− 0.001	− 0.054	0.146
$\sqrt{\text{auto}}$	− 0.086	− 0.117	0.001
$\sqrt{\text{CQADiff}}$	− 0.004	− 0.005	0
x_{81}	− 0.016	− 0.011	− 0.030
x_{100}	− 0.069	− 0.067	− 0.077
x_{115}	1.207	1.968	− 0.891
x_{117}	0.034	− 0.020	0.183
x_{auto}	− 0.111	0.100	0.142
x_{CQADiff}	− 0.836	− 1.009	− 0.360

autoregressive term to capture the dynamic effects in mAb#2. Approach 6.4 first uses tensorial methods based only on time series data to predict the CQA, and then applies different second-order tensor methods to the batch data and the prediction difference that are suitable given the present characteristics of multicollinearity and nonlinearity in the dataset. From these second-order tensor methods, ALVEN performed best by leveraging the prediction difference and allowing for the consideration of different types of linear and nonlinear transformations of key features.

In summary, different strategies and approaches to combine second- and third-order tensorial data are proposed and evaluated based on the application to two monoclonal antibody production processes. Incorporating thid-order tensorial or timeseries data into the model-building process can result in significant improvement in prediction accuracy if the predicted variable shows strong batch-to-batch correlations as shown in the case of mAb#2. However, the effect is significantly smaller if those batch-to-batch correlations are insignificant as for mAb#1. In this case, the additional time dimension does not improve the predictions significantly as the additional time resolution adds less value.

5. Conclusions

For the objective of model prediction, multiple tensorial approaches are proposed to handle combinations of both second- and third-order tensorial data. Subsequently, these different approaches are applied to two different monoclonal antibody production processes in order to predict the relevant CQAs. The mAb#1 CQA shows no significant batch-to-batch correlation whereas the mAb#2 CQA shows significant batch-to-batch correlation for up to 3 lags as quantified by autocorrelation. Comparing the performance of the proposed approaches to a base case considering only batch data modeled using a smart process analytics prediction software, applying autocorrelation in the predicted variables was a useful guide as to whether to use third-order tensorial methods. mAb#1 did not benefit from the proposed approaches due to its lack of significant batch-to-batch correlation. Consequently, no significant improvements can be observed compared to the base case. However, the prediction for mAb#2, which shows significant batch-to-batch correlation, can be substantially improved, yielding a mean RMSE of 0.0168 compared to 0.0405 in the base case. This improvement shows the strong potential of using third-order tensorial methods for data that have significant batch-to-batch correlation. Consequently, autocorrelation analysis is a useful prior analysis to apply before applying third-order tensorial methods, as well as the proposed methods that combine second- and third-order tensorial data. Additionally, we recommend considering the inclusion of an autoregressive term to further extract information from the batch-to-batch effects for datasets in which the autocorrelation is observed for the predicted variable.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This study is supported by the U.S. Food and Drug Administration, Contract No. 75F40121C00090. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the financial sponsor.

References

Andersson, C.A., Bro, R., 2000. The N-way toolbox for MATLAB. Chemometric. Intell. Lab. Syst. 52 (1), 1–4.
Bro, R., 1996. Multiway calibration. multilinear PLS. J. Chemometric. 10 (1), 47–61.
Bro, R., Rinnan, Å., Faber, N.K.M., 2005. Standard error of prediction for multilinear PLS: 2. Practical implementation in fluorescence spectroscopy. Chemometric. Intell. Lab. Syst. 75 (1), 69–76.
Chiang, L.H., Russell, E.L., Braatz, R.D., 2000. Fault Detection and Diagnosis in Industrial Systems. Springer Science & Business Media, London, UK.
Cutler, C.R., Ramaker, B.L., 1980. Dynamic Matrix Control – a computer control algorithm. J. Environ. Sci. Health, Part B: Pesticides Food Contaminants Agric. Wastes 1 (17), 72.
Evaluate, July 2021. World Preview 2022: Overview to 2026, 14th. Evaluate Ltd., London, UK.
Farid, S.S., 2007. Process economics of industrial monoclonal antibody manufacture. J. Chromatogr. B 848 (1), 8–18.
FDA, 2004. Guidance for Industry: PAT–A Framework for Innovative Pharmaceutical Development, Manufacturing, and Quality Assurance. U.S. Food and Drug Administration, Rockville, MD.
Hong, M.S., Mohr, F., Castro, C.D., Smith, B.T., Wolfrum, J.M., Springs, S.L., Sinskey, A. J., Hart, R.A., Mistretta, T., Braatz, R.D., 2023. Smart process analytics for the end-to-end batch manufacturing of monoclonal antibodies. Comput. Chem. Eng. 179, 108445.
Hong, M.S., Severson, K.A., Jiang, M., Lu, A.E., Love, J.C., Braatz, R.D., 2018. Challenges and opportunities in biopharmaceutical manufacturing control. Comput. Chem. Eng. 110, 106–114.
Hu, K., Yuan, J., 2009. Batch process monitoring with tensor factorization. J. Process Control 19 (2), 288–296.
Jiao, J., Yu, H., Wang, G., 2015. A quality-related fault detection approach based on dynamic least squares for process monitoring. IEEE Trans. Ind. Electron. 63 (4), 2625–2632.
Lewis, R., Reinsel, G.C., 1985. Prediction of multivariate time series by autoregressive model fitting. J. Multivariate Anal. 16 (3), 393–411.
Lloyd, I., 2021. Pharma R&D Annual Review 2021. Informa PLC, London, UK.
Lopez-Fornieles, E., Brunel, G., Rancon, F., Gaci, B., Metz, M., Devaux, N., Taylor, J., Tisseyre, B., Roger, J.-M., 2022. Potential of multiway PLS (N-PLS) regression method to analyse time-series of multispectral images: a case study in agriculture. Remote Sens. 14 (1), 216.
Luo, L., Bao, S., Gao, Z., 2015. Quality prediction based on HOPLS-CP for batch processes. Chemometric. Intell. Lab. Syst. 143, 28–39.
Luo, L., Bao, S., Gao, Z., Yuan, J., 2013. Batch process monitoring with tensor globallocal structure analysis. Ind. Eng. Chem. Res. 52 (50), 18031–18042.
Luo, L., Bao, S., Gao, Z., Yuan, J., 2014. Batch process monitoring with GTucker2 model. Ind. Eng. Chem. Res. 53 (39), 15101–15110.
Luo, L., Bao, S., Mao, J., Tang, D., 2016. Quality prediction and quality-relevant monitoring with multilinear PLS for batch processes. Chemometric. Intell. Lab. Syst. 150, 9–22.
Maruthamuthu, M.K., Rudge, S.R., Ardekani, A.M., Ladisch, M.R., Verma, M.S., 2020. Process analytical technologies and data analytics for the manufacture of monoclonal antibodies. Trend. Biotechnol. 38 (10), 1169–1186.
Nikita, S., Thakur, G., Jesubalan, N.G., Kulkarni, A., Yezhuvath, V.B., Rathore, A.S., 2022. AI-ML applications in bioprocessing: ML as an enabler of real time quality prediction in continuous manufacturing of mAbs. Comput. Chem. Eng. 164, 107896.
Rathore, A.S., Kumar Singh, S., Pathak, M., Read, E.K., Brorson, K.A., Agarabi, C.D., Khan, M., 2015. Fermentanomics: Relating quality attributes of a monoclonal antibody to cell culture process variables and raw materials using multivariate data analysis. Biotechnol. Progr. 31 (6), 1586–1599.
Rosas, J.G., Blanco, M., González, J.M., Alcalà, M., 2012. Real-time determination of critical quality attributes using near-infrared spectroscopy: a contribution for Process Analytical Technology (PAT). Talanta 97, 163–170.
Salvador, S., Chan, P., 2007. Toward accurate dynamic time warping in linear time and space. Intell. Data Anal. 11 (5), 561–580.
Severson, K., VanAntwerp, J.G., Natarajan, V., Antoniou, C., Thömmes, J., Braatz, R.D., 2015. Elastic net with Monte Carlo sampling for data-based modeling in biopharmaceutical manufacturing facilities. Comput. Chem. Eng. 80, 30–36.

Severson, K.A., VanAntwerp, J.G., Natarajan, V., Antoniou, C., Thömmes, J., Braatz, R. D., 2018. A systematic approach to process data analytics in pharmaceutical manufacturing: the data analytics triangle and its application to the manufacturing of a monoclonal antibody. Multivariate Analysis in the Pharmaceutical Industry. Elsevier, pp. 295–312.
Shibata, R., 1976. Selection of the order of an autoregressive model by Akaike's information criterion. Biometrika 63 (1), 117–126.
Singh, S., Tank, N.K., Dwiwedi, P., Charan, J., Kaur, R., Sidhu, P., Chugh, V.K., 2018. Monoclonal antibodies: a review. Curr. Clin. Pharmacol. 13 (2), 85–99.
Stubbs, S., Zhang, J., Morris, J., 2018. Bioprocess performance monitoring using multiway interval partial least squares. Comput. Aided Chem. Eng. 41, 243–259.
Sun, W., Braatz, R.D., 2021. Smart process analytics for predictive modeling. Comput. Chem. Eng. 144, 107134.
Tsay, R.S., 1984. Order selection in nonstationary autoregressive models. Annal. Stat. 12 (4), 1425–1433.
Wasalathanthri, D.P., Feroz, H., Puri, N., Hung, J., Lane, G., Holstein, M., Chemmalil, L., Both, D., Ghose, S., Ding, J., 2020. Realtime monitoring of quality attributes by inline fourier transform infrared spectroscopic sensors at ultrafiltration and diafiltration of bioprocess. Biotechnol. Bioeng. 117 (12), 3766–3774.
Wei, W.W.S., 2013. Time series analysis. In: Little, T.D. (Ed.), The Oxford Handbook of Quantitative Methods in Psychology: Volume 2: Statistical Analysis. Oxford University Press, Oxford, UK, pp. 458–485.
Westerhuis, J.A., Kourti, T., MacGregor, J.F., 1999. Comparing alternative approaches for multivariate statistical analysis of batch process data. J. Chemometric. 13 (3-4), 397–413.
Wu, H.-L., Wang, T., Yu, R.-Q., 2020. Recent advances in chemical multi-way calibration with second-order or higher-order advantages: multilinear models, algorithms, related issues and applications. TrAC Trend. Anal. Chem. 130, 115954.