

Effect of data preprocessing and machine learning hyperparameters on mass spectrometry imaging models

Cite as: J. Vac. Sci. Technol. A 41, 063204 (2023); doi: 10.1116/6.0002788

Submitted: 26 April 2023 · Accepted: 22 August 2023 ·

Published Online: 20 September 2023



Wil Gardner,¹ David A. Winkler,^{2,3,4} David L. J. Alexander,⁵ Davide Ballabio,⁶ Benjamin W. Muir,⁷ and Paul J. Pigram^{1,a)}

AFFILIATIONS

¹Centre for Materials and Surface Science and Department of Mathematical and Physical Sciences, La Trobe University, Melbourne, Victoria 3086, Australia

²La Trobe Institute for Molecular Sciences, La Trobe University, Melbourne, Victoria 3086, Australia

³Monash Institute of Pharmaceutical Sciences, Monash University, Parkville, Victoria 3052, Australia

⁴Advanced Materials and Healthcare Technologies, School of Pharmacy, University of Nottingham, Nottingham NG7 2RD, United Kingdom

⁵CSIRO Data61, Clayton, Victoria 3168, Australia

⁶Milano Chemometrics and QSAR Research Group, Department of Earth and Environmental Sciences, University of Milano-Bicocca, Piazza della Scienza 1, Milano 20126, Italy

⁷CSIRO Manufacturing, Clayton, Victoria 3168, Australia

Note: This paper is part of the Special Topic Collection: Reproducibility Challenges and Solutions II with a Focus on Surface and Interface Analysis.

^{a)}Author to whom correspondence should be addressed: p.pigram@latrobe.edu.au

ABSTRACT

The self-organizing map (SOM) is a nonlinear machine learning algorithm that is particularly well suited for visualizing and analyzing high-dimensional, hyperspectral time-of-flight secondary ion mass spectrometry (ToF-SIMS) imaging data. Previously, we compared the capabilities of the SOM with more traditional linear techniques using ToF-SIMS imaging data. Although SOMs perform well with minimal data preprocessing and negligible hyperparameter optimization, it is important to understand how different data preprocessing methods and hyperparameter settings influence the performance of SOMs. While these investigations have been reported outside of the ToF-SIMS field, no such study has been reported for hyperspectral MSI data. To address this, we used two labeled ToF-SIMS imaging datasets, one of which was a polymer microarray dataset, while the other was semisynthetic hyperspectral data. The latter was generated using a novel algorithm that we describe here. A grid-search was used to evaluate which data preprocessing methods and SOM hyperparameters had the largest impact on the performance of the SOM. This was assessed using multiple linear regression, whereby performance metrics were regressed onto each variable defining the preprocessing-hyperparameter space. We found that preprocessing was generally more important than hyperparameter selection. We also found statistically significant interactions between several parameters studied, suggesting a complex interplay between preprocessing and hyperparameter selection. Importantly, we identified interesting trends, both dataset specific and dataset agnostic, which we describe and discuss in detail.

© 2023 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1116/6.0002788>

22 September 2023 00:17:56

I. INTRODUCTION

The self-organizing map (SOM) was first described by Kohonen¹ as a tool for visualizing and interpreting the topology of a high-dimensional dataset. The SOM is a type of artificial neural network (ANN) that uses unsupervised training of a (typically) 2D interconnected network of neurons to produce a low-dimensional topological map of the dataset. Detailed descriptions of the SOM have been published elsewhere.^{1–4}

Our group has demonstrated the utility of the SOM for the analysis of time-of-flight secondary ion mass spectrometry (ToF-SIMS) data.^{4–14} ToF-SIMS is an analytical technique for analyzing surface chemistry with nanometer depth resolution and sub-micrometer spatial resolution, depending on instrument design and parameters. ToF-SIMS data are hyperspectral because an entire mass spectrum is associated with every pixel in the scan area. Such rich datasets provide enormous analytical potential. However, this potential is hampered by the complexity and size of the data.

More recently, we developed a way of using SOMs to visualize hyperspectral ToF-SIMS images.^{4,6} Conspicuously, by incorporating the relational perspective map (RPM),¹⁵ we have demonstrated the power and robustness of SOMs for generating accurate models of ToF-SIMS images, including both 2D⁷ and 3D¹⁴ hyperspectral images.

Despite these successes, we have not formally investigated how data preprocessing, such as scaling and/or normalization, commonly used in the analysis of ToF-SIMS data^{16–18} and SOM hyperparameter selection, affects performance. Here, we use a grid-search approach to address this deficiency, identifying which preprocessing steps and hyperparameters have the most impact on SOM performance, based on a range of metrics. As part of the preprocessing search space, we also include feature extraction (FE) using a convolutional autoencoder (CNAE) that we have previously applied to ToF-SIMS data.¹⁹ We opted to apply the CNAE, rather than other FE methods commonly applied to ToF-SIMS data, based on its demonstrated efficacy in our study.

We quantify the impacts of preprocessing methods and hyperparameters on SOM performance using multiple linear regression for two ToF-SIMS datasets. Although we use an unsupervised SOM, both datasets are labeled, providing a more informative and accurate analysis of data preprocessing and SOM hyperparameter selection. The first dataset is a hyperspectral image of a polymer microarray previously analyzed for other purposes.^{6,20} It was chosen because it contained ground truth information, in that each pixel in the hyperspectral image could be assigned to one of the 70 polymers in the microarray. The second dataset was generated by combining independent ToF-SIMS images acquired from seven different nylon polymers, using a novel algorithm described below. This algorithm generates labeled ToF-SIMS datasets with specific levels of spectral mixing and spatial autocorrelation derived from real acquired data (denoted as the semisynthetic data here). In the broader machine learning (ML) literature, semisynthetic data are routinely generated to augment real data and to enhance or investigate ML performance. For example, in medical imaging, semisynthetic images have been used to improve computer vision-based classification performance.^{21–23} These semisynthetic datasets are meaningfully distinct from purely synthetic data, and it is this

distinction that enables them to improve classification accuracy (in these examples). In our case, the algorithm we developed enables the generation of spatially well-characterized ToF-SIMS images, with the benefit of maintaining the properties (noise, instrument effects, etc.) of real data. This presents a valuable methodology for testing the performance of so-called spatially aware ML algorithms (such as the CNAE¹⁹ and spatial k-means²⁴), which consider spatial relationships between pixels.

An important caveat of this study is that, while we explore data preprocessing and SOM hyperparameter selection together, improved SOM performance should not be conflated with the general superiority (or not) of any given preprocessing pipeline. Indeed, we explicitly warn against making such generalizations. Rather, this study is intended to demonstrate why careful consideration of data preprocessing steps is important, while also showing that preprocessing and SOM hyperparameter selection are not independent. Given this, we discuss ways in which the study outcomes may be valuable more generally in the paper.

II. EXPERIMENT

A. Microarray printing and ToF-SIMS

Polymer microarray printing and ToF-SIMS experimental details for the sample studied have been described previously.^{6,20} Briefly, the microarray comprised 70 unique polymer spots printed onto a poly(hydroxy ethylmethacrylate)-coated slide.^{6,20}

ToF-SIMS data were acquired using an IONTOF TOF.SIMS 4 instrument. An analysis area of $9.2 \times 9.2 \text{ mm}^2$ (with a pixel size of $10 \times 10 \mu\text{m}^2$) was scanned using a stage raster with 25 keV Bi_3^+ primary ion beams and a negative ion detection mode. A low-energy electron flood gun was employed to counteract sample charging.

Peaks were automatically detected in the data using a count threshold of >100 counts using the SurfaceLab6 peak search function. A total of 717 peaks were identified and selected in this way, the summed intensities of which then constituted the hyperspectral dataset. From this dataset, only pixels from within the polymer dots were analyzed, which were selected by drawing elliptical regions of interest based on the total ion count (TIC) image. This resulted in a total of 52 440 pixels being included in the analysis.

B. Nylon sample preparation and ToF-SIMS

Nylon sample preparation and ToF-SIMS experimental details have been described previously.⁸ Briefly, seven chemically similar but distinct nylon (polyamide) materials were supplied in the pellet form, which were cut with a scalpel blade to expose a clean, flat surface. Samples were secured to the ToF-SIMS mount using a double-sided tape.

ToF-SIMS data were acquired using an IONTOF TOF.SIMS 5 instrument, using pulsed 30 keV Bi_3^+ primary ions in bunched mode. A range of images were collected using positive and negative polarities, covering $100 \times 100 \mu\text{m}$ analysis areas at 128×128 pixels; however, only a single positive image from each nylon type was used in this study. A low-energy electron flood gun was used to counteract charging.

22 September 2023 00:17:56

Data were binned using 0.1 m/z mass intervals over the range of 1–300 m/z. The summed intensities of these intervals for each pixel then constituted the hyperspectral dataset.

C. Semisynthetic hyperspectral data

The algorithm developed is designed to generate a semisynthetic ToF-SIMS data cube by mixing C real ToF-SIMS data cubes, each corresponding to one of the C unique classes. Let $\mathbf{X} \in \mathbb{R}_0^{+h \times w \times p \times C}$ be a concatenation of C such real data cubes, where h and w represent the two spatial dimensions and p represents the spectral dimension (either the number of m/z channels/bins or the number of mass peaks). We seek to generate a semisynthetic ToF-SIMS data cube, $\tilde{\mathbf{X}} \in \mathbb{R}_0^{+h \times w \times p}$, by mixing the data from \mathbf{X} using a class membership array, $\mathbf{M} \in [0, 1]^{h \times w \times C}$.

Formally, we calculate the spectrum of the pixel at spatial coordinates (i, j) in $\tilde{\mathbf{X}}$ as

$$\tilde{\mathbf{X}}_{ij,*} = \sum_{c=1}^C \mathbf{M}_{ij,c} \cdot \mathbf{X}_{ij,*c}, \quad (1)$$

$$i \in [1, h] \text{ and } j \in [1, w],$$

where $\mathbf{M}_{ij,c}$ is the (scalar) membership fraction of class c for the pixel and $\mathbf{X}_{ij,*c}$ is the corresponding (vector) pixel spectrum from the k th data cube in \mathbf{X} . Note also that Eq. (1) could equally be replaced by a nonlinear mixing function. For example, such a function could be designed to model ion suppression or enhancement between interacting classes due to matrix effects, in the case of ToF-SIMS data.

In addition to \mathbf{X} , we also need to calculate a suitable \mathbf{M} . The algorithm we used to generate \mathbf{M} is outlined below and is separated into two distinct phases. This algorithm and the corresponding phases are also detailed in Fig. S1 in the supplementary material.³⁷ Briefly, phase 1 iteratively adds to \mathbf{M} by randomly selecting pixel coordinates from a spatially uniform probability distribution. Phase 2, in contrast, iteratively adds to \mathbf{M} by assigning pixels that are close together to the same class, thereby increasing the spatial autocorrelation of the class assignments.

Formally, in phase 1 (Fig. S1A),³⁷ we initialize $\mathbf{M}^0 = [\mathbf{O}]_{h \times w \times C}$ and $\mathbf{A}^0 = [\mathbf{1}]_{h \times w}$, where $[\mathbf{O}]_{h \times w \times C}$ and $[\mathbf{1}]_{h \times w}$ are arrays of all zeros and all ones of size $h \times w \times C$ and $h \times w$, respectively. We define \mathbf{A} as the pixel availability matrix. We then iteratively select a set of spatial pixel coordinates, (i, j) , and a class index, c , where $i \in [1, h]$, $j \in [1, w]$, and $c \in [1, C]$. This is done by first randomly drawing c from a 1D uniform distribution along the class dimension. We then draw (i, j) from a 2D probability distribution along the spatial dimensions. At iteration t and for class c , this distribution is given by the matrix

$$\mathbf{P}_c^t = \text{Norm}(\mathbf{A}^{t-1}), \quad (2)$$

where $\text{Norm}(\cdot)$ returns the input matrix normalized to unity. Note that \mathbf{P}_c^t is a uniform distribution across the available pixels that are represented as ones in \mathbf{A}^{t-1} .

We then add a 2D Gaussian distribution (neglecting the standardizing constant, which is not needed as we normalize later)

with standard deviation σ to \mathbf{M}_c^{t-1} , centered around the pixel at (i, j) . That is, at iteration t , we set

$$\mathbf{M}_{i',j',c}^t = \mathbf{M}_{i',j',c}^{t-1} + \exp\left(-\frac{(i'-i)^2 + (j'-j)^2}{2\sigma^2}\right), \quad (3)$$

$$i' \in [1, h] \text{ and } j' \in [1, w].$$

Note that to reduce the computational complexity in practice, we only calculate Eq. (3) for those (i', j') within some neighborhood of (i, j) , outside of which $\mathbf{M}_{i',j',c}^t \approx \mathbf{M}_{i',j',c}^{t-1}$ according to Eq. (3).

In addition, we set

$$\mathbf{A}_{i',j'}^t = \begin{cases} 0 & \text{if } (i', j') = (i, j) \\ \mathbf{A}_{i',j'}^{t-1} & \text{else} \end{cases} \quad (4)$$

$$i' \in [1, h] \text{ and } j' \in [1, w]$$

to remove the pixel from the availability matrix \mathbf{A} . We represent the total number of iterations in phase 1, and therefore, the total number of pixels selected in this phase as T_1 .

Phase 2 (see Fig. S1B)³⁷ is almost identical to phase 1; however, the selection of a set of spatial pixel coordinates, (i, j) , is modified to increase the spatial autocorrelation of the class membership maps in \mathbf{M} . Specifically, at iteration t , we first randomly select c as in phase 1. We then use the c th class membership map, \mathbf{M}_c^{t-1} , and the pixel availability matrix, \mathbf{A}^{t-1} , from the previous iteration to generate the 2D probability distribution matrix \mathbf{P}_c^t , from which we draw (i, j) . Specifically, \mathbf{P}_c^t is calculated as

$$\mathbf{P}_c^t = \text{Norm}\left(\mathbf{M}_c^{t-1} \odot \mathbf{A}^{t-1}\right), \quad (5)$$

where \odot represents the Hadamard product (i.e., element-wise multiplication). Note that the inclusion of \mathbf{M}_c^{t-1} is the only difference between Eq. (5) in phase 2 and Eq. (2) in phase 1.

We then use Eq. (3) to add a 2D Gaussian distribution to \mathbf{M}_c^{t-1} and Eq. (4) to remove the pixel from \mathbf{A}^{t-1} as in phase 1. Phase 2 is repeated until $\mathbf{A}^t = [\mathbf{O}]_{h \times w}$, i.e., until each pixel has been drawn once and only once. Finally, we normalize \mathbf{M} to unity along the class dimension, such that $\sum_{c=1}^C \mathbf{M}_{ij,c} = 1$ for all i, j .

Rather than selecting a single value for σ , for all 2D Gaussian distributions, we draw a new σ^t at each iteration from a Rayleigh distribution, which has a probability density function

$$p(x; \sigma_r) = \frac{x}{\sigma_r^2} \exp\left(-\frac{x^2}{2\sigma_r^2}\right), \quad x \geq 0, \quad (6)$$

where σ_r is the scale parameter. Drawing σ^t at each iteration using Eq. (6) introduces additional spatial complexity into \mathbf{M} .

Therefore, the algorithm outlined above is controlled by two key parameters: T_1 and σ_r . Together, these parameters control the level of spatial autocorrelation in the class membership maps in \mathbf{M} as well as the degree of mixing between classes within each pixel. Given that \mathbf{M} is used to calculate $\tilde{\mathbf{X}}$ [see Eq. (1) and Fig. S1C],³⁷ these parameters consequently control the degree of interclass

spectral mixing in $\tilde{\mathbf{X}}$. Additionally, under the assumption of high spatial autocorrelation in each of the C real ToF-SIMS datasets, they also control the autocorrelation of individual ion images in $\tilde{\mathbf{X}}$. Therefore, it is possible to consider a range of values for T_1 and σ_r to generate \mathbf{M} (and, therefore, $\tilde{\mathbf{X}}$) with different levels of spatial autocorrelation and spectral mixing (Fig. S2).³⁷

To quantify spatial autocorrelation of the maps in \mathbf{M} , we used Moran's I .²⁵ To quantify spectral mixing, we propose a spectral purity metric, SP , for a given \mathbf{M} , defined as

$$SP = \frac{\text{MeanMax}(\mathbf{M}) - \frac{1}{C}}{1 - \frac{1}{C}}, \quad (7)$$

where $\text{MeanMax}(\mathbf{M})$ returns the mean of the maximum membership fractions for each pixel in \mathbf{M} . Note that if \mathbf{M} is normalized to unity along the class dimension (as in our algorithm), SP is bounded between 0 and 1: when there is no spectral mixing (i.e., when \mathbf{M} contains only zeroes and ones), $\text{MeanMax}(\mathbf{M}) = 1$, and therefore, $SP = 1$. With increased spectral mixing, $\text{MeanMax}(\mathbf{M})$ and, therefore, SP are reduced. Under the condition of maximum spectral mixing, each element in \mathbf{M} will be equal to $1/C$, such that $\text{MeanMax}(\mathbf{M}) = 1/C$ and $SP = 0$.

D. Data preprocessing and SOM training

This study used a grid search to investigate the effects of a range of data preprocessing methods and SOM hyperparameters on the performance of the ToF-SIMS hyperspectral data models. Figure 1 summarizes the preprocessing methods and SOM hyperparameters that were investigated in the grid-search. The hyperspectral imaging data, after unfolding, were analyzed by several preprocessing pipelines (Fig. 1). These involved either no processing or normalization of each pixel to TIC, plus one of the following scaling methods: min-max scaling where ion images were scaled between 0 and 1; Poisson scaling where ion images were scaled by the square root of their mean to account for Poisson noise;^{26,27} or standardization (z-scaling) where images were mean-centered (except for when the data were encoded by the CNNAE, which

enforces nonnegativity) then scaled to unit standard deviation. Data were also analyzed without applying any scaling method, with and without normalization to TIC. After preprocessing, data were either analyzed directly or used to train a CNNAE, designed to extract latent features from a hyperspectral dataset, as has been described previously.¹⁹ We used an identical architecture (number of layers, size of convolutional filters, etc.) as described previously.¹⁹ We selected 100 latent features for the encoding, and the CNNAE was constructed using Tensorflow²⁸ (with GPU) with the Keras API²⁹ in PYTHON. In total, 16 different preprocessing pipelines were employed.

For each preprocessing pipeline, a range of SOMs were trained with various hyperparameters, as outlined in Fig. 1. These included: square or hexagonal topologies (i.e., 8 or 6 neighbors for each neuron); planar or toroidal boundaries; map sizes of 10×10 , 20×20 , 30×30 , or 40×40 neurons; and 1, 2, 4, 5, 8, 10, 20, 50, 100, 200, or 500 training epochs. This resulted in 2816 combinations of SOM hyperparameters and data preprocessing methods. Three replicate SOMs (with random weight initialization) were trained for each combination, resulting in a total of 8448 models.

All SOM models were constructed using the Kohonen and CP-ANN Toolbox for MATLAB, with GPU support.^{2,3} A Dell Precision 3650 Tower workstation was used for all calculations, with an Intel Xeon W-1390P processor, 128 GB RAM, and an NVIDIA Quadro RTX 5000 GPU. With this system and toolbox, the SOM training time was ~ 0.1 – 2 s per epoch (~ 50 – 1000 s for the 500 epoch models), depending on the SOM size and dataset. We note that the computation time was specific to the implementation itself, such that other SOM implementations (e.g., in Python) may exhibit slower or faster training times with the same settings.

E. SOM performance evaluation

We used three label-based performance metrics to quantify SOM performance: homogeneity (as part of the V-measure³⁰ metric); the Jaccard similarity index;³¹ and the class scatter index.³² We also employed one label-free metric, topographic error,³³ to compute SOM topology preservation.³⁴ For brevity, we only give a high-level overview of these metrics, although we provide a more thorough mathematical description of the V-measure score in the SI.³⁷

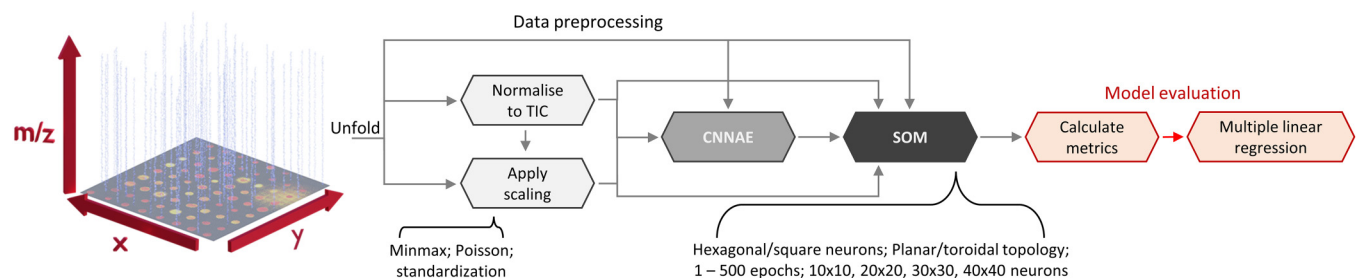


FIG. 1. Schematic showing the specifics of the grid-search used to evaluate the performance of the SOM for ToF-SIMS imaging data. Acquired hyperspectral ToF-SIMS data are first unfolded, passed through one of the several preprocessing pipelines, and then used to train a new SOM. Various hyperparameter combinations are used for different SOMs, as shown. Finally, each SOM model is evaluated using several external performance metrics, which are then used to construct a multiple linear regression model to evaluate the influence of each variable on SOM performance.

22 September 2023 00:17:56

V-measure is an entropy-based measure of the overall performance of clustering algorithms. It is defined as the weighted harmonic mean of the homogeneity and completeness scores. The completeness score is a measure of how effectively the clustering has assigned a class (in this case, the polymer type) to a single cluster (in this case, a neuron on the SOM). Inversely, the homogeneity score measures how effectively the clustering has assigned a cluster to a single class. Therefore, the V-measure score attempts to balance these two scores by using their harmonic mean. As will be discussed in Sec. III, the homogeneity score is more important than the completeness score for the SOM. This is because it is not necessarily undesirable for the SOM to assign multiple neurons to the same class, given its self-organizing and topology-preserving nature. As such, we only consider this score in our evaluations. Furthermore, to be consistent with other metrics used in our evaluation for which a smaller score is better, we convert the homogeneity, h , to what we call heterogeneity, given simply as $1 - h$. In this form, a heterogeneity of zero is considered ideal.

The Jaccard similarity index,³¹ $J(A, B)$, which we have used previously to evaluate SOM performance on the same microarray dataset,⁶ is a straightforward measure of how well two given classes, A and B , are distinguished. Specifically, the index is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \quad (8)$$

Note that the Jaccard index only measures similarity between pairs of classes. Hence, to evaluate the *overall* performance of the SOM for the entire set of classes, we calculated the *mean* Jaccard index for every pair of classes.

The class scatter index (CSI) was proposed specifically for the SOM³² and measures the mean number of clusters assigned to each class. For a given class c , neighboring neurons are considered part of the same cluster if they are associated with one or more samples (pixels) in class c . The CSI equates fewer clusters with better SOM performance, based on topology preservation.

Finally, we also calculated the topographic error,³³ TE , for each SOM. This is a label-free metric designed to measure SOM topology preservation. Mathematically, TE is given by

$$TE = \frac{1}{n} \sum_{i=1}^n t(\mathbf{x}_i, \mathbf{W}), \quad (9)$$

$$t(\mathbf{x}, \mathbf{W}) = \begin{cases} 0 & \text{if } \mu_1(\mathbf{x}, \mathbf{W}) \text{ and } \mu_2(\mathbf{x}, \mathbf{W}) \text{ are neighbors,} \\ 1 & \text{otherwise} \end{cases}$$

where \mathbf{x}_i is the i th pixel spectrum (after scaling), \mathbf{W} is the weights matrix of the SOM, and $\mu_1(\mathbf{x}, \mathbf{W})$ and $\mu_2(\mathbf{x}, \mathbf{W})$ return the closest and second closest (based on Euclidean distance) neurons to \mathbf{x} , respectively.

III. RESULTS AND DISCUSSION

A. Generation of semi-synthetic ToF-SIMS data

This study of the effects of preprocessing and hyperparameter selection on SOM model performance uses the CNNAE as part of

data preprocessing. While we focus on SOM and CNNAE algorithms specifically, there is an interesting and general question about the importance of preprocessing and hyperparameter selection, which is applicable to all unsupervised ML methods used to analyze hyperspectral imaging data. One of the key challenges is the lack of accurately labeled datasets, where each pixel is (reliably) assigned to one of a discrete number of classes.

The microarray format provides one solution to this problem. As each spot corresponds to a single polymer, it can, therefore, be labeled reliably. The drawback of this approach is that the format does not provide insight into how spatially aware algorithms (such as the CNNAE) perform when pixels from different classes are adjacent to one another and/or spectrally mixed to varying degrees.

While it is possible to prepare such materials experimentally, there is, generally, a trade-off between the degree of interclass mixing and the reliability of pixel labeling. That is, it becomes increasingly difficult to reliably label each pixel in a ToF-SIMS image as the complexity of the physical sample increases. To address this problem, we developed a novel algorithm to mix spectra from C discrete ToF-SIMS data cubes at the individual pixel level. This algorithm [Eqs. (1)–(6) and Fig. S1]³⁷ enables highly complex data to be generated from real data (hence, the use of the term *semisynthetic*) with reliable pixel labeling. Furthermore, the algorithm parameters can be tweaked to increase or decrease spectral mixing and/or spatial autocorrelation or to use nonlinear class mixing.

For example, in Fig. S2,³⁷ we present class membership maps in \mathbf{M} , generated using a range of values for T_1 (number of pixels assigned in phase 1) and σ_r (scale parameter). We also estimate the spatial autocorrelation of each map using Moran's I measure and the spectral purity, SP , defined in Eq. (7). Clearly, the algorithm generates a diverse range of semisynthetic datasets for an arbitrary number of classes C . We anticipate that this approach will be of value to those interested in exploring spatially aware ML algorithms with ToF-SIMS (or other hyperspectral) data. Here, we used $T_1 = 0.005n$, where n is the total number of pixels, and $\sigma_r = 0.5$ (Fig. S2).³⁷

B. Evaluating preprocessing and hyperparameter importance

We used multiple linear regression (MLR) to quantify the relationships between preprocessing methods, hyperparameters, and SOM performance using heterogeneity, Jaccard index, CSI, and topographic error metrics. This MLR-based approach is commonly used for design of experiments (DoE)^{35,36} but is equally applicable to our study. MLR was performed using the Statistics and Machine Learning Toolbox in MATLAB. For added interpretability, we broadly classified the four metrics into two types: class-cluster similarity (heterogeneity and Jaccard index) and topology preservation (CSI and topographic error). For each metric, a smaller value (a more negative coefficient) is considered better.

Recall that we trained SOM models with various training epochs, ranging from 1 to 500. We did this to ensure convergence of the models based on the commonly used quantization error metric. From these results, we concluded that training for 500 epochs was generally sufficient for convergence. Nevertheless, we opted to build MLR models at 10, 100, and 500 epochs separately

22 September 2023 00:17:56

for completeness and additional comparison, and we have included all of these in the SI (as detailed later).³⁷ We have also included a range of example figures in the supplementary material showing the progression of SOM training for each metric used, focusing on a selection of the preprocessing methods and hyperparameters studied (Figs. S3–S10).³⁷ We provide these as additional points of reference for the remainder of the discussion. However, they are not critical as the central focus of this study is on the converged SOMs.

We first constructed models without interaction terms. MLR regression coefficients extracted from these models for both datasets are summarized in Tables S1 and S2,³⁷ along with the adjusted R^2 values for each model. While there were many statistically significant coefficients in these models, it is generally important to consider whether interactions between variables were present, which would render these coefficients uninterpretable. Hence, we constructed similar models allowing for all first-order interactions. We used stepwise subset selection (with combined forward and backward steps) based on adjusted R^2 to identify the subset of variables to use for each model. We only report results for the 500 epoch model here, while the complete set of results is provided in the SI (Tables S3 and S4).³⁷

The standardized regression coefficients from the 500 epoch models for the polymer microarray and nylon datasets, along with their adjusted R^2 values, are presented in Tables I and II, respectively. Variables not included in the models are presented as NA in the tables. The large increase in adjusted R^2 values (compared with models without interactions) provides strong evidence for the presence of interaction effects in both datasets. Before looking more closely at these interactions, on a higher level, it is important to note that the presence of substantial interactions is critically important, as it indicates that the choice of preprocessing methods and hyperparameter selection is not independent.

More specifically, Tables I and II identify several interesting trends. Notably, preprocessing interactions were generally much stronger than both hyperparameter and preprocessing-hyperparameter interactions. This indicates that, at least for these data and SOM models, decisions about preprocessing were most important and highly complex. Figures 2 and 3 further visualize these variables and their influence on each metric for the microarray and nylon datasets, respectively. These figures show each metric (rows; A–D) as a function of the SOM size, for each scaling method (columns). Overlaid in each plot are results for raw data (black circles), data normalized to TIC (red squares), encoded data (green diamonds), and data normalized to TIC and then encoded (blue stars). Figures 2 and 3 clearly demonstrate interactions between these variables (discussed in more detail later), explaining why the MLR models with interaction terms yield higher adjusted R^2 values. Collectively, Tables I and II and Figs. 2 and 3 together provide a wealth of information about how key variables (SOM size, scaling method, TIC normalization, and encoding) influence SOM performance across all four metrics, both individually and through their interactions. With these as a reference, we now proceed with a systematic breakdown and evaluation of key findings.

SOM size, on its own, tended to have a similar effect across both datasets (Figs. 2 and 3). Namely, increasing SOM size

TABLE I. Standardized regression coefficients from MLR models of the microarray dataset, trained using various preprocessing methods and hyperparameters, as well as their interactions. Bolded entries are statistically significant at $p < 0.05$. Stars represent significance levels: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$. Intercepts are shaded from red, through white, to green, where red is worse, white is neutral, and green is better performance. Note that coloring is relative, per model.

Type	Metric	Adj R ²	Preprocessing						Hyperparameters						Preprocessing Interactions																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
			Intercept	TIC	norm	Minmax	Poisson	Standard	Encoded	Toroidal	Hexagon	SOM size	TIC	norm	Minmax	Poisson	Standard	encoded	Minmax	encoded	Poisson	encoded	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon	Standard	encoded	hexagon

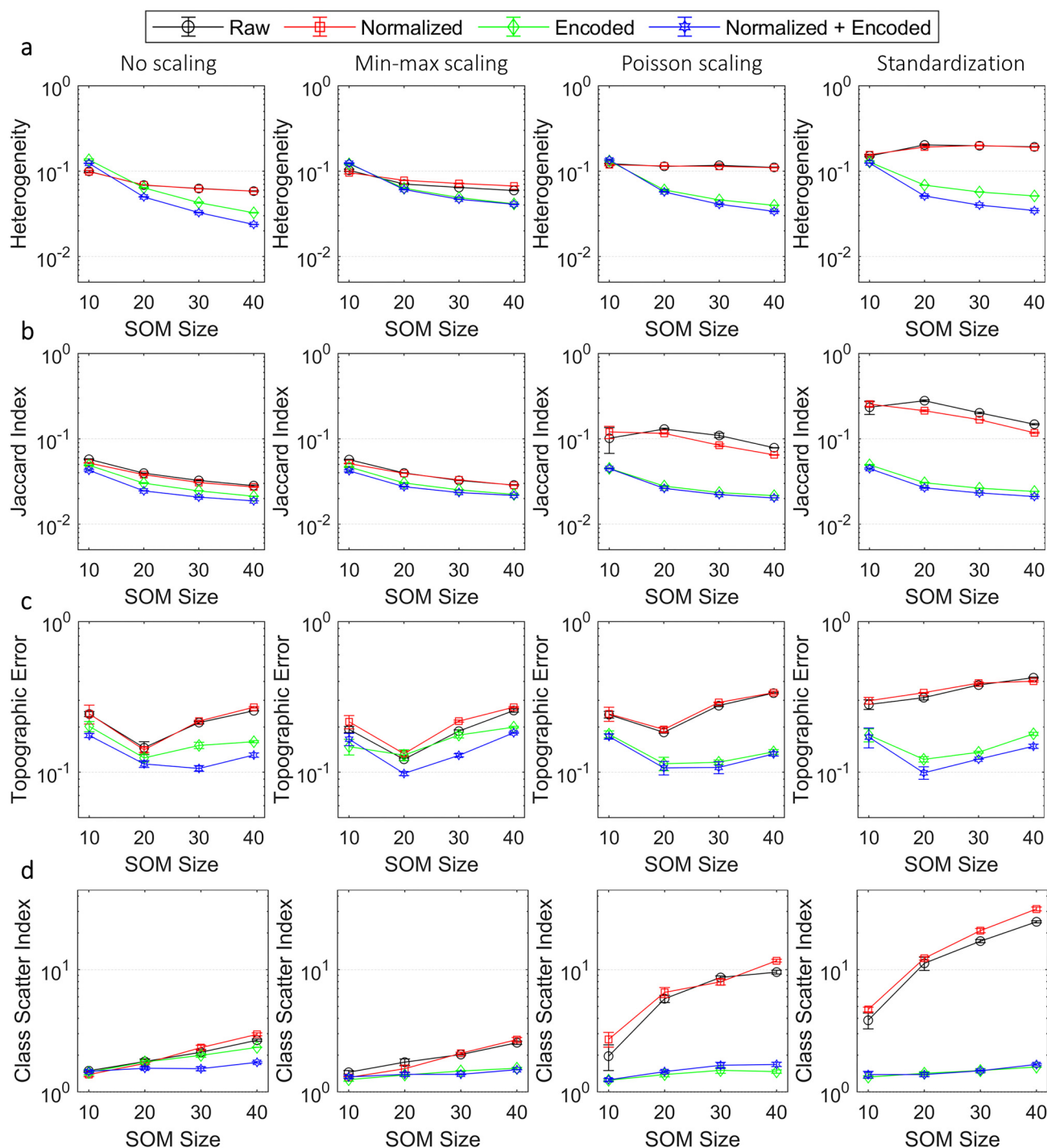
TABLE II. Standardized regression coefficients from MLR models of the nylon dataset, trained using various preprocessing methods and hyperparameters, as well as their interactions. Bolded entries are statistically significant at $p < 0.05$. Stars represent significance levels: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$. Intercepts are shaded in gray, whereas coefficients are shaded from red, through white, to green, where red is worse, white is neutral, and green is better performance. Note that coloring is relative, per model.

Preprocessing															Hyperparameters										Preprocessing interactions																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
Type	Metric	Adj R ²	TIC			SOM			Toroidal	Hexagon	Size	TIC		SOM		Toroidal	Hexagon	Size	TIC		SOM		Toroidal	Hexagon	Size	TIC		SOM		Toroidal	Hexagon	Size																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
			Intercept	norm	minmax	Poisson	Standard	minmax				Poisson	Standard	minmax	Poisson				Standard	minmax	Poisson	Standard				minmax	Poisson	Standard	minmax				Poisson	Standard	minmax	Poisson	Standard	minmax	Poisson	Standard	minmax	Poisson	Standard	minmax																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
Class-Cluster similarity	Heterogeneity	0.82	0.040***	0.0033	-0.028***	0.0071*	0.0030	0.060***	-0.0021	NA	-0.012***	0.011***	-0.032***	0.016***	-0.014***	0.0022	0.031***	0.00038	0.00038	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031***	0.0022	0.031

improved performance according to the heterogeneity and Jaccard index metrics (class-cluster similarity) but worsened performance according to the TE and CSI metrics (topology preservation). This suggests that larger SOMs did a better job of differentiating classes; however, they tended to be less topologically correct. An exception to this trend is evident in the TE metric for the microarray dataset (Fig. 2), which initially decreased when the SOM size was increased from 10×10 to 20×20 neurons (even though CSI increased). At larger sizes, both CSI and TE increased. Given that there were 70 classes in this dataset, this could suggest that the 10×10 SOM (100 neurons) was not sufficiently large to correctly model topology, such that an increase in the SOM size led to better topology preservation. This is in contrast to the nylon dataset, with only seven classes, for which both TE and CSI almost exclusively monotonically increased in relation to the SOM size. These results suggest that the optimal SOM size depends on the number of distinct classes in the data. While this is not typically known for unsupervised analyses, it is, nevertheless, important to consider.

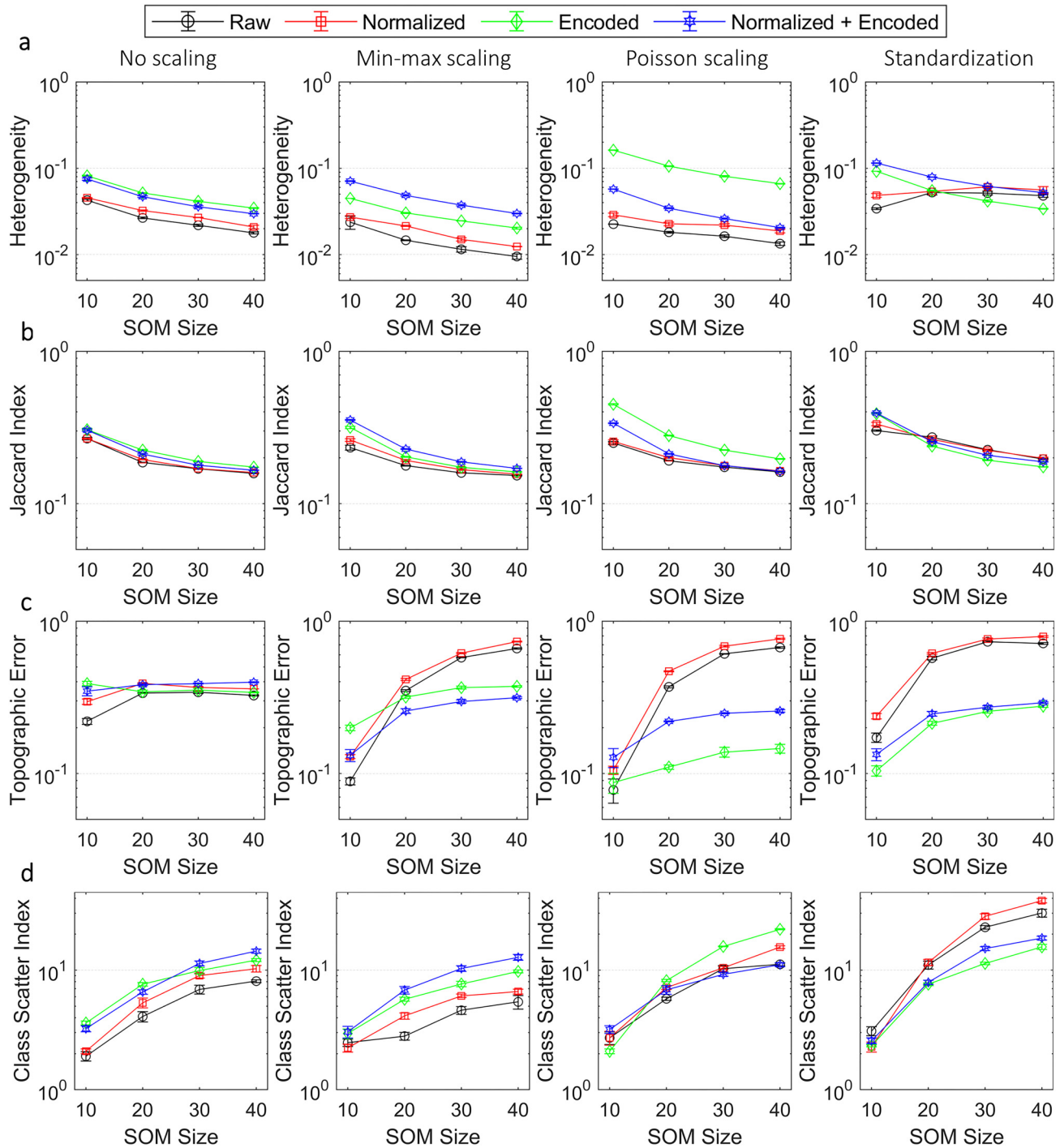
Aside from the SOM size, normalization to TIC and data encoding using the CNNAE led to significantly different outcomes, depending mostly on both the dataset and the scaling method used. With regard to TIC normalization, this is not unexpected, since the efficacy depends entirely on the system being studied and on the aims of the analysis. For example, for the microarray dataset, Table I and Fig. 2 show that both normalization and encoding generally improve performance across all metrics, indicating improved class-cluster similarity and topology preservation. Furthermore, Table I highlights a negative and significant interaction between these variables (for all metrics other than TE), indicating that the benefit of encoding was increased through normalization (and vice versa). In contrast, Table II and Fig. 3 show that, for the nylon dataset, normalization and encoding tend to reduce performance, according to the class-cluster similarity metrics. However, depending on the scaling method used, encoding sometimes led to improved performance according to the topology preservation metrics, most notably TE. It is important to note that we only considered encoding to 100 features. It is likely that modifying this as a hyperparameter of the CNNAE model would change these outcomes; however, given that this study was focused on SOM hyperparameters, this was outside the scope of this study. Nevertheless, it is an important and ongoing area of study.

Generally, across both datasets, there were clearly strong interactions between the scaling method and encoding (Figs. 2 and 3, and Tables I and II). For the microarray dataset, the interaction between standardization and encoding was the strongest—standardization in the absence of encoding led to worse performance across all four metrics used, whereas encoding mitigated this effect. Similar outcomes were observed for Poisson scaling and encoding. It is important to emphasize that this appeared to be mostly due to the poor performance of these scaling methods without encoding, rather than their interaction producing superior performance to other scaling methods. For the nylon dataset, interactions between encoding and Poisson scaling or standardization were mixed, with mixed positive and negative interactions for class-cluster similarity metrics, and mostly, negative interactions for topology metrics (as mentioned earlier).



22 September 2023 00:17:56

FIG. 2. Example results from the grid-search of the preprocessing-hyperparameter space for the polymer microarray dataset. Plots show the heterogeneity score (a), Jaccard index (b), topographic error (TE) (c), and class scatter index (CSI) (d) for a range of converged SOMs of different sizes, trained using data scaled using different methods. In each case, square neurons with toroidal topology were used. Each plot compares the metrics as a function of the SOM size, using either raw data (black circles), data normalized to total ion count (TIC) (red squares), data encoded to 100 features using the CNAE (green diamonds), or data normalized and then encoded (blue stars). Error bars show a standard deviation of three replicates, and the y-axis scale is logarithmic.



22 September 2023 00:17:56

FIG. 3. Example results from the grid-search of the preprocessing-hyperparameter space for the nylon dataset. Plots show the heterogeneity score (a), Jaccard index (b), topographic error (TE) (c), and class scatter index (CSI) (d) for a range of converged SOMs of different sizes, trained using data scaled using different methods. In each case, square neurons with toroidal topology are used. Each plot compares the metrics as a function of the SOM size, using either raw data (black circles), data normalized to total ion count (TIC) (red squares), data encoded to 100 features using the CNAE (green diamonds), or data normalized and then encoded (blue stars). Error bars show a standard deviation of three replicates, and the y-axis scale is logarithmic.

Given that standardization and Poisson scaling both appeared to reduce SOM performance, it is important to discuss these in more detail. Both methods involve the division of ion images/features by a statistical measure of that feature. For standardization, this is the standard deviation, while for Poisson scaling, it is the square root of the feature mean. If the data contain several features with means close to zero (e.g., noise m/z bins in the nylon dataset), then dividing by the square root of the mean leads to strong upscaling of noise, indicating that this method may be unsuitable for such data. Furthermore, Poisson scaling is based on the assumption that noise in the data follows a Poisson distribution. However, particularly if other preprocessing steps are applied prior to Poisson scaling (for example, normalization to TIC), such an assumption can be invalidated. We included such statistically invalid preprocessing pipelines in the empirical grid-search only for completeness. Finally, Poisson scaling is designed to account for heteroscedastic noise related to Poisson statistics. Division of features by the square root of their mean can transform the data into a space in which the noise is more uniform. For methods that focus on data variance, such as principal component analysis (PCA), scaling has been demonstrated to be highly effective in improving the interpretability of the PCA model.^{26,27} However, for the SOM, it is less clear whether heteroscedastic noise is as much of an issue. Combined with the adverse effects associated with low signal features, this could explain why Poisson scaling did not perform well for these datasets and SOM models. Our results emphasize the importance of considering which preprocessing method is used; Poisson scaling is effective for some ML methods and datasets, but this should not be assumed in general.

Of all the scaling methods, min-max scaling appeared to give the best performance across both datasets. It is important to emphasize again, however, that particular outcomes from this empirical investigation are specific to these data and to the SOM itself. Like standardization and Poisson scaling, min-max scaling has limitations, such as the potential to skew data distributions or emphasize noise. Therefore, we advise that such limitations should always be considered specifically for each dataset and statistical/machine learning algorithm being applied.

Another important interaction occurred between encoding and SOM size, which was strongly negative (and significant) across both datasets and all metrics, except for the Jaccard index for the microarray dataset. This must be interpreted carefully: these results do not imply that larger SOMs combined with encoding produced globally superior outcomes with regard to topology preservation. Indeed, it is clear from Figs. 2 and 3 (and as per the earlier discussion) that larger SOMs were associated with poorer topology preservation, regardless of whether data were encoded. Rather, these results suggest that if a large SOM is desired (for some reason other than topology preservation, e.g., if the data are expected to contain many classes), then it may be preferable to also encode the data to mitigate the loss of topology preservation. It is worth pointing out, however, that there also appear to be higher-order interactions that occurred (Figs. 2 and 3), such as between the scaling method, SOM size, and encoding. Such interactions precisely demonstrate the complexity of identifying the optimal combination of preprocessing methods and model hyperparameters, especially for unsupervised analyses. Note that these results may apply to

dimensionality reduction in general, but we encoded the data using the CNNAE only. Comparison against other feature extraction methods is outside the scope of this study and is left for future exploration.

Another noteworthy outcome is that the interaction between toroidal topology and SOM size was exclusively negative and significant for the topographic error metric. Like the interaction between encoding and SOM size, this does not indicate that this combination of hyperparameters achieves optimal topology preservation. Rather, it suggests that, if using a larger SOM, using toroidal topology aids in topology preservation. Furthermore, the same interaction was also negative and significant with regard to CSI for the nylon dataset (the interaction was not included for the microarray dataset). Thus, the detrimental effect of increased SOM size on CSI was again mitigated somewhat by using toroidal topology.

IV. CONCLUSIONS

We have demonstrated that preprocessing and hyperparameter selection can have a significant impact on the performance of the SOM applied to the analysis of ToF-SIMS images. We also showed that semisynthetic ToF-SIMS data, generated from real ToF-SIMS data, are useful for comparing the performance of ML algorithms, particularly those that are spatially aware. While real datasets with reliable ground truth labels are still considered the gold standard, such datasets are much more difficult and time-consuming to acquire. Therefore, semisynthetic data represent a valuable complementary source of labeled data that are much more readily available.

The results from this study indicate the importance of carefully considering preprocessing and hyperparameters when applying the SOM. Unfortunately, for unsupervised algorithms such as the SOM, ground truth information is typically not available, making it much more difficult to choose the optimal combination of preprocessing methods and hyperparameters. Therefore, we summarize those trends that were general across both datasets studied.

First, we note that increasing SOM size tended to improve the so-called class-cluster similarity of the models, whereby they better captured the underlying classes present in the data (especially when many classes were present). However, increasing SOM size also appeared to reduce topology preservation, such that there was a trade-off between these two outcomes.

Second, we note that the use of toroidal topology and data encoding (in this case, by a CNNAE) mitigated the loss of topology preservation for larger SOMs. This is important, as it implies that, if one wishes to use a large SOM, it is advisable to also use toroidal topology and to reduce the dimensionality of the data through encoding. Of course, the effect of encoding is likely to depend on the dimensionality of the original data and the number of features extracted, which must also be considered.

Third, we note that, in almost all cases, Poisson scaling and standardization performed either no better than, or worse than, no scaling. This suggests no clear benefit to these scaling methods. We emphasize that this outcome was specific to the SOM and may be specific to these datasets. Nevertheless, this does prompt further research in this area focusing on other ML models and datasets.

Finally, while these trends were consistent across both datasets studied, it is important to emphasize that this does not necessarily imply generality and that these trends may change for different datasets. Nevertheless, this study offers a useful starting point for extended research in this important area.

SUPPLEMENTARY MATERIAL

See the supplementary material for supplementary tables and figures and a complete mathematical description of the V-measure score.

ACKNOWLEDGMENTS

This work was supported by the Office of National Intelligence, National Intelligence and Security Discovery Research Grant (No. NI210100127) funded by the Australian Government. This work was performed in part at the Australian National Fabrication Facility (ANFF), a company established under the National Collaborative Research Infrastructure Strategy, through the La Trobe University Centre for Materials and Surface Science. The authors thank Robert Sikos, La Trobe University, for underpinning contributions in the use of self-organizing maps in the interpretation of ToF-SIMS data and the collection of the nylon datasets. The authors thank Morgan Alexander and Andrew Hook, Nottingham University, for providing the microarray ToF-SIMS dataset analyzed in this work. The authors acknowledge the Milano Chemometrics and QSAR Research Group for the development of the Kohonen and CP-ANN Toolbox for MATLAB.^{2,3}

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Wil Gardner: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Software (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **David A. Winkler:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Writing – original draft (equal); Writing – review & editing (equal). **David L. J. Alexander:** Conceptualization (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal). **Davide Ballabio:** Formal analysis (equal); Investigation (equal); Methodology (equal); Software (equal); Writing – original draft (equal); Writing – review & editing (equal). **Benjamin W. Muir:** Conceptualization (equal); Funding acquisition (equal); Investigation (equal); Project administration (equal); Writing – original draft (equal); Writing – review & editing (equal). **Paul J. Pigram:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project

administration (equal); Software (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in Open At La Trobe at <https://doi.org/10.26181/22671022>, Ref. 38.

REFERENCES

- ¹T. Kohonen, *Biol. Cybern.* **43**, 59 (1982).
- ²D. Ballabio, V. Consonni, and R. Todeschini, *Chemom. Intell. Lab. Syst.* **98**, 115 (2009).
- ³D. Ballabio and M. Vasighi, *Chemom. Intell. Lab. Syst.* **118**, 24 (2012).
- ⁴W. Gardner, S. M. Cutts, B. W. Muir, R. T. Jones, and P. J. Pigram, *Anal. Chem.* **91**, 13855 (2019).
- ⁵W. Gardner, S. M. Cutts, D. R. Phillips, and P. J. Pigram, *Biopolymers* **112**, e23400 (2020).
- ⁶W. Gardner, A. L. Hook, M. R. Alexander, D. Ballabio, S. M. Cutts, B. W. Muir, and P. J. Pigram, *Anal. Chem.* **92**, 6587 (2020).
- ⁷W. Gardner, R. Maliki, S. M. Cutts, B. W. Muir, D. Ballabio, D. A. Winkler, and P. J. Pigram, *Anal. Chem.* **92**, 10450 (2020).
- ⁸R. M. T. Madióna, S. E. Bamford, D. A. Winkler, B. W. Muir, and P. J. Pigram, *Anal. Chem.* **90**, 12475 (2018).
- ⁹R. M. T. Madióna, N. G. Welch, S. B. Russell, D. A. Winkler, J. A. Scoble, B. W. Muir, and P. J. Pigram, *Surf. Interface Anal.* **50**, 713 (2018).
- ¹⁰R. M. T. Madióna, D. A. Winkler, B. W. Muir, and P. J. Pigram, *Appl. Surf. Sci.* **487**, 773 (2019).
- ¹¹R. M. T. Madióna, D. A. Winkler, B. W. Muir, and P. J. Pigram, *Appl. Surf. Sci.* **478**, 465 (2019).
- ¹²N. G. Welch, R. M. T. Madióna, T. B. Payten, C. D. Easton, L. Pontes-Braz, N. Brack, J. A. Scoble, B. W. Muir, and P. J. Pigram, *Acta Biomater.* **55**, 172 (2017).
- ¹³N. G. Welch, R. M. T. Madióna, T. B. Payten, R. T. Jones, N. Brack, B. W. Muir, and P. J. Pigram, *Langmuir* **32**, 8717 (2016).
- ¹⁴W. Gardner, D. A. Winkler, D. Ballabio, B. W. Muir, and P. J. Pigram, *Biointerphases* **15**, 061004 (2020).
- ¹⁵J. X. Li, *Information Visual.* **3**, 49 (2004).
- ¹⁶A. Henderson, J. S. Fletcher, and J. C. Vickerman, *Surf. Interface Anal.* **41**, 666 (2009).
- ¹⁷B. Tyler, G. Rayal, and D. Castner, *Biomaterials* **28**, 2412 (2007).
- ¹⁸M. S. Wagner, D. J. Graham, and D. G. Castner, *Appl. Surf. Sci.* **252**, 6575 (2006).
- ¹⁹W. Gardner, D. A. Winkler, S. M. Cutts, S. A. Torney, G. A. Pietersz, B. W. Muir, and P. J. Pigram, *Anal. Chem.* **94**, 7804 (2022).
- ²⁰A. L. Hook, P. M. Williams, M. R. Alexander, and D. J. Scurr, *Biointerphases* **10**, 019005 (2015).
- ²¹V. V. Danilov, D. Y. Kolpashchikov, O. M. Gerget, N. V. Laptev, A. Proutski, L. A. Hernández Gómez, F. Alvarez, and M. J. Ledesma-Carbayo, *Comput. Med. Imaging Graph.* **106**, 102188 (2023).
- ²²L. Liu, A. Johansson, Y. Cao, J. Dow, T. S. Lawrence, and J. M. Balter, *Phys. Med. Biol.* **65**, 125001 (2020).
- ²³M. Shi, T. Zhao, S. J. West, A. E. Desjardins, T. Vercauteren, and W. Xia, *Photoacoustics* **26**, 100351 (2022).
- ²⁴T. Alexandrov and J. H. Kobarg, *Bioinformatics* **27**, 1230 (2011).
- ²⁵P. A. P. Moran, *Biometrika* **37**, 17 (1950).
- ²⁶M. R. Keenan and P. G. Kotula, *Surf. Interface Anal.* **36**, 203 (2004).
- ²⁷M. R. Keenan and P. G. Kotula, *Appl. Surf. Sci.* **231–232**, 240 (2004).
- ²⁸M. Abadi *et al.*, *arXiv:1603.04467v2* (2016).
- ²⁹F. Chollet, *Keras* (GitHub, San Francisco, CA, 2015), see <https://github.com/fchollet/keras>

- ³⁰A. Rosenberg and J. Hirschberg, *V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure* (Association for Computational Linguistics, Prague, 2007).
- ³¹P. Jaccard, *Bull. de la Soc. Vaud. des Sci. Nat.* **37**, 547 (1901).
- ³²L. Elend and O. Kramer, in *Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization*, edited by A. Vellido, K. Gibert, C. Angulo, and J. D. Martín Guerrero (Springer International, Cham, 2020), pp. 23–32.
- ³³K. Kiviluoto, *Proceedings of International Conference on Neural Networks (ICNN'96)*, Washington, DC, 3–6 June 1996 (IEEE, Piscataway, NJ 1996), Vol. 1, Vol. 291, pp. 294–299.
- ³⁴F. Forest, M. Lebbah, H. Azzag, and J. Lacaille, [arXiv:2011.05847](https://arxiv.org/abs/2011.05847) (2020).
- ³⁵R. Carlson and J. E. Carlson, “1.11—The study of experimental factors★,” in *Comprehensive Chemometrics*, 2nd ed., edited by S. Brown, R. Tauler, and B. Walczak (Elsevier, Amsterdam, 2020), pp. 251–285.
- ³⁶L. A. Sarabia, M. C. Ortiz, and M. S. Sánchez, “1.12—Response surface methodology★,” in *Comprehensive Chemometrics* 2nd ed., edited by S. Brown, R. Tauler, and B. Walczak (Elsevier, Amsterdam, 2020), pp. 287–326.
- ³⁷See supplementary material online for supplementary tables and figures and a complete mathematical description of the V-measure score.
- ³⁸W. Gardner (2023). “Data set for article: Effect of data preprocessing and machine learning hyperparameters on mass spectrometry imaging models,” [Open at La Trobe \(OPAL\)](#).