BIOINFORMATICS

Efficient spatial segmentation of large imaging mass spectrometry datasets with spatially aware clustering
Theodore Alexandrov1,2,* and Jan Hendrik Kobarg1
1Center for Industrial Mathematics, University of Bremen, 28359 Bremen and 2Steinbeis Innovation Center for Scientific Computing in Life Sciences, 28211 Bremen, Germany

ABSTRACT

Motivation: Imaging mass spectrometry (IMS) is one of the few measurement technologies of biochemistry which, given a thin sample, is able to reveal its spatial chemical composition in the full molecular range. IMS produces a hyperspectral image, where for each pixel a high-dimensional mass spectrum is measured. Currently, the technology is mature enough and one of the major problems preventing its spreading is the under-development of computational methods for mining huge IMS datasets. This article proposes a novel approach for spatial segmentation of an IMS dataset, which is constructed considering the important issue of pixel-to-pixel variability.

Methods: We segment pixels by clustering their mass spectra. Importantly, we incorporate spatial relations between pixels into clustering, so that pixels are clustered together with their neighbors. We propose two methods. One is non-adaptive, where pixel neighborhoods are selected in the same manner for all pixels. The second one respects the structure observable in the data. For a pixel, its neighborhood is defined taking into account similarity of its spectrum to the spectra of adjacent pixels. Both methods have the linear complexity and require linear memory space (in the number of spectra).

Results: The proposed segmentation methods are evaluated on two IMS datasets: a rat brain section and a section of a neuroendocrine tumor. They discover anatomical structure, discriminate the tumor region and highlight functionally similar regions. Moreover, our methods provide segmentation maps of similar or better quality if compared to the other state-of-the-art methods, but outperform them in runtime and/or required memory.

Contact: theodore@math.uni-bremen.de

1 INTRODUCTION

Given a thin sample (usually a tissue slice), imaging mass spectrometry (IMS) measures high-dimensional mass spectra at its spatial points, providing a hyperspectral image with a mass spectrum measured at each pixel (Fig. 1). Each mass spectrum dimension represents the abundance of molecules with this molecular mass. Currently, IMS is one of the few biochemical technologies able to establish the spatial biochemical composition of the sample in the full molecular range (small and large molecules, e.g. metabolites, lipids and proteins). Since 1970s, secondary ion mass spectrometry was the main IMS technique for surface analysis (Benninghoven and Loebach, 1971), although being unable to measure large molecules (e.g. peptides and proteins). With the advent of Matrix-assisted laser desorption ionization (MALDI) imaging mass spectrometry (Stoeckli et al., 2001), the measurement of peptides and proteins became possible what opened IMS a door to the variety of biological and biomedical problems.

Currently, IMS is one of the most promising innovative measurement techniques in biochemistry. IMS has proven its potential in discovery of new drugs (Solonen et al., 2010; Yang et al., 2009) and cancer biomarkers (Cazares et al., 2009; Rauser et al., 2010) just to mention a few important applications. IMS was used in numerous studies leading to understanding chemical composition and biological processes, see recent reviews on this topic (Amstalden van Hove et al., 2010; Watrous et al., 2011). As for many modern biochemical techniques, in particular in proteomics (Patterson, 2003), the development of computational methods for IMS is lagging behind the technological progress. Two unsupervised methods are currently considered in IMS data processing: (i) data representation using principal component analysis (PCA) and its variants (Klerk et al., 2007), the technique standardly used for processing SIMS data, and (ii) spatial segmentation of an IMS dataset by means of spectra (or pixels) clustering (Alexandrov et al., 2010; Deininger et al., 2008; McCombie et al., 2005).

Concerning the second problem, the spatial segmentation of an IMS dataset, several approaches have been proposed. First, straightforward clustering of mass spectra, e.g. k-means (McCombie et al., 2005). Second, feature extraction with PCA and then hierarchical clustering of features (Deininger et al., 2008). Hierarchical clustering has an advantage of interactive analysis of the clustering dendrogram but needs to keep in memory the full distance matrix; several work-arounds have been proposed (Zhang et al., 1996) but none is accepted as a standard. The main drawback of using straightforward clustering of mass spectra is that it is negatively affected by the pixel-to-pixel variability issue (Alexandrov et al., 2010; Watrous et al., 2011), which is especially serious in the most popular IMS techniques, SIMS and MALDI-IMS. Underestimation of this issue leads to strong noise in the resulted segmentation maps (Alexandrov et al., 2010). As we show later, the method combining PCA with hierarchical clustering is also prone to this problem.

We have recently proposed a spatial segmentation method solving this important issue, where edge-preserving spatially-adaptive denoising (Grasmair, 2009) is applied to gray-scale images of selected masses prior to clustering (Alexandrov et al., 2010). The produced segmentation maps are significantly better than those produced without using denoising, in terms of lack of noise, discrimination of anatomical and histological details, and the number of visible regions. However, these improvements are achieved in full at a cost of using a computationally intensive
edge-preserving spatially adaptive denoising method and slow high dimensional discriminant clustering. Use of a simpler methods improves the segmentation maps but not significantly due to strong and multiplicative noise in data.

In this article, we propose another original approach for spatial segmentation of IMS data with strong pixel-to-pixel variability. In general, this approach can be applied to any hyperspectral data, not necessarily IMS, but we consider segmentation of only IMS data. In (Alexandrov et al., 2010), the motivation for using the prior-to-clustering denoising was that neighbor pixels should have similar intensity at a peak mass (a mass, where not just noise is represented). Results shown by Alexandrov et al. (2010) confirm that a procedure taking into account spatial relations between pixels delivers a better quality than a straightforward pixels clustering. In this article, our approach is to take into account the spatial relations between pixels by incorporating this information directly into a clustering method. We propose two new approaches to pixels clustering, when each pixel is considered together with its neighbors. One is non-adaptive, where each pixel is clustered together with pixels from its neighborhood; neighborhoods are selected in the same manner for all pixels. The second structure-adaptive way respects the structure observable in data and, for a pixel, its neighborhood is defined taking into account similarities of spectra measured at pixels around it. In order to build efficient clustering methods based on these two ideas (non-adaptive and structure-adaptive), we use the FastMap method (Faloutsos and Lin, 1995), applying it for an efficient dimensionality reduction and, given a distance matrix, for finding points in euclidean space with similar inter-distances. Importantly, we do not keep the full distance matrix in memory using a special FastMap trick explained later in the paper.

We evaluate the proposed methods on two MALDI-IMS datasets, a rat brain coronal section without pathology (20 185 spectra) where of our interest is to recover the anatomical structure of the brain, and a section of a neuroendocrine tumor (NET) invading the small intestine (27 360 spectra) where the processing was performed using our original implementation of all methods, including FastMap. The rat brain dataset comprises 20 185 spectra acquired within the slice area (120 × 201 pixels), each of 3045 data points covering 3.2–18 kDa. For examples of intensity images for masses 4966 Da and 6717 Da for the rat brain dataset, see Figure 1; more examples are in (Alexandrov et al., 2010). The spectra were baseline-corrected with the TopHat algorithm (minimal baseline width set to 10%, the default value in ClinProTools). No normalization or binning was done. Then spectra were saved into ASCII files and loaded in Matlab R2010b (The Mathworks Inc., Natick, MA, USA), where the processing was performed using our original implementation of all methods, including FastMap. The rat brain dataset comprises 20 185 spectra acquired within the slice area (120 × 201 pixels), each of 3045 data points covering the mass range 2.5–10 kDa; the NET dataset comprises 27 360 spectra (171 × 239 pixels) each of 5027 data points covering 3.2–18 kDa. For examples of intensity images for masses 4966 Da and 6717 Da for the rat brain dataset, see Figure 1; more examples are in (Alexandrov et al., 2010).

**2 METHODS**

2.1 Data measurement and preprocessing

Samples preparation and IMS measurements of both the rat brain and NET datasets are described in detail in Alexandrov et al. (2010). Shorty, the cryosections of 10 µm thickness were cut on a cryostat, transferred to a conductive indium-tin-oxide-coated glass slide (Bruker Daltonik GmbH, Bremen, Germany) and measured using a MALDI-TOF instrument (Autoflex III, Bruker Daltonik GmbH) using flexControl 3.0 and flexImaging 2.1 software (Bruker Daltonik GmbH). The lateral resolution was set to 80 µm.

The pre-processing was done in the ClinProTools 2.2 software (Bruker Daltonik). The spectra were baseline-corrected with the TopHat algorithm (minimal baseline width set to 10%, the default value in ClinProTools). No normalization or binning was done. Then spectra were saved into ASCII files and loaded in Matlab R2010b (The Mathworks Inc., Natick, MA, USA), where the processing was performed using our original implementation of all methods, including FastMap. The rat brain dataset comprises 20 185 spectra acquired within the slice area (120 × 201 pixels), each of 3045 data points covering the mass range 2.5–10 kDa; the NET dataset comprises 27 360 spectra (171 × 239 pixels) each of 5027 data points covering 3.2–18 kDa. For examples of intensity images for masses 4966 Da and 6717 Da for the rat brain dataset, see Figure 1; more examples are in (Alexandrov et al., 2010).

2.2 Peak picking

Selection of dataset frequent peaks: for the peak picking, we improved the approach proposed by Alexandrov et al. (2010). First, as by Alexandrov et al. (2010), for each 10th spectrum we picked its peaks with orthogonal matching pursuit (OMP) algorithm (Denis et al., 2009). This greedy algorithm searches for the specified number $p_0$ of peaks which simultaneously are (i) high and (ii) fit at best the given peak shape (the Gaussian shape is used). OMP is not just selection of the $p_0$ peaks most correlated with the Gaussian function. In comparison with this naive greedy approach, OMP has better theoretical properties of reconstruction peaks hidden in the noise; for more details, see (Denis et al., 2009). The peak picking for each 10th spectrum assigns to each $m/z$-value a number of spectra in which this $m/z$-value was selected as a peak. Finally, we take the most frequent peaks which were selected in more than $f_p$% of considered spectra. The parameters of this peak picking approach were selected as $p_0 = 20$ and $f_p = 0.5$. In total, we got 305 frequent peaks for both datasets.
We also use the FastMap algorithm as an efficient dimensionality reduction. The FastMap algorithm (Faloutsos and Lin, 1995), given a distance matrix and Lin, 1995) but implemented by Faloutsos and Lin in their C package. For the NET dataset (et al., 2009) but is more efficient having the linear in n complexity of O(n^2). Moreover, although it is not well-known, for finding projections of n points, FastMap needs to calculate only n(2q + 1) pairwise distances. Computing them on-the-fly, one significantly reduces the required memory. This trick is not described in the original paper (Faloutsos and Lin, 1995) but implemented by Faloutsos and Lin in their C package. We also use the FastMap algorithm as an efficient dimensionality reduction.

2.3 FastMap: distance-preserving projection

The FastMap algorithm (Faloutsos and Lin, 1995), given a distance matrix of size n × n and a euclidean space of dimension q, finds q points in this space with inter-distances close to those given in the matrix. It resembles multidimensional scaling (Hastie et al., 2009) but is more efficient having the linear in n complexity of O(nq). Moreover, although it is not well-known, for finding projections of n points, FastMap needs to calculate only n(2q + 1) pairwise distances. Computing them on-the-fly, one significantly reduces the required memory. This trick is not described in the original paper (Faloutsos and Lin, 1995) but implemented by Faloutsos and Lin in their C package. We also use the FastMap algorithm as an efficient dimensionality reduction.

2.4 Spatially aware clustering

Probably, the most apparent way to embed the spatial relations between pixels into a clustering algorithm is to use a distance-based clustering where the distance between two spectra is measured using the FastMap algorithm for dimensionality reduction, projecting the mapped spectra into a euclidean space of lower dimension. As a last step, reducing redundancy in the mapped data, we apply the FastMap algorithm for dimensionality reduction, projecting the mapped spectra into a euclidean space of lower dimension. Finally, clustering with an efficient vectorial clustering algorithm is performed. We propose using the k-means clustering algorithm.

Algorithm 1 Spatially-aware clustering (SA)

Parameters: pixel neighborhood radius r, FastMap desired dimension q, number of clusters k
1. Given r, create weights \( w_i \) as in Equation (2)
2. For each spectrum \( s \), \( m = \Phi(s) \) using weights \( w_i \) [map a spectrum into the feature space using Equation (4)]
3. Given q, project mapped spectra \( \{m_i\}_{i=1}^{N} \) into \( \mathbb{R}^q \) using FastMap obtaining \( \{\tilde{m}_i\}_{i=1}^{N} \) [reduce the dimensionality up to q]
4. Cluster the projected mapped spectra \( \{\tilde{m}_i\}_{i=1}^{N} \) into k groups using k-means

2.5 Spatially aware structure-adaptive clustering

Applying the spatially-aware clustering proposed in the previous section to IMS data, we realized that it improves the segmentation maps as compared to the straightforward clustering. However, it can smooth the edges between the anatomical or histological regions or eliminates small details; more on this in Section 3. Using a smaller pixel neighborhood radius r solves this problem only partially, since for a smaller r the noise in the resulted segmentation map is stronger.

So, we propose another method, where for a pixel, the weights of pixels in its neighborhood are not simply Gaussian but calculated adaptively, taking into account similarities of the pixels. The key idea is as follows. For each pixel in the neighborhood, we consider the distance between its spectrum and the spectrum in the center of the neighborhood. The larger the distance (the less similar the spectra are), the smaller is the weight. This idea is adapted from the bilateral filtering (Tomasi and Manduchi, 1998), an edge-preserving approach are: (i) \( \sigma_p \), the standard deviation of the Gaussian shape of a peak, (ii) \( \sigma_n \), the number of peaks selected per spectrum, and (iii) \( \tau \), the percentage of spectra a peak should be found in.
color image denoising method, where the adaptively calculated weights are used afterwards for averaging. First, for a pixel with coordinates \((x, y)\), we introduce

\[
\rho_k(x, y) = \exp\left(-\frac{d_k(x, y)^2}{\sigma_k^2}\right), \quad -\infty \leq k \leq \infty,
\]

in line with Tomasi and Manduchi (1998), where \(\lambda\) is a parameter and \(\lambda_k = \sqrt{\sum_{ij} \rho_k(x, y)^2} / \max_{ij} \rho_k(x, y)^2\). Then the distance between two spectra at coordinates \((x_1, y_1)\) and \((x_2, y_2)\) is as follows

\[
d_{ij} = \frac{\sum_{k=1}^{\min(\lambda_k, \lambda_s)} \rho_k(x_1, y_1) \rho_k(x_2, y_2)}{\sum_{k=1}^{\lambda_k, \lambda_s} \rho_k(x_1, y_1) + \sum_{k=1}^{\lambda_k, \lambda_s} \rho_k(x_2, y_2)},
\]

where \(\lambda_s\) is the neighborhood size and \(\lambda_s\) is the smallest value for \(\rho_k(x, y)\) for a given \(q\) so that the pairwise distances are similar to those calculated using Equation (6). Note that we do not calculate all \((n^2-n)/2\) pairwise distances but only \(m(2q+1)\) thanks to the FastMap trick. Finally, the points found by FastMap are clustered with \(k\)-means.

**Algorithm 2 Spatially-aware structure-adaptive clustering (SASA)**

**Parameters:** pixel neighborhood radius \(r\), FastMap desired dimension \(q\), number of clusters \(k\)

1. Given \(r\), create weights \(\bar{w}_{ij}\) as in Equation (2)
2. Given the distance function (6), project the spectra into \(\mathbb{R}^q\) using FastMap, where, when necessary, calculate the distance between spectra on-the-fly
3. Cluster the projected spectra into \(k\) groups using \(k\)-means

Figure 4 summarizes and illustrates the proposed segmentation methods, SA and SASA. Note that each of them has only three parameters, the pixel neighborhood radius \(r\), the dimension \(q\) of the space where FastMap projects the mapped data to, and the number of clusters \(k\). Later on, we will show that \(q\) is important, thus each method has only two easily interpretable parameters: \(r\) adjusts the smoothness of the segmentation map and \(k\) is simply the number of colors of the map. Both methods together with FastMap were implemented in Matlab 2010b.

### 3 RESULTS

In this section, we analyze in detail the results for the rat brain dataset. Then, we show segmentation maps for the neuroendocrine tumor dataset.

#### 3.1 Rat brain dataset

A rat brain section is a standard example tissue in IMS. According to Watrous et al. (2011), 43% of publications on IMS of tissues consider a brain tissue. This is due to the well-known anatomical structure and clear and well-separated anatomical regions. In this section, we compare the segmentation maps produced for the rat brain slice with a schematic of the rat brain corresponding to the coronal section \(z = 4.16\) mm from Bregma drawn based on the rat brain atlas (Paxinos and Watson, 2007). The peak picking procedure with alignment of masses to peaks finds \(p = 71\) peaks.
3.1.1 Overview Each of our proposed segmentation methods, SA (spatially adaptive, with Gaussian weights used) and SASA (spatially adaptive, with structure-adaptive weights), has only three parameters: the pixel neighborhood radius \( r \), the dimension \( q \) of the space where FastMap projects the mapped data into, and the number of clusters \( k \).

We consider segmentation maps produced for \( r=2, 3, 4 \) The FastMap dimension is \( q=20 \). The number of clusters (i.e. map colors) is \( k=10 \), what by Alexandrov et al. (2010) was found to be representative for this dataset. Figure 5 shows an optical image (A), the schematic of the anatomical structure (B), a segmentation map produced with straightforward clustering of spectra when no spatial relations between spectra are taken into account (C), and maps for SA (D–F) and SASA methods (G–I).

First, one can see that for the segmentation maps produced with both SA and SASA methods reflect the anatomical structure. Some anatomical regions (cortex, hippocampus, corpus callosum and internal capsule, amygdala) are very well represented. Note that the hippocampus has different parts (one in the middle and another close to amygdala) which still have the same color in the map (mid blue). Some regions are not well represented, e.g. a thin part of thalamus which goes around hypothalamus is not visible. However, as discussed by Alexandrov et al. (2010), this might be not a computational problem but an underrepresentation of these regions in the processed IMS dataset.

Second, our methods significantly outperform the straightforward clustering (Fig. 5C) where strong noise hides details and the whole anatomical regions. For example, in Figure 5C amygdala are not separated from hippocampus; hippocampus from the inner part of cortex and from paraventricular nuclei. Importantly, the noise in the segmentation map is a technological and computational artifact but not a property of the brain tissue; for more details on noise in MALDI-imaging, see (Alexandrov et al., 2010).

Thus, we conclude that the overall quality of the produced segmentation maps for the rat brain dataset is good. Note the blue small region interrupting the left part of cortex (Fig. 6, region A). This represents a tissue slice preparation defect (visible in the optical image as well) when the thin 10 µm tissue slice was folded during transferring it onto a glass slide.

3.1.2 Efficiency The efficiency of the segmentation method was the ultimate goal for us because existing advanced segmentation methods run several tens of minutes for a dataset. Tens of minutes seems acceptable because it is still less than the dataset acquisition time (several hours). However, this does not allow one to use segmentation interactively, what is of very importance in imaging applications. Moreover, at the present moment datasets with higher lateral resolution of 20 µm are becoming to be measured (Lagarrigue et al., 2010). If the rat brain slice would be measured with 20 µm resolution (instead of 80 µm used in this article), this would result in...
Both methods need only fly. The latter is especially important if such memory inefficient pixel, more pixels around it are taken into account when calculating radius

3.1.3 The role of the pixel neighborhood radius $r$

Figure 5D–I

SASA method stores additionally and $n_q$ distances calculated on-the-fly. For the FastMap algorithm has linear complexity $O_2$ and in the SASA $q$-means is used; $r \leq 3$, $q=20$, $k=10$.

Equations (2) or (5) what helps to reduce the pixel-to-pixel variability. On the other side, small details can be smoothed out, especially by using SA method with non-adaptive weights. For example, the central part of hippocampus in the right half loses its details visible in Figure 5D–E but not in Figure 5F, as well as the red dot in central part of the cortex (data not attributed).

3.1.5 SA-artifacts Comparison of the SA- and SASA-segmentation maps reveals an artifact produced by the SA method. It is a layer of chartreuse yellow pixels along hippocampus which are not visible in the SASA maps and cannot be attributed to any anatomical region. Figure 6, regions B and C, highlights the areas of the artifact. We hypothesize that this is an 'averaging' artifact due to weights $a_{ij}$ in Equation (2) are not adaptive to the data. This is confirmed by the absence of this layer in the SA map with the smallest pixel neighborhood radius $r = 2$ and in the SASA maps.

3.1.6 The role of the FastMap dimension $q$ The parameter $q$, the dimension of the space FastMap projects the spectra into, is the most tricky among three parameters of the SA and SASA methods. Naturally, increase of $q$ makes the problem high-dimensional and, thus, prone to the curse of dimensionality issue. On the other hand, for any distance-preserving algorithm the quality of projection reduces with decrease of the dimension $q$. We propose to select $q$ not greater than $p$ as FastMap would project from $\mathbb{R}^p$ into $\mathbb{R}^q$. For the SA method, this is motivated by the assumption that the mapping (4) into $p(2r+1)^2$-dimensional space introduces much redundancy.

Figure 7 shows the segmentation maps for the SASA method with $r = 3$ and $k = 10$, for different values of the FastMap dimension $q = 10$, 20, 50. The values of $q$ were selected to be smaller than $p = 71$. One can see that the maps for $q = 20$ and $50$ are very similar. The map for $q = 10$ looks noisier with a possibly artifact due to weights $a_{ij}$ in Equation (2) are not adaptive to the data. This is confirmed by the absence of this layer in the SA map with the smallest pixel neighborhood radius $r = 2$ and in the SASA maps.

Table 1. Runtimes for producing a segmentation map

<table>
<thead>
<tr>
<th>Dataset (n x p)</th>
<th>Method</th>
<th>Neighborhood radius</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r=2</td>
<td>r=3</td>
</tr>
<tr>
<td>Rat brain (20185 x 71)</td>
<td>SA</td>
<td>17 s</td>
</tr>
<tr>
<td></td>
<td>SASA</td>
<td>19 s</td>
</tr>
<tr>
<td></td>
<td>Denoising+HDDC*</td>
<td>17 min</td>
</tr>
<tr>
<td></td>
<td>Denoising+skmeans*</td>
<td>2 min</td>
</tr>
<tr>
<td></td>
<td>PCA+hierarchical*</td>
<td>25 s</td>
</tr>
<tr>
<td>Tumor (27960 x 62)</td>
<td>SA</td>
<td>23 s</td>
</tr>
<tr>
<td></td>
<td>SASA</td>
<td>25 s</td>
</tr>
</tbody>
</table>

Runtimes on ThinkPal laptop with Intel i5 Core 2.4 GHz; data pre-processing and peak picking are not included; $q=20$, $k=10$.

*No neighborhood is exploited.

Table 2. Detailed runtimes for SA and SASA methods

<table>
<thead>
<tr>
<th>Substep</th>
<th>Rat brain (s)</th>
<th>Tumor (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scaling</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Weights</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Fastmap</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>$k$-means</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>$k$-means</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Scaling</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Weights</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Fastmap</td>
<td>32</td>
<td>31</td>
</tr>
<tr>
<td>$k$-means</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$k$-means</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Fig. 6. Segmentation map (SA method) for the rat brain dataset, $r = 3$, $q = 20$, $k = 10$. Region A shows a tissue slice preparation defect (see in text, Section 3.1.1). Regions B and C highlight an SA-specific artifact, a layer of chartreuse yellow pixels along hippocampus (see in text, Section 3.1.4).

Table 2. Detailed runtimes for SA and SASA methods

<table>
<thead>
<tr>
<th>Substep</th>
<th>Rat brain (s)</th>
<th>Tumor (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scaling</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Weights</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Fastmap</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>$k$-means</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>$k$-means</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Scaling</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Weights</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Fastmap</td>
<td>32</td>
<td>31</td>
</tr>
<tr>
<td>$k$-means</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$k$-means</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Next, we compare our methods with advanced segmentation methods. (A) Prior-to-clustering edge-preserving denoising produces a similarly noisy map. We have tried the average linkage as well; it produces a similarly noisy map. The complete linkage was exploited. One can see that the segmentation map is very noisy, possibly because the pixel-to-pixel variability in spectra. We have tried the average linkage as well; it produces a similarly noisy map.

3.1.7 Comparison with other methods PCA with hierarchical clustering We have already shown that our segmentation maps excel the maps produced with straightforward clustering of spectra (Fig. 5C), mostly due to reduction of the pixel-to-pixel variability. Let us consider the method proposed by Deininger et al. (2008), where hierarchical clustering was applied to low-dimensional features extracted with PCA of spectra. Figure 8A shows the segmentation map produced with this method. As recommended in (Deininger et al., 2008), seven PCA components explaining 70% variance and the euclidean distance between extracted features were used. For the Ward linkage need much memory (for this dataset 8 GB was not enough), the complete linkage was exploited. One can see that the segmentation map is very noisy, possibly because the pixel-to-pixel variation in spectra. We have tried the average linkage as well; it produces a similarly noisy map.

3.1.8 Segmentation with prior-to-clustering edge-preserving denoising Next, we compare our methods with advanced segmentation proposed in (Alexandrov et al., 2010), where prior-to-clustering edge-preserving denoising of mass images was used to reduce the pixel-to-pixel variability. The edge-preserving denoising due to (Grasmair, 2009) requires about 2 min for 71 images of the rat brain dataset. The segmentation map from (Alexandrov et al., 2010) for $k=10$ is shown in Figure 8. For clustering, High Dimensional Discriminant Clustering (HDDC) was used, the clustering method designed specially for high-dimensional data. However, its main disadvantage is the long runtimes due to the expectation-maximization algorithm. Moreover, during an M-step it can be unstable when a cluster has a few elements or no elements, what requires starting it several times with random initializations. Although a new version of HDDC fixing these issues is planned to be included soon in the MIXMOD software (Biernacki et al., 2006), at the present time HDDC is slow. For the rat brain, one iteration takes about 50 s For $k=10$, at least 20 iterations are necessary because of the mentioned instability, which sums up to approximately 15 min.

For this reason, we replaced HDDC with $k$-means in the method proposed in (Alexandrov et al., 2010). The resulted segmentation map is shown in Figure 8. One can see that the map produced with HDDC is comparable to the maps presented in Figure 5. Moreover, as also discussed in (Alexandrov et al., 2010), $k$-means seems to be worse than HDDC (artifacts, less regions, detailness). The SA- and SASA-maps look better than those after $k$-means with prior-to-clustering denoising. The runtimes for methods considered in this section are given in Table 1, which are much longer than those for the SA and SASA methods.

3.2 The neuroendocrine tumor dataset

In this section, we briefly consider the segmentation maps for the second dataset, the neuroendocrine tumor invading the small intestine (ileum). This dataset differs from the rat brain dataset in the following respects: (i) it represents pathology (tumor), (ii) the tissue is more complicated, the difference between anatomical regions is not that clear, (iii) the tumor area is a heterogeneous composition of tumor cells, tumor stroma, and connective tissue. All this poses a complex challenge for a segmentation algorithm.

Figure 9 shows the optical image after H&E-staining together with 3D structure of the tissue and optical image with main functional structures as well as the segmentation maps. First, the tumor region is separated from the rest and is represented in three colors: blue, red and chartreuse yellow. This corresponds to results shown in (Alexandrov et al., 2010), although there the blue and red regions have not been separated. Moreover, the anatomical structure is represented, although the tissue flattened when put on
Efficient spatial segmentation of IMS data

Fig. 9. Neuroendocrine tumor dataset. (A) Optical image after H&E staining, the tumor region is selected. (B) 3D structure of the tissue and optical image with main functional structures, reproduced from (Alexandrov et al., 2010) with permission from the American Chemical Society. (C–I) Segmentation maps, $k = 10$. (C) Straightforward clustering of spectra. (D–E) SA method. (G–I) SASA method.

4 DISCUSSION

Clustering algorithm: $k$-means was selected as a clustering algorithm after the FastMap projection because it is a fast and reliable algorithm. Moreover, $k$-means optimizes the euclidean distances between the points, see e.g. Chapter 14.3.6 of (Hastie et al., 2009) (publicly available online). Thus, performing $k$-means in the space after FastMap projection (for both SA and SASA) is equivalent to minimize the within-point scatter between spectra where the distance between two spectra is calculated using Equations (1) or (6).

No mass-wise processing: in Alexandrov et al. (2010), denoising of each gray-scale image corresponding to a mass (channel) selected after peak picking is performed. Naturally, a channel-wise processing may be criticized as being prone to lose information presented in a combination of channels. In our methods SA and SASA, we never do channel-wise processing but consider the full spectra.

FastMap dimension $q$: as discussed in Section 3.1.6, the FastMap dimension $q$ is the most tricky parameter. We have done a computational study investigating the properties of the FastMap projection and observed that increase of $q$ changes the distances between projections, but only until some value. After this value, the distances between projections stay almost unchanged. Investigating this question can lead to a way of choosing $q$, and, more generally, to the way of finding the intrinsic dimension of a set of points.

Evaluation: the evaluation of produced results is an important problem, especially in an unsupervised framework, where no simple criterion (like total recognition rate) can be computed. We have tested the silhouette criterion (Rousseeuw, 1987) of separation between found clusters but have not found correspondence between the value of criterion and the visual quality of the maps. Probably, this might be explained by no clear separation between clusters.

In general, the evaluation of a spatial segmentation remains an important and unsolved problem in imaging mass spectrometry data processing, where no reference or simulated data are provided yet. Other publications on IMS segmentation, e.g. (Alexandrov et al., 2010; Deininger et al., 2008), do not consider this question and present their methods as a data mining tool with extensive support from histologists or biologists estimating the quality of the produced segmentation maps. This is explained by the novelty of this problem and the lack of existing problems and research groups solving this problem. Certainly, in the nearest future the formal evaluation will be necessary, in particular for comparing results of different segmentation. However, any formal evaluation is a complicated task since it requires the ground truth maps for a complex enough dataset. We are working in this direction and hope to present some results soon.

In this article, which is based mostly on work by Alexandrov et al. (2010), our aim was to construct an efficient algorithm delivering segmentation maps of at least comparable quality for the datasets from Alexandrov et al. (2010) where comparison is done visually...
taking into account the anatomical or histological structure of the tissue sample.

Application to other hyper-spectral or multi-channel data: the proposed methods can be applied to other IMS modalities, especially those with with strong pixel-to-pixel variability (e.g. MALDI ion source, SIMS). Moreover, the SASA method can be of use for processing MALDI-IMS data obtained with the recent FT/ICR mass analyzer (Cornett et al., 2008) which produces spectra much longer than MALDI-TOF-IMS (considered in this article), of order 10^6. Recall that the memory space and computational complexity of the SASA method does not depend on the dimensionality p of spectra but only on number of spectra n and the FastMap dimension q.

Our spatial segmentation methods can be applied for segmenting other hyper-spectral or multi-channel data, for example for terahertz imaging (Brun et al., 2010), or hyper-spectral imaging (Tarabalka et al., 2010) like that one used in German Hyperspectral Satellite Mission (Stufler et al., 2007). The SASA method is recommended, since for this type of data usually no peak picking is performed and the dimensionality p of data is high. Moreover, in the SASA method, more appropriate distance between spectra can be selected instead of the euclidean distance in Equation (6).

3D imaging mass spectrometry: our methods can be used in spatially 3D IMS, where a spectrum is measured for a voxel with spatial coordinates (x, y, z). In this case, the number of spectra increases another order of magnitude reaching n = 10^p. For such n, pure distance-based methods which need to keep the full distance matrix in memory become inappropriate in contrast to our methods which are linear in n in memory space.

ACKNOWLEDGEMENTS

We gratefully thank Prof. Herbert Thiele (Bruker Daltonik) for providing the rat brain dataset, Prof. Ferdinand von Eggeling (University Hospital Jena, Germany) for the tumor dataset, Michael Becker and Sören Deininger (both Bruker Daltonik) for doing IMS measurements and additionally Michael for drawing the schematic of the rat brain. We also thank Chen Sagiv (SagivTech Ltd., Israel) for introducing the bilateral filtering, and Oliver Keszöcze of the rat brain. We also thank Chen Sagiv (SagivTech Ltd., Israel) for the tumor dataset, Michael McCombie, T. Alexandrov and J.H. Kobarg providing the rat brain dataset.

REFERENCES


