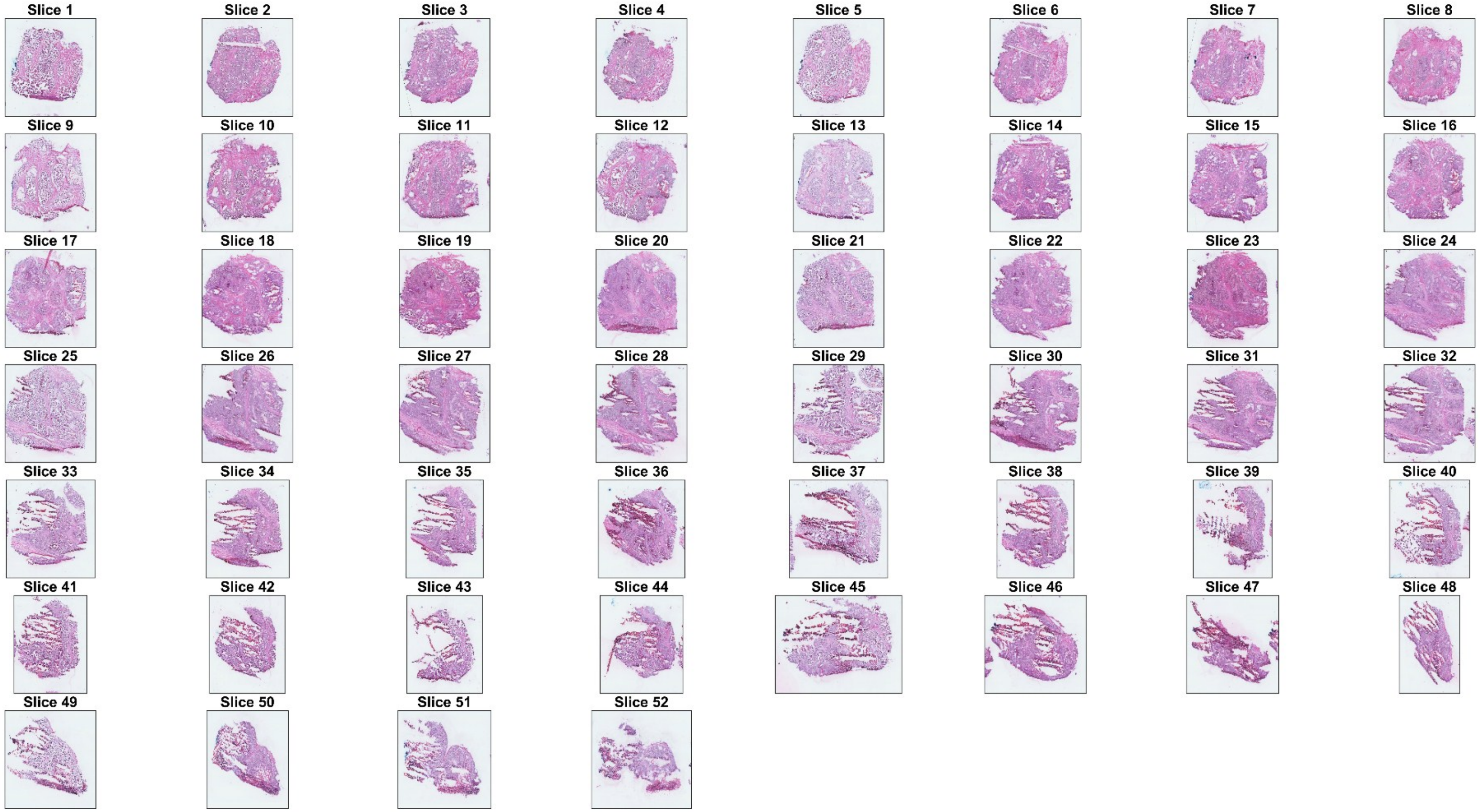
Supplementary Materials



Supplementary Fig 1 - Sequence of the H&E stained tissue sections from the human colorectal adenocarcinoma biopsy.

Parametric t-SNE

Parametric t-SNE is an unsupervised dimensionality reduction technique based on a (deep) neural network topology in which the deepest layer consists of a t-SNE feed-forward network[1]. The objective of this technique is to define a non-linear mapping between the high-dimensional original feature space and a low-dimensional (often 2-dimensional) latent space where the data points are placed according to their mutual similarities in the high-dimensional space. The topology of parametric t-SNE consists of a deep encoder that projects the data to a lower dimensional space, followed by a t-SNE mapping. The large number of weights (several millions) represents the main difficulty of training deep neural networks, because back propagation may get stuck in poor local minima that depend strongly on the initial values of the weights. An approach which has been shown to successfully overcome this problem is based on a greedy layer-wise training procedure[2, 3] where the deep neural network is seen as a combination of simpler neural networks. In the case of parametric t-SNE, restricted Boltzmann machines (RBM) represent these building blocks. The training procedure consists of four steps: (1) multiple RBMs are trained to reconstruct the input data, and their hidden layers are used as the input layers of the successive RBMs, (2) RBMs are stacked together and unfolded to generate a deep autoencoder, (3) weights are learnt to reconstruct the input data, (4) the encoder with a t-SNE layer added on top is fine-tuned with back-propagation to minimise the objective function.

Pre-training. A greedy layer-wise pre-training step consists of stacking together a set of RBMs. The procedure is performed through a greedy layer-wise unsupervised learning algorithm. A set of RBMs is iteratively trained to reconstruct the input, and hidden layers are used as input for the successive neural network (Supplementary Fig 2-A, 2-B). After training, all the RBMs are stacked together to generate a deep neural network (Supplementary Fig 2-C), which is fine-tuned to reconstruct the input through back-propagation.

Restricted Boltzmann Machine. A RBM is a generative stochastic neural network consisting of a visible and a hidden layer with a symmetric connection between them. The nodes of an RBM are usually Bernoulli distributed, but can be extended to being Gaussian distributed[4]. Let W=(w_{ij}) be the (m x n)-dimensional weight matrix, v_i and h_j the visible and the hidden activations, and b_i and c_j the respective biases. Since an RBM can be seen as a Markov Random Field, it can be associated to an energy function:

$$E(v,h) = -\sum_{i=1}^{n} b_i v_i - \sum_{j=1}^{m} c_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i w_{ij} h_j$$

from which the probability distributions for the hidden and visible states are defined as

$$P(v,h) = \frac{1}{Z} e^{-E(v,h)}$$

where Z represents the partition function. Activation functions are usually non-linear, such as the logistic sigmoid. The learning approach of RBMs is performed through a single-step contrastive divergence (CD-1) method[5]. In practice, the input activations generate the hidden activation through the transfer function σ (ex. logistic sigmoid),

$$h_j = \sigma\left(\sum_{i=1}^{m} w_{ij} v_i + c_j\right).$$

Therefore, the reconstructions of visible and hidden activations may be calculated,

$$v'_i = \sigma\left(\sum_{j=1}^{n} w_{ij}h_j + b_i\right)$$

$$h'_j = \sigma\left(\sum_{i=1}^{m} w_{ij}v'_i + c_j\right)$$

where i∈{1, …, m} and j∈{1, …, n}.

The weight matrix is updated as follows,

$$\Delta w_{ij} = \epsilon\left(v_i h_j - v'_i h'_j\right)$$

where the variables v', h' correspond to the reconstructed activations from the hidden and visible layers respectively, and ϵ is the learning rate. Bias updates are defined analogously. When batch learning is employed, the average of the product between visible and hidden activations over the batch is used for the calculation of the updated weights.

Fine-tuning. After pre-training RBMs independently, they are stacked together, and fine-tuning is performed by adding on top of the inner layer a t-SNE[6] non-linear mapping (Supplementary Fig 2-D). This ensures that during the fine-tuning process the network will learn a low-dimensional manifold in which the spectra that were similar in the original high-dimensional space are placed close together. For this purpose, the pair-wise distances in the original high-dimensional space and the low-dimensional latent space are converted into probabilities, defined by an isotropic Gaussian distribution centred on each data sample, and computing the density of other data samples under this distribution. The conditional probabilities are defined as

$$p\left(x_j|x_i\right) = \frac{\exp\left(-\|x_i - x_j\|^2/2\sigma_i^2\right)}{\sum_{k \neq i}\exp\left(-\|x_i - x_k\|^2/2\sigma_i^2\right)}$$

$$p\left(x_i|x_i\right) = 0,$$

where the variance σ of the Gaussian is set specifically so that the *perplexity* of the conditional distributions is constant. The perplexity is a free parameter that can be interpreted as the number of neighbours considered for each data sample. The conditional distribution is symmetrised

$$p(x_i,x_j) = \frac{p(x_j|x_i) + p(x_i|x_j)}{2n}$$

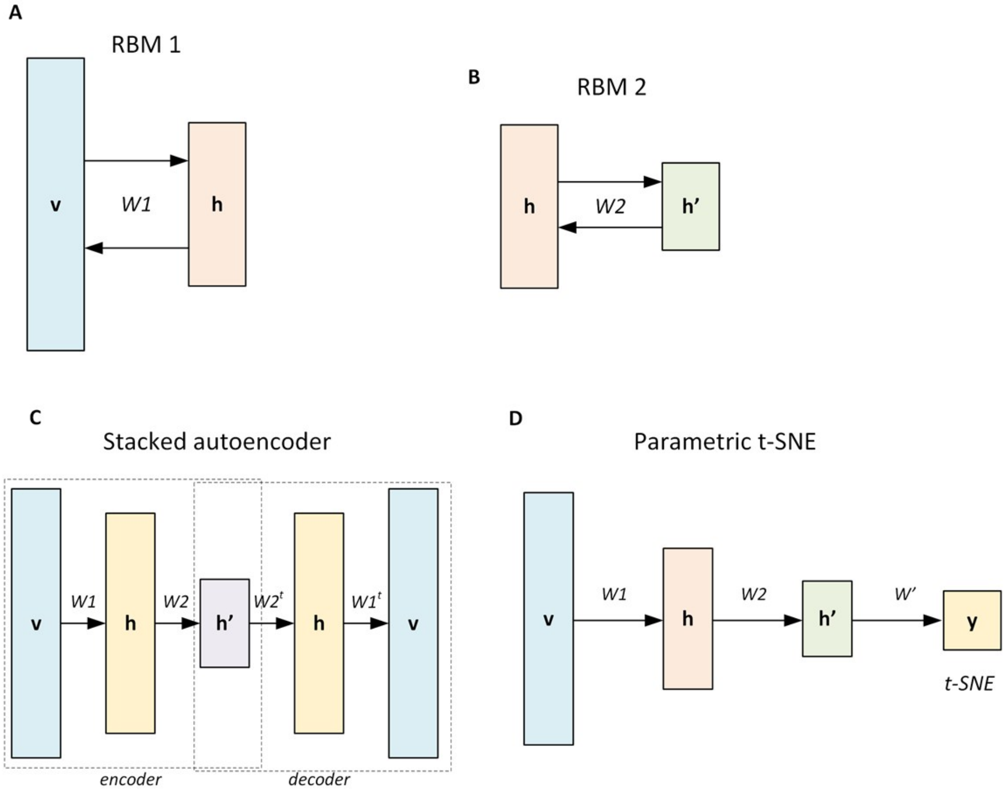
Then, p can be interpreted as the similarity measure between the data samples and, setting a small perplexity value, the local structure can be captured. In the latent space, the similarity of the low-dimensional data samples is based on a Student's t-distribution

$$q(x_i,x_j) = \frac{\left(1 + \|f(x_i|W) - f(x_j|W)\|^2/\alpha\right)^{-\frac{\alpha+1}{2}}}{\sum_{k \neq l}\left(1 + \|f(x_k|W) - f(x_l|W)\|^2/\alpha\right)^{-\frac{\alpha+1}{2}}}$$

where $f(x_i|W)$ is the representation of the data sample x$_i$ through the feed-forward neural network, an α is the number of degrees of freedom of the Student's t-distribution. The learning process is performed through back propagation to minimise the Kullback-Leibler divergence between p and q,

$$KL(P||Q) = \sum_{i \neq j} p(x_i,x_j)\log\frac{p(x_i,x_j)}{q(x_i,x_j)}$$

using batches (of thousands) of samples.



Supplementary Fig 2 - Example of a 4-layer parametric t-SNE model. Two RBMs (A-B) are pre-trained through contrastive divergence. The hidden layer of the first RBM is used as input for the second RBM (B). A stacked autoencoder is defined combining the RBMs (C). Fine-tuning of network weights and biases is performed after adding a t-SNE layer on top (D) of the encoder by backpropagation.

OPTICS

Density based clustering. Density-based clusters are defined by objects where neighbours of radius ε contain at least a minimum number of data points MinPts. In order to define the clusters, some definitions are required. A data point p is defined as directly density-reachable from another object q if p belongs to the ε-neighbourhood $N_\varepsilon(q)$ of q and its cardinality, $|N_\varepsilon(x)|$, is larger than MinPts. Those objects that satisfy the second property are labelled as *core*. Two objects p and o are said to be *density-reachable* if there is a chain of objects that are all directly density-reachable with respect to ε and

MinPts. Finally, an object p is *density-connected* to data point q if there is an object o such that both p and q are density-reachable from o.

A *density-based cluster* is defined as a subset C of D of density-connected data points which satisfy the following conditions

Maximality: ∀p,q∈D: if p∈C and q is *density-reachable* from p wrt ε and MinPts, then q∈C.

Connectivity: ∀p,q∈C: p∈C is *density-connected* to q wrt ε and MinPts.

All the points not contained in any clusters are labelled as *noise*.

OPTICS algorithm. In OPTICS, given a dataset D, the data points are characterised by two values: the *core-distance* and the *reachability-distance*. Let p be a data point from D, and let ε' be the minimum radius of the closed-ball containing MinPts neighbours. Then, the *core-distance* is defined as

$$\text{core - } distance_{\varepsilon,MinPts}(p) = \begin{cases} UNDEFINED & if\ |N_{\varepsilon}(p)| < MinPts \\ \varepsilon' & otherwise \end{cases}$$

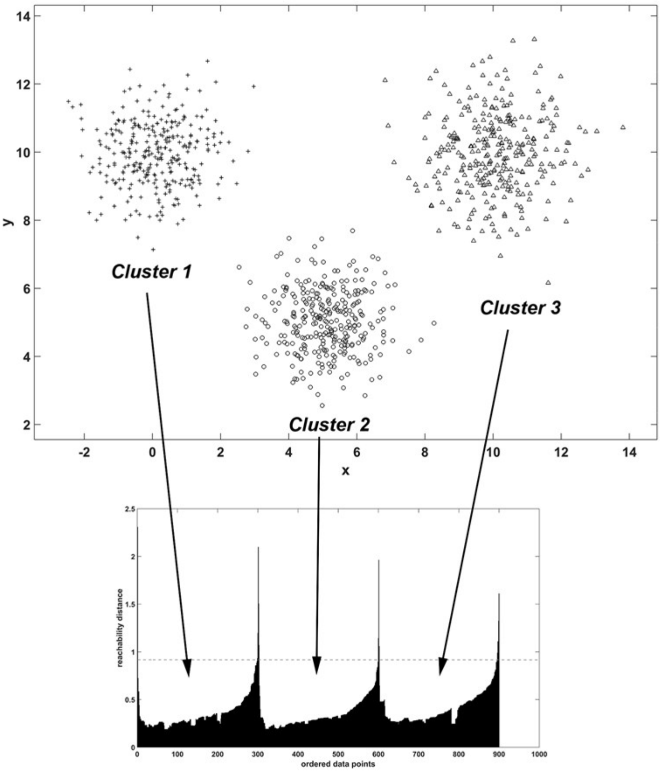
The reachability-distance between two data points p and o from D is defined as follows. Let $N_{\varepsilon}(o)$ be the ε-neighbourhood of o. Then the reachability-distance of p with respect to o is defined as

$$\text{reachability - } distance_{\varepsilon,MinPts}(p,o) = \begin{cases} UNDEFINED \\ max(\text{core - distance}_{\varepsilon,MinPts}(p),\text{distance}(o,p)) \end{cases}$$

The algorithm first orders the data points following the *ExpandClusterOrder* procedure. This procedure retrieves the ε-neighbourhood of a data point, sets its reachability-distance to *UNDEFINED* and calculates the core-distance. The data point is then pushed into an ordered list. If the data point is a core object (there are at least MinPts data points in its ε-neighbourhood), then all the directly density-reachable data points are pushed into an *OrderSeeds* list. These objects are sorted by their reachability-distance to the closest core object. At each iteration, the object from OrderSeeds having the smallest reachability-distance is selected. The ε-neighbourhood and its core-distance are calculated and are added to the ordered list together with its core-distance and its current reachability-distance. If the current object is itself a core object, then further candidates are added to the OrderSeeds list.

The ordered list of objects together with their reachability-distance can then be clustered using MinPts and a clustering-distance ε’≤ε.

The visualisation of the reachability-distances through the *reachability plot* permits the identification of structures in the data, and partitions can be generated assigning a particular threshold to the reachability-distance. Furthermore, the reachability plot can be seen as a special case of a dendrogram. An example of the application of OPTICS on a simulated dataset is shown in Supplementary Fig 3.



Supplementary Fig 3 - An example of clustering using OPTICS. The three clusters (top) are identified in the reachability-plot (bottom) by setting an opportune threshold value for the reachability distance (represented by the grey dashed line).

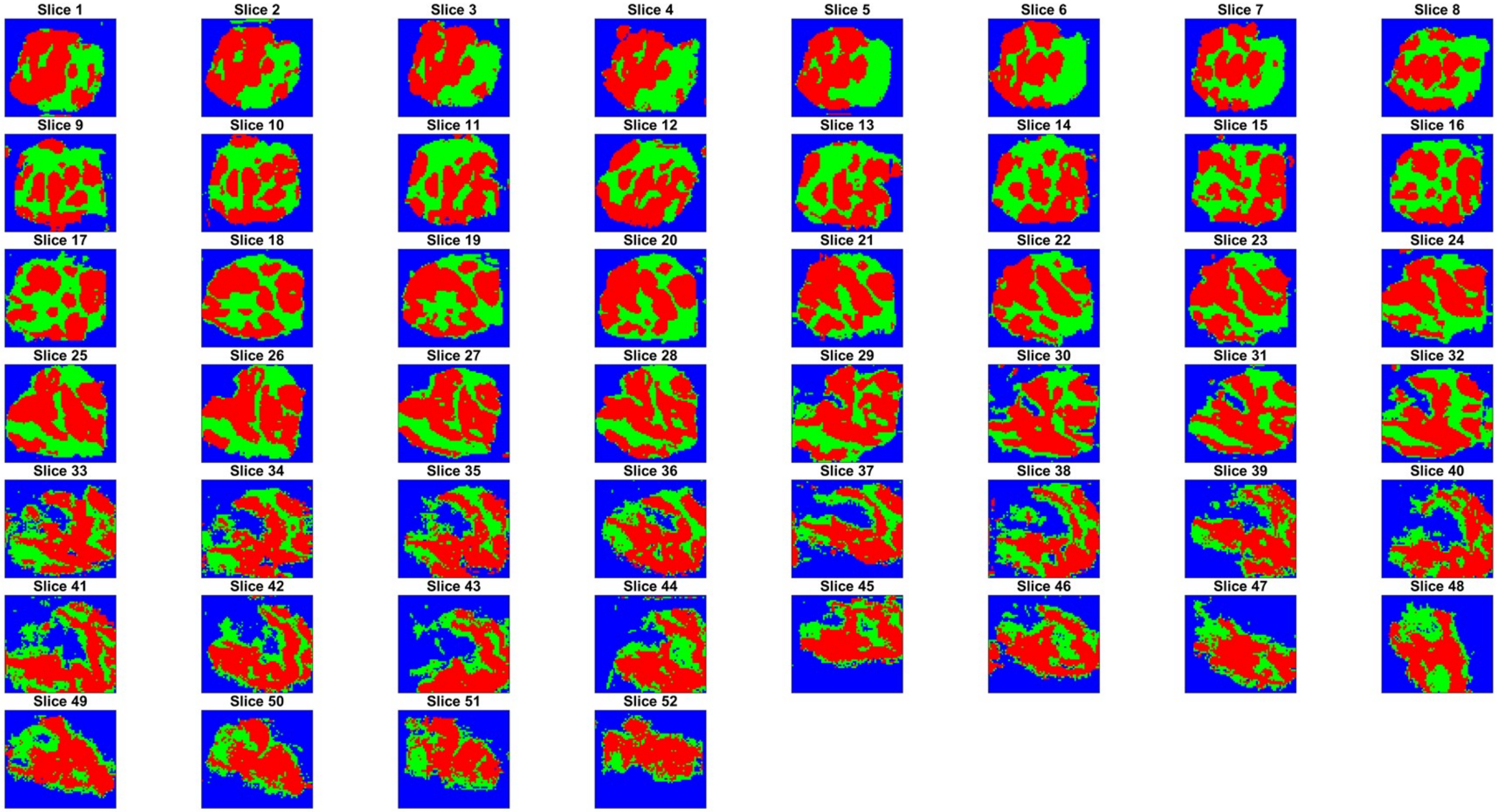
Random Forest:	
Number trees:	400
Min. leaf size	1

Stacked sparse autoencoder:	
Topology:	391-250-50-3
Activation function:	Logistic sigmoid
Max epochs pre-training:	1000
Max epochs fine-tuning:	1000
Loss function (pre-tr. and fine tun.):	Sparse MSE
Weight regularization:	0.001
Sparsity proportion:	0.05
Sparsity regularization:	1
Back propagation:	Conjugate gradient

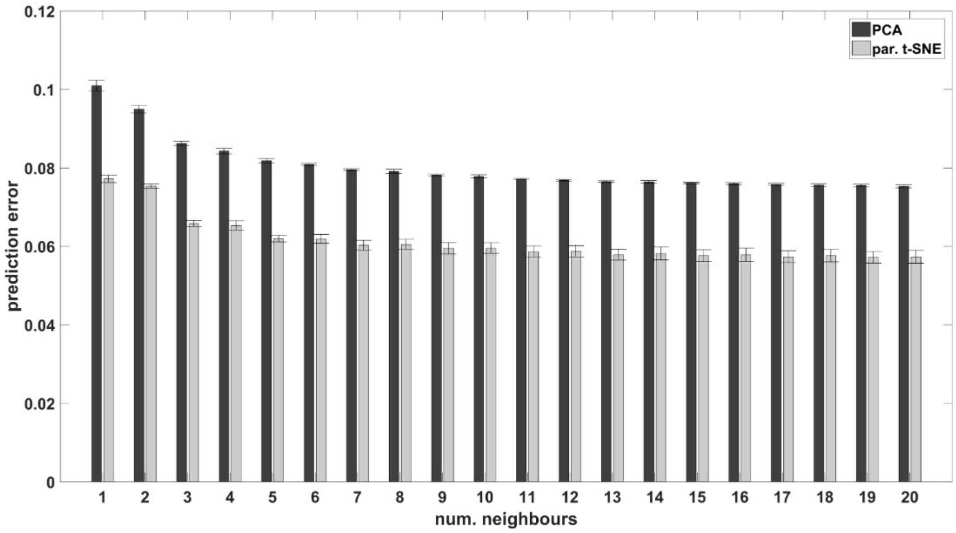
Supplementary Table 1 – Parameters for Random forest and stacked sparse autoencoder classifiers. Also, linear SVM and MMC-LDA were tested for supervised segmentation.

Method	***Mean accuracy +/- st. dev.***
Linear SVM	**0.99976 (0.00025)**
SSAE	0.99946 (0.00025)
Random Forest	0.99940 (0.00042)
MMC-LDA	0.99443 (0.00151)

Supplementary Table 2 – Results of the supervised segmentation. Four classifiers were tested with a 30% hold-out cross-validation. Mean accuracy and its standard deviation were calculated over 5 repetitions.



Supplementary Fig 4 - Results of the linear SVM classification on the entire dataset. The three main classes are reported: tumour (red), healthy (green), background (blue). The segmented regions are compatible with contiguous tissue slices.



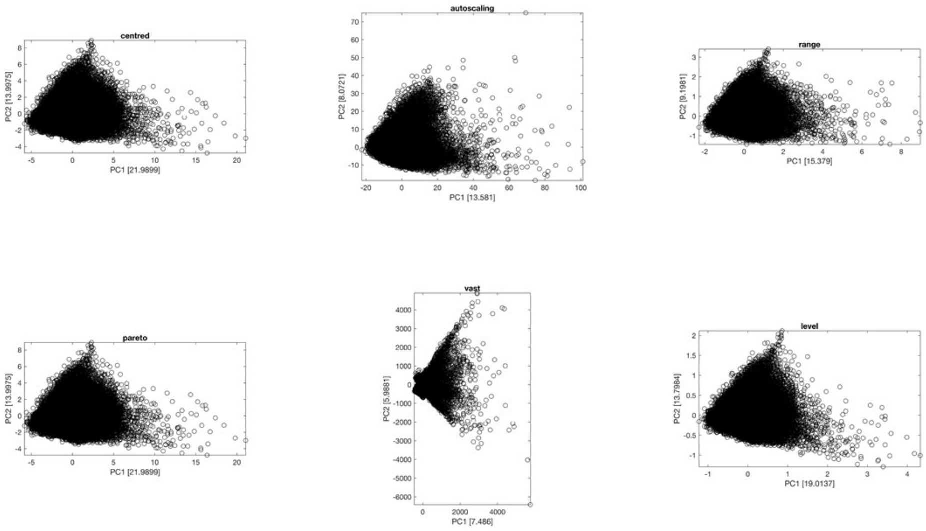
Supplementary Fig 5 - Prediction error for the k-NN classifier of the 2-dimensional data points. Average prediction errors and their standard deviations are calculated over 5 repetitions. For all the tested values of k varying in the range of 1-20 the prediction error of the 2-dimensional parametric t-SNE representation was lower than that of the 2 first principal components scores. This result confirms that the 2-dimensional parametric t-SNE data points are mapped closer if their spectral profiles are similar in the original high- dimensional space.

Scaling method	***Mean trustworthiness***	***St. Dev. Trustworthiness***
Centring	**0.78257**	**0.00170**
Autoscaling	0.74378	0.00287
Range scaling	0.76172	0.00694
Pareto Scaling	**0.78257**	**0.00170**
Vast scaling	0.68432	0.03424
Level scaling	0.75520	0.01319

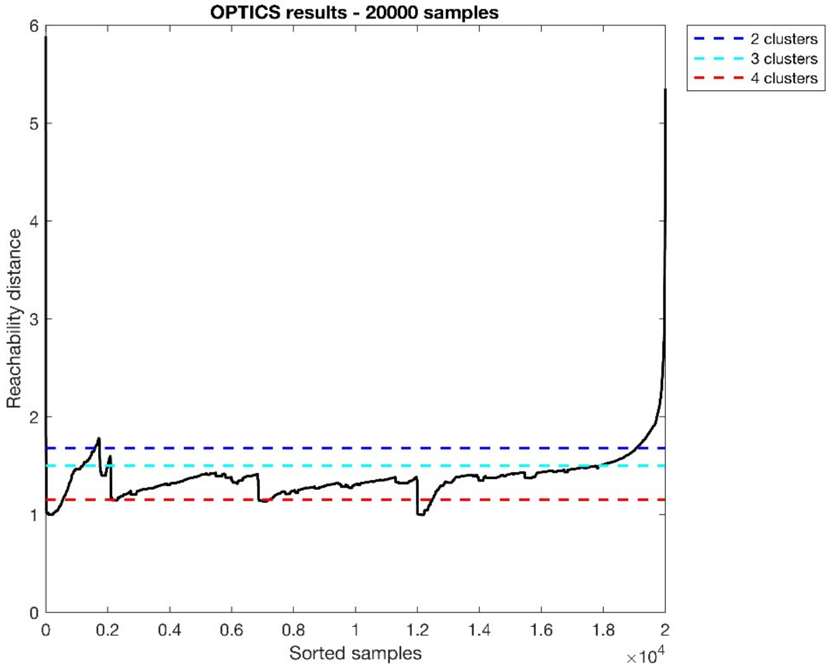
Parametric t-SNE trustworthiness: mean = 0.82396, st.dev = 0.00514

Supplementary Table 3 – Mean trustworthiness and its standard deviation for the scores of the first 2 principal components using different scaling methods, compared with the results of parametric t-SNE. In all the cases, parametric t-SNE provides a more faithful low-

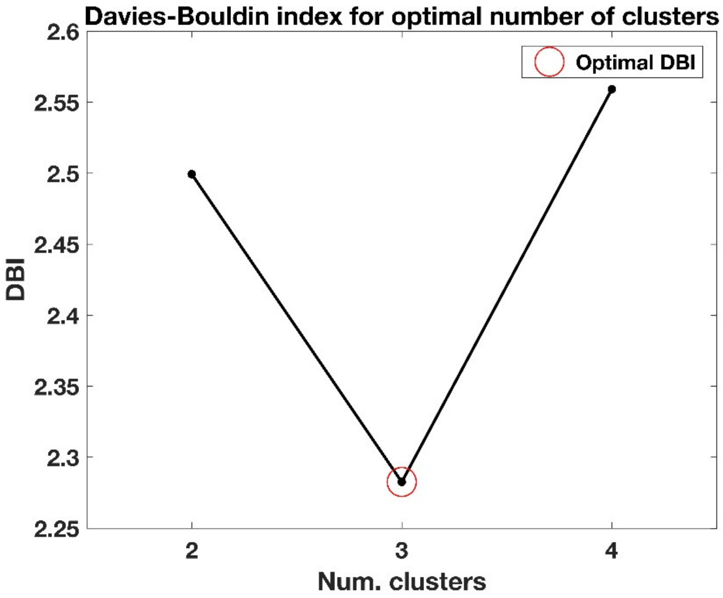
dimensional representation of the high-dimensional similarity relationships.



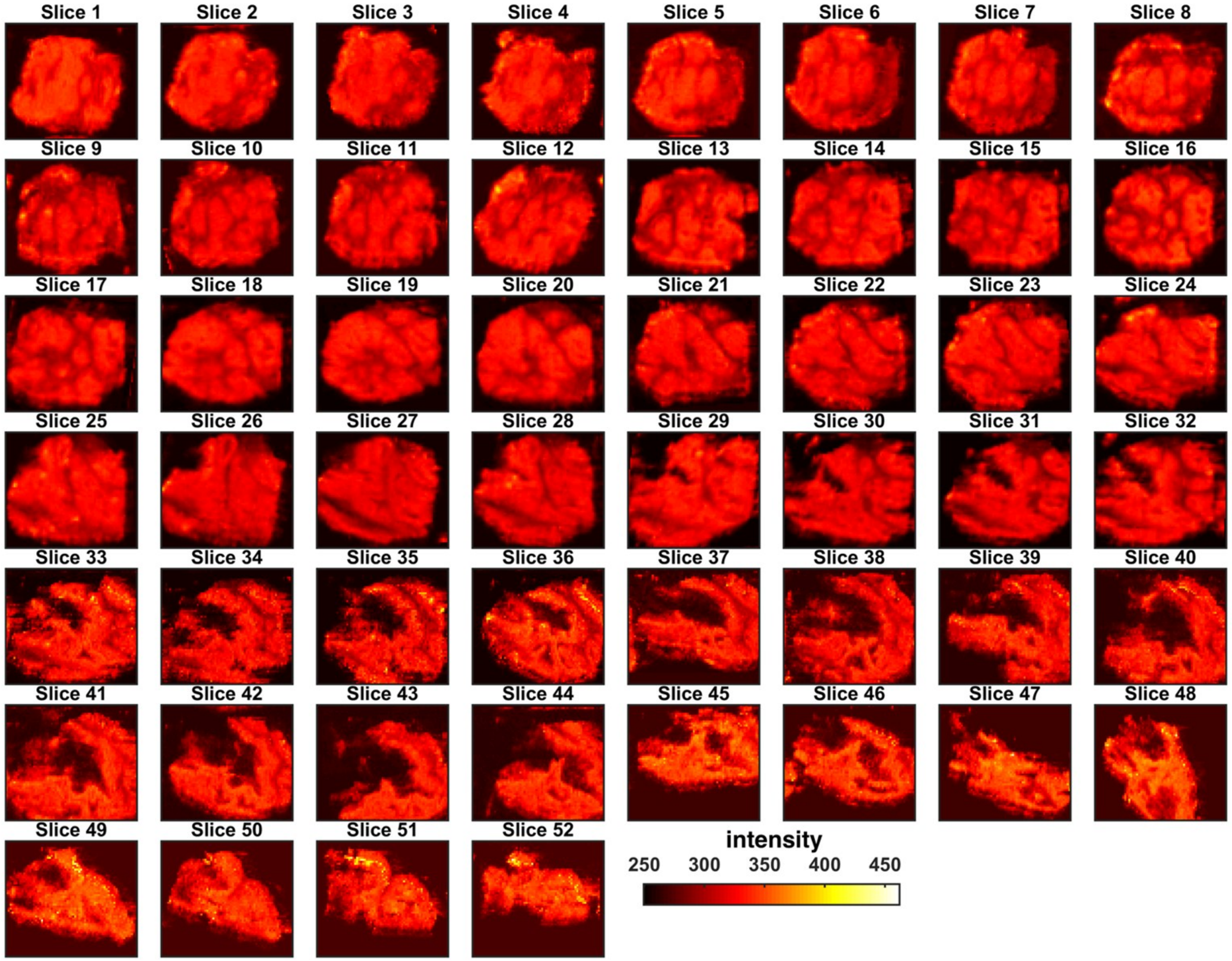
Supplementary Fig 6 – Scatter plot of the first 2 principal components scores after applying different data scaling method. No clusters are visible.



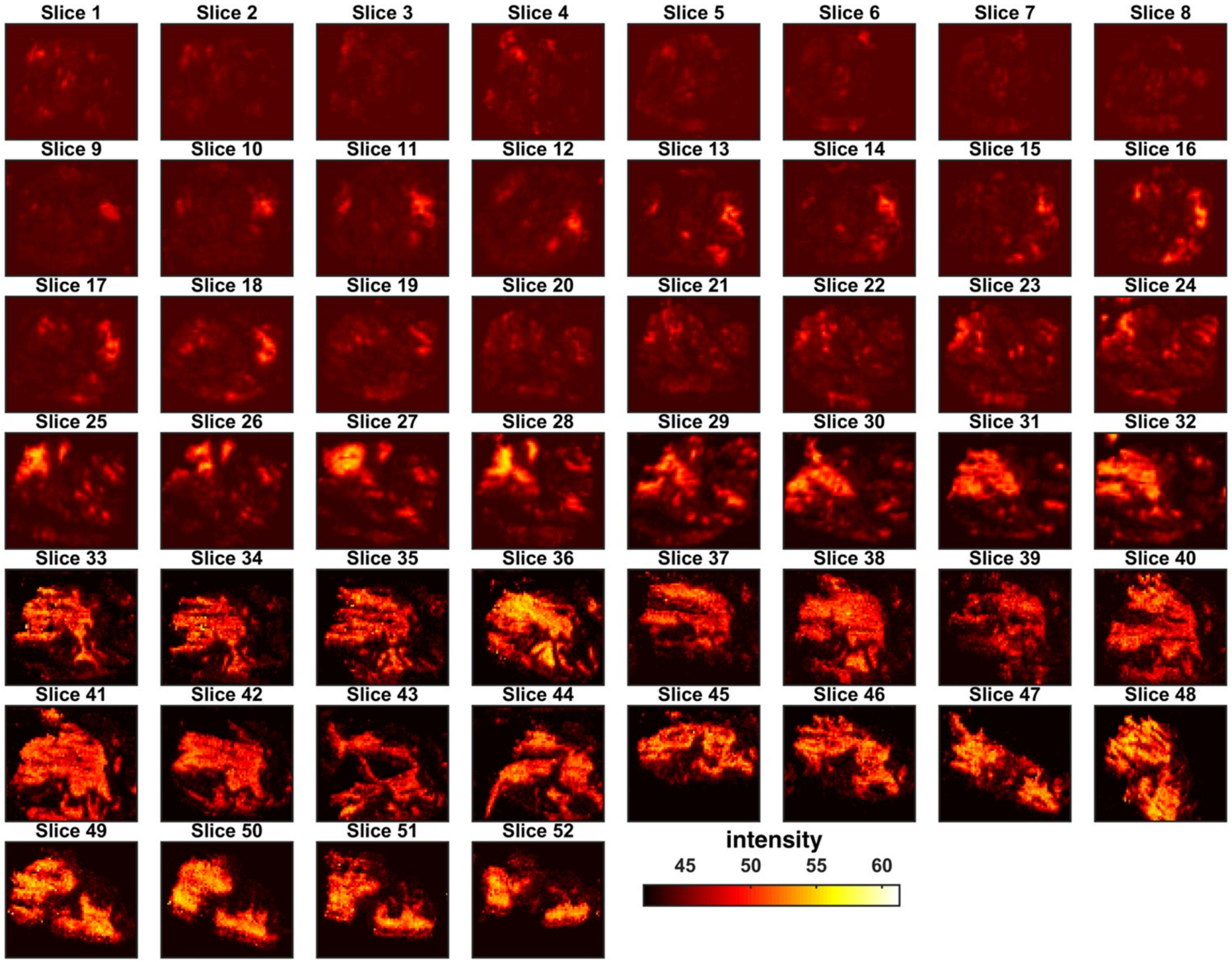
Supplementary Fig 7 – Reachability plot for 20,000 randomly selected 2-dimensional data points. A MinPts of 200 was used. Candidate partitions with 2, 3 and 4 clusters are identified through the reachability distance.



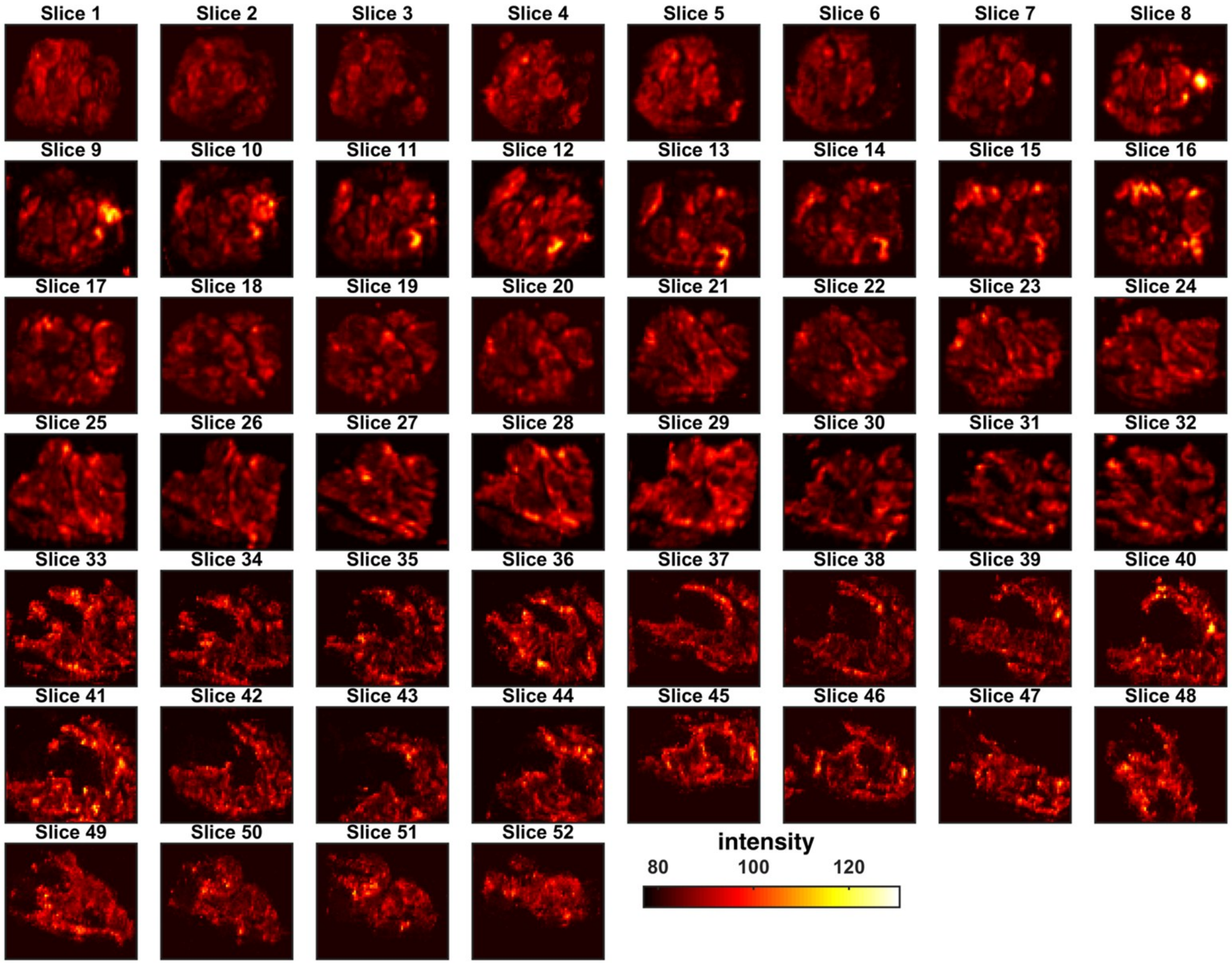
Supplementary Fig 8 – Davies-Bouldin indices corresponding to the 3 candidate partitions. The minimum value is reached with 3 clusters.



Supplementary Fig 9 – SSI images corresponding to the first sub-network.



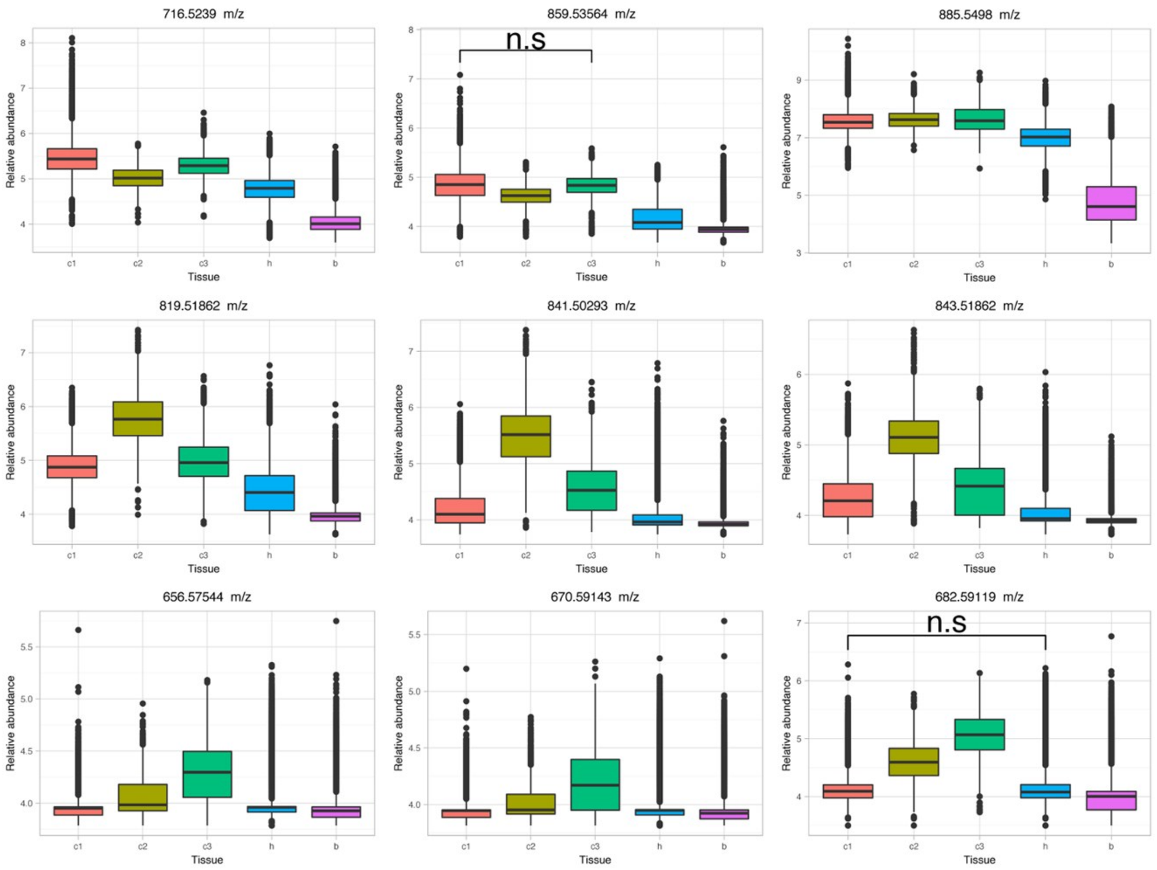
Supplementary Fig 10 - SSI images corresponding to the second sub-network.



Supplementary Fig 11 - SSI images corresponding to the third sub-network.

Query m/z	Percentage
885.5458	65.45%
859.5355	62.27%
716.5239	46.63%
744.5542	34.45%
768.5544	50.96%
770.5745	60.48%
698.5137	64.15%
740.5241	61.92%
714.5038	96.60%
722.5139	41.19%
819.515	77.96%
841.5053	80.74%
843.5153	84.05%
817.505	66.64%
796.5248	78.65%
793.5047	73.21%
865.5055	80.39%
845.5353	90.93%
821.535	66.89%
869.5356	93.16%
682.5935	81.34%
670.5933	93.81%
656.5732	92.82%
684.6035	91.98%
646.6131	65.01%
620.6028	98.84%
702.5437	78.74%

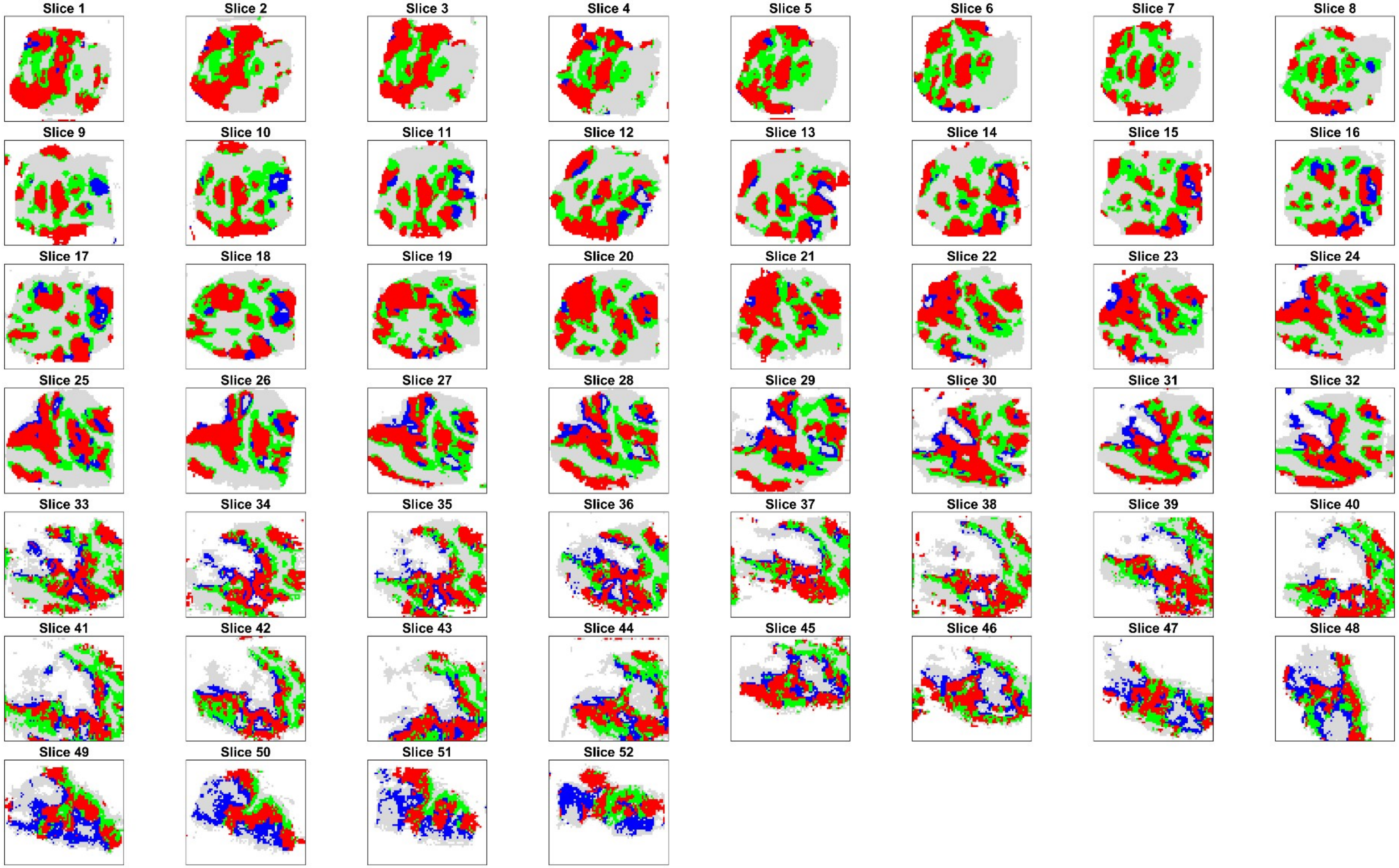
Supplementary Table 4 – Percentage of peaks found in the search window of +/- 5 ppm corresponding to the sub-networks top ions. Left column represents the common m/z value that was searched in the raw data, and the second column represents the percentage of times at least one peak was found in the window +/- 5 ppm.



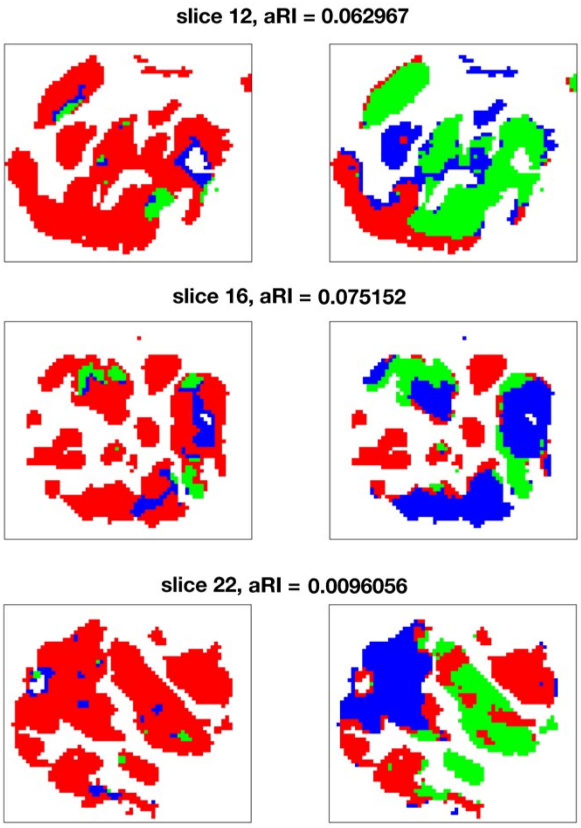
Supplementary Fig 12 - Box plots of 3 ions with largest degree values in each sub-network. Multiple comparison Dunn's test shows that ions from first sub-network (first row) are more ubiquitous in the entire tumour region, with 859.5356 m/z being expressed in both the first (c1) and third cluster (c3). Ions from the other two sub-networks instead are significantly more abundant in cluster 2 (c2) and 3 respectively. All the ions are less abundant in healthy (h) tissue and background (b). All the pairwise tests result in a Bonferroni-Hochberg corrected p-value < 0.05, except those reported with n.s. where p-value was not significant.

	Sub network 1 [# nodes = 66]				Sub network 2 [# nodes = 20]				Sub network 3 [# nodes = 11]			
rank	**m/z [Da]**	**error [ppm]**	**name**	**Degree**	**m/z [Da]**	**error [ppm]**	**name**	**Degree**	**m/z [Da]**	**error [ppm]**	**name**	**Degree**
1	885.5498	0	PI(38:4) [M-H]-	15	819.5186	0	PG(40:7) [M-H]-	16	682.5912	0	Cer(d18:1/24:1) [M+Cl]-	10
2	859.5356	1	PI(36:3) [M-H]-	14	841.5029	0	PG(42:8) [M-H]-	15	670.5914	0	Cer(d18:1/23:0) [M+Cl]-	6
3	716.5239	0	PE(37:1) [M-H]-	13	843.5186	0	PG(42:9) [M-H]-	15	656.5754	0	Cer(d18:1/22:0) [M+Cl]-	6
4	744.5554	0	PE(36:1) [M-H]-	12	817.5031	0 3	PG(40:8) [M-H]- PI(O-31:0) [M+Cl]-	14	684.6063	0	Cer(d18:1/24:0) [M+Cl]-	6
5	768.5557	1	PE(35:3) [M-H]-	11	796.5223	5	PS(O-35:1) [M+Cl]-, PS(P-35:0) [M+Cl]-	13	646.6146	0	Cer(d18:1/24:1) [M-H]-	2
6	770.5712	0	PE(38:2) [M-H]-	11	793.5032	0	PG(38:6) [M-H]-	11	620.6001	2	Cer(d18:1/22:0) [M-H]-, Cer(d18:0/22:1) [M-H]-, Cer(d14:1/26:0), [M-H], Cer(d16:1/24:0) [M-H]-	1
7	698.5133	0	PE(37:2) PE(O-34:3) [M-H]-, PE(P-34:2) [M-H]-	10	865.5031	0	PG(44:12) [M-H]-	11	702.5443	0	PE(O-34:1) [M-H]-, PE(P-34:0) [M-H]-	1
8	740.5243	0	PE(33:3) [M-H]-	10	845.5351	1	PG(42:8) [M-H]-	10	-	-	-	-
9	714.5070	1	PE(34:2) [M-H]-	9	821.5351	1	PG(40:6) [M-H]-	7	-	-	-	-
10	722.5134	0	PE(36:4) [M-H]-	9	869.5350	1 3	PG(44:8) [M-H]- PI(O-35:2) [M+Cl]- PI(P-35:1) [M+Cl]-	7	-	-	-	-

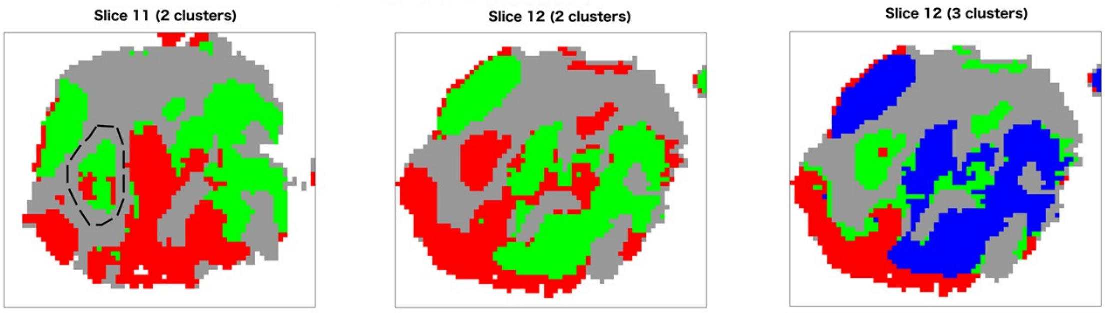
Supplementary Table 5 - Identified molecules for the top 10 ions in the three sub-networks corresponding to the three OPTICS clusters. Ions are ranked accordingly with a descent value of their node degree value (Column "Degree"). In cluster 3 only 7 of the 11 molecules belonging to the cluster were identified. The ions annotation was performed using the Lipid maps online search engine. It is evident that cluster 1 is characterised by an abundance of phosphoethanolammines, whereas the cluster 2 is characterised by an abundance of phosphatidylglycerols and the cluster 3 is characterised by an abundance of ceramides.



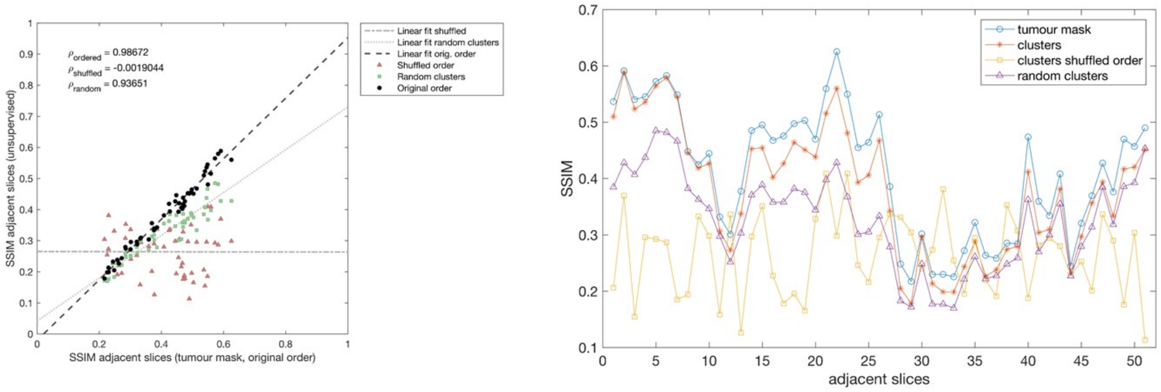
Supplementary Fig 13 – Results of k-means with 3 clusters (correlation distance), similarity with clusters found by OPTICS results in aRI = 0.2647. OPTICS clusters 2 and 3 are here combined in one cluster (ex. blue cluster in Slice 10 and 16), instead they are found to be distinct by co-expression network analysis.



Supplementary Fig 14 – Example of 3 slices where the clusters found in the 3D dataset (left) were significantly different from those found when analysing the single 2D slice (right). In all the cases, aRI confirmed these results.



Supplementary Fig 15 – Example of clusters found by parametric t-SNE+OPTICS analysing individual 2D slices. The 2 optimal clusters found in the slice 11 are not topologically compatible with the candidate partitions (2 and 3 clusters) found in the slice 12. The region *delineated by the dashed line is assigned to a completely different cluster in the slice 12.*



Supplementary Fig 16 – Analysis of the SSIM sequences. Left: the OPTICS clusters show a highly correlated SSIM sequence with that of the entire tumour region, whereas the shuffled order clusters result in a poor correlation. The sequence generated by the 3 randomly assigned clusters is still highly correlated with that of the entire tumour because they share the overall shape, but its values are significantly lower than those of OPTICS clusters (right) because, being the clusters labels randomly assigned, the internal structures are not preserved in the adjacent slices.

References

1. L. van der Maaten, *RBM*, 2009, **500**, 500.
2. Y. Bengio, P. Lamblin, D. Popovici and H. Larochelle, *Advances in neural information processing systems*, 2007, **19**, 153.
3. G. E. Hinton, S. Osindero and Y.-W. Teh, *Neural Computation*, 2006, **18**, 1527-1554.
4. G. Hinton, *Momentum*, 2010, **9**, 926.
5. M. A. Carreira-Perpinan and G. Hinton, *AISTATS*, Citeseer, 2005.
6. L. Van der Maaten and G. Hinton, *Journal of Machine Learning Research*, 2008, **9**, 85.