SUPPORTING INFORMATION

Enhancing the Throughput of FT Mass Spectrometry Imaging Using Joint Compressed Sensing and Subspace Modeling

Yuxuan Richard Xie[1][4], Daniel C. Castro[3][4], Stanislav S. Rubakhin[2][4], Jonathan V. Sweedler[1][2][3][4]*, Fan Lam[1][4]*

[1] Department of Bioengineering, [2] Department of Chemistry, [3] Department of Molecular and Integrative Physiology, [4] Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States. *Corresponding Authors.

Table of Contents

Supporting Methods

Supporting Figures

SUPPORTING METHODS

Computational Details

All computational experiments presented in this paper was conducted on a standard PC setup with an Intel i7-8700K CPU and 64 GBs memory. Algorithms were implemented in Python 3.8.2 with Numpy 1.19.5, Scikit-learn 0.24.2 and Scipy 1.6.3 open-source packages.

Model and algorithm.

The subspace model

The subspace model assumes that each transient can be accurately approximated by a linear combination of a smaller number of basis functions (or basis transients) $\{\phi_l(t)\}_{l=1}^{L}$ with pixel dependent spatial coefficients $\{c_l(r)\}_{l=1}^{L}$ (r the pixel location), where L is the model order (i.e., the number of basis functions). Accordingly, the entire can then be expressed as:

$$s(r,t) = \sum_{l=1}^{L} c_l(r)\phi_l(t). \qquad (2)$$

Note that the basis transients derived from high-resolution data are of the same dimensionality with the same number of temporal data points as the standard long transients, denoted as N_T, thus providing the same mass resolving power. With $\{\phi_l(t)\}$, spatial coefficients can be effectively determined using a much smaller number of temporal data points $N_{T'}$ (with $L < N_{T'} \ll N_T$) by solving:

$$\hat{c} = \underset{c}{\operatorname{argmin}}\ ||s' - c\boldsymbol{\Phi}'||_2^2 + \alpha||c||^2 \qquad (3)$$

where s' is the measured short transient with $N_{T'}$ points, $\boldsymbol{\Phi}'$ is the $L \times N_{T'}$ basis matrix for fitting (truncated from the original long basis for computational and memory efficiency) , and α is a regularization parameter. Once the spatial coefficients are determined by solving eq. (3), transients with the desired resolution are reconstructed:

$$\hat{s}(r,t) = \sum_{l=1}^{L} \hat{c}_l(r)\phi_l(t) \qquad (4)$$

This formulation avoids the need to sample transients with N_T points uniformly for all pixels to achieve the target mass resolution, which is time consuming and can generate a large amount of data. Once the basis is predetermined, spatial coefficients can be estimated using short transients with $N_{T'}$ points measured by a shorter acquisition to form the final reconstruction using eq.(4). Because $N_{T'} \ll N_T$, the subspace model enables drastic reduction in data acquisition time through the short-time acquisition followed by reconstruction.

Integrating subspace model with compressed sensing (CS)

With spatial sparse sampling, all the pixels need to be jointly reconstructed in order to leverage the spatiotemporal correlation to interpolate the missing pixels effectively. To this end, we formulated a sparsity constrained subspace fitting for all pixels as follows:

$$\min_{\boldsymbol{C}} ||\boldsymbol{D} - \boldsymbol{\Omega}\boldsymbol{C}\boldsymbol{\Phi}'||_F^2 + \lambda\left\|\boldsymbol{W}\boldsymbol{M}^T\boldsymbol{C}\right\|_1 \qquad (5)$$

where $\boldsymbol{D}$ is the data matrix containing short transients with $N_{T'}$ points, $\boldsymbol{\Omega}$ is the designed binary measurement matrix with one at the sampled pixel index and zero otherwise, $\boldsymbol{C}$ is the unknown spatial coefficient matrix for all the desired pixels, $\boldsymbol{\Phi'}$ is the pre-estimated basis matrix, $\boldsymbol{W}$ is a wavelet transform operator, and $\boldsymbol{M}^T$ is an operation that arranges pixel indices into the proper spatial locations to form an image. Compared to ℓ2 regularization, ℓ1 regularization encourages the solution to be sparse in a transform domain (wavelet in this case), effectively addressing the ill-posedness of the reconstruction problem. However, solving eq. (5) requires a nonlinear optimization procedure. We propose to use the alternating direction method of multipliers (ADMM) algorithm to solve the equivalent form of eq. (5):

$$\min_{\boldsymbol{C},\boldsymbol{G}} \|\boldsymbol{D}-\boldsymbol{\Omega}\boldsymbol{C}\boldsymbol{\Phi'}\|_F^2+\lambda\|\boldsymbol{G}\|_1 \quad s.t. \quad \boldsymbol{G}=\boldsymbol{W}\boldsymbol{M}^T\boldsymbol{C} \tag{6}$$

with the augmented Lagrangian function:

$$\|\boldsymbol{D}-\boldsymbol{\Omega}\boldsymbol{C}\boldsymbol{\Phi'}\|_F^2+\lambda\|\boldsymbol{G}\|_1+\langle\boldsymbol{Y},\boldsymbol{G}-\boldsymbol{W}\boldsymbol{M}^T\boldsymbol{C}\rangle+\frac{\rho}{2}\left\|\boldsymbol{G}-\boldsymbol{W}\boldsymbol{M}^T\boldsymbol{C}\right\|_F^2 \tag{7}$$

where $\boldsymbol{G}$ is an introduced auxiliary variable, $\boldsymbol{Y}$ is the Lagrangian multiplier and ρ is a penalty parameter. Eq. (7) can then be solved by alternatively solving the following subproblems:

$$\boldsymbol{G}^{(l+1)}=\min_{\boldsymbol{G}} \lambda\|\boldsymbol{G}\|_1+\frac{\rho}{2}\left\|\boldsymbol{G}-\boldsymbol{W}\boldsymbol{M}^T\boldsymbol{C}^{(l)}+\frac{1}{\rho}\boldsymbol{Y}^{(l)}\right\|_F^2 \tag{8}$$

$$\boldsymbol{C}^{(l+1)}=\min_{\boldsymbol{C}} \lambda\|\boldsymbol{D}-\boldsymbol{\Omega}\boldsymbol{C}\boldsymbol{\Phi'}\|_F^2+\frac{\rho}{2}\left\|\boldsymbol{G}^{(l+1)}-\boldsymbol{W}\boldsymbol{M}^T\boldsymbol{C}+\frac{1}{\rho}\boldsymbol{Y}^{(l)}\right\|_F^2 \tag{9}$$

$$\boldsymbol{Y}^{(l+1)}=\boldsymbol{Y}^{(l)}+\rho\left(\boldsymbol{G}^{(l+1)}-\boldsymbol{W}\boldsymbol{M}^T\boldsymbol{C}^{(l+1)}\right) \tag{10}$$

Solution to eq. (8) for $\boldsymbol{G}^{(l+1)}$ can be obtained through simple soft-thresholding:[1]

$$\boldsymbol{G}^{(l+1)}=\mathcal{S}\left(\boldsymbol{W}\boldsymbol{M}^T\boldsymbol{C}^{(l)}-\frac{1}{\rho}\boldsymbol{Y}^{(l)};\frac{\lambda}{\rho}\right) \tag{11}$$

where $\mathcal{S}$ is the soft-thresholding operator,

$$\mathcal{S}(\boldsymbol{Q}_{n,m};\alpha)=\begin{cases} 0, \ \boldsymbol{Q}_{n,m}<\alpha \\ \dfrac{\boldsymbol{Q}_{n,m}}{|\boldsymbol{Q}_{n,m}|}(|\boldsymbol{Q}_{n,m}|-\alpha), \boldsymbol{Q}_{n,m}\geq\alpha \end{cases}$$

Solution to eq. (9) can be derived from a linear equation:

$$\boldsymbol{\Omega}^T\boldsymbol{\Omega}\boldsymbol{C}\boldsymbol{\Phi}'\boldsymbol{\Phi}'^T+\frac{\rho}{2}\boldsymbol{M}\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{M}^T\boldsymbol{C}=\boldsymbol{\Omega}^T\boldsymbol{D}\boldsymbol{\Phi}'^T+\frac{\rho}{2}\boldsymbol{M}\boldsymbol{W}^T\boldsymbol{G}^{(l+1)}+\frac{1}{2}\boldsymbol{M}\boldsymbol{W}^T\boldsymbol{Y}^{(l)}. \tag{12}$$

Eq. (12) can then be decoupled into solving for sampled pixels and non-sampled pixels, which can be computed row by row and is parallelizable, if orthogonal wavelet transform is used ($\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}$)Let b_i to be the i-th row of the summation matrix on the right-hand side in eq. (12), the spatial coefficients are solved as follows:

$$\begin{cases} c_i\boldsymbol{\Phi}'\boldsymbol{\Phi}'^T+\frac{\rho}{2}c_i=b_i, \quad if\ i.th\ pixel\ is\ sampled \\ \frac{\rho}{2}c_i=b_i, \ if\ i.th\ pixel\ is\ not\ sampled \end{cases} \tag{13}$$

The Lagrangian multiplier $\boldsymbol{Y}$ is updated in eq. (10). The algorithm cycles through eqs. (11), (12) and (10) until convergence, by checking the relative change of the $\boldsymbol{C}$ and $\boldsymbol{Y}$ at each step (Figure S10).

<u>*Basis estimation*</u>

We can estimate the basis ($\boldsymbol{\Phi}$) from a collection of high-resolution transients with N_T points. First, a Casorati matrix is formed by arranging these transients as follows:

$$\boldsymbol{S} = \begin{bmatrix} s(r_1,t_1) & s(r_1,t_2) & \cdots & s(r_1,t_{N_T-1}) & s(r_1,t_{N_T}) \\ s(r_2,t_1) & s(r_2,t_2) & \cdots & s(r_2,t_{N_T-1}) & s(r_2,t_{N_T}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s(r_N,t_1) & s(r_N,t_2) & \cdots & s(r_N,t_{N_T-1}) & s(r_N,t_{N_T}) \end{bmatrix}$$

We then apply singular value decomposition (SVD) to $\boldsymbol{S}$. Given a $N \times N_T$ matrix, it can be decomposed through the compact SVD:

$$\boldsymbol{S} = \boldsymbol{U\Sigma V^T} \tag{14}$$

where $\boldsymbol{U}$ is a $N \times N_T$ matrix, $\boldsymbol{V}$ is an $N_T \times N_T$ matrix, and $\boldsymbol{\Sigma}$ is a $N_T \times N_T$ square matrix with non-negative real numbers on the diagonal as singular values. Model order L is selected heuristically based on the decay of singular values, giving the basis transients expressed as:

$$\boldsymbol{\Phi} = \boldsymbol{V}_L^T \tag{15}$$

where $\boldsymbol{V}_L^T$ contains the top L rows of $\boldsymbol{V^T}$ that correspond to the L largest singular values. In our experimental implementation, the basis transients were estimated from 4,000 randomly sampled transients from the fully sampled high-resolution dataset. More details and analysis on the basis estimation can be found in the reference[2]. A scikit-learn implementation of truncated SVD with ARPACK solver was used. With the selected number of sampled high-resolution transients and model order, the algorithm took about 15 mins to complete.

Experimental Details

Animals. Ten- to 12-week old male Sprague-Dawley® outbred rats (Rattus norvegicus) were obtained from Envigo (https://www.envigo.com/). Animal euthanasia was performed in accordance with the Illinois Institutional Animal Care and Use Committee and both federal and ARRIVE guidelines for the humane care and treatment of animals.

Tissue sectioning and matrix application. Rats were euthanized using CO2 asphyxiation. Immediately after euthanasia, transcardiac perfusion of the vascular system was performed using ice cold modified Gey's balanced salt solution (mGBSS) containing (in mM) 1.5 CaCl2, 4.9 KCl, 0.2 KH2PO4, 11 MgCl2, 0.3 MgSO4, 138 NaCl, 27.7 NaHCO3, 0.8 Na2HPO4, 25 HEPES and 10 glucose, pH 7.2. The entire rat brain was rapidly excised and frozen on dry ice. 16-µm-thick sagittal (right hemisphere) and coronal (both hemispheres) brain slices were prepared at –20 ˚C using Leica cryostat (Leica). The tissue slices were thaw-mounted onto indium-tin oxide glass slides. 30 mg/mL in 70% methanol 2,5-dihydroxybenzoic acid (DHB) MALDI matrix was applied using an HTX-M5 Sprayer (HTX Technologies, Chapel Hill, NC). MALDI matrix solution was sprayed at a flow rate of 100 µL/min, a temperature of 75 ^{0}C, and nebulizing gas pressure of 10 psi. The distance of the sprayer nozzle to sample surface was 50 mm, row spacing 2.5 mm, and sprayer nozzle velocity of 1200 mm/min.

FT-ICR MSI and sampling implementation. MSI was performed on a solariX 7T MALDI FT-ICR mass spectrometer (Bruker Corp., Billerica, MA) equipped with a dual ESI/MALDI source. The fully sampled high-resolution datasets for retrospective sparse sampling were acquired with a mass window of *m/z* 150–1200 and 1,048,576 data points per transient, yielding 0.731 s transient duration, which resulted in a resolving power of 160,000 at *m/z* 400. MALDI mass spectra were acquired in positive-mode using a Smartbeam-II UV laser (Bruker Corp.) in 'minimum' mode with a 50-µm raster width. Each MALDI acquisition consisted of 400 laser shots at a frequency of 1000 Hz. External calibration was performed using PepMix II (Bruker Corp.). For the experimental implementation of the integrated subspace and CS based sparse sampling, short transients were collected with 65,536 data points, yielding 0.045 s transient duration and a resolving power of 10,000 at *m/z* 400 prior to reconstruction. The mass resolution was calculated theoretically by:

$$\frac{m}{\Delta m} = \frac{1.274 \times 10^7 z B_0 T_{aqn}}{m}$$

where $\boldsymbol{B_0}$ is the magnetic field strength (7T in our case), and $\boldsymbol{T_{aqn}}$ is the transient acquisition time. The peak resolution R provided was given by calculating the full width at half maximum (FWHM) of the particular Lorentzian peaks. Tissues were imaged with a 25 µm raster width, and the number of laser shots was set to 300. Spatial sparse sampling requires preselection of random pixel locations to scan over a defined tissue region. Tissue sections placed on ITO slides were imaged at 1200 dpi with a flatbed scanner (Canon U.S.A. Inc., Melville, NY). Optical images were loaded into Bruker flexImaging software for tissue region of interest (ROI) selection. For each tissue ROI, a geometry file mapping the relative pixel position to the physical stage position and an autoXecute file defining the acquisition sequence were generated. A customized Python script was used to modify the autoXecute file by randomly selecting a subset of scanning positions (e.g. 30% of all scanning positions) from the acquisition sequence. The modified autoXecute file was then read by the autoXecuteRun program to start a single (or batch) acquisition.

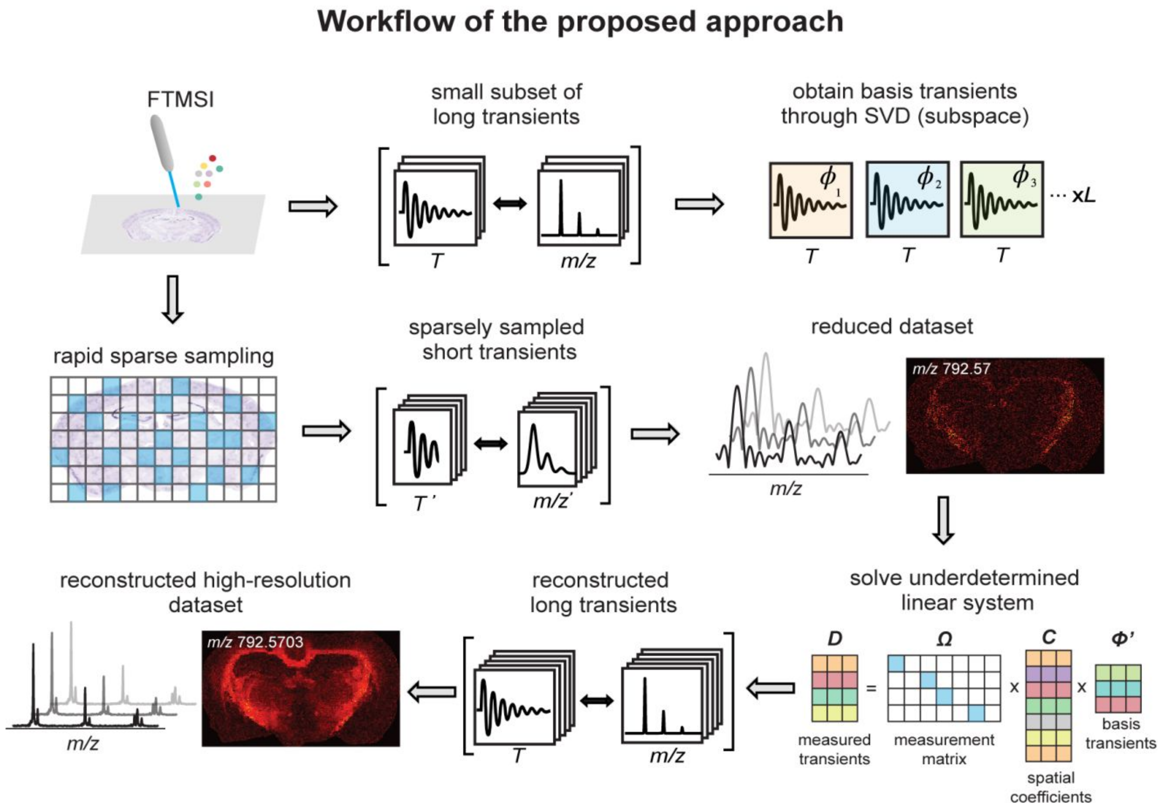

Figure S1. General workflow of the proposed method to accelerated FT MSI using a joint compressed sensing and subspace modeling approach.

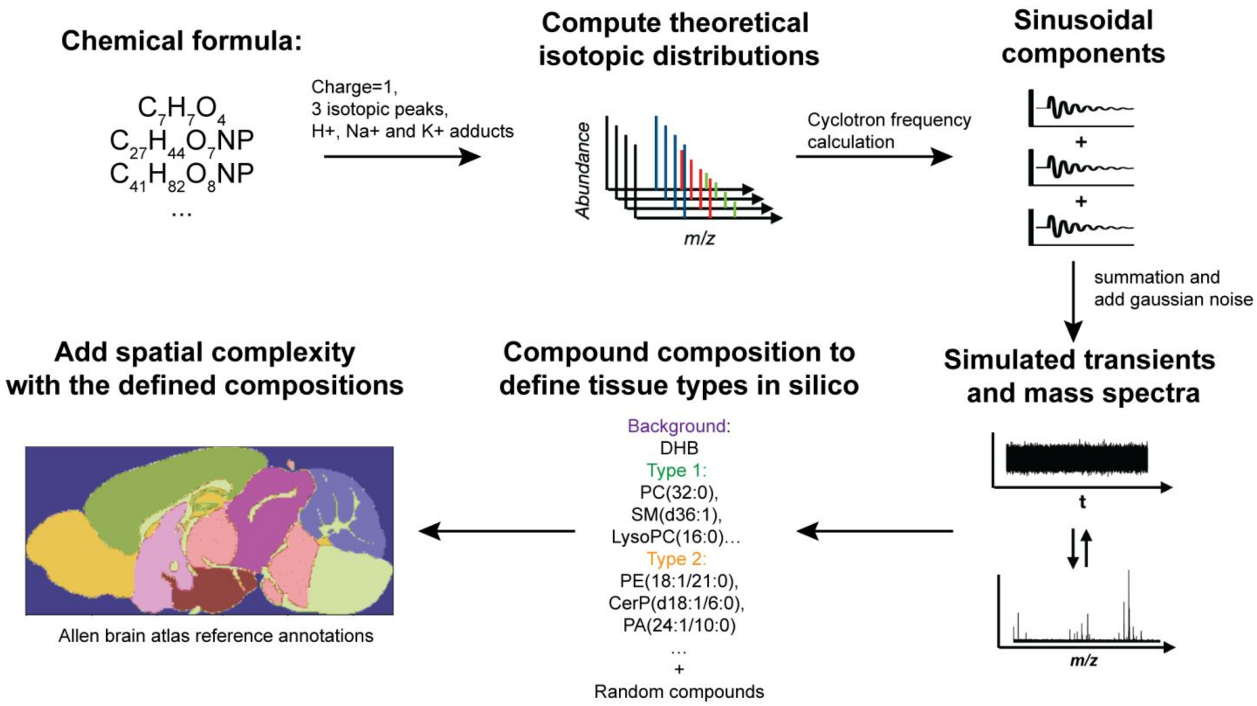

Figure S2. The workflow for the simulated FT-ICR MSI data, described in the supporting methods.

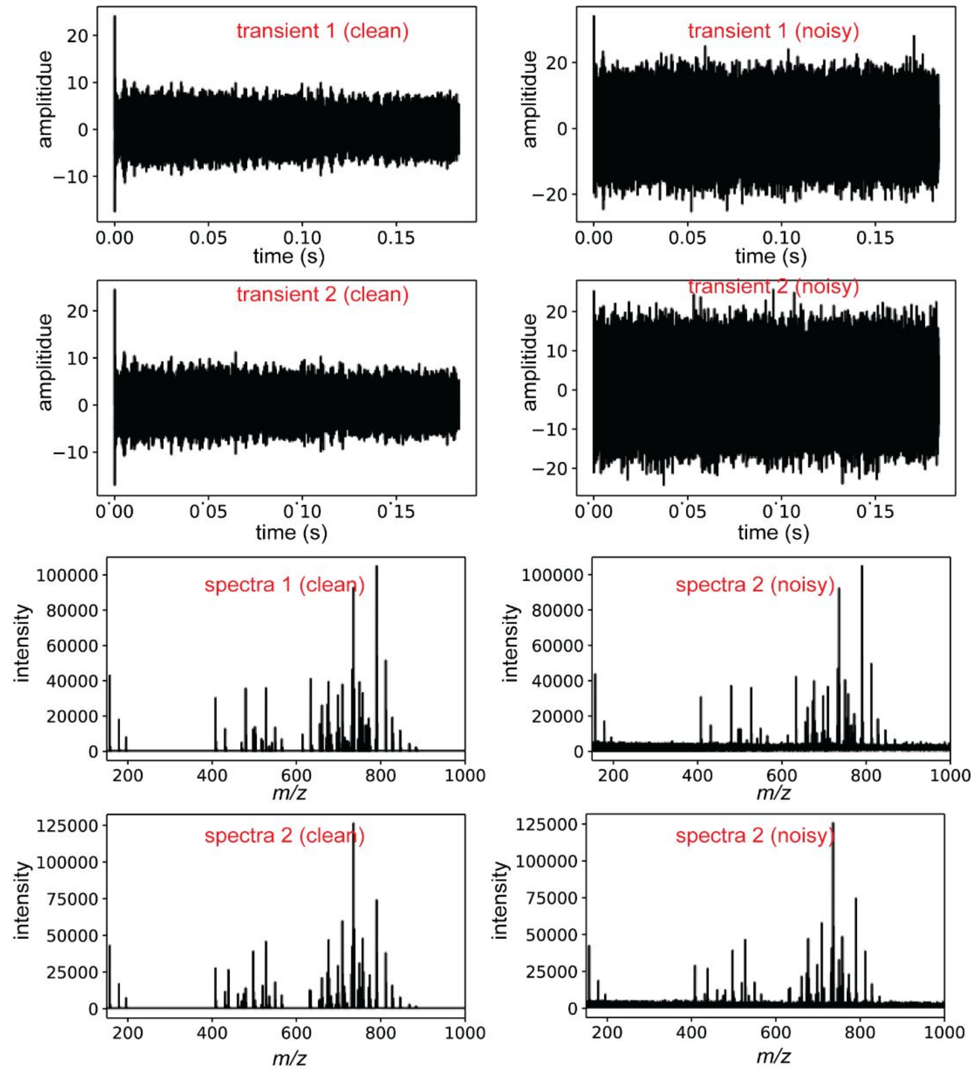

Figure S3. Representative simulated transients and spectra. Clean simulated transients (left column, top two) and the correspondingly mass spectra (left column, bottom two) from two simulated spatial substructures were displayed. The same data with Gaussian white noise added were shown on the right column.

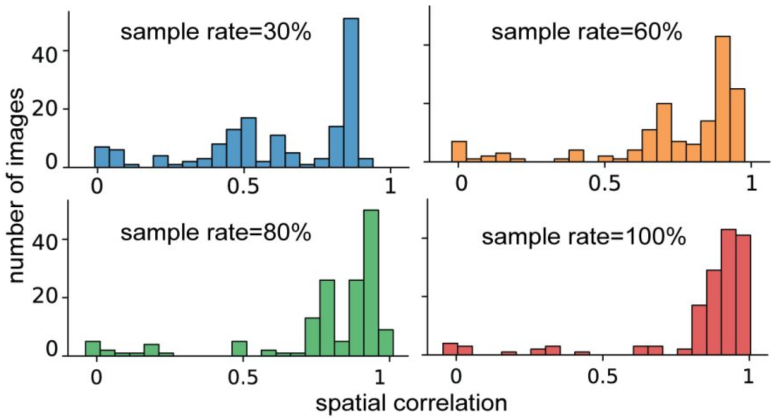

Figure S4. Evaluation on the simulated FT-ICR MSI dataset. The distributions of the spatial correlation between original and reconstructed image pairs at various sample rates using the noisy basis.

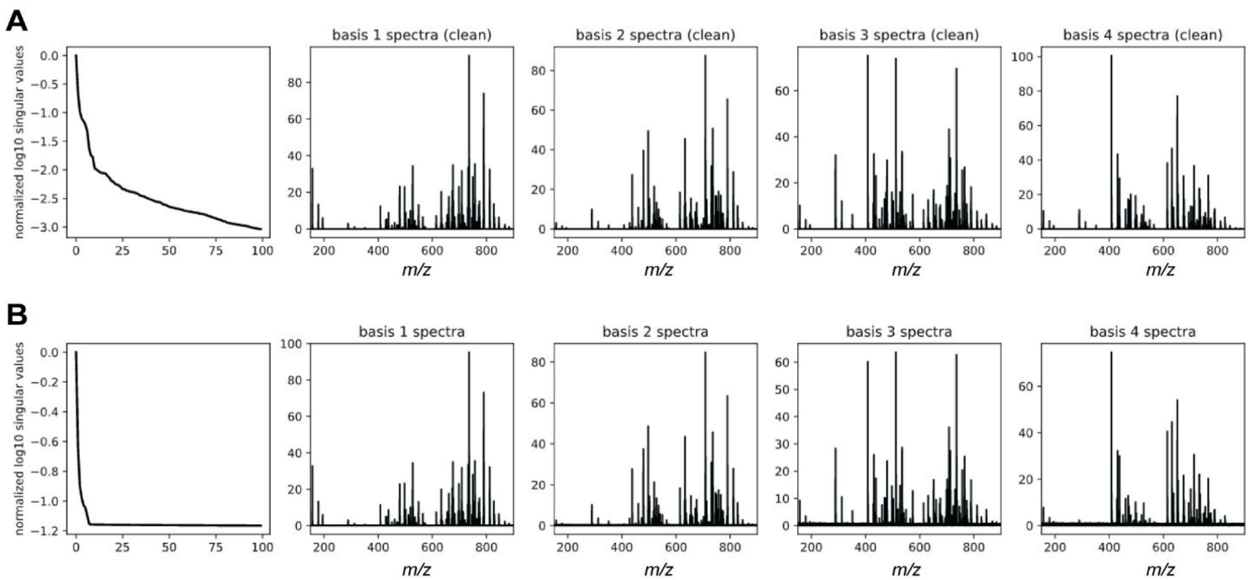

Figure S5. Extracted basis transients from (A) the clean dataset and (B) the noisy dataset. The singular values decay faster for the clean data than the noisy data.

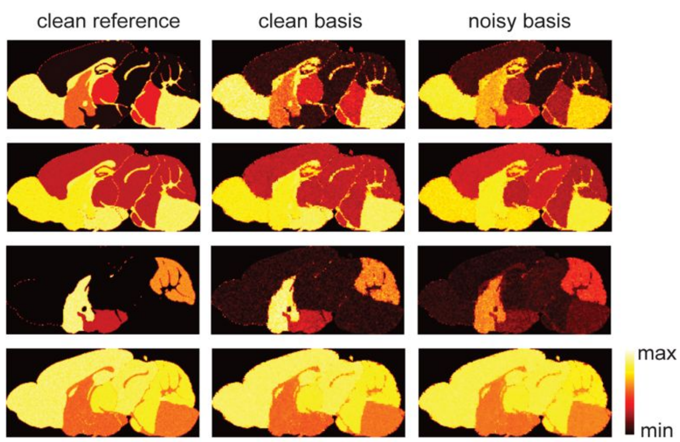

Figure S6. Original ion images, reconstructed images using the clean basis, and the reconstructed images using the noisy basis were compared at different *m/z* channels. Images shown were reconstructed using a 60% sample rate.

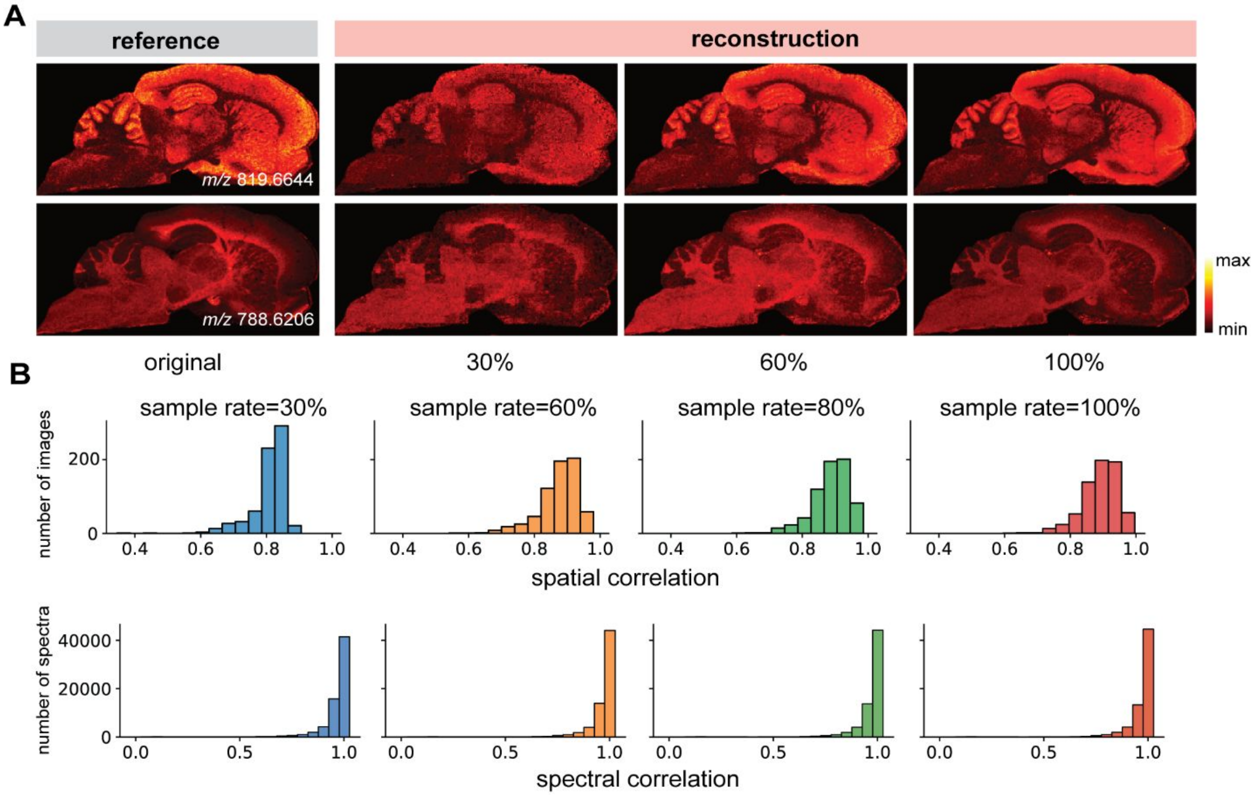

Figure S7. Reconstruction from data generated by a retrospective sparse sampling of a fully sampled high-resolution FT-ICR MSI dataset for a rat brain sagittal section. (A) Selected ion images at *m/z* 819.6644 and *m/z* 788.6206 are shown as the reference (original data, column 1) and reconstruction from data with 30, 60 and 100% pixels sampled (columns 2 to 4 respectively). (B) The histograms of the spatial and spectral correlation measures of ion images and the mass spectra from reference and reconstructed data.

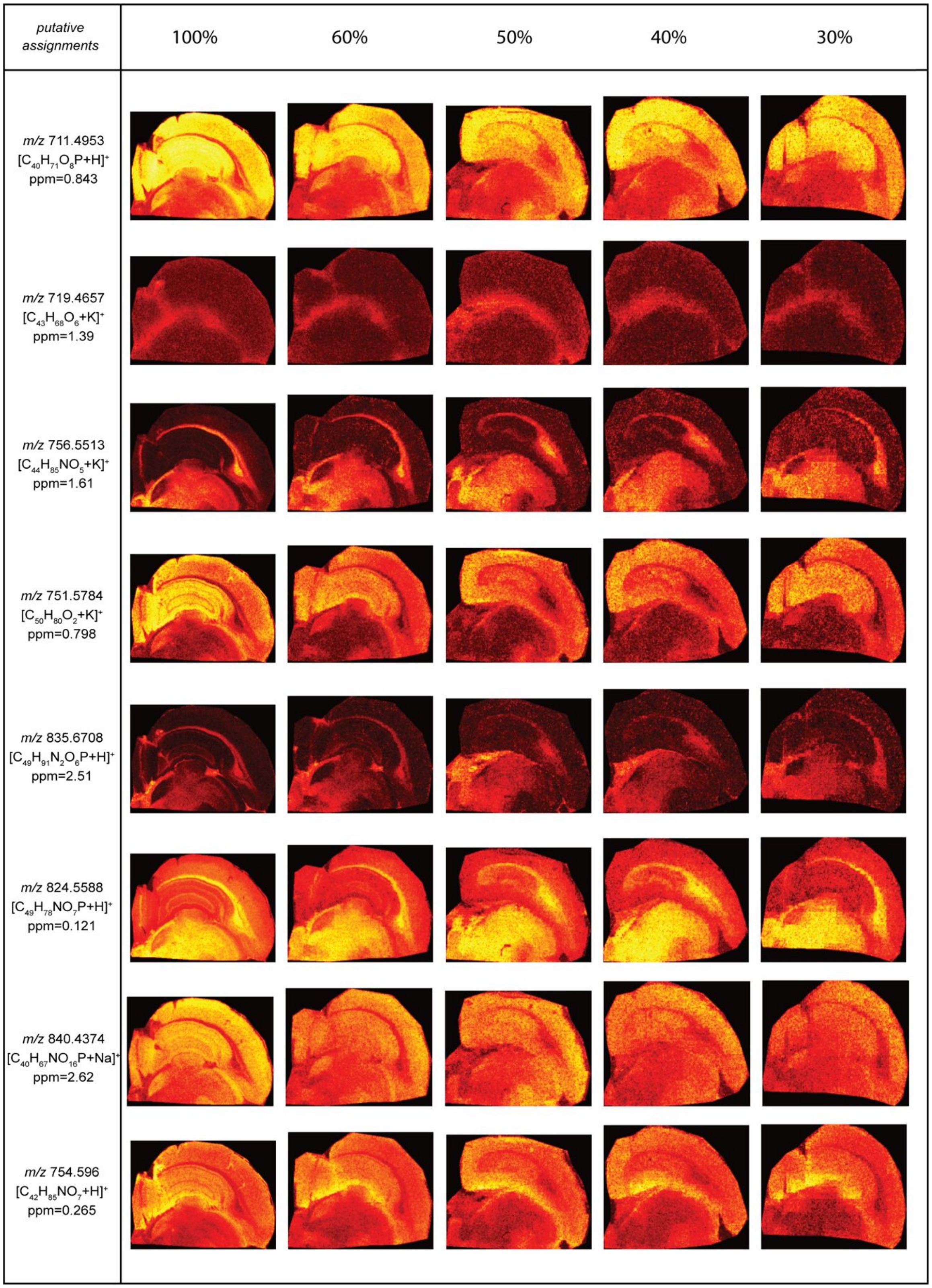

Figure S8. Reconstructed ion images from the dataset 2-6 listed in Table 1. In each row, images from datasets measured by different acquisition settings were putatively annotated with chemical formulas and ppm errors.

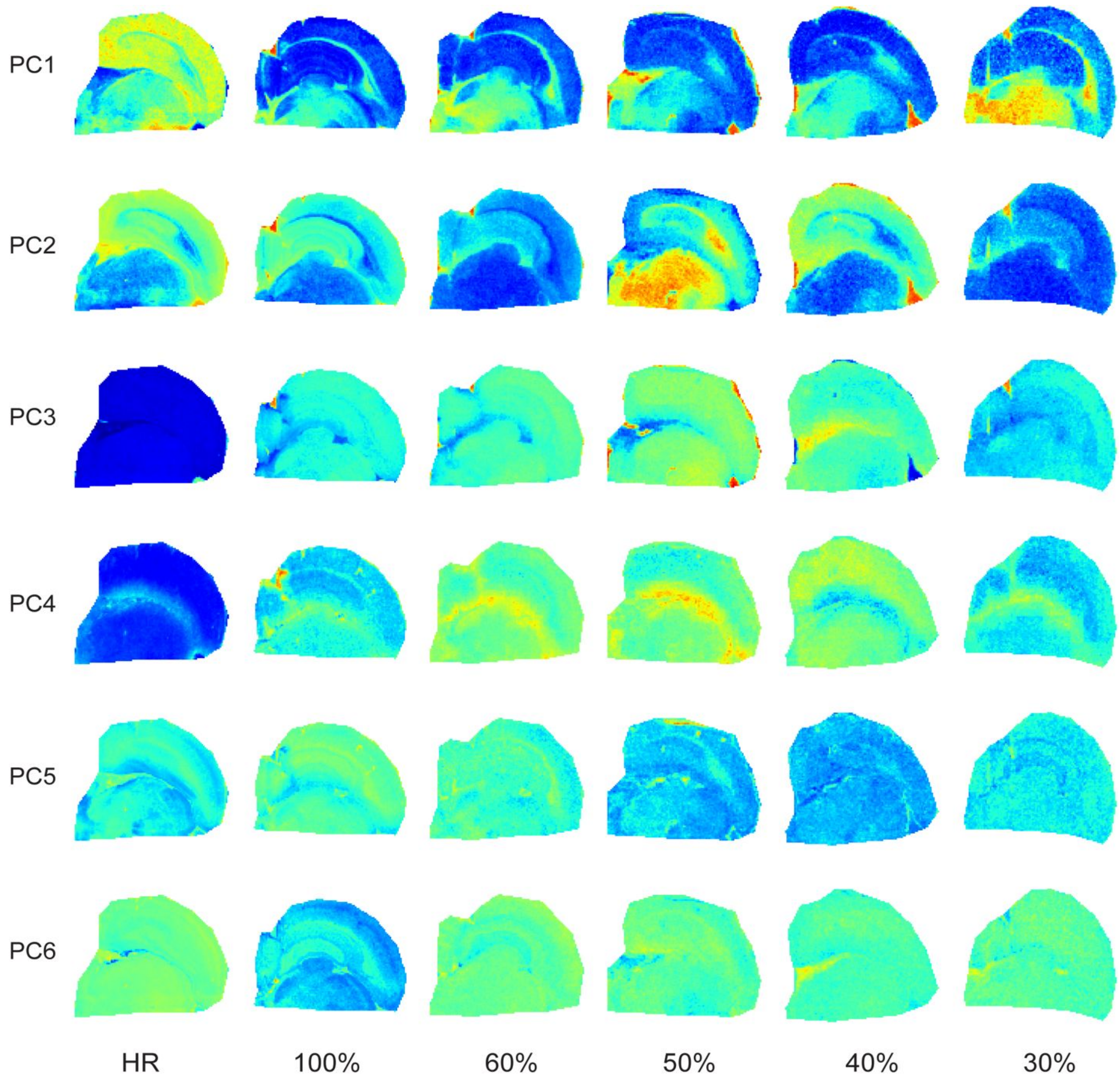

Figure S9. PCA of the reconstructed datasets listed in Table 1. Scores for the top 6 PCs were arranged into images, revealing the spatial variations of the contributions from different PCs.

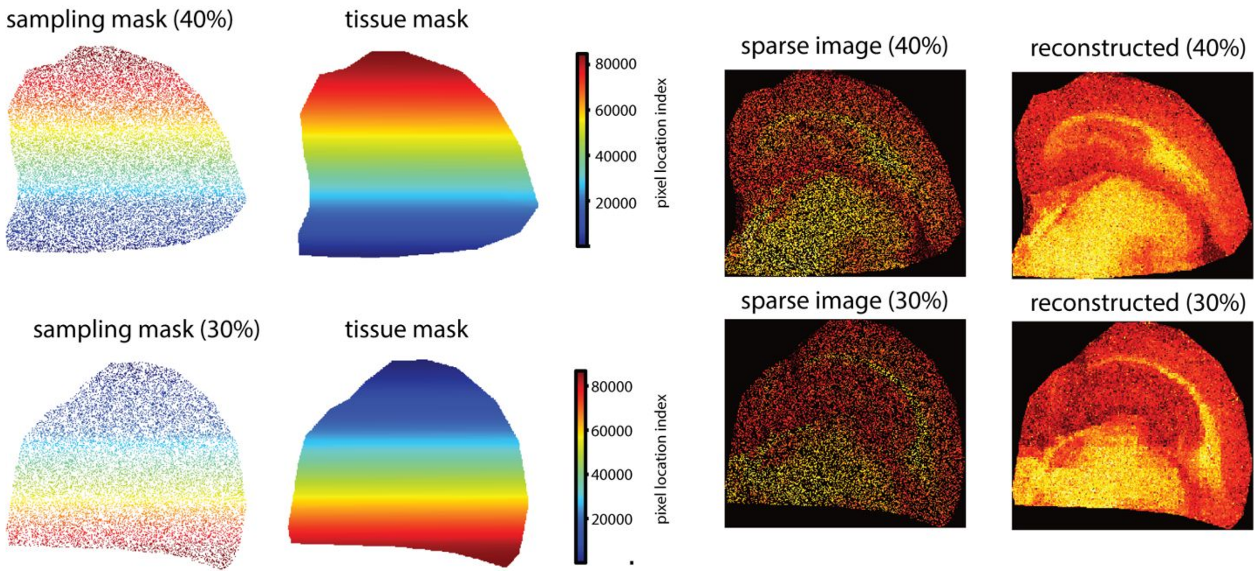

Figure S10. Demonstration of the spatial sparse sampling. Random spatial locations were selected for scanning by generating sampling masks from the whole tissue mask (left two columns; colors indicate the pixel location index). Ion images are obtained from raw data (right two columns, first column) and reconstructed data (second column).

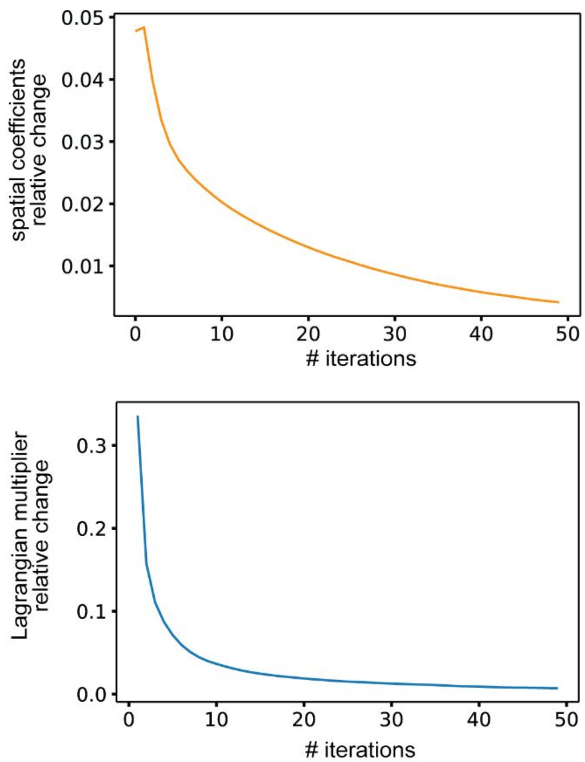

Figure S11. Convergence analysis of spatial coefficients and the Lagrangian multipliers for the proposed algorithm.

Table S1. Chemical formulas, ion forming adducts, *m/z* and relative intensities of the corresponding isotopic peaks for the simulation. Peak patterns consisting of three isotopic peaks are simulated for each singly charged cation.

chemical formulas	adduct	isotopic peak 1 *m/z*	isotopic peak 1 intensity	isotopic peak 2 *m/z*	isotopic peak 2 intensity	isotopic peak 3 *m/z*	isotopic peak 3 intensity
{'H': 80, 'C': 40, 'O': 8, 'N': 1, 'P': 1}	H	735.5773	0.63964	736.5806	0.286972	737.5836	0.073389
{'H': 80, 'C': 40, 'O': 8, 'N': 1, 'P': 1}	Na	757.5592	0.639708	758.5626	0.286929	759.5656	0.073363
{'H': 80, 'C': 40, 'O': 8, 'N': 1, 'P': 1}	K	773.5331	0.61141	774.5365	0.274313	775.5363	0.114276
{'H': 83, 'C': 41, 'O': 6, 'N': 2, 'P': 1}	H	732.614	0.633047	733.6173	0.29291	734.6204	0.074043
{'H': 83, 'C': 41, 'O': 6, 'N': 2, 'P': 1}	Na	754.5959	0.633115	755.5992	0.292868	756.6023	0.074017
{'H': 83, 'C': 41, 'O': 6, 'N': 2, 'P': 1}	K	770.5699	0.605384	771.5732	0.280117	772.5731	0.114499
{'H': 50, 'C': 24, 'O': 7, 'N': 1, 'P': 1}	H	497.3476	0.756622	498.3509	0.205622	499.3536	0.037756
{'H': 50, 'C': 24, 'O': 7, 'N': 1, 'P': 1}	Na	519.3295	0.756706	520.3329	0.205558	521.3355	0.037737
{'H': 50, 'C': 24, 'O': 7, 'N': 1, 'P': 1}	K	535.3035	0.71744	536.3068	0.194981	537.3048	0.087579
{'H': 44, 'C': 27, 'O': 7, 'N': 1, 'P': 1}	H	527.3006	0.734009	528.304	0.222787	529.3067	0.043204
{'H': 44, 'C': 27, 'O': 7, 'N': 1, 'P': 1}	Na	549.2826	0.73409	550.2859	0.222727	551.2887	0.043183
{'H': 44, 'C': 27, 'O': 7, 'N': 1, 'P': 1}	K	565.2565	0.697077	566.2599	0.211584	567.2582	0.091339
{'H': 67, 'C': 40, 'O': 8, 'P': 1}	H	708.4725	0.642698	709.4759	0.285035	710.4789	0.072268
{'H': 67, 'C': 40, 'O': 8, 'P': 1}	Na	730.4544	0.642766	731.4578	0.284991	732.4609	0.072243
{'H': 67, 'C': 40, 'O': 8, 'P': 1}	K	746.4283	0.614204	747.4317	0.272404	748.4315	0.113392
{'H': 86, 'C': 44, 'O': 8, 'N': 1, 'P': 1}	H	789.6242	0.614332	790.6276	0.302619	791.6306	0.083049
{'H': 86, 'C': 44, 'O': 8, 'N': 1, 'P': 1}	Na	811.6062	0.614397	812.6095	0.302581	813.6126	0.083023
{'H': 86, 'C': 44, 'O': 8, 'N': 1, 'P': 1}	K	827.5801	0.588247	828.5835	0.289776	829.5836	0.121978
{'H': 71, 'C': 37, 'O': 8, 'P': 1}	H	676.5038	0.662052	677.5072	0.272441	678.5102	0.065507
{'H': 71, 'C': 37, 'O': 8, 'P': 1}	Na	698.4857	0.662123	699.4891	0.272394	700.4921	0.065483
{'H': 71, 'C': 37, 'O': 8, 'P': 1}	K	714.4596	0.631857	715.4631	0.260022	716.4626	0.108121
{'H': 54, 'C': 31, 'O': 10, 'N': 1, 'P': 1}	H	633.3636	0.699921	634.367	0.244326	635.3697	0.055753
{'H': 54, 'C': 31, 'O': 10, 'N': 1, 'P': 1}	Na	655.3456	0.699997	656.3489	0.244272	657.3516	0.055731
{'H': 54, 'C': 31, 'O': 10, 'N': 1, 'P': 1}	K	671.3195	0.666261	672.3229	0.232583	673.3218	0.101156
{'H': 82, 'C': 41, 'O': 8, 'N': 1, 'P': 1}	H	749.5929	0.633158	750.5963	0.291058	751.5993	0.075784
{'H': 82, 'C': 41, 'O': 8, 'N': 1, 'P': 1}	Na	771.5749	0.633226	772.5782	0.291016	773.5813	0.075759
{'H': 82, 'C': 41, 'O': 8, 'N': 1, 'P': 1}	K	787.5488	0.605486	788.5522	0.278343	789.5521	0.116171
{'H': 48, 'C': 24, 'O': 6, 'N': 1, 'P': 1}	H	479.337	0.758246	480.3404	0.2056	481.3431	0.036154
{'H': 48, 'C': 24, 'O': 6, 'N': 1, 'P': 1}	Na	501.319	0.75833	502.3223	0.205536	503.325	0.036134
{'H': 48, 'C': 24, 'O': 6, 'N': 1, 'P': 1}	K	517.2929	0.7189	518.2962	0.194939	519.2942	0.086161
{'H': 89, 'C': 51, 'O': 8, 'N': 1}	H	845.6739	0.573465	846.6773	0.326103	847.6804	0.100432
{'H': 89, 'C': 51, 'O': 8, 'N': 1}	Na	867.6559	0.573524	868.6592	0.326071	869.6624	0.100405
{'H': 89, 'C': 51, 'O': 8, 'N': 1}	K	883.6298	0.55067	884.6332	0.313146	885.6339	0.136184
{'H': 74, 'C': 38, 'O': 6, 'N': 1, 'P': 1}	H	673.5405	0.655216	674.5438	0.278835	675.5469	0.065948

{'H': 74, 'C': 38, 'O': 6, 'N': 1, 'P': 1}	Na	695.5224	0.655287	696.5258	0.27879	697.5288	0.065923
{'H': 74, 'C': 38, 'O': 6, 'N': 1, 'P': 1}	K	711.4964	0.625627	712.4997	0.26625	713.4993	0.108123
{'H': 71, 'C': 36, 'O': 6, 'N': 2, 'P': 1}	H	660.5201	0.666502	661.5234	0.271426	662.5264	0.062073
{'H': 71, 'C': 36, 'O': 6, 'N': 2, 'P': 1}	Na	682.502	0.666574	683.5053	0.271378	684.5083	0.062048
{'H': 71, 'C': 36, 'O': 6, 'N': 2, 'P': 1}	K	698.476	0.635908	699.4793	0.258974	700.4787	0.105118
{'H': 69, 'C': 39, 'O': 8, 'P': 1}	H	698.4881	0.649024	699.4915	0.28097	700.4945	0.070006
{'H': 69, 'C': 39, 'O': 8, 'P': 1}	Na	720.4701	0.649093	721.4735	0.280925	722.4765	0.069982
{'H': 69, 'C': 39, 'O': 8, 'P': 1}	K	736.444	0.619979	737.4474	0.268403	738.4471	0.111619
{'H': 79, 'C': 39, 'O': 8, 'P': 1}	H	708.5664	0.64833	709.5698	0.281415	710.5728	0.070255
{'H': 79, 'C': 39, 'O': 8, 'P': 1}	Na	730.5483	0.648399	731.5517	0.281371	732.5548	0.07023
{'H': 79, 'C': 39, 'O': 8, 'P': 1}	K	746.5222	0.619346	747.5257	0.268841	748.5254	0.111813
{'H': 80, 'C': 44, 'O': 6, 'N': 2, 'P': 1}	H	765.5905	0.614645	766.5938	0.304126	767.5969	0.081229
{'H': 80, 'C': 44, 'O': 6, 'N': 2, 'P': 1}	Na	787.5724	0.61471	788.5758	0.304088	789.5789	0.081202
{'H': 80, 'C': 44, 'O': 6, 'N': 2, 'P': 1}	K	803.5464	0.588533	804.5497	0.291213	805.5499	0.120254
{'H': 35, 'C': 19, 'O': 7, 'P': 1}	H	408.2271	0.801208	409.2306	0.170101	410.2329	0.028691
{'H': 35, 'C': 19, 'O': 7, 'P': 1}	Na	430.2091	0.801297	431.2125	0.170028	432.2149	0.028675
{'H': 35, 'C': 19, 'O': 7, 'P': 1}	K	446.183	0.757406	447.1864	0.16081	448.1837	0.081784
{'H': 50, 'C': 26, 'O': 4}	H	428.386	0.747976	429.3894	0.215865	430.3924	0.036159
{'H': 50, 'C': 26, 'O': 4}	Na	450.368	0.748059	451.3714	0.215802	452.3744	0.036139
{'H': 50, 'C': 26, 'O': 4}	K	466.3419	0.709662	467.3453	0.204815	468.3433	0.085524
{'H': 29, 'C': 15, 'O': 4, 'N': 1}	H	289.2248	0.838366	290.228	0.143246	291.2305	0.018388
{'H': 29, 'C': 15, 'O': 4, 'N': 1}	Na	311.2067	0.838461	312.21	0.143166	313.2124	0.018373
{'H': 29, 'C': 15, 'O': 4, 'N': 1}	K	327.1806	0.790529	328.1839	0.135081	329.1805	0.07439
{'H': 37, 'C': 20, 'O': 8, 'P': 1}	H	438.2377	0.791121	439.2411	0.176999	440.2435	0.03188
{'H': 37, 'C': 20, 'O': 8, 'P': 1}	Na	460.2197	0.791209	461.2231	0.176928	462.2254	0.031863
{'H': 37, 'C': 20, 'O': 8, 'P': 1}	K	476.1936	0.748385	477.197	0.167446	478.1945	0.084169
{'H': 46, 'C': 24, 'O': 7, 'P': 1}	H	479.3132	0.75962	480.3166	0.203312	481.3193	0.037068
{'H': 46, 'C': 24, 'O': 7, 'P': 1}	Na	501.2952	0.759704	502.2986	0.203247	503.3012	0.037049
{'H': 46, 'C': 24, 'O': 7, 'P': 1}	K	517.2691	0.720135	518.2725	0.192751	519.2704	0.087114
{'H': 64, 'C': 33, 'O': 9, 'N': 1, 'P': 1}	H	651.447	0.686327	652.4503	0.254954	653.4532	0.058718
{'H': 64, 'C': 33, 'O': 9, 'N': 1, 'P': 1}	Na	673.4289	0.686402	674.4323	0.254903	675.4351	0.058695
{'H': 64, 'C': 33, 'O': 9, 'N': 1, 'P': 1}	K	689.4029	0.653931	690.4062	0.242927	691.4053	0.103142
{'H': 40, 'C': 23, 'O': 7, 'N': 1, 'P': 1}	H	475.2693	0.765359	476.2727	0.198838	477.2753	0.035803
{'H': 40, 'C': 23, 'O': 7, 'N': 1, 'P': 1}	Na	497.2513	0.765444	498.2546	0.198772	499.2572	0.035784
{'H': 40, 'C': 23, 'O': 7, 'N': 1, 'P': 1}	K	513.2252	0.725291	514.2285	0.188436	515.2264	0.086273
{'H': 43, 'C': 23, 'O': 10, 'P': 1}	H	512.2745	0.763346	513.2779	0.196662	514.2803	0.039992
{'H': 43, 'C': 23, 'O': 10, 'P': 1}	Na	534.2564	0.76343	535.2599	0.196596	536.2623	0.039974
{'H': 43, 'C': 23, 'O': 10, 'P': 1}	K	550.2304	0.723483	551.2338	0.1864	552.2317	0.090118
{'H': 77, 'C': 41, 'O': 8, 'N': 1}	H	713.58	0.633495	714.5834	0.290848	715.5864	0.075657
{'H': 77, 'C': 41, 'O': 8, 'N': 1}	Na	735.562	0.633562	736.5653	0.290806	737.5684	0.075632
{'H': 77, 'C': 41, 'O': 8, 'N': 1}	K	751.5359	0.605794	752.5393	0.278136	753.5392	0.11607

{'H': 37, 'C': 19, 'O': 12, 'N': 1}	H	473.2467	0.790253	474.25	0.172349	475.2521	0.037397
{'H': 37, 'C': 19, 'O': 12, 'N': 1}	Na	495.2286	0.790341	496.2319	0.172277	497.234	0.037382
{'H': 37, 'C': 19, 'O': 12, 'N': 1}	K	511.2026	0.747609	512.2059	0.163057	513.2036	0.089334
{'H': 49, 'C': 31, 'O': 10, 'P': 1}	H	614.3214	0.702719	615.3248	0.242331	616.3276	0.05495
{'H': 49, 'C': 31, 'O': 10, 'P': 1}	Na	636.3034	0.702796	637.3068	0.242277	638.3095	0.054928
{'H': 49, 'C': 31, 'O': 10, 'P': 1}	K	652.2773	0.668796	653.2807	0.23064	654.2796	0.100565
{'H': 52, 'C': 31, 'O': 10, 'N': 1, 'P': 1}	H	631.348	0.700073	632.3513	0.244218	633.3541	0.055709
{'H': 52, 'C': 31, 'O': 10, 'N': 1, 'P': 1}	Na	653.3299	0.700149	654.3333	0.244164	655.336	0.055687
{'H': 52, 'C': 31, 'O': 10, 'N': 1, 'P': 1}	K	669.3039	0.666399	670.3072	0.232477	671.3062	0.101124
{'H': 66, 'C': 37, 'O': 9, 'N': 1, 'P': 1}	H	701.4626	0.659025	702.466	0.273475	703.4689	0.067499
{'H': 66, 'C': 37, 'O': 9, 'N': 1, 'P': 1}	Na	723.4446	0.659096	724.4479	0.273429	725.4508	0.067475
{'H': 66, 'C': 37, 'O': 9, 'N': 1, 'P': 1}	K	739.4185	0.629099	740.4219	0.261064	741.4214	0.109837

(1) Donoho, D. L. De-Noising by Soft-Thresholding. *IEEE Transactions on Information Theory* **1995**, *41* (3), 613–627. https://doi.org/10.1109/18.382009.

(2) Xie, Y. R.; Castro, D. C.; Lam, F.; Sweedler, J. V. Accelerating Fourier Transform-Ion Cyclotron Resonance Mass Spectrometry Imaging Using a Subspace Approach. *J. Am. Soc. Mass Spectrom.* **2020**, *31* (11), 2338–2347. https://doi.org/10.1021/jasms.0c00276.