A study of the application of graphite MALDI to the analysis of lanthanide salts and deconvolution of the isobaric species observed.

Ulric Conway[1#], Alexander D. Warren[1,2] and Paul J. Gates[1*]

Affiliations:

1. School of Chemistry, University of Bristol, Cantock's Close, Bristol, BS8 1TS, U.K.

2. Interface Analysis Centre, School of Physics, University of Bristol, Tyndall Avenue, Bristol, BS8 1TL, U.K.

# current address: Kratos Analytical, Trafford Park, Manchester, M17 1GP, U.K.

Supplementary Information:

Table S1. The sources and descriptions of the lanthanides analysed in this study. Sources were SPEX CertiPrep (New Jersey, USA), Sigma Aldrich (Gillingham, UK), Fluka Analytical (Gillingham, UK) and Fisher Scientific (Loughborough, UK).

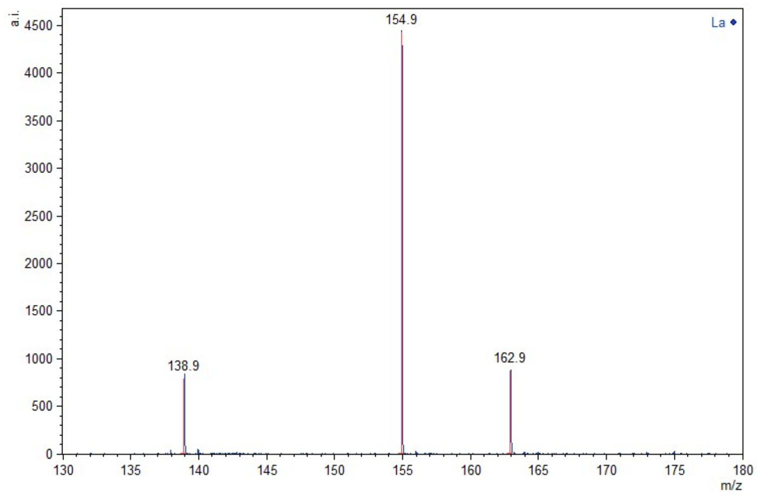
Element	**Description**	**Ox$^{\underline{n}}$**	**Code**	**Description**	**Supplier**
Lanthanum (La)	1,000 mg/L in 2% HNO_3	III	PLLA2-2Y	Assurance Grade	SPEX CertiPrep
Cerium (Ce)	1,000 mg/L in 2% HNO_3	III	PLCE2-2Y	Assurance Grade	SPEX CertiPrep
Praseodymium (Pr)	1,000 mg/L in 2% HNO_3	III	PLPR2-2Y	Assurance Grade	SPEX CertiPrep
Neodymium (Nd)	1,000 mg/L in 2% HNO_3	III	PLND2-2Y	Assurance Grade	SPEX CertiPrep
Samarium (Sm)	1,000 mg/L in 2% HNO_3	III	PLSM2-2Y	Assurance Grade	SPEX CertiPrep
Europium (Eu)	1,000 mg/L in 2% HNO_3	III	PLEU2-2Y	Assurance Grade	SPEX CertiPrep
Gadolinium (Gd)	1,000 mg/L in 2% HNO_3	III	PLGD2-2Y	Assurance Grade	SPEX CertiPrep
Terbium (Tb)	1,000 mg/L in 2% HNO_3	III	PLTB2-2Y	Assurance Grade	SPEX CertiPrep
Dysprosium (Dy)	1,000 mg/L in 2% HNO_3	III	68839	ICP Standard	Fluka Analytical
Holmium (Ho)	1,000 mg/L in 2% HNO_3	III	SJ/56025/100	ICP Standard	Fisher Chemicals
Erbium (Er)	1,000 mg/L in 2% HNO_3	III	05693	ICP Standard	Fluka Analytical
Thulium (Tm)	1,000 mg/L in 2% HNO_3	III	PLTM2-2Y	Assurance Grade	SPEX CertiPrep
Ytterbium (Yb)	1,000 mg/L in 2% HNO_3	III	39956	ICP Standard	Sigma Aldrich
Lutetium (Lu)	1,000 mg/L in 2% HNO_3	III	PLLU2-2Y	Assurance Grade	SPEX CertiPrep

Table S2. Experimental parameters and settings used for the 4700 Proteomics Analyzer.

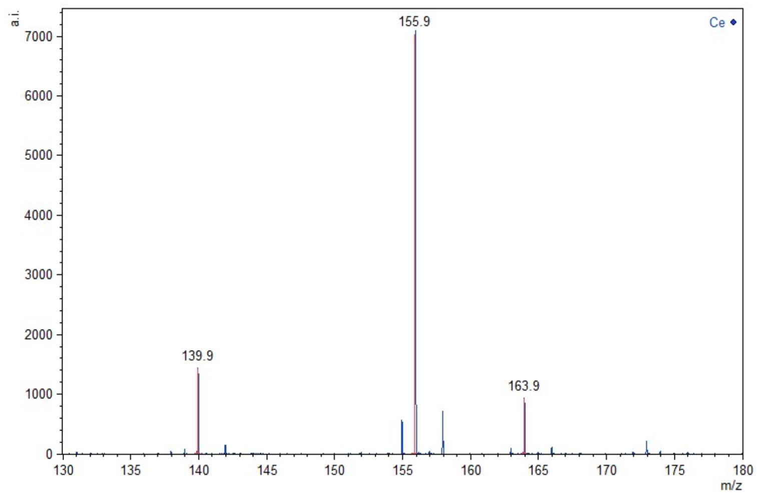
Parameter	**Setting**
Detection mode	Reflectron
Ion mode	Positive
m/z range	50-600
Laser fluence	50-60%
Spectra summed	1000 shots per 20 sub-spectra
Spots per sample	10
Repeats per spot	3

Figure S1: Replotted positive ion mode MALDI-MS analysis of the lanthanide period. The spectra are expanded to only show the regions of interest (M^+, MO^+, MOH^+, MC_2^+ etc)

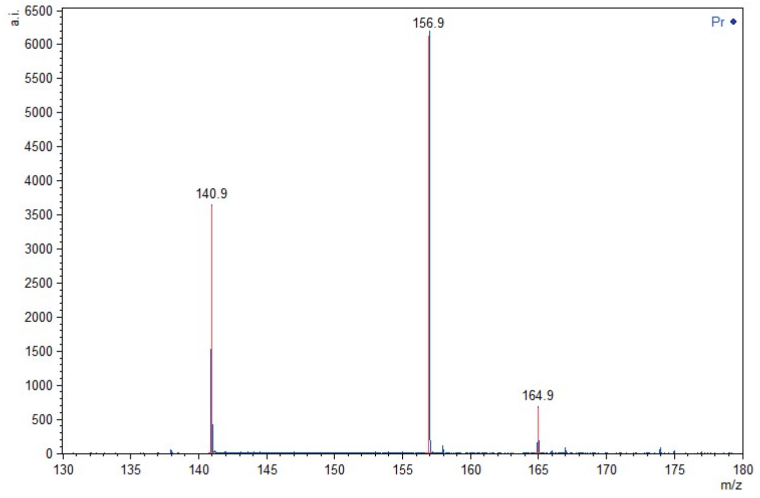
(a) Lanthanum (La) (Mw $M^+ = 138.9$, $MO^+ = 154.9$, $MC_2^+ = 162.9$)



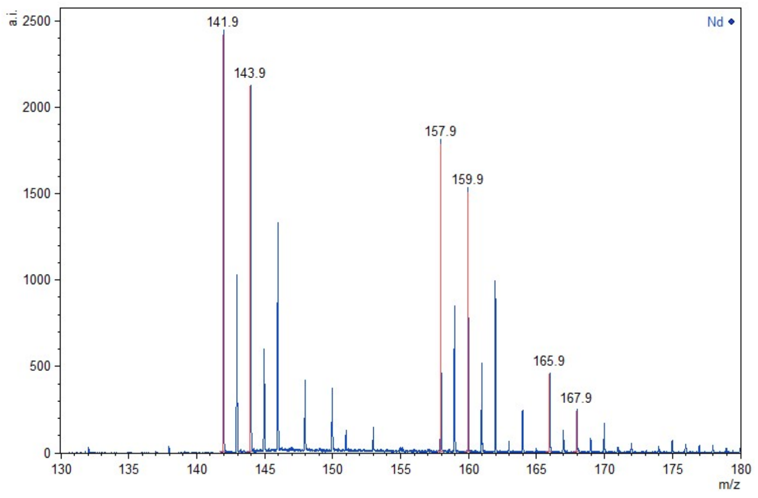
(b) Cerium (Ce) (Mw $M^+ = 139.9$, $MO^+ = 155.9$, $MC_2^+ = 163.9$)



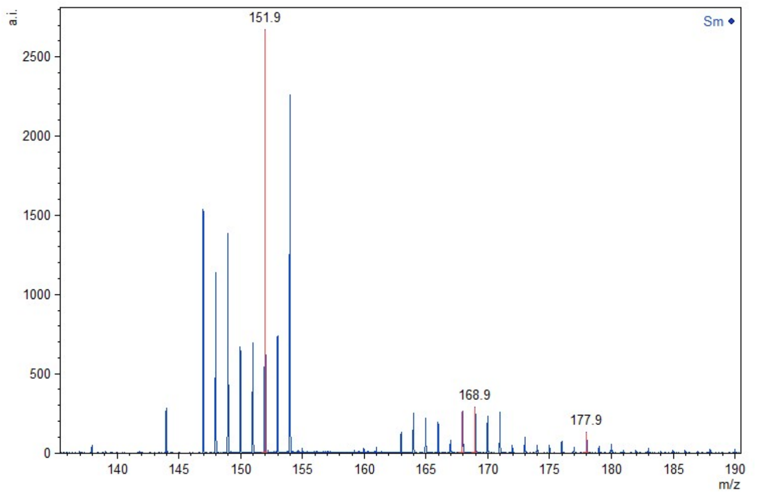
(c) Praseodymium (Pr) (Mw $M^+ = 140.9$, $MO^+ = 156.9$, $MC_2^+ = 164.9$)



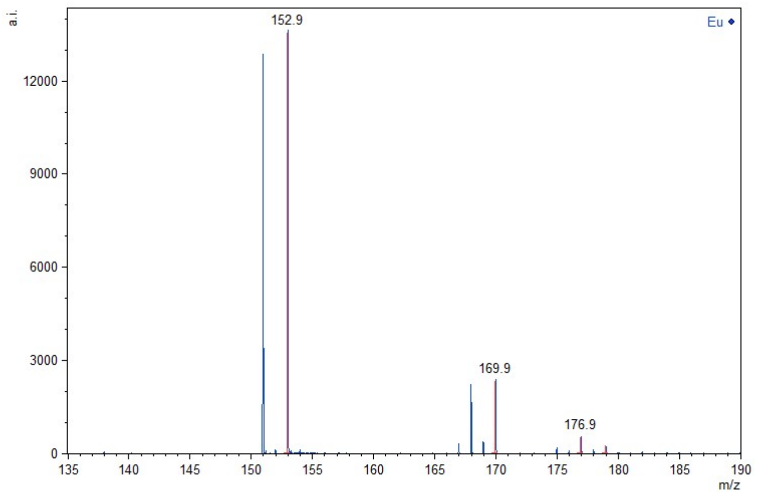
(d) Neodymium (Nd) (Mw M^+ = 141.9, MO^+ = 157.9, MC_2^+ = 165.9)



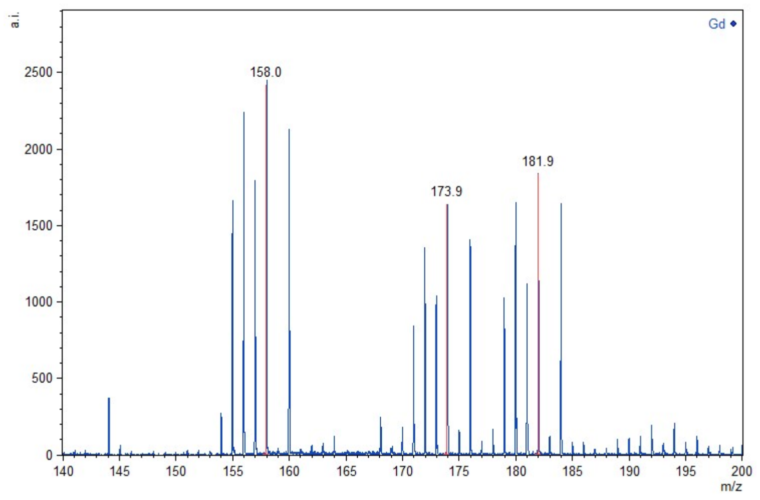
(e) Samarium (Sm) (Mw M^+ = 151.9, MO^+ = 167.9, MOH^+ = 168.9, $MC_2H_2^+$ = 177.9)



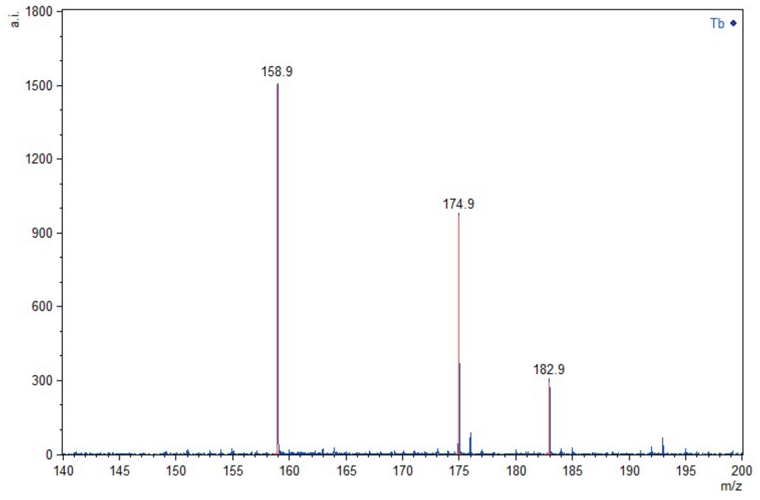
(f) Europium (Eu) (Mw M^+ = 152.9, MO^+ = 168.9, MOH^+ = 169.9, MC_2^+ = 176.9, $MC_2H_2^+$ = 178.9)



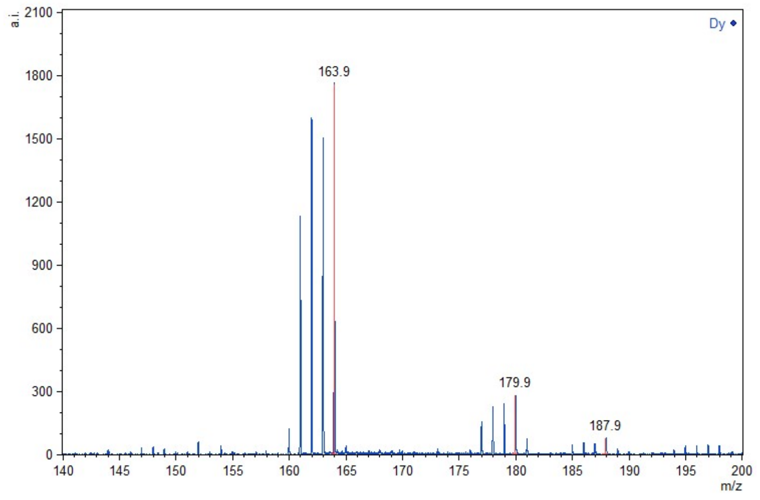
(g) Gadolinium (Gd) (Mw $M^+ = 158.0$, $MO^+ = 173.9$, $MC_2^+ = 181.9$)



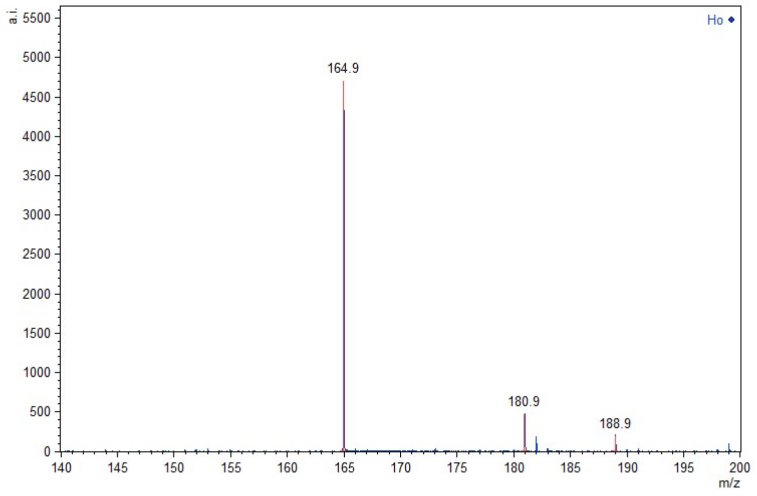
(h) Terbium (Tb) (Mw $M^+ = 158.9$, $MO^+ = 174.9$, $MC_2^+ = 182.9$)



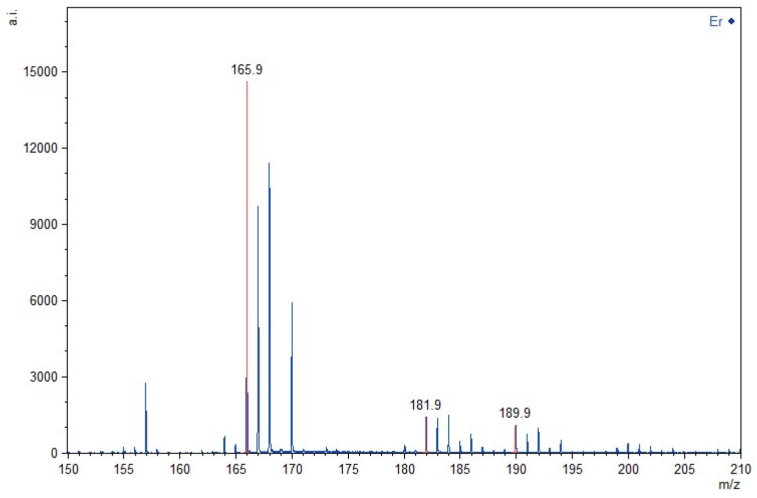
(i) Dysprosium (Dy) (Mw $M^+ = 163.9$, $MO^+ = 179.9$, $MC_2^+ = 187.9$)



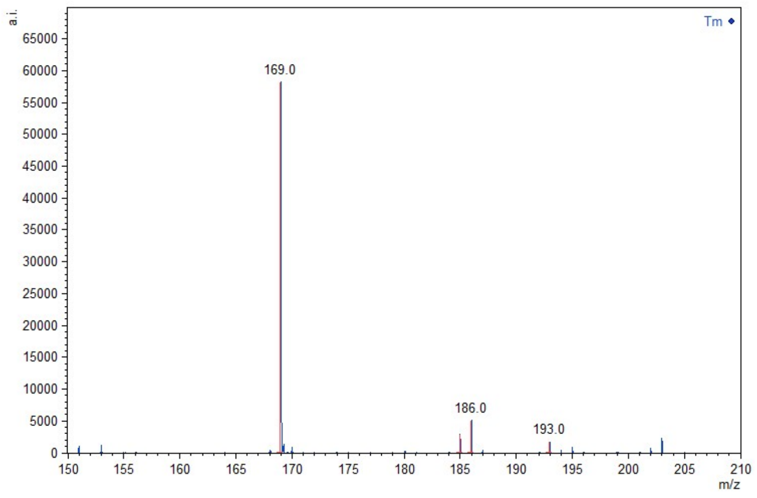
(j) Holmium (Ho) (Mw M^+ = 164.9, MO^+ = 180.9, MC_2^+ = 188.9)



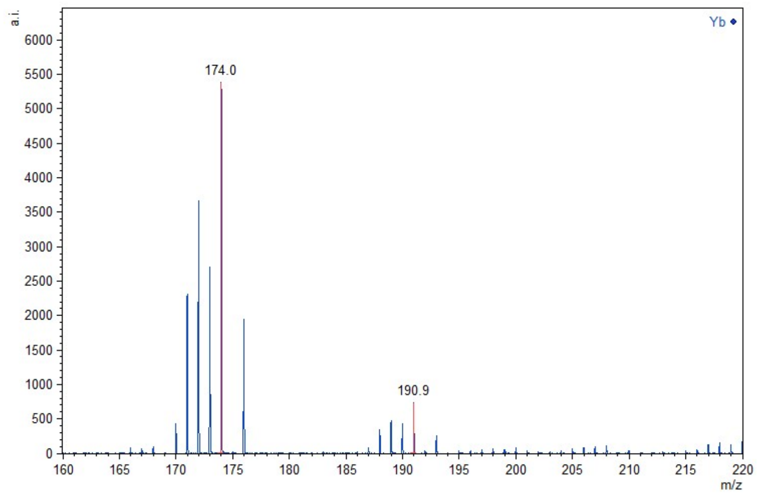
(k) Erbium (Er) (Mw M^+ = 165.9, MO^+ = 181.9, MC_2^+ = 189.9)



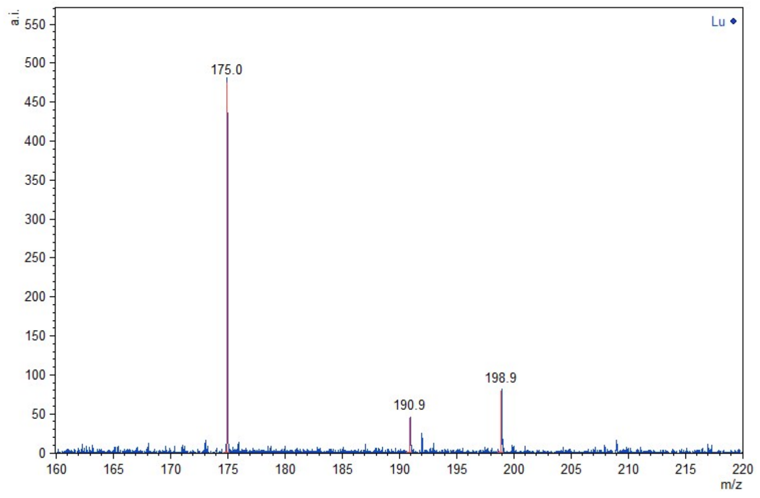
(l) Thulium (Tm) (Mw M^+ = 169.0, MO^+ = 185.0, MOH^+ = 186.0, MC_2^+ = 193.0)



(m) Ytterbium (Yb) (Mw M^+ = 174.0, MO^+ = 190.0)



(n) Lutetium (Lu) (Mw M^+ = 175.0, MO^+ = 190.0, MC_2^+ = 198.9)



Deconvolution theory:

Observed mass spectra isotopic patterns can be represented as a column vector **b**, where the row represents the *m/z* and the value represents the intensity. If the value of the abundance ***bj*** at each mass ***j*** consists of a number of contributions from different species, then it can be written as a system of linear equations:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= bm \end{aligned} \qquad \text{(S1)}$$

where ***x*** represents the contribution of isobaric component ***a*** to the observed intensity ***b***. This system can be written as a matrix equation with the $m \times n$ matrix **A** which contains the theoretical isotope patterns of the ***n*** different components:

$$Ax = b \qquad \text{(S2)}$$

where:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix},\ x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},\ b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Solution of the matrix equation will yield the column vector **x**, whose components $\boldsymbol{x_i}$ can be used to find the percentage contribution of each component ($\boldsymbol{X_i}$) in the observed spectrum:

$$X_i = \frac{x_i}{\sum_i x_i} x\, 100 \qquad \text{(S3)}$$

Generally speaking, equation S2 will lack a unique solution and so it becomes a least squares problem. This is equivalent to saying that there will be some deviation $\boldsymbol{\varepsilon}$ between the calculated abundances and what is actually observed:

$$Ax + \varepsilon = b \qquad \text{(S4)}$$

So, although we might not be able to directly solve **Ax = b**, we can try to minimise the *residual* $\varepsilon = (\mathbf{b} - \mathbf{Ax})$. The sum of the squares of the residuals can be written as (using $^{\boldsymbol{T}}$ to denote the transpose of a matrix):

$$\sum \varepsilon^2 = \varepsilon^T\varepsilon = (b - Ax)^T(b - Ax) \quad \text{(S5)}$$

$$= \mathbf{b}^T\mathbf{b} - 2\mathbf{x}^T\mathbf{A}^T + \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{b} \quad \text{(S6)}$$

To minimise the sum of squares we set the derivatives with respect to ***x*** equal to zero, and then rearrange the equation to obtain the ***x*** which we seek:

$$\frac{\partial}{\partial x}\sum \varepsilon^2 = -2\mathbf{A}^T\mathbf{b} + 2\mathbf{A}^T\mathbf{A}\mathbf{x} = 0 \quad \text{(S7)}$$

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b} \quad \text{(S8)}$$

$$\mathrm{x} = (\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{A}^T\mathbf{b} \quad \text{(S9)}$$

For an inverse of $\mathbf{A}^T\mathbf{A}$ to exist we require that $\boldsymbol{m} \geq \boldsymbol{n}$, or in other words, the number of possible components in the spectral pattern is smaller than or equal to the number of observations (peaks). The standard expression for the error in the elements of x is:

$$s_{x_i}^2 = s_r^2 \,.\, (A^TA)^{-1} \quad \text{(S10)}$$

where s_r is standard deviation of the residuals. Taking into account the propagation of errors through equations S3 and S8 results in the following equation for the standard deviation of $\mathbf{X}_i$, as used by Meija and Caruso [xx]:

$$\frac{s_{X_i}}{X_i} = \sqrt{\frac{1}{(m-1)}(b - Ax)^T(b - Ax)} \,.\, \sqrt{\left(\frac{1}{x_i}(A^TA)^{-1}_{ii}\right)^2 + \sum_{k=1}^{n}\left(\frac{1}{x_k}(A^TA)^{-1}_{kk}\right)^2} \quad \text{(S11)}$$

Overlapping species can therefore be separated from each other in a quantitative manner by translating equation S9 into computer code (see Python code below).

For example, when analysing samarium-containing analytes one frequently observes both SmO^+ and $SmOH^+$ which are one *m/z* unit apart and interfere with each other due to the complex isotopic pattern of samarium. The relative amounts of each species can be determined because the theoretical patterns of each of the contributors can be calculated before-hand, forming the matrix **A** in equation S9. Clearly this highlights an important limitation in this technique: that the individual ionic components of the observed pattern must be known *a priori*. Alternatively, the experimenter can attempt to obtain the isobaric components by educated guesswork. In any case, the resulting statistical error (from equation S11) will provide information about how closely the experimentally observed data matches the theoretically inputted data. When combined with visual inspection of the spectra, deconvolution of isobaric components in this way is still a powerful technique.

Deconvolution application - for Sm (see figure 2):

Raw data from the MALDI analysis was imported into Origin where the peaks of interest were integrated to form the vector **b**, from equation S2 above. Formation of the **A** matrix in this case is shown in table S3.

Table S3. Matrices showing the known components of the matrix equation to be solved. Matrix **A** contains the theoretical abundances (shown to 4 s.f.) of the components, where the three columns correspond to SmO^+, $SmOH^+$ and $SmC_2H_2^+$ respectively. Matrix **b** contains the peak heights of each spectral feature (shown rounded to nearest whole number).

$$\begin{array}{rl} m/z = & 160 \\ & 161 \\ & 162 \\ & 163 \\ & 164 \\ & 165 \\ & 166 \\ & 167 \\ & 168 \\ & 169 \\ & 170 \\ & 171 \\ & 172 \\ & 173 \\ & 174 \\ & 175 \\ & 176 \\ & 177 \\ & 178 \\ & 179 \\ & 180 \\ & 181 \end{array} \qquad \mathbf{A} = \begin{bmatrix} 3.07 & - & - \\ - & 3.07 & - \\ - & - & - \\ 14.97 & - & - \\ 11.23 & 14.97 & - \\ 13.86 & 11.23 & - \\ 7.41 & 13.86 & - \\ 0.03 & 7.41 & - \\ 26.73 & 0.03 & - \\ - & 26.73 & - \\ 22.81 & - & 3.07 \\ - & 22.81 & 0.07 \\ 0.05 & - & - \\ - & 0.05 & 14.98 \\ - & - & 11.56 \\ - & - & 14.08 \\ - & - & 7.69 \\ - & - & 0.16 \\ - & - & 26.73 \\ - & - & 0.58 \\ - & - & 22.77 \\ - & - & 0.50 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 701 \\ 514 \\ 0 \\ 3254 \\ 4560 \\ 4536 \\ 3466 \\ 1146 \\ 5555 \\ 3721 \\ 4733 \\ 3614 \\ 630 \\ 1163 \\ 872 \\ 520 \\ 1224 \\ 368 \\ 1825 \\ 398 \\ 914 \\ 339 \end{bmatrix}.$$

From here, the least squares optimised **x** could be calculated using equations S3-S11. This was achieved by manipulating the matrices in the code building environment of Origin, called "Origin C". Functions were programmed so that the algorithms could work directly on the peak data already stored in Origin.

Python Code for Deconvolution

The computational method used to solve the matrix equation Ax = b for x; in other words, solve the following equation:

$$x = (\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{A}^T\mathbf{b}. \qquad \text{(S12)}$$

This method uses the "Origin C" programming language, which is based on C/C++ and which can be used to manipulate data imported into Origin software, including matrices. The program used to solve the equation was developed inhouse by the authors and is presented here followed by some notes.

```
#include <Origin.h>

void deconvolve()
{
    int m, n, o, p;
    double sum = 0.0, kSUM = 0.0, m_doub = 0.0;

    vector sumvec;
    Matrix matF, matY;
    matrix matFt, matFtF, matFtF_inv, matA, summat, matFA;
    matrix matRES, matRESt, matStderr, matRESq, matDISP;

    /* Get the theoretical patterns */
    matF.Attach("F");

    /* Get the experimental data */
    matY.Attach("Y");

    /* Get matrix dimensions */
    m = matF.GetNumRows();
    n = matF.GetNumCols();
    o = matY.GetNumRows();
    p = matY.GetNumCols();
    //printf(" Matrix F is %d rows by %d columns\n", m , n);
    //printf(" Matrix Y is %d rows by %d columns\n", o , p);

    /* Check dimensions */
    if(((m<n)) || (!(m==o)) )
    {
        printf("Check matrix dimensions!\n");
    }

    /* Take transpose of F */
matFt = matF;
matFt.Transpose();

/* Multiply F with its transpose (to obtain information matrix) */
matFtF = matFt*matF;

/* Take the inverse (to obtain inverse information matrix) */
matFtF_inv = matFtF;
matFtF_inv.Inverse();

/* Solution of least squares gives ... */
matA = matFtF_inv*matFt*matY;
```

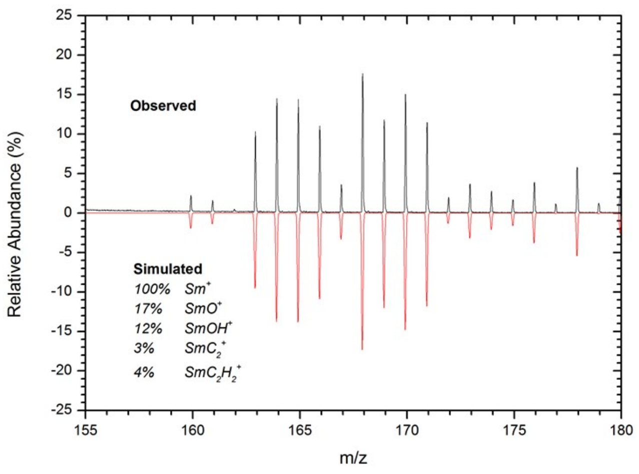
```

/* Calculate percentages of each component */
matA.SumColumns(sumvec);
sum = sumvec[0];
summat = (matA/sum)*100;

/* Find the residuals and the sum of their squares */
matFA = matF*matA;
matRES = matY - matFA;
matRESt = matRES;
matRESt.Transpose();
matRESq = matRESt*matRES;
matStderr = matA;

/* Change m to double type for error calc */
    m--;
    m_doub = (double) m;

    /* Calculate residual standard deviation */
for(int l=0; l<n; l++)
    kSUM += pow( (1/matA[l][0])*matFtF_inv[l][l],2);
for(int k=0; k<n; k++)
    matStderr [k][0] = sqrt (matRESq[0][0]/m_doub)*sqrt( pow
    ((1/matA[k][0])*matFtF_inv[k][k],2) + kSUM);

/* Output results to script window */
printf("Percentage of each component:\n");
for(int i=0; i<n; i++)
   printf("%d - %f (+/-%d)%%\n", (i+1), summat[i][0], (int)
   ceil(matStderr[i][0]*summat[i][0]));
```

```

/* Choose to display matrices */
    //MatrixPage MatPg1;
//MatPg1.Create("Origin");
//MatrixLayer MatLy1 = MatPg1 . Layers (0);
//Matrix DISP (MatLy1);
//DISP = matStderr;
//
//printf("%f\n", m_doub);

//MatrixPage MatPg2;
//MatPg2.Create("Origin");
//MatrixLayer MatLy2 = MatPg2.Layers(0);
//Matrix matFtF(MatLy2);

//MatrixPage MatPg3;
//MatPg3.Create("Origin");
//MatrixLayer MatLy3 = MatPg3.Layers(0);
//Matrix matFtF_inv(MatLy3);

//MatrixPage MatPg4;
//MatPg4.Create("Origin");
//MatrixLayer MatLy4 = MatPg4.Layers(0);
//Matrix matA(MatLy4);

//MatrixPage MatPg5;
//MatPg5.Create("Origin");
//MatrixLayer MatLy5 = MatPg5.Layers(0);
//Matrix matDISP(MatLy5);
```

```

}
```

After all the variables in the program have been declared, lines 13-17 define the two input matrices: **matF** containing the theoretical abundances calculated from mMass software [S1] and **matY** containing the experimental abundances. The program then proceeds by gaining the dimensions of the matrices and some simple error checking in lines 19-31. After this a number of intermediate matrices are set up to make the calculation easier to check throughout each stage, followed by the calculation on line 45 which essentially mirrors equation S12. Lines 52-68 contain the calculation of the error in finding the least squares, which is followed by the output of the results to a window in Origin. The remainder of the program contains options to display many of the different matrices in a separate window throughout the calculation, for checking by eye.

Figure S2: Plots of the observed peaks (black) against deconvoluted simulations (red) for the lanthanides (a) samarium, (b) neodymium and (c) europium.

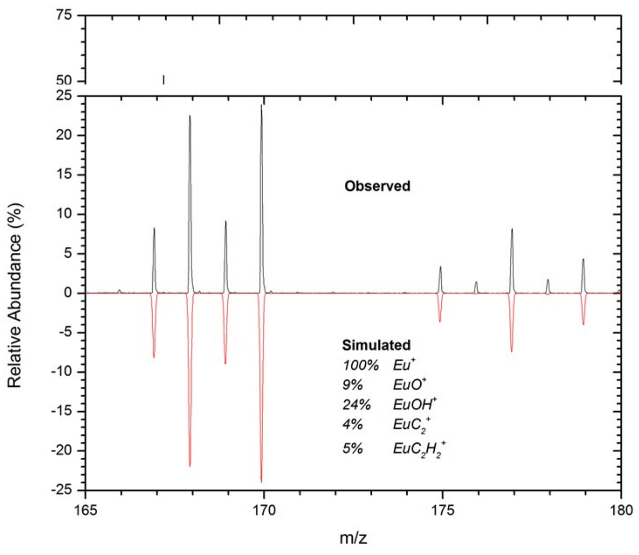


(a)

(b)

(c)

[S1] M. Strohalm, D. Kavan, P. Novak, M. Volny, V. Havlicek, mMass 3: A Cross-Platform Software Environment for Precise Analysis of Mass Spectrometric Data, *Anal. Chem*. **2010,** *82*, 4648-4651.


Observed
Simulated
100% Eu^+
9% EuO^+
24% $EuOH^+$
4% EuC_2^+
5% $EuC_2H_2^+$
Relative Abundance (%)
m/z