Software

MassFeatGen

Generation of
Numerical Features from Mass Spectra

for Multivariate Data Analyses
and Spectra Similarity Searches

User Guide

Version 1.2

Laboratory for **C**hemo**M**etrics

Institute of Chemical Engineering
Vienna University of Technology
Austria

April 2005

AUTHORS

Wilhelm **Demuth** and Kurt **Varmuza**

Laboratory for ChemoMetrics
Vienna, Austria

CORRESPONDENCE

Prof. Dr. Kurt Varmuza

kvarmuza@email.tuwien.ac.at

Laboratory for ChemoMetrics
c/o Institute of Chemical Engineering
Vienna University of Technology
Getreidemarkt 9/166
A-1060 Vienna, Austria

www.lcm.tuwien.ac.at

LIMITED WARRANTY

No warranties are made by the authors that the Program or User Guide are free of errors. The user relies on the results of the Program solely at his/her own risk. The authors are not liable to any damages caused by bugs or ambiguities in the Program, the delivered data files or the User Guide.

MassFeatGen-UserGuide-1r.doc 2005-04-25

CONTENTS

1 Introduction

Software MassFeatGen is a scientifically oriented tool for chemoinformatis, spectroscopy, and chemometrics. MassFeatGen is running under the operating systems Microsoft Windows 2000/XP.

1.1 Overview

Main function of MassFeatGen is the calculation of numerical spectral features from low resolution mass spectra (peak lists). A mass spectrum is thereby represented by a vector with the vector components being the spectral features. Typically, 14 to ca 1000 features are generated.

A spectral feature is a number - characteristic for the spectrum - that can be automatically computed from spectral data. Aim of this data transformation is to obtain a set of variables that are, hopefully, closer related to chemical structure properties than the original spectral data. Typically, nonlinear mathematical transformations are applied, considering spectroscopic ideas to some extent. MassFeatGen is primarily dedicated to be used with electron impact mass spectra.

Advantages of this approach for multivariate data analyses of mass spectra have been shown in several applications [1-5]. A pioneering paper on this subject by Crawford and Morrison [6] dates even back to year 1968. Successful uses of spectral features in mass spectra similarity searches have been described, for instance, by Clerc et al. [7,8], and by McLafferty and co-workers [9,10]. Later applications have been reported, for instance, by Drablos [11], Lebedew and Cabrol-Bass [12], and Varmuza et al. [13-15].

One of the first numerical transformations successfully used for mass spectra is the summation of intensities at masses differing by a multiple of 14 (*modulo-14 features*), which in most cases corresponds to CH_2 groups [6]. Characteristic fragments in a homologous series of compounds may be shifted by a multiple of 14 mass units; these features collect corresponding signals into the same variable. In other words, a mass spectrum is transformed into a set of 14 features by adding the peak heights at masses $m + 14z$ with $m = 1, 2, ... 14$, and $z = 1, 2, ...$

It is known for instance that mass spectra of fatty acid ethyl esters (class 1) and α-methyl-substituted fatty acid methyl esters (class 2) are very similar. If an unknown compound belongs to one of these classes, the spectra similarity hitlist usually contains compounds from both classes [16]. To test the capability of modulo-14 features and multivariate classification, a data set has been selected from a mass spectral database, containing 34 ethyl esters and 49 methyl esters. Each spectrum has been transformed into a vector containing the 14 modulo-14 features. The scatter plot in Fig. 1 uses a discriminant variable as abscissa and the scores of the first principal component as ordinate; the scatter plot is a projection of a 14-dimensional space spanned by the features. The two substance classes appear well separated. Using peak heights instead of modulo-14 features would not give such a good class separation. Data from two compounds (**1**, **2**) not used in the training are

correctly classified. Multivariate classification based on modulo-14 features is successful in this example; it is possible to discriminate the two classes of compounds that could not be distinguished by library search.

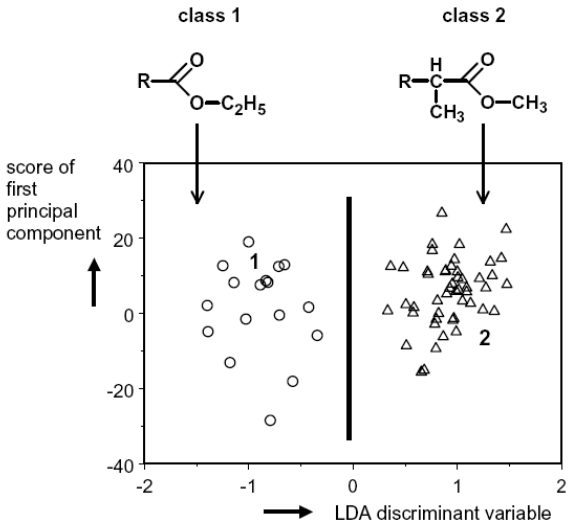


Fig. 1. Discrimination of two classes of fatty acid esters using modulo-14 features. Two unknowns are correctly assigned:
1, $C_{17}H_{35}COOC_2H_5$;
2, $C_{16}H_{33}CH(CH_3)COOCH_3$.

Another successful application of spectral features has been recently reported for spectra similarity searches with the aim to obtain hitlists that contain reference compounds with chemical structures highly similar to the structure of the unknown [15]. This strategy is essential if the unknown is not in the library.

1.2 Operating modes

MassFeatGen can be used in two different operating modes.

- **Interactive** mode with a typical Windows user interface.
- **Remote/batch** mode by calling MassFeatGen from another program. A command file (in text format) is used to transfer all parameters to MassFeatGen. So called semaphore files are used for a communication between the calling program and MassFeatGen (error messages, interrupt, etc.). In this mode no window is opened by MassFeatGen.

MassFeatGen requires two **input files** in text format (Fig. 2):

One input file contains a single mass spectrum or several mass spectra with peaklist data (integer masses and intensities). Only the widely used JCAMP-MS format (see below) is supported; a test file with mass spectra in JCAMP-MS format is provided.

The other input file contains codes for the spectral features to be generated; for typical and successful applied features such files are provided.

MassFeatGen makes two **output files** in text format (Fig. 2):

One output file contains the generated features with a row for each transformed mass spectrum and a column for each feature (typically tab-delimited, other formats can be selected by the user).

The other output file is optional and contains names for the generated features.

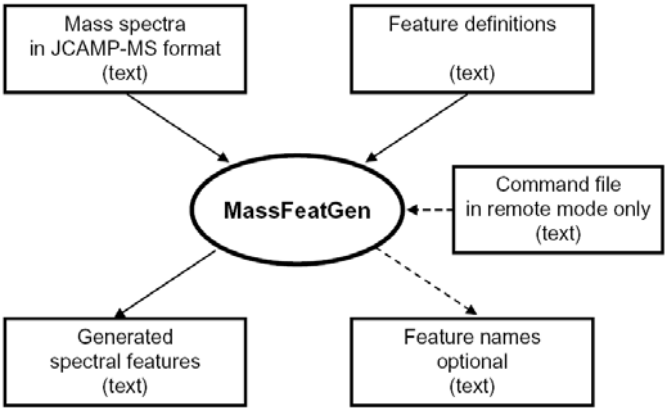


Fig. 2. Input and output files for MassFeatGen.

Example for a Feature Definition File (for generation of the 14 modulo-14 features from peaks in mass range 31 to 500, see Section 3.4:

```
MD 14 31 500 M
```

2 Installation

MassFeatGen is running under the operating systems Microsoft Windows 2000/XP. No special requirements for the personal computer and no special installation are necessary.

Look at the README.TXT file.

(1) Create (or select) a folder (or several folders) for the use of MassFeatGen.

(2) Copy the program file MassFeatGen.EXE and the other provided files into this (these) folder(s).

(3) Start the program by a double-click at MassFeatGen.EXE.
Eventually create an icon for MassFeatGen by a method available in the used operating system.

Computing time for feature generation depends on the feature type and the lengths of the mass spectra. Typical computing time for 500 spectral features and 1000 mass spectra is 1 second (Pentium 4, 2.6 GHz); the generated feature file in binary format float32 is 2.6 MB.

3 Mass Spectral Features

The features implemented in MassFeatGen are divided into eight groups and are described in Sections 3.1 to 3.8. In Section 3.0 the implemented scaling of peak intensities is explained. Section 3.9 contains a summary of the feature definitions and Section 3.10 two examples for FeatureDefinitionFiles.

The generated features are in the range 0 to 100, except a special scaling is applied.

A **FeatureDefinitionCode** defines a group of features (or a single feature) to be generated by MassFeatGen. The code consists of a keyword (for instance "MD" for modulo-14 features) and parameters; it is written in one line; separating character is the blank (space).

A **FeatureDefinitionFile** (*.txt) contains one or several FeatureDefinitionCodes.

General variables used for the definition of features are as follows.

m	Mass number (integer).
I(m)	Peak intensity (% base peak) at mass number *m.*
x	Generated feature.

3.0 Scaling of intensities (SCI)

The peak intensities of the mass spectrum can be scaled before the calculation of features. Note, however, that not all type of scaling are appropriate for all features. A given scaling (keyword SCI) in a FeatureDefinitionFile is valid for all following feature definitions unless changed by another SCI. Default is "no scaling".

An intensity threshold can be applied, a power of the intensity can be calculated, and a weighting by the mass of the peak is possible. Parameters are as follows.

I_0	Intensity threshold (in % base peak intensity); Peak intensities < I_0 are set to zero; + I_0 the range I_0 ...100 is scaled to range 0 ... 100; - I_0 no re-scaling.
exp_int	Exponent for intensity.
exp_mass	Exponent for mass number.
norm	Normalization mode for scaled intensities, M normalization to maximum 100 (default), N no normalization.

The scaling algorithm can be described in several steps as follows.

(1) Cut off (intensities below I_0 are set to zero).

IF $I < I_0$ THEN $I^* = 0$ ELSE $I^* = I$

(2) Optional range scaling of interval I_0 ... 100 to interval 0 ... 100.

If threshold I_0 is given as a negative number ($-I_0$) then I^* is not re-scaled.

$I^{**} = I^*$

If I_0 is given as a positive number, then

$I^{**} = 100\ (I^* - I_0) / (100 - I_0)$

Examples: $I^* = I_0$ gives $I^{**} = 0$
$I^* = 100$ gives $I^{**} = 100$

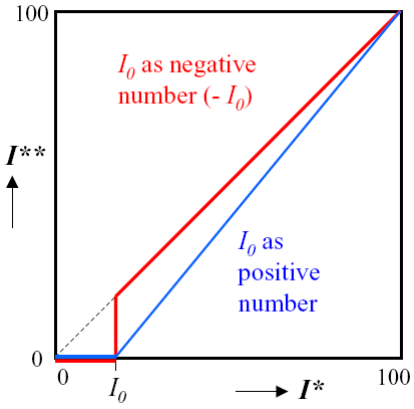


Fig. 3. Scaling.

(3) Power of intensity and weighting by power of mass.

$I^{***} = m^{exp\_mass}\ (I^{**})^{exp\_int}$

(4) Normalization to maximum 100 if parameter *norm* is "M" (recommended for most applications).

$I_{scaled} = 100\ I^{***} / \max(I^{***})$

The maximum of I^{***} is determined from all I^{***} calculated for the current spectrum. I_{scaled} is the final scaled peak intensity.

FeatureDefinitionCode

SCI	I_0	*exp_int*	*exp_mass*	*norm*

Default **`SCI`** Equivalent to **`SCI 0 1 0 M`** (no scaling).

Examples **`SCI 1 0.333 0 M`** Intensities < 1% base peak intensity are deleted; cubic root of intensities; features normalized to maximum 100.

`SCI -5 2 0.5 M` Intensities < 5% base peak intensity are deleted; intensity range 5 - 100% is re-scaled to 0 - 100%; squared intensities are weighted by square root of mass number; features normalized to maximum 100.

3.1 Intensities at selected masses or averaged intensities in mass ranges (IM)

A simple type of features are peak intensities at selected masses; thereby so called key fragments can be used as features. The averaged intensity in mass ranges can be used to characterize the shape of a spectrum or the distribution of peaks in lower and higher mass ranges. Use of the cubic root of intensities showed advantages in some applications [15].

The only parameter for this feature type is a ***MassRangeMenu*** which is explained by an example:

`IM 43,45/48,55-57` generates five features:

MassRangeMenu

(a) intensity at mass 43 (one feature),
(b) averaged intensity in mass range 45 to 48 (inclusive) (one feature),
(c) intensities at masses 55, 56, and 57 (three features).

Use a "/" between mass numbers for averaging the intensities.
Use a "-" between mass numbers for single peak intensities at consecutive mass numbers.
No blank character must be used within a *MassRangeMenu*.

FeatureDefinitionCode	**IM**	*MassRangeMenu*

Examples

`IM 51,77,105` Intensities (% base peak) at mass numbers 51, 77, and 105 are selected as features.

`IM 33/50,51/70,71/100` Averaged intensities (% base peak) of the three mass intervals 33-50, 51-70, and 71-100 are generated as features.

`IM 33-150` Intensities (% base peak) at mass numbers 33, 34, ... 150 are used as features (118 features).

```
SCI 0 0.3333 0 M
IM 33-150
```

Cubic root of intensities (scaled to range 0 - 100) at mass numbers 33 to 150 are used as features.

3.2 Intensities at selected masses in % of local intensity sum (IML)

Features of this group emphasize isolated peaks even if possessing only low intensities. The idea to these features - also called "peak intensities normalized to local ion current" - dates back to 1972 [7]. The local ion current is the sum of peak intensities in a mass interval $\pm\Delta m$ around a considered mass m. Parameters used are as follows.

Δm	One-side mass interval. Local ion current is calculated for mass interval $m - \Delta m$ to $m + \Delta m$.
MassRangeMenu	Defines the mass numbers used. See Section 3.1 (IM).

$$x = 100\, I(m) / \Sigma\, I(k) \qquad k = (m - \Delta m) \ldots (m + \Delta m)$$

If the local ion current, $\Sigma\, I(k)$, is zero, the feature cannot be calculated (division by zero) and is set to zero.

FeatureDefinitionCode

IML	Δm	*MassRangeMenu*

Example `IML 3 31,33-100` Mass interval for local ion current is 3. Normalized intensities are calculated for masses 31, 33, 34, ... 100 (69 features).

3.3 Spectra type features (TYP)

The distribution of peaks across the mass range is characteristic for some compound classes. Parameters used are as follows.

mass_low	Lower limit of considered mass interval.
mass_high	Higher limit of considered mass interval.
feature_name	Defines one of the three implemented features DUST, IBAS, EVEN (see below) or all of them (ALL).

The total intensity sum is

$$I(all) = \Sigma\, I(m) \qquad m = mass\_low \ ... \ mass\_high$$

Three heuristic features are defined as follows.

feature_name	Definition
DUST	This feature characterizes the relative peak intensities in the low mass range up to 78. $x(DUST) = 100\ \Sigma\, I(m) / I(all) \qquad m = 1 \ldots 78$
IBAS	This feature is the base peak intensity, *I(base)*, in % of the total intensity sum. $x(IBAS) = 100\ I(base) / I(all)$
EVEN	This feature measures the relative peak intensities at even mass numbers. $x(EVEN) = 100\ \Sigma\, I(m) / I(all) \qquad m = 2, 4, 6, \ldots$ (even numbers)

In case the considered mass range contains no peaks (*I(all)* = 0), the feature is set to 0 to avoid division by zero.

FeatureDefinitionCode

TYP	*feature_name*	*mass_low*	*mass_high*

Examples

`TYP EVEN 31 800` Calculates feature *x(EVEN)* for mass interval 31 to 800.

`TYP ALL 31 800` Calculates all three spectra type features for mass interval 31 to 800.

3.4 Modulo summation features (MD)

The remainder of an integer division with a denominator *z* is called "modulo *z*". For instance 43/14 and 57/14 both give a remainder of 1. Summation of peak intensities at masses with equal remainder for division by 14 play an important role in mass spectrometry [6]; such features are called "modulo-14 features". A mass difference of 14 corresponds in most cases to a CH_2-group. Characteristic fragments in a homologous series of compounds may be shifted by a multiple of 14 mass units, but modulo-14 features collect corresponding signals into the same variable.

With "modulo 14" the summation of peak intensities is possible in 14 ways:

$$
\begin{aligned}
s1 &= I(1) + I(15) + I(29) + \ldots \\
s2 &= I(2) + I(16) + I(30) + \ldots \\
&\ldots \\
s14 &= I(14) + I(28) + I(42) + \ldots
\end{aligned}
$$

From these 14 sums the 14 modulo-14 features are computed by scaling to a maximum of 100.

$$
s\_max = \max(s1, s2, s3, \ldots s14)
$$

$$
\begin{aligned}
x1 &= 100\ s1 / s\_max \\
x2 &= 100\ s2 / s\_max \\
&\ldots \\
x14 &= 100\ s14 / s\_max
\end{aligned}
$$

If all sums are zero the features are set to zero avoiding division by zero. Other types of normalization are possible but do not give features in the range 0 to 100 (see parameter *norm*).

For mass spectra usually only modulo-14 features are used, however, MassFeatGen is able to calculate this feature type for any denominator *z* (mass difference).

Parameters used are as follows:

z	Remainder of modulo function (typically 14).
mass_low	Lower limit of considered mass interval.
mass_high	Higher limit of considered mass interval.
norm	Normalization. M Maximum 100 (recommended), S Sum of modulo features is 100, N No normalization (default).

FeatureDefinitionCode

MD	*z*	*mass_low*	*mass_high*	*norm*

Example `MD 14 31 800 M` Calculates 14 modulo-14 features (maximum 100) using peak intensities from mass range 31 to 800.

3.5 Logarithmic intensity ratio features (LR)

This feature group reflects the better reproducibility of intensity ratios compared to absolute intensities. The logarithmic intensity ratio, L, of two peaks at mass m and Δm is basically defined by

$$L = \ln (I(m) / I(m+\Delta m))$$

To avoid arithmetic problems intensities below a threshold I_0 (typical 1 % base peak intensity) are set to I_0 . Features can be calculated in three different modes (0, 1, 2). Parameters used are as follows.

Δm	Mass difference (positive, typical 1 or 2).
MassRangeMenu	Defines the mass numbers, m, used. For format see Section 3.1 (IM); averaging is not allowed.
mode	0, 1, 2 (see below). Default = 0.
I_0	Intensity threshold (>0). Default = 1 % base peak intensity.

Thresholding

$$I = \max (I, I_0)$$

Mode 0

After thresholding (with $I_0 = 1$) the logarithmic intensity ratios

$$L_0 = \ln (I(m) / I(m+\Delta m))$$

are in the range -4.6 (-ln 100) to 4.6 (ln 100). A feature in the range 0 to 100 is computed by

$$x_0 = 50 + 50\, L_0 / \ln 100$$

Disadvantage of this feature is the fact that no peaks at masses m and $m+\Delta m$ give the value 50. This problem can be overcome by using mode 1 and 2.

Mode 1

The feature is defined in a way to obtain positive values only if $I(m) > I(m+\Delta m)$, otherwise 0. The resulting feature is scaled to a maximum of 100.

$$L_1 = \ln \{ I(m) / \min [I(m) , I(m+\Delta m)] \}$$

$$x_1 = 100\ L_1 / \ln 100$$

Mode 2

The feature is defined in a way to obtain positive values only if $I(m) < I(m+\Delta m)$, otherwise 0. These features are complementary to features obtained by mode 1. The resulting feature is scaled to a maximum of 100.

$$L_2 = \ln \{ I(m+\Delta m) / \min [I(m) , I(m+\Delta m)] \}$$

$$x_2 = 100\ L_2 / \ln 100$$

FeatureDefinitionCode

LR	Δm	*MassRangeMenu*	*mode*	I_0

Examples

`LR 1 43,77` Calculates two features from intensity ratios $I(43)/I(44)$ and $I(77)/I(78)$ in mode 0. Default parameters (*mode* = 0, I_0 = 1) are applied.

`LR 2 41-100 1 1` Calculates 60 features from intensity ratios $I(41)/I(43)$, $I(42)/I(44)$, ... $I(100)/I(101)$ in mode 1.

Remarks

The following three tables show the values of LR features (rounded to integer) for
$I(m)$ = 0, 10, 20, ... 100;
$I(m+\Delta m)$ = 0, 10, 20, ... 100;
I_0 = 1; *mode* = 0, 1, 2.

Mode 0

$I(m)$ =	0	10	20	30	40	50	60	70	80	90	100
$I(m+\Delta m)$											
100	0	25	33	37	40	42	44	46	48	49	50
90	1	26	34	38	41	44	46	47	49	50	51
80	2	27	35	39	42	45	47	49	50	51	52
70	4	29	36	41	44	46	48	50	51	53	54
60	6	31	38	42	46	48	50	52	53	54	56
50	8	33	40	44	48	50	52	54	55	56	58
40	10	35	42	47	50	52	54	56	58	59	60
30	13	38	46	50	53	56	58	59	61	62	63
20	17	42	50	54	58	60	62	64	65	66	67
10	25	50	58	62	65	67	69	71	73	74	75
0	50	75	83	87	90	92	94	96	98	99	100

Mode 1

$I(m+\Delta m)$ \ $I(m)$ =	0	10	20	30	40	50	60	70	80	90	100
100	0	0	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0	0	2
80	0	0	0	0	0	0	0	0	0	3	5
70	0	0	0	0	0	0	0	0	3	5	8
60	0	0	0	0	0	0	0	3	6	9	11
50	0	0	0	0	0	0	4	7	10	13	15
40	0	0	0	0	0	5	9	12	15	18	20
30	0	0	0	0	6	11	15	18	21	24	26
20	0	0	0	9	15	20	24	27	30	33	35
10	0	0	15	24	30	35	39	42	45	48	50
0	0	50	65	74	80	85	89	92	95	98	100

Mode 2

$I(m+\Delta m)$ \ $I(m)$ =	0	10	20	30	40	50	60	70	80	90	100
100	100	50	35	26	20	15	11	8	5	2	0
90	98	48	33	24	18	13	9	5	3	0	0
80	95	45	30	21	15	10	6	3	0	0	0
70	92	42	27	18	12	7	3	0	0	0	0
60	89	39	24	15	9	4	0	0	0	0	0
50	85	35	20	11	5	0	0	0	0	0	0
40	80	30	15	6	0	0	0	0	0	0	0
30	74	24	9	0	0	0	0	0	0	0	0
20	65	15	0	0	0	0	0	0	0	0	0
10	50	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0

3.6 Autocorrelation features (AC)

Loss of small stable molecules is a prominent ion fragmentation reaction. Characteristic mass differences between peaks as well as periodicities in a spectrum can be described by autocorrelation features. Parameters used are as follows.

MassDiffMenu	Defines the mass differences Δm to be used for calculation of feature (see examples).
mass_low	Lower limit of considered mass range.
mass_high	Higher limit of considered mass range.

An autocorrelation feature for a mass difference Δm is defined by

$$x(\Delta m) = 100 \sum I(m)\, I(m+\Delta m) \,/\, \sum I(m)\, I(m) \qquad m = mass\_low \ldots mass\_high$$

FeatureDefinitionCode

AC	*mass_low*	*mass_high*	*MassDiffMenu*

Examples

`AC 31 500 2,35` Calculates two features, one for Δm = 2, the other for Δm = 35; peak intensities between mass 31 and 500 (incl.) are used.

`AC 1 900 2-14,18` Calculates 14 features for Δm = 2, 3, ... 14, and Δm = 18; peak intensities between mass 1 and 900 (incl.) are used.

3.7 Peak group features (PG)

Features of this group indicate the joint presence of target peaks at defined mass numbers. Because the joint presence of peaks is more important than their intensities, scaled intensities I^{exp_int} ($0 < exp\_int <= 1$) are used in the calculation of the feature, instead of I. Furthermore, intensities equal or smaller than an intensity threshold, I_0, are set to zero, and the transformed intensities are finally scaled to the range 0 to 100. A scaled intensity, I_{scaled}, is calculated by

$$I_{scaled} = 100\,(I - I_0)^{exp\_int} / (100 - I_0)^{exp\_intc} \quad \text{IF} \quad I > I_0$$

$$I_{scaled} = 0 \quad \text{IF} \quad I <= I_0$$

Features, x, are calculated by

$$x = (1/g)\,\Sigma\, I_{scaled}(m)$$

m: sum over all *g* target peaks

Note that target peaks are only defined by their masses. Parameters used are as follows.

exp_int	Exponent for scaling peak intensities ($0 < exp\_int <= 1$).
I_0	Intensity threshold (% base peak intensity).
mode	A string containing *code* "MP", *exp_int*, and I_0 (each with 7 char.). for instance **MP00000.30000001** *exp_int* = 0.3 I_0 = **1** %
MassMenu	Masses of target peaks (see examples, no blanks).
name	A user-defined string with a maximum of 10 characters (no blanks); dedicated to identify a set of target peaks (only a comment).

FeatureDefinitionCode

PG	*name*	*mode*	*MassMenu*

Examples **PG BENZOYL MP00000.50000002 77,105**

name — *mode* — target peaks (*MassMenu*)

intensity scaling uses a threshold of I_0 = **2** %, and *exp_int* = 0.5

masses 77 and 105

Calculates one feature; value of the feature is high if at mass 77 and at mass 105 are high peaks.

```
PG AROMATIC MP00000.50000002 38,39,50,51,63,64,74-76
```

mode target peaks (*MassMenu*)

Calculates one feature; value of the feature is high if at masses 38, 39, 50, 51, 63, 64, 74, 75, and 76 are high peaks (low-aromatic series according to Mc Lafferty [17]. Scaling as in the previous example.

Remark The rather strange coding of the parameters for scaling has historical reasons.

3.8 Peak pattern features (PPS)

Features of this group indicate the presence of a specified target peak pattern - for instance an isotope peak pattern. The target pattern is defined by masses and theoretical peak intensities. The target pattern is shifted across the spectrum and the maximum correlation coefficient between target pattern and spectral peaks gives the feature. The considered mass range can be defined; it can also be fixed to a single position.

Parameters used are as follows.

name	A user-defined string with a maximum of 10 characters (no blanks); dedicated to identify a set of target peaks (only a comment).
mode	In the current version only the correlation coefficient can be used (see below) with mode **CORR**.
mass_low	Lower limit of considered mass range.
mass_high	Higher limit of considered mass range. IF *mass_low* = 0 and *mass_high* = 0 the similarity between target pattern and spectral peaks is calculated only at the mass position given in the target pattern.
mass_j, int_j	Mass and theoretical intensity of target peak *j*.
g	Number of target peaks

Similarity between target pattern and spectral peaks

$I_{th}(i)$	Theoretical intensity of peak *i* in the target pattern.
$I(i)$	Intensity of peak *i* in the spectrum.
	$i = 1 \ldots g$ (number of target peaks).

The correlation coefficient, r_m, between target peak pattern and corresponding spectral peaks (with the first peak at mass *m*) is given by

$$r_m = \Sigma\, I_{th}(i)\, I(i) \,/\, \{\, \Sigma\, [I_{th}(i)]^2 \;\; \Sigma\, [I(i)]^2 \,\}^{0.5} \qquad i = 1 \ldots g$$

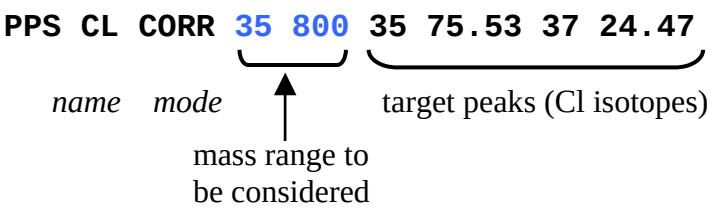
Because $I_{th}(i)$ and $I(i)$ are positive, *r* is between 0 and 1. The target pattern is shifted mass by mass across the spectrum (within the interval *mass_low* and *mass_high*, see parameter description) and the maximum correlation coefficient gives the feature

$$x = 100 \max(r_m)$$

FeatureDefinitionCode

PPS	*name*	*mode*	*mass_low*	*mass_high*	*mass_1*	*int_1*	*mass_2*	*int_2*

Examples



This feature definition calculates one feature that may be characteristic for the presence of an isotope peak pattern of chlorine. The theoretical isotope pattern (given by the target peaks) is shifted across the spectrum, starting at mass 35 and ending at mass 798. At each position the squared correlation coefficient is calculated, and the maximum correlation coefficient gives the value of the feature.

PPS HEROIN CORR 0 0 268 43 310 44 327 100 369 76

The target peaks are four prominent peaks in the mass spectrum of diacetylmorphine (masses 268, 310, 327, 369). Because the mass range is defined by "**0 0**" the correlation coefficient is calculated only at the mass position given by the target peaks. The feature can be used as a similarity measure between the mass spectrum of heroin and the investigated spectrum.

3.9 Summary of mass spectral features

0.	**Scaling of intensities**
	SCI ***I_0*** ***exp_int*** ***exp_mass*** ***norm***

I_0 Intensity threshold (in % base peak intensity), intensities < I_0 are set to 0;
+ I_0 the range I_0 ...100 is scaled to range 0 ... 100;
- I_0 no re-scaling.

exp_int Exponent for intensity.

exp_mass Exponent for mass number.

$$I_{scaled} = m^{exp\_mass} (I - I_0)^{exp\_int}$$

norm Normalization mode for scaled intensities,
M normalization to maximum 100 (default),
N no normalization.

```
SCI 1 0.333 0 M
SCI -5 2 0.5 M
```

1.	**Intensities at selected masses or averaged intensities in mass ranges**
	IM ***MassRangeMenu***

MassRangeMenu List of mass numbers to be used.
/ average of intensities in interval (incl.); e.g. **45/48**
- single intensities at consecutive mass numbers; e.g. **55-57**

```
IM 43,45/48,55-57
IM 51,77,105
```

2.	**Intensities at selected masses in % of local intensity sum**
	IML **Δm** ***MassRangeMenu***

Δm One-side mass interval. Local ion current is calculated for mass interval $m - \Delta m$ to $m + \Delta m$.

MassRangeMenu List of mass numbers to be used.
- single intensities at consecutive mass numbers; e.g. **55-57**

```
IML 31,33-100
```

3.	**Spectra type features**
	TYP ***feature_name*** ***mass_low*** ***mass_high***

feature_name Defines one of the three implemented features DUST, IBAS, EVEN or all of them (ALL).

mass_low Lower limit of considered mass interval.

mass_high Higher limit of considered mass interval.

```
TYP ALL 31 800
```

4.	**Modulo summation features**
	MD ***z mass_low mass_high norm***

z	Remainder of modulo function (typically 14).
mass_low	Lower limit of considered mass interval.
mass_high	Higher limit of considered mass interval.
norm	Normalization, M Maximum 100 (recommended), S Sum of modulo features is 100, N No normalization (default).

```
MD 14 31 800 M
```

5.	**Logarithmic intensity ratio features**
	LR ***Δm MassRangeMenu mode*** I_0

Δm	Mass difference (positive, typical 1 or 2).
MassRangeMenu	List of mass numbers to be used. - single intensities at consecutive mass numbers; e.g. **55-57**
mode	0, 1, 2. Default = 0.
I_0	Intensity threshold (> 0). Default = 1 % base peak intensity.

```
LR 1 43,77
LR 2 41-100 1 1
```

6.	**Autocorrelation features**
	AC ***mass_low mass_high MassDiffMenu***

mass_low	Lower limit of considered mass range.
mass_high	Higher limit of considered mass range.
MassDiffMenu	Defines the mass differences *Δm* to be used. - consecutive mass differences; e.g. **2-14**

```
AC 31 500 2,35
LR 31 500 2-14,18
```

7.	**Peak group features**
	PG ***name*** ***mode*** ***MassMenu***

name A user-defined string with a maximum of 10 characters (no blanks); dedicated to identify a set of target peaks (only a comment).

mode A string containing *code* "MP", *exp_int*, and I_0 (each with 7 characters). for instance **MP00000.30000001**

exp_int = 0.3 I_0 = 1 %

exp_int Exponent for scaling peak intensities ($0 < c <= 1$).

$I_{scaled} = (I - I_0)^{exp\_int}$

I_0 Intensity threshold (% base peak intensity).

MassMenu Masses of target peaks.

PG BENZOYL MP00000.50000002 77,105

8.	**Peak pattern features**
	PPS ***name*** ***mode*** ***mass_low*** ***mass_high*** ***mass_1*** ***int_1*** ***mass_2*** ***int_2***

name A user-defined string with a maximum of 10 characters (no blanks); dedicated to identify a set of target peaks (only a comment).

mode CORR (correlation coefficient).

mass_low Lower limit of considered mass range.

mass_high Higher limit of considered mass range.

IF *mass_low* = 0 and *mass_high* = 0 the similarity between target pattern and spectral peaks is calculated only at the mass position given in the target pattern.

mass_j, int_j Mass and theoretical intensity of target peak *j*.

PPS CL CORR 35 800 35 75.53 37 24.47

PPS HEROIN CORR 0 0 268 43 310 44 327 100 369 76

3.10 Examples for FeatureDefinitionFiles

Feature set 658

This FeatureDefinitionFile generates 658 features; scaling, peak group features, and peak pattern features are not used. Provided file is **`FeatureDefinition-658.txt`**.

```
IM 31,33-150
IM 33/50
IM 51/70
IM 71/100
IM 101/150
IML 3 31,33-150
TYP ALL 31 800
MD 14 31 800 M
MD 14 31 120 M
MD 14 121 800 M
LR 1 39-150
LR 2 39-150
AC 31 800 1,2,14-60
AC 31 120 1,2,14-60
AC 100 800 1,2,14-60
```

Feature set 862

This FeatureDefinitionFile generates 862 features successfully used for interpretative spectra similarity searches [15]. Provided file is **`FeatureDefinition-862.txt`**.

Feature Group No.	Feature Description	*n*	*exp_int*
IM	Intensities at masses 12, 13, 15, 17, 19-27, 29-31, 33-200	184	0.333
IM	Averaged intensities of mass intervals 33-50, 51-70, 71-100, 101-150	4	0.333
IML	Intensities normalized to local ion current for $\Delta m = \pm 3$ at masses 12, 13, 15, 17, 19-27, 29-31, 33-200	184	2
TYP	Spectra type	3	0.333
MD	Modulo-14 summation for mass intervals 31-120, 121-800, 31-800	42	0.333
LR	Logarithmic intensity ratios for mass differences of 1 and 2, and masses 39-150	224	1 *
AC	Autocorrelation for mass differences 1, 2, 14-60, and mass intervals 31-120, 100-800, 31-800	147	1
PG	Characteristic peak groups [17]	54	1 *
PPS	Isotope peak patterns for Cl_1 - Cl_5, Br_1 - Br_5, and Cl_xBr_y (x+y = 2, 3, 4, 5) up to mass 800	20	1 *
	Sum	862	-

n, number of features; *exp_int*, optimum exponent for preceding peak intensity transformation.
* Intensity transformation not reasonable for this group or included in feature definition.

```
LR 1 39-150
LR 2 39-150
AC 31 800 1,2,14-60
AC 31 120 1,2,14-60
AC 100 800 1,2,14-60

PPS Cl CORR 35 800 35 75.53 37 24.47
PPS Cl2 CORR 70 800 70 57.05 72 36.97 74  5.99
PPS Cl3 CORR 105 800 105 43.09 107 41.88 109 13.57 111  1.47
PPS Cl4 CORR 140 800 140 32.54 142 42.17 144 20.50 146  4.43 148  0.36
PPS Cl5 CORR 175 800 175 24.58 177 39.82 179 25.80 181  8.36 183  1.35 185 0.09
PPS Br CORR 79 800 79 50.54 81 49.46
PPS Br2 CORR 158 800 158 25.54 160 49.99 162 24.47
PPS Br3 CORR 237 800 237 12.91 239 37.90 241 37.09 243 12.10
PPS Br4 CORR 316 800 316  6.52 318 25.54 320 37.49 322 24.46 324  5.99
PPS Br5 CORR 395 800 395  3.30 397 16.13 399 31.58 401 30.91 403 15.13 405 2.96
PPS ClBr CORR 114 800 114 38.17 116 49.73 118 12.10
PPS Cl2Br CORR 149 800 149 28.83 151 46.90 153 21.31 155  2.96
PPS ClBr2 CORR 193 800 193 19.29 195 44.01 197 30.71 199  5.99
PPS Cl3Br CORR 184 800 184 21.77 186 42.48 188 27.57 190  7.45 192 0.72
PPS Cl2Br2 CORR 228 800 228 14.57 230 37.96 232 33.97 234 12.04 236 1.47
PPS ClBr3 CORR 272 800 272  9.75 274 31.78 276 37.29 278 18.22 280 2.96
PPS Cl4Br CORR 219 800 219 16.45 221 37.41 223 31.22 225 12.38 227 2.37 229 0.18
PPS Cl3Br2 COR3 263 800 263 11.00 265 32.24 267 34.94 269 17.40 271  4.05 273
0.36
PPS Cl2Br3 COR3 307 800 307  7.36 309 26.39 311 35.94 313 22.88 315  6.69 317
0.72
PPS ClBr4 COR3 351 800 351  4.93 353 20.88 355 34.57 357 27.65 359 10.51 361 1.46

PG MCL01 MP00000.20000005 31,50,69,100,119,131,169,181,193
PG MCL02 MP00000.20000005 38,39,50,51,63-65,74-76
PG MCL03 MP00000.20000005 39,40,51,52,65-67,77-79
PG MCL04 MP00000.20000005 87-89,99-101,112,113,125-127,138,139,150-152
PG MCL05 MP00000.20000005 45,57-59,69,71,83-85,97,98,109-112
PG MCL06 MP00000.20000005 69,81-84,95-97,107-110
PG MCL07 MP00000.20000005 73,147,207,221,281,295,355
PG MCL08 MP00000.20000005 76,90,104,118,132
PG MCL09 MP00000.20000005 77,91,105,119,133
PG MCL10 MP00000.20000005 105,119,133
PG MCL11 MP00000.20000005 63,77,91
PG MCL12 MP00000.20000005 49,63,77,91,105
PG MCL13 MP00000.20000005 77,91,105,119,133
PG MCL14 MP00000.20000005 92,106,120,134
PG MCL15 MP00000.20000005 78,92,106,120,134
PG MCL16 MP00000.20000005 79,93,107,121,135
PG MCL17 MP00000.20000005 107,121,135
PG MCL18 MP00000.20000005 93,107,121,135
PG MCL19 MP00000.20000005 94,108,122
PG MCL20 MP00000.20000005 66,80,94
PG MCL21 MP00000.20000005 66,80,94,108,122,136
PG MCL22 MP00000.20000005 39,53,67,81,95,109,123,137
PG MCL23 MP00000.20000005 81,95,109
PG MCL24 MP00000.20000005 40,54,68,82,96,110,124,138
PG MCL25 MP00000.20000005 54,68,82,96,110,124,138
PG MCL26 MP00000.20000005 83,97,111,125
PG MCL27 MP00000.20000005 41,55,69,83,97,111,125,139
PG MCL28 MP00000.20000005 55,69,83,97,111,125,139
PG MCL29 MP00000.20000005 126,140
PG MCL30 MP00000.20000005 42,56,70,84,98,112,126,140
PG MCL31 MP00000.20000005 70,84,98,112,126,140
PG MCL32 MP00000.20000005 98,112,126,140
PG MCL33 MP00000.20000005 56,70,84,98
PG MCL34 MP00000.20000005 56,70,84,98,112,126
PG MCL35 MP00000.20000005 43,57,71,85,99,113
PG MCL36 MP00000.20000005 43,57,71,85,99
```

```
PG MCL37 MP00000.20000005 85,99,113
PG MCL38 MP00000.20000005 44,58,72,86,100
PG MCL39 MP00000.20000005 44,58,72,86,100,114,128
PG MCL40 MP00000.20000005 72,86,100,114
PG MCL41 MP00000.20000005 31,45,59,73,87,101
PG MCL42 MP00000.20000005 45,59,73,87,101
PG MCL43 MP00000.20000005 87,101,115,129
PG MCL44 MP00000.20000005 59,101,115
PG MCL45 MP00000.20000005 59,73,87,101,115
PG MCL46 MP00000.20000005 102,116,130
PG MCL47 MP00000.20000005 46,60,74
PG MCL48 MP00000.20000005 60,74,88,102
PG MCL49 MP00000.20000005 46,60,74,88,102
PG MCL50 MP00000.20000005 75,89,103,117,131
PG MCL51 MP00000.20000005 33,47,61,75,89,103,117
PG MCL52 MP00000.20000005 33,47,61,75,89,103
PG MCL53 MP00000.20000005 33,34,35,45,47,61,75,89,103
PG MCL54 MP00000.20000005 47,61,75,89,103,117

SCI 0 0.33333333 0 M
MD 14 31 800 M
MD 14 31 120 M
MD 14 121 800 M

TYP ALL 31 800

IM 12-13,15,17,19-27,29-31,33-200
IM 33/50
IM 51/70
IM 71/100
IM 101/150

SCI 0 2 0 M
IML 3 12-13,15,17,19-27,29-31,33-200
```

4 Menu for Interactive Mode

4.1 File

Print... Prints displayed mass spectrum

Print Preview Preview of print page

Print Setup ... Selection of printer and print parameters

Exit

4.2 View

Toolbar Switch on/off display of Toolbar below MainMenu.

Status Bar Switch on/off Status Bar at bottom of window.

Page Format Change of format of screen display,
Window,
A4 Page,
Full Screen (1024 x 768 pixel).

Page Font Size Change of font size of spectrum lettering,
10 points (Document),
15 points (Poster),
or user-given number of points.

Page Line Width Change of line width in spectrum display,
10 (for fonts with 10 points, Document),
15 (for fonts with 15 points, Poster),
or user-given number.

Page Zoom Change of size of spectrum display,
100, 120, 150 %,
or user-given number.

4.3 Database

Import to BIB Database Import of a file with mass spectra in JCAMP format (*.TXT) and creation of a database in binary BIB format (*.SSD). No other input formats are allowed (For JCAMP format see Section 5). The imported database is automatically opened.

Open BIB Database Open of a mass spectral database in binary BIB format (*.SSD).

Close BIB Database Close of the opened mass spectral database in BIB format.

4.4 Features

Load Definition Load of a file (*.TXT) containing feature definitions (see Section 3)

Create and Save Features Calculates features as given in the loaded feature definitions for all mass spectra in the opened BIB database, and creates an output file containing the features.

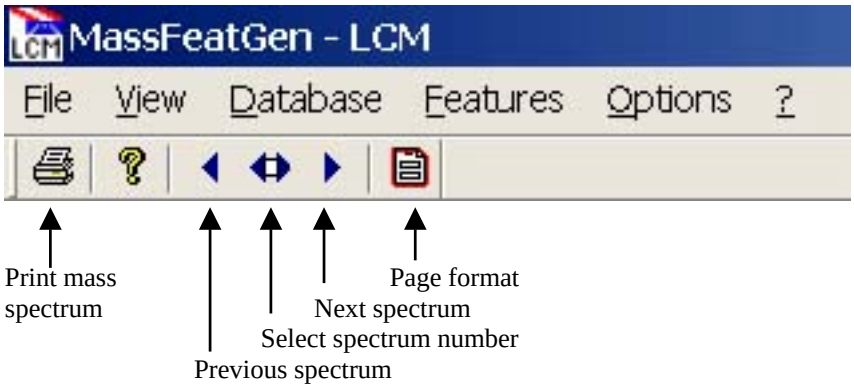
Savemode
(.) with header describing the file format and user selected data format [TEXT F 12.5, or other, FLOAT (32 bit) or DOUBLE (64 bit)]; output file *.CDM;
(.) text F20.15;
(.) text F20.5;
(.) text F12.5 (recommended);
(.) Float 32 (32 bit);
(.) Uint8 (1 byte, range 0-255).

Save Feature Names Writes feature names to a text file (*.TXT).

4.5 Options

Load Parameters Reads (eventually edited) INI file.

4.6 Toolbar

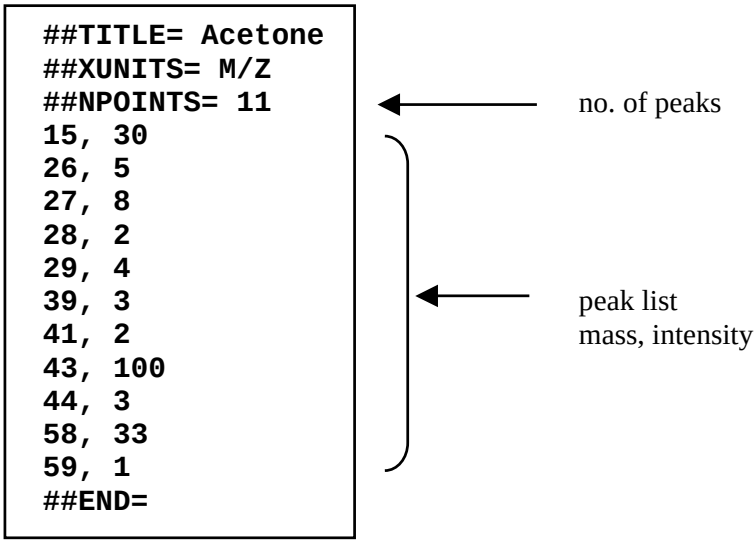


5 Mass Spectra Import

5.1 JCAMP format for mass spectra

In the current version of MassFeatGen the import of mass spectra is only possible in JCAMP format. JCAMP-DX is a widely used format for the exchange of spectra using text files. The format for mass spectra has for instance been described by Lampen et al. [18], based on previous JCAMP formats for IR [19] and NMR [20]. JCAMP uses keywords (starting with ##) to indicate different data types. Only a few keywords are needed and supported by MassFeatGen.

Example for a mass spectrum in a simple JCAMP format:



Non-integer masses are rounded by MassFeatGen (see Section 5.2).
Peak intensities are scaled to % base peak intensity.

Different versions of the JCAMP format are in use. Although most keywords not necessary for feature generation are ignored by MassFeatGen some JCAMP files may require a re-formatting for a successful import by MassFeatGen.

An example file containing 10 mass spectra in JCAMP format is provided together with the software.

During import of mass spectra a database in so called BIB format is created. The BIB format is a compact binary format for spectral and structural databases developed at the Laboratory for Chemometrics. A database file has extension *.SSD, the automatically generated index file *.SSI. Feature generation works with BIB databases.

5.2 Mass conversion

During import of mass spectra non-integer masses are rounded to integer mass numbers. Three different modes are available; mode 0 is default, mode 1 and 2 can be used by setting appropriate parameters in the INI file (MassFeatGen.ini).

m°	mass in import file
*m**	corrected mass
m	integer mass number after conversion

Mode 0 (default)

*m** = *m°* - 0.2	Fixed correction of 0.2 because many experimental mass spectra show masses slightly higher than the nominal mass.
m = rounded *m**	

Mode 1

*m** = *m°* - *dm*	Application of a constant user-defined correction *dm* (*dm* is equal to variable **MS_m_mmfa** in INI file).
m = rounded *m**	

Mode 2

*m** = *m°* - *c* . *m°*	Application of a user-defined correction that is proportional *m°* (*c* is equal to variable **MS_m_mmff** in INI file).
m = rounded *m**	

Mode 1 or 2 is active only if parameters **MS_mode**, **MS_m_mmfa**, and **MS_m_mmff** are properly defined in the INI file. The part of the INI file which is relevant for mass conversion is shown below. Note that "//" starts a comment.

```
// ********** MS Import Parameters **********

// MS mass round to integer
// mode 0     default = constant 0.2     m = m - 0.2
// mode 1     constant                   m = m - MS_m_mmfa
// mode 2     linear                     m = m - MS_m_mmff*m

MS_mode=0                                // set MS mode

// Parameters (examples)
//   mean mass fraction add, mode 1
// MS_m_mmfa=0.2
//   mean mass fraction factor, mode 2
// MS_m_mmff=1.0003
```

6 Example

The example shows the use of MassFeatGen in interactive mode step by step. A set of 10 mass spectra is used and 8 spectral features will be calculated. The spectra selected for this example and the features generated do not claim any spectroscopic relevance but are chosen for simplicity. All input files and created output files are provided together with the software. For own tests it is recommended to use other names for the output files for not overwriting the provided files.

Mass spectra in JCAMP format

File: Spec10-JCAMP-demo-a.txt

No	Compound	Brutto formula	Mol.weigth	Class
1	Hexane	C6 H14	86	1
2	Hexane, 2,3-dimethyl	C8 H18	114	1
3	1-Heptene	C7 H14	98	2
4	Cyclopropane, pentyl-	C8 H16	112	2
5	4-Octanone	C8 H16 O	128	3
6	4-Octanone, 7-methyl-	C9 H18 O	142	3
7	Chlorobenzene	C6 H5 Cl	112	4
8	Phenol, 4-chloro-3-methyl-	C7 H7 O Cl	142	4
9	Butylbenzene	C10 H14	134	5
10	Benzene, 3-butenyl-	C10 H12	132	5

The data set contains five substance classes, each with two spectra: class 1 contains alkanes, class 2 hydrocarbons with one double bond equivalent, class 3 aliphatic ketones, class 4 aromatic chloro compounds, and class 5 benzyl compounds.

Feature definitions

File: FeatureDefinition-demo.txt

```
SCI 0 0.5 0 M
IM 91
SCI
TYP ALL 1 900
LR 1 91 1 1
PG ALKANE MP00000.50000000 43,57,71
PG ALKENE MP00000.50000000 41,55,69
PPS CL1 CORR 1 900 35 75.5 37 24.5
```

This file defines eight features. First, **`SCI 0 0.5 0 M`** defines for the next feature (**`IM 91`**) a scaling with square roots of the intensities normalized to range 0 to 100. For all other features no scaling is used (**`SCI`**).

No.	Feature description
1	Intensity (scaled) at mass 91.
2	Spectra type feature DUST.
3	Spectra type feature IBAS.
4	Spectra type feature EVEN.
5	Logarithmic intensity ratio *I*(91)/*I*(92), mode 1, threshold 1%.
6	Peak group feature for presence of peaks at masses 43, 57, and 71 (alkanes).
7	Peak group feature for presence of peaks at masses 41, 55, and 69 (alkenes).
8	Peak pattern feature for one Cl-atom anywhere in the spectrum.

Work with MassFeatGen

Menu items are written in Arial; input in **`Courier`**.

1. Start program.

2. Import JCAMP file: Database | Import to BIB Database
Select file **`Spec10-JCAMP-demo-a.txt`**
A BIB database with name Spec10-JCAMP-demo-a.ssd is generated and opened. The first spectrum is displayed; other spectra may be displayed by using the arrow icons in the Toolbar.

3. Load feature definitions Features | Load Definition
Select file **`FeatureDefinition-demo.txt`**

4. Generate feature Features | Create and Save Features
Select an output format in the displayed list;
[**`Text%12.5`**] is recommended.
Enter a filename for the feature output file; for the provided file **`Feature-n10-p8-demo.txt`** has been used - use another file name for your test.

The features are generated and written to the specified file (note the message).

5. Generate feature names Optionally, a file containing names for the generated features can be created.
Features | Save Feature Names
Enter a filename; for the provided file **`FeatureNames-p8-demo.txt`** has been used - use another file name for your test.

6. Look at created files using an editor software.

The created feature file (**`Feature-n10-p8-demo.txt`** or other) is

Feature	1	2	3	4	5	6	7	8
	0.00000	97.32912	17.57160	24.35424	0.00000	70.37371	40.82412	99.85750
	3.16228	98.48934	26.97599	25.86998	0.00000	69.66106	36.87509	99.76339
	3.16228	96.46806	15.35627	36.48649	0.00000	35.34522	76.36592	99.99683
	0.00000	86.14200	17.10864	47.08298	0.00000	46.26309	77.42817	99.17346
	0.00000	84.50660	15.54002	20.31080	0.00000	93.03446	40.42821	99.80399
	0.00000	87.21874	23.19647	22.59337	0.00000	70.61784	33.63502	99.96016
	0.00000	49.67499	34.21143	58.80944	0.00000	5.56341	3.16228	99.99991
	0.00000	45.30410	21.49151	37.56716	0.00000	7.96817	5.05525	99.97254
	100.00000	28.23735	34.90401	34.20593	13.10063	7.81527	11.25892	99.77883
	100.00000	39.72125	34.84321	27.87456	42.69360	0.00000	8.16497	99.99683

The created feature name file (**`FeatureNames-p8-demo.txt`** or other) is

```
IM 91 91
TYP DUST 1 900
TYP IBAS 1 900
TYP EVEN 1 900
LR 91 1 1 1
PG ALKANE MP00000.50000000 43,57,71
PG ALKENE MP00000.50000000 41,55,69
PPS CL1 CORR 1 900 35 75.5 37 24.5
```

The feature file contains 10 rows (for 10 spectra) and 8 columns (for 8 features). The feature names contain an identification and the parameters used.

Discussion of the generated features

An over-interpretation of spectral features should be avoided - the pragmatic way to test them for desired applications is recommended. Only a short discussion of some of the generated features is tried here.

Feature 1 is the scaled intensity at mass 91; as expected it is high for the two benzyl compounds and low for the others.

Features 2 (DUST), 3 (IBAS), and 4 (EVEN) do not show evident information about the used substance classes.

Feature 5 is based on the logarithmic intensity ratio $I(91)/I(92)$; as expected it is high for the two benzyl compounds and low for the others.

Feature 6 reflects the presence of C_nH_{2n+1} ions; highest values appear with alkanes and aliphatic ketones. Feature 7 reflects the presence of C_nH_{2n-1} ions; highest values appear with alkenes.

Feature 8 is considered to be sensitive for the isotope peak pattern of Cl_1. However, feature values show that all 10 spectra contain a peak group similar to the chlorine isotope peaks.

Multivariate data analysis with the generated features

The features constitute a 10x8 matrix. Principal component analysis (PCA) is a widely used method in chemometrics to visualize such data [21-23]. The score plot in Fig. 4 shows a tendency of forming clusters for the five substance classes (software for multivariate data analysis is not contained in MassFeatGen). Other features may give another clustering.

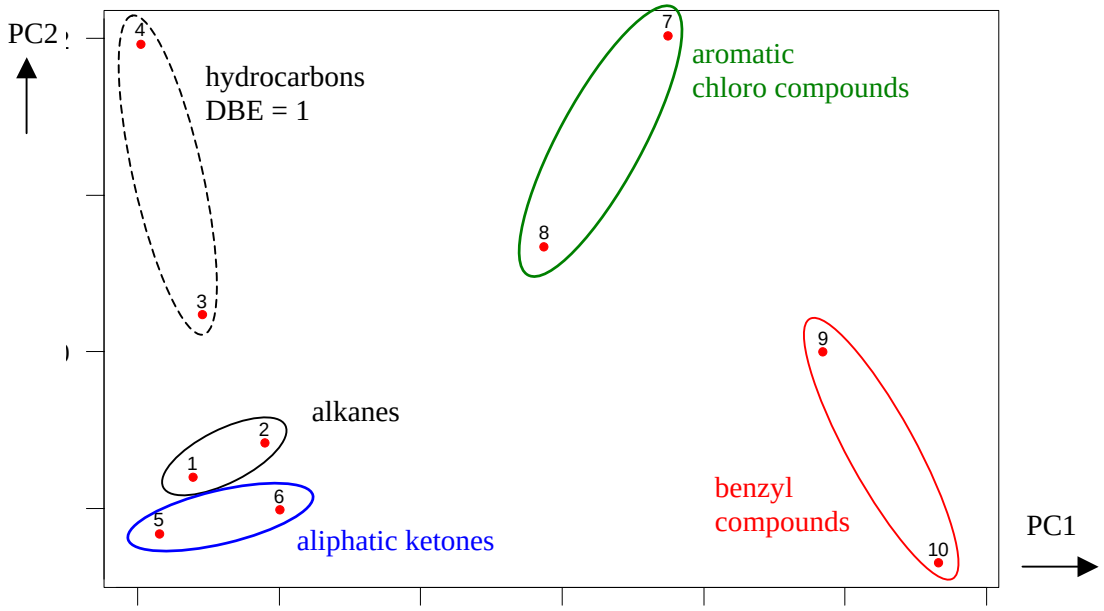


Fig. 4. PCA score plot of 10 mass spectra represented by 8 spectral features (autoscaled). Variances preserved by first (PC1) and second (PC2) principal component are 55.4 and 18.9 % of total variance, respectively. Each point corresponds to a compound; clustering of substance classes has been manually indicated by ellipses.

An alternative method to PCA in this example is hierarchical cluster analysis. Fig. 5 shows a resulting dendrogram with a similar clustering as obtained by PCA.

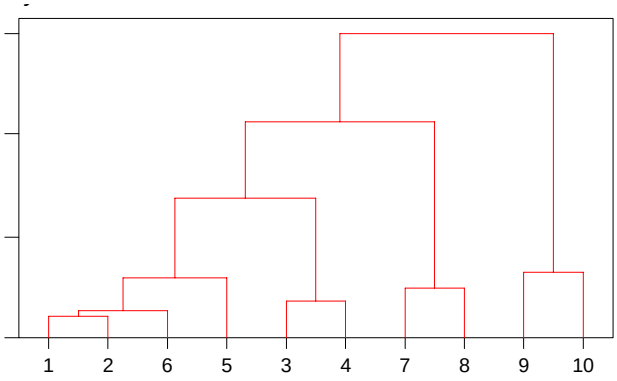


Fig. 5. Dendrogram from hierarchical cluster analysis (Euclidean distance of original features, agglomerative and complete linkage) of 10 mass spectra represented by 8 spectral features.

7 Remote/Batch Mode

MassFeatGen can be executed by calling it from another program. A command file (in text format) is used to transfer parameters to MassFeatGen. So called semaphore files are used for a simple communication between the calling program and MassFeatGen (progress, interrupt, termination). In remote mode no window is opened by MassFeatGen. Next sections describe how to call MassFeatGen from another program (in C++, Basic, Matlab, DOS batch file) and how to prepare a command file.

7.1 Calling MassFeatGen

<path1>	path for MassFeatGen.exe file
<path2>	path for command file
<command_file>	command file in text format (see section 7.3)

- Calling from a C++ program

```
system("<path1>massfeatgen.exe <path2><command_file>")
```

- Calling from a Basic program

```
shell "<path1>massfeatgen.exe <path2><command_file>"
```

- Calling from a DOS batch file

```
<path1>massfeatgen.exe <path2><command_file>
```

- From a Matlab program (see section 7.4)

```
dos('<path1>massfeatgen.exe <path2><command_file> &')
```

Remarks for the Matlab command:

* exe in lower case avoids opening of a new window for the started MassFeatGen.
* EXE in upper case opens a new window for the started MassFeatGen; this window has to be closed manually.
* **&** causes the Matlab program to continue after the start of MassFeatGen.

7.2 Communication files (semaphore files)

Progress file

MassFeatGen creates this file and writes to it the progress of computation (in %, a single integer number). If an error during the execution of MassFeatGen occurs an error code is written to the progress file (in this version code -1 is used for all detected errors). After a normal end of executing MassFeatGen this files contains "100".

Semaphore end file

MassFeatGen creates this file after closing the result file and shortly before termination. The content ("1") of this file is irrelevant. However, existence of this file tells the calling program that the result file is ready to be opened and read.

Stop file

This file can be used to terminate the execution of MassFeatGen. The file has to be created by the calling program. The content of the stop file is irrelevant.

Attention

After MassFeatGen has finished, the calling program may have to delete three files that have been created by MassFeatGen (eventually after checking their existence):

Progress file,
Semaphore end file,
Stop file.

Error handling

Errors detected by MassFeatGen during execution are reported - if possible - via the progress file by a negative error code (instead of % progress). The only error code used in this version is -1. In some error cases, the progress file cannot be opened by MassFeatGen and the error is reported in an error message box. Note that the calling program cannot (easily) check such error messages; the error message box has to be closed manually.

7.3 Command file

The command file transfers to MassFeatGen:	Names of files to be used or created, and parameters.

The command file is a text file; it consists of data lines. Each data line has the format

<keyword> = **<data>**

Some keywords contain blanks; these blanks must not be omitted; lower or/and upper case characters may be used. Blanks before and after the “=” are allowed. Comment lines start with "//".

Long filenames are supported. It is recommended to give the full path for each file name.

keyword	data

Import JCAMP	Import file with mass spectra in JCAMP format (path and file name). Optionally more than one file name can be given (using for each a new line).
BIB File	Spectra file in BIB format (path and file name).
Feature File	Feature definition file (path and file name).
Result File	Output file with generated spectral features (path and file name). For Savemode = -10 use "CDM" as extension in file name.
Progress File	Progress (%) of computation, see section 7.2 (path and file name, optional).
Sem End File	File for indication of end of feature generation, see section 7.2 (path and file name).
Stop File	File to stop execution of MassFeatGen, see section 7.2 (path and file name, optional).
Savemode	Output format (optional)* -10 CDM format ("chemometric data matrix", with header describing the file format as given by keyword **CDM format**) -4 text %20.5f with line feed after each feature, -3 text %20.5f with line feed after each spectrum, -2 text %20.15f with line feed after each feature, -1 text %20.15f with line feed after each spectrum, **0 text %12.5f with line feed after each spectrum (default) ***, 1 binary float32, 2 binary uint8 scaled to 0-255.
CDM format	This command is only valid if **Savemode** = -10. Format for output file in CDM format can be defined in three modes: **FLOAT** (32 bit) **DOUBLE** (64 bit) **TEXT <tab> %<z>.<d>f** <z> is total number of columns, <d> is number of columns after decimal point, with a tabulator character between TEXT and % For instance: TEXT<tab>%9.5f or TEXT<tab>%20.10f

* Default output format is "text %12.5f". That means a fixed width of 12 characters (columns) per feature (xxxxxx.xxxxx), and a line feed (next line) after the last feature of a spectrum (one row per spectrum).

Examples for command files

CommandFile-demo-1.txt

```
// Command File demo-1   19 April 2005
//
// Imports mass spectra from a file in JCAMP format (*.txt).
// Makes a BIB file (*.ssd) with the imported mass spectra.
// Generates spectral features using feature definitions given
//   in a feature definition file (*.txt).
// Writes generated features to a result file in format %12.5f
//   (one line per spectrum, default format).
// Uses a progress file and a semaphore end file.
//
Import JCAMP  = D:\mass\Spec10-JCAMP-demo-a.txt
BIB file      = D:\mass\Spec10-JCAMP-demo-a-bib.ssd
Feature File  = D:\mass\FeatureDefinition-demo.txt
Result File   = D:\mass\Features-generated.txt
Progress File = C:\TEMP\progress.txt
Sem End File  = C:\TEMP\finished.txt
```

CommandFile-demo-2.txt

```
// Command File demo-2   19 April 2005
//
// Uses an already existing BIB file with mass spectra (*.ssd).
// Generates spectral features using feature definitions given
//   in a feature definition file (*.txt).
// Writes generated features to a result file in format %20.15f
//   (Savemode = -2, one line per feature value).
//
BIB file      = D:\mass\Spec10-JCAMP-demo-a-bib.ssd
Feature File  = D:\mass\FeatureDefinition-demo.txt
Savemode      = -2
Result File   = D:\mass\Features-generated-2.txt
Sem End File  = C:\TEMP\end-semaphore.txt
```

7.4 Calling MassFeatGen from a Matlab program

A simple source code in Matlab and some comments (in blue color) are given for calling MassFeatGen, and then checking the progress file and finally terminating MassFeatGen. No paths for files are used in this example.

```
% prepare   command file (command.txt),
%           JCAMP file with mass spectra,
%           Feature definition file
% define variables   progressFile     name of progress file
%                    endFile          name of end semaphore file

% start submat
dos('massfeatgen.exe command.txt &')       % exe in lower case !

% loop during execution of MassFeatGen

endloop = -1;
while endloop == -1

   pause(1);          % pause of 1 second (check every second the
                      % progress file and the end semaphore file)

   % check if progress file exists
   % if yes then read the progress file
   progressFileExist = exist(progressFile,'file');
   if progressFileExist == 2
      fidProgress = fopen(progressFile,'rt');          % open file
      lin = fgetl(fidProgress);                        % read file
      progressNumber = str2num(lin);          % convert to number
      % if number read is positive then continue
      % else terminate loop because of an error
      if  progressNumber > 0
            % output progress in percent ...
      else
            % output error message ...
          endloop = 1;
      end
      fclose(fidProgress)                              % close file
   end

   % check if end semaphore file exists
   % if yes then terminate loop
   endExist = exist(endFile,'file');
   if endExist == 2
      delete(endFile);
      endloop = 1;
   end

end
delete(progressFile)
% open and read result file
```

8 References

[1] D. Cabrol-Bass, C. Cachet, C. Cleva, A. Eghbaldar, T. P. Forrest, Can. J. Chem. 73 (1995) 1412-1426.
[2] K. Varmuza, W. Werther, J. Chem. Inf. Comput. Sci. 36 (1996) 323-333.
[3] K. Varmuza, Anal. Sci. 17 (2001) i467-i470.
[4] H. Yoshida, R. Leardi, K. Funatsu, K. Varmuza, Anal. Chim. Acta 446 (2001) 483-492.
[5] W. Werther, W. Demuth, F. R. Krueger, J. Kissel, E. R. Schmid, K. Varmuza, J. Chemometrics 16 (2002) 99-110.
[6] L. R. Crawford, J. D. Morrison, Anal. Chem. 40 (1968) 1469-1474.
[7] F. Erni, J. T. Clerc, Helv. Chim. Acta 55 (1972) 489-500.
[8] P. R. Naegeli, J. T. Clerc, Anal. Chem. 46 (1974) 739A-744A.
[9] K. S. Haraki, R. Venkataraghavan, F. W. McLafferty, Anal. Chem. 53 (1981) 386-392.
[10] F. W. McLafferty, S. Y. Loh, D. B. Stauffer, in: H. L. C. Meuzelaar (Ed.), Computer-enhanced analytical spectroscopy, Vol. 2, Plenum Press, New York, NY, 1990, p. 163-181.
[11] F. Drablos, Anal. Chim. Acta 256 (1992) 145-151.
[12] K. S. Lebedew, D. Cabrol-Bass, J. Chem. Inf. Comput. Sci. 38 (1998) 410-419.
[13] K. Varmuza, J. Kissel, F. R. Krueger, E. R. Schmid, in: E. Gelpi (Ed.), Advances in Mass Spectrometry, Vol. 15, Wiley & Sons, Chichester, 2001, p. 229-246.
[14] K. Varmuza, P. He, K. T. Fang, J. Data Science 1 (2003) 391-404.
[15] W. Demuth, M. Karlovits, K. Varmuza, Anal. Chim. Acta 126 (2004) 75-85.
[16] K. Varmuza, in: P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, I. H. F. Schaefer, P. R. Schreiner (Eds.), The encyclopedia of computational chemistry, Vol. 1, Wiley, Chichester, 1998, p. 346-366.
[17] F. W. McLafferty, Interpretation of mass spectra. University Science Books, Mill Valley, CA, 1980.
[18] P. Lampen, H. Hillig, A. N. Davies, M. Linscheid, Appl. Spectrosc. 48 (1994) 1545-1552.
[19] R. S. McDonald, P. A. J. Wilks, Appl. Spectrosc. 42 (1988) 151-162.
[20] A. N. Davies, P. Lampen, Appl. Spectrosc. 47 (1993) 1093-1099.
[21] D. L. Massart, B. G. M. Vandeginste, L. C. M. Buydens, S. De Jong, J. Smeyers-Verbeke, Handbook of chemometrics and qualimetrics: Part A. Elsevier, Amsterdam, 1997.
[22] B. G. M. Vandeginste, D. L. Massart, L. C. M. Buydens, S. De Jong, J. Smeyers-Verbeke, Handbook of chemometrics and qualimetrics: Part B. Elsevier, Amsterdam, 1998.
[23] K. Varmuza, in: J. Gasteiger (Ed.), Handbook of Chemoinformatics, Vol. 3, Wiley-VCH, Weinheim, 2003, p. 1098-1133.