



DEPARTMENT OF ANALYTICAL CHEMISTRY  
UNIVERSITY OF PLOVDIV, BULGARIA

## **INFRARED SPECTRA INTERPRETATION SYSTEM**

# **IRIS**

**Version 2.2**

**Short User Guide**

January 2008

**AUTHORS: Dr. Plamen Nikolov Penchev**

Department of Analytical Chemistry  
University of Plovdiv, 24 Tsar Assen Str.  
BG-4000 Plovdiv, Bulgaria  
Tel.: +359 32 2615 447  
Fax: +359 32 235 049  
email: plamen@ulcc.uni-plovdiv.bg  
URL: <http://argon.uni-plovdiv.bg/plamen>

**Credits:** Some of the features of the design of the software are proposed and implemented by Nikolay T. Kochev, University of Plovdiv, in the program IRLIB, v.1.1, © P. Penchev, N. Kochev, 1995 - 2000. Prof. Dr. Kurt Varmuza (email: [kvarmuza@email.tuwien.ac.at](mailto:kvarmuza@email.tuwien.ac.at)) heavily influenced and contributed to some of the program algorithms.

**VERSION:** 2.2, January 2008.

**LIMITED WARRANTY:** No warranties are made by the author 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 author is not liable to any damage caused by bugs or ambiguities in the Program, the delivered Data Files or the User Guide.

**COPYRIGHT:** Plamen Penchev, 2008. The program can be used freely for education and business but a citation has to be written as *Plamen Penchev; IRIS: an Infrared Library Search System, 2008*; <http://www.kosnos.com/spectroscopy/iris>.

The three given IR libraries are composed by the spectra recorded in author's lab by himself and several of his coworkers. Some of the spectra are of a very bad quality. These IR libraries can be used only with this program and **any extraction of spectra from these three libraries is a severe violation of copyright law!**

**We are proud to announce that the IRIS program is used since 1998 for education of the students in University of Plovdiv, Plovdiv, Bulgaria.**

## Contents

Installation of IRIS on the Computer .....	3
Application Overview .....	4
Reference Section .....	5
Background Section .....	19
References .....	22

### Installation of IRIS on the Computer

To install the software make a directory on disk *c:* (or *d:*) called IRIS (or another name) and extract all files from IRIS.ZIP archive file to the created directory.

Create a separate directory, e.g. **D:\IRIS\LIBS**, for the files of spectral libraries. Copy (or extract) all available libraries to it.

To start the program in *Windows* environment first start the explorer, choose the corresponding drive, and then the directory where the program file is. The program can be started by double clicking with the mouse on the file icon. When the program is started for the first time, select menu item **OPTION**, select page **FILES AND DIRECTORIES**, and set the **LIBRARIES' DIRECTORY** edit box to this directory, i.e. **D:\IRIS\LIBS**.

Because IRIS is a 16 bit Windows program, it shows some inconveniencies by working on 32 bit Windows versions.

### Application Overview

*IRIS* is a program designed for searching in infrared spectral libraries and interpretation of infrared spectra. It is a user friendly menu driven program working in Microsoft Windows environment.

The program can perform the following operations:

- loading an unknown or a library spectrum in one of the three available buffers;  
The JCAMP-DX file format can also be used to import spectra;

- viewing of full or zoomed spectra in absorbance or transmittance units, peak tables, and the corresponding compounds' structures;
- peak search using three algorithms: forward, reverse, and scalar product.
- full spectral search applying four algorithms: sum of squared differences, sum of absolute value differences, scalar product, and correlation coefficient;
- search for a given compound's chemical name;
- interactive analysis of spectra of mixtures with the aid of multilinear regression, and subsequent graphic representation of the results;
- peak-picking from a spectral curve of an unknown spectrum, as well as creating of peak table file from a spectral library one;
- printing of the active spectrum or search results;
- classification of IR spectra according to the presence or absence of chemical substructures.

The spectra classification is performed with the aid of an artificial neural network, linear discriminant function and k-nearest neighbors' classifiers. The implemented classifiers give evidence for presence or absence of chemical substructures in compounds with unknown chemical structure.

## Reference Section

This section gives details on all the commands available in IRIS.

FILE MENU COMMANDS	
LOAD UNKNOWN SPECTRUM	
LOAD LIBRARY SPECTRUM	
LOAD HIT LIST	
LOAD HQI FILE	
IMPORT SPECTRUM	CTRL-Z
SAVE SPECTRUM	
SAVE HIT LIST	
EXIT	

**LOAD UNKNOWN SPECTRUM.** Loads an unknown spectrum in one of the three buffers A, B and C. The spectrum is loaded in the buffer that is active: see the menu item **VIEW | BUFFERS**. Buffer D is reserved for search results' spectra. The spectrum must be in a library-spectrum format (see the Background section). The

identification string of the loaded spectrum is shown on the bottom of the screen. Another alternative to that menu item is **FILE | IMPORT SPECTRUM** that imports a spectrum in JCAMP format, v.4.24 or Perkin-Elmer variant of the JCAMP format.

**LOAD LIBRARY SPECTRUM.** Loads a library spectrum in one of three buffers A, B and C. The spectrum is loaded in the buffer that is active: see the menu item **VIEW | BUFFERS**. The chemical and spectral information for the library spectrum is shown on the bottom of the screen. The information fields (or so called Header) include IUPAC name, molecular formula, Wiswesser line-notation chemical formula, molecular weight, measurement technique, and melting and boiling points. All these data are recorded into a file that is different from the spectra file. See the Background section for the details on the format of different library files.

*Speed button:* 

**LOAD HIT LIST.** Loads a previously saved hitlist. A window appears with the hitlist's identification string, and the first hitlist spectrum is shown in Buffer D. The user can inspect the hitlist spectra with the aid of menu items **VIEW | PREVIOUS HIT** and **VIEW | NEXT HIT**, or shortcut buttons **<F5>** and **<F6>**. The applied search method, number of hits and hitlist identification string can be seen with the local popup menu's item **SHOW HITLIST IDENT**. The local popup menu is invoked by clicking the right mouse button over the header's fields when the Buffer D is active. The hitlist is a simple text file whose format is described in Background section.

**LOAD HQI FILE.** Loads a text file that contains the hit quality indices (HQIs) for all library spectra. The HQIs are obtained by a given full-curve spectral search. This list of HQIs is used by KNN classification of IR spectra: see the Background section for KNN algorithm details and format of the HQI file.

**IMPORT SPECTRUM.** Loads an unknown spectrum in one of the three buffers A, B and C. The spectrum is loaded into the buffer that is active. Buffer D is reserved for hitlist spectra. The spectrum must be in JCAMP-DX (v. 4.24) format, Perkin-Elmer version of this format or a simple text file in which the spectrum is represented as "Wave number / Absorbance" pairs: for details see the Background section.

*Speed button:* 

**SAVE SPECTRUM.** Saves the active buffer spectrum into library-spectrum format file. The buffer D spectrum can also be saved with this command. A window appears with the spectrum identification string, and the user can change it or leave it the same.

**This menu item is removed from the demo version!**

**SAVE HIT LIST.** Saves a loaded or obtained by search hit list. The File-Save-Dialog is opened and the user can select the file name and the directory where the file is to be saved. If the program options “MAKE LIS FILE” and/or “MAKE HQI FILE” are switched on, the corresponding LIS and/or HQI files are created.

**Exit.** Close the program.

<b>SEARCH MENU COMMANDS</b>	
NAME SEARCH	F2
PEAK SEARCH	
SPECTRAL SEARCH	
MIXTURE ANALYSIS	F3

**NAME SEARCH.** Performs a no-case-sensitive search of a set of name fragments in library compounds names. The user can use the logic operators AND, OR and NOT and round brackets: “(“ and “)”. The logical operators can be also written in low case letters but they need to be separated from name fragments by a bracket or at least one empty space. The user can select several spectral libraries to be searched. The obtained hitlist is usually longer than that composed by the spectral and peak search. Its limit is determined by the “MAX NUMBER OF HITS” field in the program options: the last varies between 100 and 999. If there are more entries than the limit a message appears informing the user.

**PEAK SEARCH.** Performs a peak search. The search-parameters-window is opened. The user can select the spectral libraries to be searched, the spectral similarity measure and both peak-match tolerances: in abscissa (wave number) and ordinate (absorbance). The peaks can be entered manually, loaded from a peak-table file or pasted from the peak table of the active buffer. They can also be saved into a peak-table file. The peak-table file is a simple text file whose format is described in Background section. If the peak table of the unknown spectrum is to be searched the user must first perform the peak-picking proce-

dure with the menu command **TOOLS | PEAK-PICKING**. Three peak-match algorithms are available: forward and reverse searches, as well as scalar product match. For the calculation details see Background section of this manual.

*Speed button:* 

**SPECTRAL SEARCH.** Performs a full-curve spectral search. The user can select the searched spectral libraries and spectra-match algorithm in the search-parameters-window. If **SPECTRAL FILE** radio button is checked the user is prompted to load the unknown spectrum that will be searched in the selected libraries. Four different spectral similarity measures can be applied: sum of squared differences, sum of absolute value differences, scalar product, and correlation coefficient. For the calculation details see Background section of this manual.

*Speed button:* 

**MIXTURE ANALYSIS.** Performs mixture analysis of an unknown spectrum with the aid of the hitlist spectra. The user can set several parameters in the opened window: number of processed hitlist spectra,  $N_S$ , and spectral region where the calculations will be done. Clicking on the **REGRESSION** button calculates regression analysis of the active buffer spectrum with the first  $N_S$  hitlist spectra. The obtained results can be saved into a text file. Clicking on the **HITLIST SCAN** button starts a row of regression calculations with an increasing number of the hitlist spectra. The calculations stop when the number of used hitlist spectra reaches the preset value of  $N_S$  or when a strong linear dependence of hitlist spectra is obtained. The newly opened window shows hitlist compounds' pseudo-concentrations in the analyzed mixture. The straight lines in the middle of each sub-window correspond to pseudo-concentrations equal to zero. Clicking on each sub-window gives the mean, standard deviation and r.s.d. of the corresponding pseudo-concentration. The regression results can be shown in a table, saved into a simple text file, or loaded from such a file. The statistics of these results can also be saved into a text file. For the calculation details see Background section of this manual.

VIEW MENU COMMANDS	
<u>P</u> EAKE TABLE	F7
<u>S</u> TRUCTURE	F8
<u>H</u> ITLIST	F4
PREVIOUS <u>H</u> IT	F5
<u>N</u> EXT HIT	F6
PREVIOUS LIB SPECTRUM	←
NEXT LIB SPECTRUM	→
<u>T</u> RANSMITANCE	
<u>O</u> VERLAID	
<u>B</u> UFFERS	CTRL-A, ETC.

**PEAK TABLE.** Shows the active buffer spectrum's peak table if present. To make a peak table of unknown spectrum use the menu command **TOOLS | PEAK-PICKING**. For a library a new peak-table file can be derived from its spectra by a given threshold with the aid of the menu item **LIBRARY | MAKE PEAK TABLE**.

**STRUCTURE.** Shows the active buffer spectrum's structure if present. The type of the structure representation is determined by the options set up through the menu command **OPTIONS**. The first three buttons on the structure window show (or hide) the structure carbon and hydrogen atoms, and aromatic bonds. The fourth button changes the structure representation: atoms shown with atomic symbols or with filled (colored) circles. The following combinations of mouse button's clicking and holding one of the Ctrl, Alt or Shift keys gives the same and additional effects (L/RMB - left/right mouse button click):

- RMB shows/hides carbon atoms
- LMB shows/hides hydrogen atoms
- Ctrl-LMB decreases atoms radii (also the width of multiple bonds)
- Shift-LMB increases atoms radii (also the width of multiple bonds)
- Ctrl-RMB sets up the atoms radii to the value defined in Options menu
- Alt-LMB removes or sets up the atoms colors

**HIT LIST.** The hitlist obtained by a search or loaded from a file is shown on the bottom of the program window if the active buffer is D. The user can use the item **SHOW HITLIST IDENT** from the local popup menu to view the hitlist identification string (the searched unknown compound's identification, the used search method and number of hits). The popup menu is invoked with clicking of the right mouse button over the header's fields. The other popup menu items are: **SHOW LARGER/NORMAL HEADER**, **VIEW/HIDE HITLIST** and **SHOW/HIDE HEADER INFO**. The



last one shows/hides two additional header fields for the library compounds: Reference and Quality (see Background section for details on them).

**PREVIOUS HIT, NEXT HIT.** Shows previous or next hitlist spectrum; the hitlist spectra are represented only in buffer D.

*Speed buttons:*  and 

**PREVIOUS LIB SPECTRUM, NEXT LIB SPECTRUM.** Shows previous or next library spectrum; the library spectra are represented only in buffers A, B and C. The user can fast navigate through the library by using the following combinations of Shift and Ctrl keys and right and left arrows (→, ←) :

→ next library spectrum;

← previous library spectrum;

*Speed buttons:*  and 

Shift - → library spectrum with a number higher by 10 than the current one;

Shift - ← library spectrum with a number less by 10 than the current one;

Ctrl - → library spectrum with a number higher by 100 than the current one;

Ctrl - ← library spectrum with a number less by 100 than the current one;



Ctrl-Shift - → library spectrum with a number higher by 1000 than the current one;

Ctrl-Shift - ← library spectrum with a number less by 1000 than the current one.

**TRANSMITTANCE.** Toggles between both representations of IR spectra: absorbance and transmittance units.

*Speed button:*  or 

**OVERLAID.** Switches alternatively between overlaid and stacked view of the spectra of all active buffers. If spectra are stacked (separated) the user can select a spectrum to be active by clicking (LMB) onto its window. If spectra are overlaid the switch between buffers is done by clicking the mouse in that part of the window that corresponds in height to one of the buffers.

*Speed button:*  or 

**BUFFERS | A ON/OFF, ETC.** Turns on/off the view of the corresponding spectral buffer. The newly shown buffer becomes active, and its spectrum header is

shown onto the bottom of the program window (if spectrum present). The same operations can easily be done with the four speed buttons shown below.

Speed buttons: 

#### **ADDITIONAL OPTIONS TO MANIPULATE THE SPECTRA VIEW:**

To zoom all shown spectra in a given spectral interval press left mouse button in the place corresponding to one of the interval limits, hold the button pressed and drag the mouse pointer to the other interval limit, release the button. To zoom out the zoomed in spectra click with right mouse button onto the spectra window.

<b>LIBRARY MENU COMMANDS</b> MAKE PEAK TABLE CREATE NEW LIBRARY DELETE LIBRARIES MERGE LIBRARIES ADD SPECTRUM TO LIBRARY REMOVE SPECTRUM FROM LIBRARY EDIT LIBRARY HEADER REPLACE LIBRARY STRUCTURE
---

**MAKE PEAK TABLE.** Makes a peak table file for a given library. The user can select the threshold for the peak-picking procedure and the library name. Our experiments show that the best threshold values are between 0.01 and 0.03 a.u.

**CREATE NEW LIBRARY.** Starts building of new spectral library. **Not implemented in the demo version!**

**DELETE LIBRARIES.** Deletes several spectral libraries. **Not implemented in the demo version!**

**MERGE LIBRARIES.** Merges several spectral libraries. **Not implemented in the demo version!**

**ADD SPECTRUM TO LIBRARY.** Adds a new spectrum to a library. **Not implemented in the demo version!**

**REMOVE SPECTRUM FROM LIBRARY.** Removes a spectrum from a library. **Not implemented in the demo version!**

**EDIT LIBRARY HEADER.** Removes a spectrum and its header from a library. **Not implemented in the demo version!**

**REPLACE LIBRARY STRUCTURE.** Replaces a structure of a library entry with another structure saved in ISISDraw Mol file. **Not implemented in the demo version!**

**TOOLS MENU COMMANDS**

PEAK-PICKING

SPECTRA DIFFERENCE

**PEAK-PICKING.** Makes a peak table for the current buffer's spectrum. The user determines peak's threshold with a scroll bar; the **PICK PEAKS** button derives a peak table from the spectrum in the active buffer, and **OK** button puts the table into the buffer. The peak table can be pasted into the peak list of the Searching-parameters windows when the Peak-Search routine is started: see menu item **SEARCH | PEAK SEARCH**. *Note: the peak table of a library spectrum is NOT replaced in the peak-table file with the newly derived peaks: scrolling the library returns the original peak table of the library spectrum.*

**SPECTRA DIFFERENCE.** Opens a parameters window for calculation of spectral difference between two spectra. The user can select the result (R), minuend (M) and subtrahend (S) buffers, as well as the multiplier (k) by which the subtrahend is multiplied:  $R = M - k \cdot S$ . The coefficient k is determined by dragging the Multiplier scroll bar. The **OK** button puts the result difference spectrum in the chosen result buffer. While Spectra difference window is active the user can zoom in and out the spectra, switch on and off buffers, as well as alternate between absorbance and transmittance or stacked and overlaid spectra display. The spectra subtraction routine is very useful for analyzing spectra of mixtures by so called stripping out the components spectra: see Background section for details.

**PRINT MENU COMMANDS**

PRINT ACTIVE SPECTRUM

PRINT ALL SPECTRA

PRINT HIT LIST

**PRINT ACTIVE SPECTRUM.** Prints the spectrum in the active buffer. The setup dialog is opened, and the user can set up which items of the spectral and chemical information to be printed. **Not all options are still implemented!**

Speed button: 

**PRINT ALL SPECTRA.** Prints all shown spectra. The spectra are printed either stacked or overlaid as they are represented currently on the program window. The setup dialog is opened, and the user can set up which items of the spectral and chemical information to be printed. **Not all options are still implemented!**

Speed button: 

**PRINT HIT LIST.** Prints the contents of a hitlist. **Not all options are still implemented!**

#### OPTIONS MENU COMMAND

It opens a multi-page dialog window for setup of the program options. There are three pages: **FILES AND DIRECTORIES**, **STRUCTURES** and **SPECTRA**.

**FILES AND DIRECTORIES** page is for setup of library files' extensions and the directory where these files are located. For the format of the library files and their purpose see the Background section of this manual.

**STRUCTURES** page is for setup of the representation of the library structures, on the screen, and by printing.

**SPECTRA** page is for setup of the representation of the library spectra. **NUMBER-OF-HITS** value determines how many hits compose the hitlist when applying the seven algorithms for **FULL-CURVE SPECTRAL** and **PEAK SEARCH**. These hits are arranged by decreasing their hit quality indices. **MAX NUMBER-OF-HITS** value determines how many hits compose the hitlist when applying the **NAME SEARCH**. As these search result are characterized by presence (or absence) of the name fragments in the compounds' names, the hitlist entries are not sorted and an extensive list is to be expected in some cases. That is why this number is usually quite higher than the previous one. **BASE-LINE-HEIGHT** (measured in pixels) determines the representation of the spectra on the program window, and the **DIFFERENCE-BASELINE HEIGHT** (in percent from the height of spectrum window) –

representation of the spectra by performing spectra subtraction. A small percentage hides negative parts of the result spectrum but expands the positive parts of all spectra. On the other side, a high percentage reveals negative parts of the result spectrum but “squeezes” all spectra.

If **STRETCHED SPECTRA** check box is checked the zoom of the spectra in abscissa leads to a zoom in ordinate (to some extent). Very small absorbance values are not fully zoomed so that the user is not confused about their magnitude. The remaining two check boxes, **MAKE LIS FILE** and **MAKE HQI FILE**, determine if two additional files are created by saving the hitlist. The LIS file is a text file with the library number of the hits and the corresponding libraries' names. This file is one of the files used by the program ToSiM [1], and can be loaded and used by the program IRIS through **CLASSIFIERS** menu. The second file (HQI file) is a text file containing HQI values of all library entries obtained by **FULL-CURVE SPECTRAL SEARCH** in only one spectral library. The file can be loaded through menu **LOAD HQI FILE** and the HQI values can be used by a KNN classification. For the details about both files see the Background section.

<b>INTERPRETATION MENU COMMANDS</b> IR SPECTRA CLASSIFICATION
--

**IR SPECTRA CLASSIFICATION.** Starts the INFRARED SPECTRA CLASSIFICATION MODULE that classifies IR spectra according to the presence or absence of a set of chemical substructures in organic compounds. The module can use classifiers based on the artificial neural networks (ANN), linear discriminant analysis (LDA) and k-nearest neighbors method (KNN). Also the simple classifiers based on expert knowledge for the characteristic intervals (EXS) can be applied. This section gives details on all the commands available in the module.

**FILE MENU COMMANDS**

LOAD CLASSIFIERS  
APPEND CLASSIFIERS  
SAVE CLASSIFIERS  
SAVE RESULTS  
EXIT

**LOAD CLASSIFIERS.** Loads classifiers into the module memory. The classifiers could be four types: ANN, LDA, EXS and KNN. First type uses artificial neural

network, the second - linear discrimination analysis, the third - characteristic intervals, and the last - KNN method. The classifiers file is a text file: see the Background Section of this manual.

**APPEND CLASSIFIERS.** Appends classifiers to the list of classifiers previously loaded in the module's memory. If there are no previously loaded classifiers this menu item is equivalent to menu item **LOAD CLASSIFIERS**.

**SAVE CLASSIFIERS.** Saves the loaded classifiers into a classifiers file.

**SAVE RESULTS.** Saves classification results obtained by classification (the menu item **INTERPRETATION | INTERPRETATION**). The results' file is a text file: see the Background Section of this manual.

**Close.** Closes the module.

<b>INTERPRETATION MENU COMMAND</b> INTERPRETATION
--

**INTERPRETATION.** Classifies an IR spectrum with a set of loaded classifiers. The threshold classification precision is determined with the scroll bar on the front panel, and is shown on the left of it. The spectrum's peak table is automatically used for classification when the user selects the menu item **INTERPRETATION | IR SPECTRA CLASSIFICATION** from IRIS program's menu. When KNN classifiers are applied the classification uses the HQI values obtained for all library entries by **FULL-CURVE SPECTRAL SEARCH**. These values can also be loaded from an HQI file: see the menu item **FILE | LOAD HQI FILE**.

<b>CLASSIFIERS MENU COMMANDS</b> REMOVE RENAME
--

**REMOVE.** Removes a classifier that is selected in the classifiers' list. The program prompts a confirm dialog box. If none of the classifiers is selected then the first one is removed.

**RENAME.** Renames a classifier that is selected in the classifiers' list. The program prompts a dialog box. If none of the classifiers is selected then the first one is renamed.

Not implemented in the demo version!

#### **ADDITIONAL REMARKS**

The KNN classification is performed with the same module that classifies IR spectra with the aid of ANN and LDA classifiers, i.e. the module started with menu command **INTERPRETATION | IR SPECTRA CLASSIFICATION**. The set of KNN classifiers is loaded with menu command **FILE | LOAD CLASSIFIERS**. The ANN and LDA classifiers interpret the spectrum in the active buffer; the KNN classifiers interpret the search results (or those loaded from HQI file). That is why the results can be presented in the window that are for two different (unknown) compounds. To avoid the misleading of the user the INTERPRETATION module gives two hints: one is the identification string of the spectrum in the active buffer, the other is that of the search results.

Before applying a KNN classifier the user has to search the unknown spectrum in the spectral library. The searched library has to be the same as that used by the generation of the classifier, and both should have the same number of spectra, also. The search method needs also to be the same as that used by the classifier generation. The results of the previously performed search, that were saved in a hitlist file, cannot be used because the classifier uses the hit quality indices of the entries from its learning set (they are with class affiliations 2 and 3), and these entries are distributed, as a rule, also outside the hitlist. That is why, when a search is performed, it is desirable the results to be saved into the so called *HQI file* (from hit quality index); this is a file containing the HQIs for all library entries. For that to happen when the hitlist is saved, the corresponding check box in SPECTRA page of OPTIONS windows (menu command **OPTIONS**) has to be checked.

## **BACKGROUND SECTION**

(some technical details are not described for the free demo version of IRIS)

1. Algorithms
  - 1.1. Spectral Search Algorithms
    - 1.1.1. Peak Search Algorithms
    - 1.1.2. Full-curve Spectral Search Algorithms
  - 1.2. Mixture Analysis
  - 1.3. IR Spectra Classification
    - 1.3.1. IR Spectra Classifier
    - 1.3.2. Spectral Features for LDA & ANN Classifiers
    - 1.3.3. Neural Network Model
    - 1.3.4. KNN Classification
    - 1.3.5. KNN Classifiers' Development
2. Format of Files
  - 2.1. Format of Library Files
    - 2.1.1. Library Index File Format
    - 2.1.2. Header File Format
    - 2.1.3. Spectra File Format
    - 2.1.4. Peak Table File Format
    - 2.1.5. Structure File Format
  - 2.2. Unknown Spectrum File Format
  - 2.3. JCAMP-DX 4.24 File Format
  - 2.4. Perkin-Elmer JCAMP-DX File Format
  - 2.5. ASCII Spectrum File Format
  - 2.6. Hitlist File Format
  - 2.7. Peak Table File Format
  - 2.8. Mixture Analysis Result File Format
  - 2.9. IR Spectra Classifiers File Formats
    - 2.9.1. Neural Network Classifier Format
    - 2.9.2. Linear Discriminant Function Classifier Format
    - 2.9.3. KNN Classifier Format
3. Sample Files



## 1. Algorithms

This section describes almost all mathematics in the implemented algorithms.

### 1.1. Spectral Search Algorithms

Seven algorithms for searching in spectral collections are implemented in the IRIS program. Three of them use spectrum bands' intensities and positions, the other four match full spectral curves. All algorithms perform sequential searches through the libraries that had been selected.

#### 1.1.1. Peak search in spectral libraries

Peak search algorithms described in the literature [2] can be generally divided into two types: *forward* ones used for identification of pure compounds, and *reverse* ones applied for identification of the components of organic mixtures. Forward peak matching accounts how many peaks of the unknown spectrum are matched to the peaks of the reference spectrum, and reverse one how many peaks of the reference spectrum are matched to the peaks of the unknown spectrum.

In IRIS program hit quality indices (HQIs) for these algorithms are three digit numbers composed of figures *A*, *B* and *C*, that are calculated independently as it is shown in the next paragraphs:

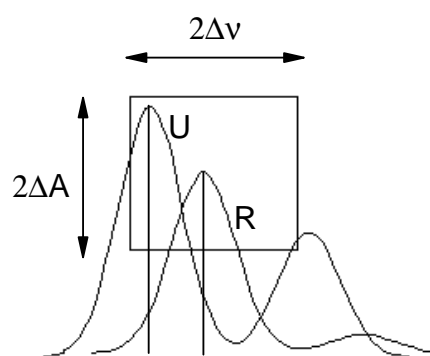
- forward:  $HQI_F = ABC$
- reverse:  $HQI_R = BAC$

The peaks in the unknown spectrum compose the set *U* (with *N* elements), these in the reference spectrum – the set *R* (with *M* elements), and *I* represents the intersection of *U* and *R*; it contains *K* elements. If the two sets, *U* and *R*, are identical the following relations are valid:

$$U = R = I \quad \text{and} \quad N = M = K$$

In the applied spectroscopy a full coincidence of both sets is usually not occurring even for two different IR spectra of the same compound. That is why the peak matching is done with a tolerance defined by the user. In IRIS program two tolerances are used – one along the abscissa (wave numbers),  $\Delta\nu$ , and one

along the ordinate (absorbance),  $\Delta A$ . The coincidence of a peak from  $U$  and a peak from  $R$  means that the  $U$ -peak is within a rectangle with side lengths  $2\Delta v$  and  $2\Delta A$  and a center in  $R$ -peak, see Figure 1.



**Figure 1.** Matching two peaks, one from the unknown spectrum ( $U$ ), the other from the reference one ( $R$ ), using tolerances along the abscissa and the ordinate.

Three numbers rounded to integers are calculated:

$$A = 9 K / M; \quad B = 9 K / N; \quad C = 9 (1 - d \sum |v_U^K - v_R^K|);$$

where  $v_U^K$  and  $v_R^K$  ( $k = 1, 2, \dots, K$ ) are wave numbers of the unknown and reference peak, and the sum is taken over all matched peaks.

$A$  shows the number of matched peaks as a percentage of peaks of the unknown spectrum,  $B$  - as a percentage of peaks of the unknown spectrum.  $C$  determines how close the matched peaks are along the abscissa. Constant  $d$  depends on the tolerance used along the abscissa,  $\Delta v$ , and number of the matched peaks,  $K$ :  $d = 1 / (K \Delta v)$ . If all matched peaks of the unknown spectrum lie on the “border” of the tolerances’ rectangle, then  $C = 0$ .

The third HQI implemented in IRIS (so called *peak scalar product* HQI) is calculated as a scalar product of two vectors: the first vector is composed of peaks of the unknown spectrum, the second one of peaks of the reference spectrum:

$$HQI_p = \frac{\sum_k A_k^U A_k^R}{\|A^U\| \cdot \|A^R\|};$$

only the matched peaks are multiplied and summed in the numerator.

### 1.1.2. Full-curve Spectral Search

Four different measures for the similarity of two IR spectra are used [3,4]. The

spectral matching uses the full-curve spectra containing 801 absorbance values between  $3700\text{ cm}^{-1}$  and  $500\text{ cm}^{-1}$  with a constant sampling interval of  $4\text{ cm}^{-1}$ .

The measures ( $HQI_1$  to  $HQI_4$ ) range between zero and 999 (the last value is obtained for identical spectra). Let  $N$  be the number of absorbance values in a spectrum (in the program  $N$  is equal to 801);  $A_k^U$  and  $A_k^R$  are the absorbances of the interval  $k$  in the spectrum of the unknown and in that of the reference (library) spectrum, respectively.

- Hit quality index  $HQI_1$  is based on the sum of the squared absorbance differences,  $S_1$ , equation (1).

$$HQI_1 = 999(1 - S_1) \quad \text{with } S_1 = \sqrt{\sum_k (A_k^U - A_k^R)^2 / N} \quad (1)$$

- Hit quality index  $HQI_2$  is calculated from the sum of the absolute absorbance differences,  $S_2$ , equation (2).

$$HQI_2 = 999(1 - S_2) \quad \text{with } S_2 = (1/N) \sum_k |A_k^U - A_k^R| \quad (2)$$

- Hit quality index  $HQI_3$  is the scalar product of two spectral vectors normalized to unit length, i.e. cosine of the angle between them, equation (3).

$$HQI_3 = 999S_3 \quad \text{with } S_3 = \frac{\sum_k A_k^U A_k^R}{|A^U| \cdot |A^R|} \quad (3)$$

- Hit quality index  $HQI_4$  is based on the correlation coefficient, equation (4).

$$HQI_4 = 999(S_4 + 1)/2 \quad \text{with } S_4 = \frac{\sum_k (A_k^U - \overline{A^U})(A_k^R - \overline{A^R})}{\sqrt{\sum_k (A_k^U - \overline{A^U})^2 * \sum_k (A_k^R - \overline{A^R})^2}} \quad (4)$$

## 1.2. Mixture Analysis

The analysis of spectrum of an organic mixture is based on the multilinear regression calculations [4]. The results are the so called pseudo-concentrations of the components in a mixture (matrix  $C_{1,H}$ ); all subscript of matrices express their dimensions. They are calculated from the spectra in the hitlist ( $S_{N,H}$ ) and mixture spectrum ( $M_{1,N}$ ), with increasing number ( $H$ ) of hitlist spectra involved, according to equation (5):

$$C_{1,H} = M_{1,N} S_{N,H} \cdot^T (S_{H,N} S_{N,H}^T)^{-1}; \quad (5)$$

“ $T$ ” and “ $-1$ ” in the superscript denote a transposed and inverse matrix, respectively, and  $N$  and  $H$  are the number of spectral points and the number of hitlist spectra involved in the calculations, respectively. The result is a matrix of  $C_i^j$  values, where the upper indices designate the regression number, and the lower - compound number:

$$\begin{array}{cccccc} C_1^1, & 0, & 0, & 0, & \dots & 0, & 0 \\ C_1^2, & C_2^2, & 0, & 0, & \dots & 0, & 0 \\ C_1^3, & C_2^3, & C_3^3, & 0, & \dots & 0, & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ C_1^n, & C_2^n, & C_3^n, & C_4^n, & \dots & C_{n-1}^n, & C_n^n \end{array} \quad (6)$$

The zeros in the upper right part of the matrix  $C$  mean that the corresponding hitlist compounds are not regarded *a priori* as mixture components, i.e. their spectra do not take part in the corresponding regression calculations. The program presents graphs  $C = f(h)$ ;  $h$  is the number of the hitlist spectra involved in the calculations, i.e. the values in first column, in second column, etc. The user can decide which compounds are in the mixture by comparing the relative stability of the corresponding curves. This stability can be estimated either by a visual inspection of the curves or by comparing the relative standard deviation (r.s.d.) of curves ordinates. The latter are calculated only for the nonzero values neglecting the upper right zeroes. The lesser the r.s.d. (more “stable” graph) the more probable is that the corresponding hitlist compound is a component of the studied mixture. We recommend the curves with negative average pseudo-concentration to be not regarded despite their low r.s.d.

Matrix  $C$  does not represent the exact concentrations because of the normalization of the library spectra in the range 0 - 1 a.u. and the differences in sample preparation: the studied spectrum and all library spectra were registered with arbitrary path length (for liquids) or arbitrary amount of compound in the KBr pellet (for solids). Therefore we call  $C$  values pseudo-concentrations, and these values have no impact on the decision of the user which hitlist compounds are mixture components.

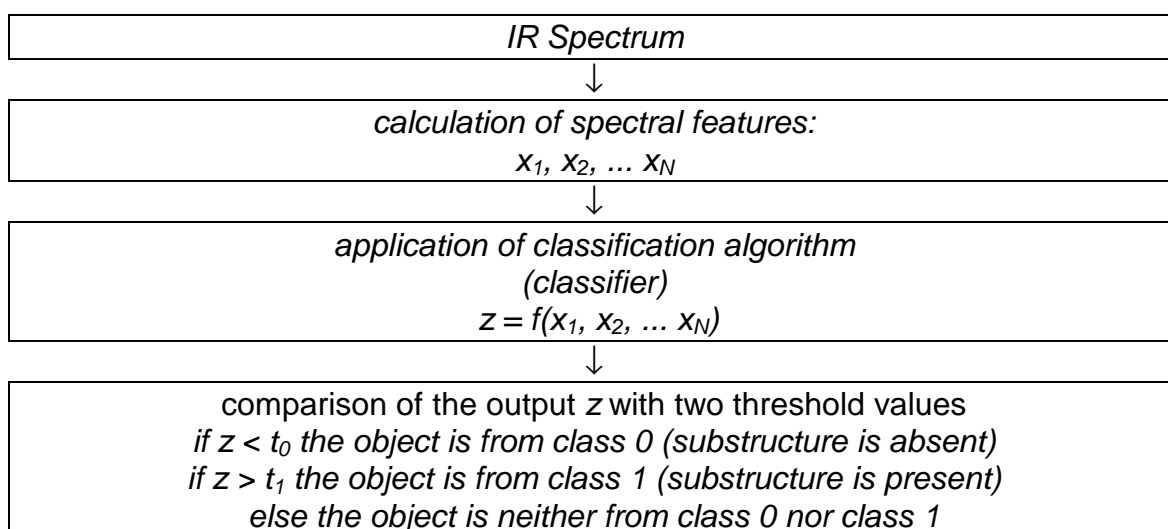
### 1.3. IR Spectra Classification

#### 1.3.1. IR Spectra Classifier

A classifier represents a mathematical algorithm for transforming the IR spectrum of unknown compound (the compound under study) to the conditional probability of presence,  $P_1$ , or absence,  $P_0$  of a given chemical substructure in this compound. This chemical substructure (e.g. methylene group) or, correspondingly, the precisely defined structural property (e.g. alkane), associated with the classifier, is called *classifier's substructure*.

For the LDA and ANN classifiers, first an array of numbers, so called features, is calculated from the IR spectrum of the unknown compound. The application of classifier function gives so called the output variable,  $z$ . For the KNN classifiers the output variable is the percentage in the hitlist of the compounds that contain the classifier's substructure.

Both conditional probabilities are tabulated in the classifier as functions of  $z$ :  $P_0 = P_0(z)$  and  $P_1 = P_1(z)$ . As the user sets the threshold values of the required classification precision,  $P_t$ , the classification is equivalent to comparison of  $z$  with two threshold values,  $t_0$  and  $t_1$ : for them it is fulfilled that  $P_0(t_0) = P_t$  and  $P_1(t_1) = P_t$ . The way the LDA and ANN classifications are performed is depicted in Figure 2. The KNN classification is described in details in one of the next paragraphs.



**Figure 2.** Application of LDA or ANN classifier.

The first step in the generation of a classifier is the composition of a *learning* (or *training*) set and a *validation set* that are relevant to the classifier's substructure.

Substructure searches in the database followed by a random selection of compounds result in two files: the first consisting of data for  $L_0 + V_0$  compounds *not* containing the substructure (class 0), and the second consisting of data for  $L_1 + V_1$  compounds containing the substructure (class 1).  $L_k$  and  $V_k$  are the numbers of entries from class  $k$  ( $k = 0, 1$ ) in the learning and validation set, respectively; a typical value for  $L_k$  and  $T_k$  is 250. For some substructures only a smaller number of compounds is available in the database. Isotopically labeled compounds and compounds containing metal atoms have been excluded from both sets.

The ‘zero’ class output values are taken as 0.0, while those of ‘one’ class as 1.0. Separation of the two classes obtained by training of a neural network does not mean automatically that the calculated functional relation

$$z = f(\text{Inp}_1, \text{Inp}_2, \dots, \text{Inp}_N)$$

has a general applicability for all data sets of objects of these abstract classes. That is why it is necessary to test the obtained mathematical model, i.e. the trained neural network, with a set of objects from both classes. It is preferable that these objects were not used by the calculation of the model, i.e. by the neural network’s training. The details of calculation of precision as function of output variable,  $z$ , are given in our paper [5].

### 1.3.2. Spectral Features for LDA & ANN Classifiers

Spectral features are a set (an array) of numbers that characterize the IR spectrum. Most of the works in IR spectra classification use features based on predetermined fixed wavelength intervals. We introduce a different approach by choosing the wavelength intervals ( $\nu_1, \nu_2$ ) individually for each classified substructure and by defining two types of features: interval ones (INT) and logarithmic ones (L12).

Feature  $INT(\nu_1, \nu_2)$  is the intensity of a spectral band as given in equation (7), with  $A_{max}$  being the maximum absorbance in this interval.



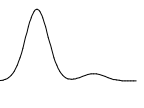


$$INT(\nu_1, \nu_2) = \begin{cases} A_{max}/100 \\ 0, \quad \text{if no peak in } (\nu_1, \nu_2) \end{cases} \quad (7)$$

This feature type is used in most knowledge-based IR spectra interpretation systems.

Feature  $L12(\nu_1, \nu_2)$  is calculated from the logarithmic absorbance ratio as given in equation (8), with  $A_{sec}$  being the absorbance of the second highest peak in the interval.

$$L12(\nu_1, \nu_2) = \begin{cases} [a - \lg(A_{max}/A_{sec})]/a; & a = 2 \\ 0, & \text{if less than two peaks are} \\ & \text{present in } (\nu_1, \nu_2) \end{cases} \quad (8)$$

This feature considers that some chemical substructures give rise to two or more characteristic bands in a given spectral interval. A similar feature was successfully used for substructure classification from mass spectra [6]. The constant  $a$  in equation (8) scales the feature to the range zero to one. If the maximum and minimum absorbances are 100 and one, respectively,  $a$  has to be equal to two. The maximum value is reached when the two largest peaks in the interval are equally sized ( $A_{max} = A_{sec}$ ), Figure 3.

peaks					no peaks 
$L12(\nu_1, \nu_2)$	1.00	0.85	0.50	0.00	0.00

**Figure 3.** Examples of the values for feature  $L12(\nu_1, \nu_2)$  for different peak absorbances in the considered wavelength interval.

Selection of appropriate wavelength intervals ( $\nu_1, \nu_2$ ) is the crucial task in feature generation. The selection of the spectral intervals for the used spectral features is discussed in details in our paper [5].

### 1.3.3. Neural Network Model

The program uses a forward-feed three-layer neural network [7] (including input layer). The neural network's coefficients were set by a back-propagation-of-error algorithm. The number of inputs neurons is equal to the number of features (variables). The number of output neurons is equal to one. This is determined by the problem posed to the program – performing a binary classification.

The training of the neural network (starting with random coefficients) had been performed with features that have been normalized with the mean  $M_j$  and variance  $V_j$  for every variable  $j$  of the training data set (features  $X_{i,j}$  of objects from classes 0 and 1) as given in equations (9).

$$M_j = \left( \sum_{i=1}^{NoObjs} X_{i,j} \right) / NoObjs \quad (9a)$$

$$V_j = \sum_{i=1}^{NoObjs} (X_{i,j} - M_j)^2 / (NoObjs - 1); \quad (9b)$$

$$Inp_{i,j} = (X_{i,j} - M_j) \sqrt{V_j}; \quad (9c)$$

for  $j = 1$  to  $NoFeats$ ; for  $i = 1$  to  $NoObjs$

where  $NoObjs$  is the sum of the numbers of objects in data classes 0 and 1, and  $NoFeats$  is the number of variables (features) of the data set.

The input values ( $Inp_j$ ;  $i = 1$  to  $NoFeats$ ) for every object  $i$  ( $i = 1$  to  $NoObjs$ ) are propagated through the net in the following way:

$$NetH_i = \sum_{j=1}^{NoFeats} W_{21_{i,j}} \cdot Inp_j; i = 1 \text{ to } NoHidds$$

$$OutH_i = f(NetH_i + Offset2_i); i = 1 \text{ to } NoHidds$$

$$NetO = \sum_{j=1}^{NoHidds} W_{32_j} \cdot OutH_j$$

$$OutO = f(NetO + Offset3);$$

where  $f(x)$  is so called "squashing" function. The program uses sigmoidal function of type:

$$f(x) = \frac{1}{1 + e^{-x}}$$

The other designations are:  $NoFeats$  - number of input neurons;  $NoHidds$  - number of hidden neurons;  $W_{21_{i,j}}$  - weight coefficients between the first (input) and the second (hidden) layer;  $W_{32_j}$  - weight coefficients between the hidden and the third (output) layer;  $Offset2_i$  - offsets in the neurons of the hidden layer;  $Offset3$  - offset of the only output neuron;  $NetH_i$  and  $OutH_i$  - "net" input and "real" output values of the hidden neurons;  $NetO$  and  $OutO$  - the same for the output neuron.



By training with a back-propagation-of-errors algorithm the following network coefficients are adjusted:  $W21_{i,j}$ ,  $W32_j$ ,  $Offset2_i$ , and  $Offset3$ . The algorithm can be easily explained by the following adapted segment of the original Pascal code:

```

Begin   { procedure Back_Propagation }
  { Sum - real variable for calculation of sums }
  { d3 - output error weighted with output values }
  { d2[i] - back-propagated-to-hidden-neurons errors
    weighted with hidden outputs' values }
  { dw21[j,i] - old changes of w21[i] coefficients }
  { dw32[i] - old changes of w32[i] coefficients }
  { dOffset2[i] - old changes of Offset2[i] coefficients }
  { dOffset3 - old change of Offset3 coefficient }
  { Eta - learning rate; Mu - momentum factor }

  { calculation of output error d3 and its weighting }
  d3 := (OutT - OutO)*OutO*(1 - OutO);

  { correction of w32 coefficients }
  for i := 1 to NoHidds do
  begin
    dw32[i] := Eta*d3*OutH[i] + Mu*dw32[i];
    w32[i] := w32[i] + dw32[i];
    Sum := d3*w32[i];
    d2[i] := Sum*OutH[i]*(1 - OutH[i]);
  end;

  { correction of Offset3 coefficient }
  dOffset3 := Eta*d3 + Mu*dOffset3;
  Offset3 := Offset3 + dOffset3;

  { correction of w21 coefficients }
  For i := 1 to NoFeats do
    For j := 1 to NoHidds do
    begin
      dw21[j,i] := Eta*d2[j]*LSet[NoSp]^ [i] +
        Mu*dw21[j,i];
      w21[j,i] := w21[j,i] + dw21[j,i];
    end;

  { correction of Offset2 coefficients }
  for i := 1 to NoHidds do
  begin
    dOffset2[i] := Eta*d2[i] + Mu*dOffset2[i];
    Offset2[i] := Offset2[i] + dOffset2[i];
  end;

End;   { procedure Back_Propagation }

```

The calculated (trained) neural network model contains not only the weight coefficients and offsets but also the means and variances of all variables in data classes 0 and 1, see the description of the classifier format.

### 1.3.4. KNN Classification

*The application of the KNN classifier includes the following steps:*

1. The IR spectrum of the compound under study is searched in a small library of

IR spectra (so called *learning set*). The classifier explicitly determines the applied method for searching, size of the processed hitlist,  $N_H$ , and the compounds in the learning set. The last is composed of  $L_0$  spectra of compounds that do not contain classifier's substructure, and  $L_1$  spectra of compounds that do contain it.

2. The obtained hitlist is composed by compounds whose spectra are most similar to the unknown spectrum in the context of applied search method (similarity measure). The algorithm determines how many of the hitlist compounds contain the classifier's substructure,  $N_+$ .

3. This value is compared with the arguments of the tabulated values of both conditional probabilities given as functions  $P_0 = f_0(N_+)$  and  $P_1 = f_1(N_+)$ . As a result of this comparison two conditional probabilities are retrieved:  $P_0$  and  $P_1$ .

4. The higher of two probabilities is called *classification precision* and is compared with the threshold value set by the user,  $P_t$ : the substructure is predicted as present (absent) if it holds that  $P_1 > P_t$  ( $P_0 > P_t$ ).

### 1.3.5. KNN Classifiers' Development

The procedure for generation of KNN classifier resembles to some extent the general scheme of ANN and LDA classifiers development that was described in a previous paper [5]. First, the user selects the method for Full-Curve Spectral Search and the number of hits in the hitlist,  $N_H$ . Second, the  $V_0 + V_1$  spectra of the validation set are searched in a small library composed only by the  $L_0 + L_1$  spectra of the learning set. The obtained  $V_0 + V_1$  hitlists (each is a sorted list of  $N_H$  entries) are processed to determine how many compounds ( $N_{+k}$ ,  $k = 1, 2, \dots, V_0 + V_1$ ) in each of them contain the classifier's substructure. These hitlists are divided into two categories: 0-hitlists obtained by searching the  $V_0$  spectra of compounds that do not contain classifier's substructure, and 1-hitlists obtained by searching the  $V_1$  spectra of compounds that contain classifier's substructure. As  $N_{+k}$  can be between 0 and  $N_H$ , the new variable (called *classifier's output*),  $X_+$  is calculated as follows:  $X_+ = N_+/N_H$ . From the array of  $N_{+k}$  values, and taking into account the hitlist categories, the following four functions are tabulated for the  $N_H + 1$  values of the output variable  $X_+$  ( $X_k = 0.00, 1/N_H, 2/N_H, \dots, 1.00$ ):

$N_{00}(X_k)$  - number of 0-hitlists that have  $X_+ = N_+/N_H$  less than or equal to  $X_k$

$N_{01}(X_k)$  - number of 0-hitlists that have  $X_+ = N_+/N_H$  higher than  $X_k$

$N_{10}(X_k)$  - number of 1-hitlists that have  $X_+ = N_+/N_H$  less than  $X_k$

$N_{11}(X_k)$  - number of 1-hitlists that have  $X_+ = N_+/N_H$  higher than or equal to  $X_k$

As a final result, the precision ( $P$ ) and recall ( $R$ ) for class 0 and class 1 are tabulated as functions of  $X_k$ :

$$P_0(X_k) = N_{00}(X_k) / ( N_{00}(X_k) + N_{10}(X_k) )$$

$$P_1(X_k) = N_{11}(X_+) / ( N_{11}(X_+) + N_{01}(X_k) )$$

and

$$R_0(X_k) = N_{00}(X_k) / ( V_0 + V_1 )$$

$$R_1(X_k) = N_{11}(X_k) / ( V_0 + V_1 )$$

## 2. Format of Files

### 2.1. Format of Library Files

The whole information for a spectral library is contained in eight files. The description of the files is given in Table 1.

**Table 1.** Description of the library files.

Name	description (contents)	default extension
library index file	contains the description of the library	LID
header file	chemical information for each compound: so called <i>header</i>	LHD
spectra file	spectra curves	LSP
peak table index file	indices to peak table	LIP
peak table file	peak tables	LPT
structures index file	indices to the atoms and bonds records	LSI
atoms file	atoms records	LAT
bonds file	bonds records	LBO

For a normal work with a spectral library all its eight files are necessary to be present. The minimum configuration consists of the first three files: from the spectra file the two peak table files can be derived by the menu command **LIBRARY | MAKE PEAK TABLE.**

### 2.1.1. Library Index File Format

Not described in the demo version!

### 2.1.2. Header File Format

Not described in the demo version!

### 2.1.3. Spectra File Format

Not described in the demo version!

### 2.1.4. Peak Table File Format

Not described in the demo version!

### 2.1.5. Structure File Format

Not described in the demo version!

## 2.2. Unknown Spectrum File Format

The unknown spectrum file consist of one record of the following type:

```
TSpecArr = array[1..801] of integer;  
TUnkSp = record  
  Ident: string[210];  
  A: TSpecArr;  
end;
```

The `Ident` field contains description of the spectrum. Then follows an array of two byte signed integer values representing the spectral curve.

## 2.3. JCAMP-DX 4.24 File Format

This file format is a standard for exchange of IR spectra between different operating systems. The full description can be found in [8].



“##IRSPECTRUM= ”. There are given, first one abscissa value, and then several ordinate values. The first ordinate value corresponds to the abscissa value, next ordinates are for abscissa shifted by  $1\text{ cm}^{-1}$ ,  $2\text{ cm}^{-1}$ , etc. In the presented excerpt of the file the shift is  $-1\text{ cm}^{-1}$  because the X-factor is equal to  $-1.0$ . Every new line begins with abscissa value, and there follow Y-values. The file ends with the keyword “##END= ”.

This is an excerpt from a Perkin-Elmer JCAMP-DX file.

```
##TITLE= 4,4'-Diamino-biphenyl in KBr, AL (NH2C6H4C6H4NH2) MW 184,24
##XYUNITS= WAVENUMBER, LOG_ABSORBANCE
##XYSTART= 4000, 0.12126
##XYFINAL= 450, 0.15647
##XYFACTORS= -1.00, 10000
##NPOINTS= 3551,
##IRSPECTRUM=
4000.0 1212 1212 1211 1210 1209 1209 1210 1209 1209 1209 1210 1211 1213 1214
3986.0 1215 1217 1218 1217 1216 1214 1213 1212 1213 1214 1216 1216 1215 1215
3972.0 1216 1216 1216 1214 1213 1213 1214 1214 1213 1213 1213 1215 1215 1215
. . . . .
472.0 1579 1567 1560 1557 1559 1563 1568 1572 1577 1582 1588 1594 1598 1602
458.0 1605 1607 1606 1599 1589 1579 1571 1567 1564
##END=
```

## 2.5. ASCII Spectrum File Format

This is a text file format used in IRIS program to import IR spectra measured in different instruments. This a very limited format with only three fields but the generation of files with this format is easy and can be done even manually from various file formats with copy-and-paste routines of a text editor. The end of the file is not marked with a keyword.

The identification string (name) of the IR spectrum is placed on the first line after the keyword “##TITLE= ”. The next keyword “##TEXT\_FILE= ” is crucial for IRIS to recognize the file format. The keyword “##IRDATA= ” initializes the spectral data field. Spectral data begin on a new line and they represent X and Y real values of the spectral curve, one X Y couple per line. X is given in  $\text{cm}^{-1}$ , Y in arbitrary absorbance units: there are no limits about the minimum and maximum of Y values because the spectrum is normalized always in 0.0 - 1.0 a.u. range. The X and Y values have to be separated with one or more white spaces. There are only three requirements for the X values: the smallest difference  $\Delta X$  to be at least  $4\text{ cm}^{-1}$ , the minimum value of X to be smaller than or equal to  $500\text{ cm}^{-1}$ , and

its maximum value larger than or equal to  $3200\text{ cm}^{-1}$ . The X data may not be equidistant and can be present in random order.

This is an excerpt from an ASCII Spectrum File.

```
##TITLE= Hydrolyseprodukt 8143
##TEXT_FILE=
##IRDATA=
393.5010.727353
395.4290.761473
397.3580.788762
399.2870.75938
. . . . .
4497.920.519521
4499.850.519889
4501.780.518945
```

## 2.6. Hitlist File Format

This is a text file. The first line contains the identity string of the searched spectrum. The next line contains the keyword "Search method: " and the method that had been used for searching. If the user changes even a letter in method description the program will not find the correct method when the hitlist file is loaded back in the program. The next line contains the keyword "Number of hits = " and the number of the hits in the file. Then follow the hits, one per a line. First number, in **I4** format, is the hit number, next one, also in **I4**, is the value of HQI, the string, in **A8** format, is the library name, and the last number, in **I6**, is the spectrum number in the library.

The hitlist file is generated by the program with the menu item **FILE | SAVE HIT LIST**, and it is read with **FILE | LOAD HIT LIST**.

## 2.7. Peak Table File Format

This is a text file. The first line contains the identity string of the peak table. The next line contains the keyword "Number of peaks = " and the number of the peaks in the file. Then follow the peaks, one per a line. First number, in **I5** format, is the wave number, in  $\text{cm}^{-1}$ , the next one, in **F6.2**, is the height of the peak, in a.u.

The peak table file is generated by the program with the button **SAVE** in the **PEAK SEARCH** window, and it is read with button **LOAD** in this window.

The following is a printing of a Peak Table file:

BUTYLAMINE	
Number of peaks = 11	
3368	0.13
3292	0.13
2960	0.92
2928	1.00
2872	0.56
1604	0.11
1464	0.19
1380	0.12
1084	0.09
968	0.09
836	0.32

## 2.8. Mixture Analysis Results File Format

Three types of text files can be generated in MIXTURE ANALYSIS window. The description of the files is given in Table 2.

**Table 2.** Files of mixture analysis results and statistics.

Name	description (contents)	default extension
regression file	contains the results from multi-linear regression	MGR
scan file	contains the results from multi-linear scanning regression calculations	MRS
statistics file	contains the statistics for multi-linear scanning regression calculations results	MST

The *regression file* contains the results from performing only one regression calculation with the unknown spectra and first  $H_S$  hitlist spectra. This file is produced by clicking on the **SAVE** button in **REGRESSION RESULTS** window – see file **MIXTURE.MRG** in **IRIS\FILES** directory. The file cannot be loaded into the program.

The identity string of the unknown spectrum is given on the first line of the file. It follows the number of hitlist spectra involved in regression calculation and the used statistical significance. Next line contains the captions of the results columns.

The results are given in three columns. The hitlist compound's name is given in the first one in **A60** format. Next is one blank space. The mean value and interval estimation are given in the next two columns, both in **F8.3** format. If the interval estimation includes 0.0 value then three stars, " \*\*\* ", are placed at the end of the



line. This means that the corresponding pseudo-concentration value is statistically insignificant.

The *scan file* contains the results from performing a row of regression calculations with increasing number of hitlist spectra. The file is produced by clicking on **SAVE** button in **RESULTS** group of the **MIXTURE SCAN RESULTS** window. The file can be loaded again into the program with **LOAD** button of the same group.

The identity string of the unknown spectrum is given on the first line of the file. On the next line there follows the maximum number of hitlist spectra involved in regression calculations,  $N_{sp}$ : as it was mentioned above, the regression calculations are performed with increasing number of hitlist spectra until reaching the preset value or a strong linear dependence between hitlist spectra, see matrix (6) in **Mixture Analysis** section. There follow  $N_{sp}$  groups of data for each of the hitlist compounds involved in the calculations. In each data group the following is given: on the first line – number of the hit, HQI, library name and spectrum number of the hitlist spectrum, chemical name, on the second line – again the chemical name. Then  $N_{sp}$  real numbers follow which are the corresponding pseudo-concentration values for the hit. They are taken from the corresponding column of the matrix in (6). As can be seen from the sample file **MIXTURE.MRS** in **IRIS\FILES** directory the last hit has only one non zero value.

The third file, *statistics file*, contains the statistics of the results from performing a row of regression calculations with increasing number of the hitlist spectra. The file is produced by clicking on the **SAVE** button in **STATISTICS** group of the **MIXTURE SCAN RESULTS** window – see file **MIXTURE.MST** in **IRIS\FILES** directory. The file cannot be loaded again into the program.

The identity string of the unknown spectrum is given on the first line of the file. On the next line there follows the maximum number of hitlist spectra involved in regression calculations,  $N_{sp}$ . Next line contains the captions of the statistics columns: they are four altogether. The following information about the hitlist spectrum is given in the first column: hit number, HQI, library name and spectrum number of the hitlist spectrum, and part of the chemical name. All this is restricted to 60 characters altogether. The next three columns contain the mean, standard deviation and r.s.d. of the pseudo-concentration values which are on

and under the main diagonal of the matrix (6). First two are in **F8.3** format, the last one is in F8.1 format. If the mean of the pseudo-concentration is a negative value (i.e. meaningless) three stars " \*\*\* " are given after it, else there are five blank spaces.

As can be seen from the sample file **MIX\_AN.MST** the last hit has only mean value and no standard deviation or r.s.d. value. This is because only one value of the pseudo-concentration is calculated for it.

## **2.9. IR Spectra Classifiers File Formats**

The classifiers file is a text file that contain one or more classifiers from the same or different type. Every classifier begins with "## CLASSNAME = " and ends with "## CLASSEND". Each classifier's item (field) is preceded with a key word. Each key word (except "## CLASSEND" ) begins with two pound signs "##" and ends with equal sign and blank space "= ". First line contains classifier's name, up to 11 symbols. This item and the next five ones begin on the line, where the corresponding keyword is, at position 16.

There follow the classifier comments field, 60 symbols, and on a new line - classifiers type, 6 symbols. Classifier initials follow on the next line; they are up to 30 symbols, and the information for the learning and validation sets (up to 60 symbols) is on the next line. In the initials field there are two letters for the initials of the person who had created the classifier, the date and time of classifier creation. In the "## CLASSFILE = " field are given the file names of the LIS-files from which the learning and validation sets are derived, and the numbers of spectra in class 0 and class 1 of the learning and validation sets.

The next lines are different for the different classifier types, and they are described in the next three paragraphs.

### **2.9.1. Neural Network Classifier Format**

Next two lines contain the number of input neurons (number of features) and the number of hidden neurons - nonnegative integers in format **I8**. The corresponding keywords are "## NOINPNEUR = " and "## NOHIDNEUR = ".

The following two lines contain the learning rate and momentum factor - real values in format **F12.6**. Their keywords are “## LEARNRATE = ” and “## MOMENFACT = ”. These numbers are only for information how the neural network is trained and do not influence the program flow anyhow.

The next two lines contain the keywords “## TRANSFACT = “ and “## MEANVALUE = “. Both have no items on the line where they were written. On the next line begin the mean values for every variable - five per a line in format F12.6. After them follows the keyword “## VARIANCES = “. On a new line begin variance values for every variable - five per a line in format **F12.6**.

The next line contains the keyword “## FEATURES = “. The features description begins on new line, one feature is given per a line. First three letters determine the feature type. There follows a number that shows the lower feature interval's limit, given in  $\text{cm}^{-1}$ . The second and fourth number are still not used by the program and they are given as zeroes. The third number is the feature interval's width, given in  $\text{cm}^{-1}$ . All four numbers are in format **I8**.

The next line contains the keyword “## NETCOEFFS = “. It tells that the network coefficients block begins. Next is the keyword “## COEFFSW21 = “ which marks the beginning of  $W_{21}$  coefficients. On a new line begins the layout of  $W_{21}$  coefficients. First, there are written the coefficients of the first input neuron, followed by those of the second input neuron and so on. The coefficients for every input neuron begin on a new line and there are 5 coefficients per a line in format **F12.6**. Altogether there are “number of input neurons” times “number of hidden neurons” coefficients.

Then follows the keyword “## OFFSETS2 = “ and after it on a new line begin the offset values for the hidden neurons. They are laid five per a line in format **F12.6**.

The next line contains only the keyword “## COEFFSW32 = “. After it on a new line begins the layout of  $W_{32}$  coefficients, five per a line in format F12.6. The following two lines contain the keyword “## OFFSETS3 = “, and the value of the offset of the output neuron. The degree of training of the network is given after the keyword “## FQ\_VALUE T= ”. This is the root mean squared deviation of

the outputs for objects in the learning set from the target values. This value is given only for information.

After the coefficients block follows the keyword '## CLASSPNTS = '. On the next line is given the number of output values for which there were calculated the precisions and recalls. This number is written in format **I8**.

The keyword "## CLASSLIM0 = " is next. Then, on a new line begin the lines containing the output value, and the precision and recall for this output value for 'zero' class. The numbers are in format **F12.6**, **F8.2** and **F8.2**, respectively. These triples are on one line each, there are as many lines altogether as the number specified after the keyword "## CLASSPNTS = ". The same follows for the 'one' class. The keyword is '## CLASSLIM1 = '.

The file ends with the keyword "## CLASSEND " written on a new line.

### **2.9.2. Linear Discriminant Function Classifier Format**

The classifier file structure is nearly the same as that of a neural network file with some changes: instead of neural network coefficients there are given the coefficients of linear discriminant function. The LDA coefficients are given on a new line after the keyword "## LDACOEFFS = ". There are 5 coefficients per a line in format **F12.6**. The keywords from ANN classifier that are not present in LDA classifier are:

```
"## LEARNRATE = ", "## MOMENFACT = ", "## NOHIDNEUR = ",  
"## NETCOEFFS = ", "## COEFFSW21 = ", "## OFFSETS2 = ",  
"## COEFFSW32 = ", "## OFFSETS3 = ", "## FQ_VALUE T= ".
```

### **2.9.3. KNN Classifier Format**

The KNN classifier is quite different from both previously described classifiers that is why it will be described thoroughly. Every classifier's item (field) is preceded with a key word. Each key word (except "## CLASSEND") begins with two pound signs "##" and ends with equal sign and blank space "=". The classifier begins with "## CLASSNAME = " and ends with "## CLASSEND". First line contains classifier's name, up to 11 symbols. This item and the next five begin on the line, where the keyword is, at position 16.

There follow the classifier comments field, 60 symbols, and on a new line - classifier type, 6 symbols. Classifier initials follow on the next line; they are up to 30 symbols, and the information for the learning and validation sets (up to 60 symbols) is on the next line. In the initials field there are two letters for the initials of the person who had created the classifier, the date and time of classifier creation. In the “## CLASSFILE = ” field are given the numbers of spectra in class 0 and class 1 of the learning (L0/1) and validation (V0/1) sets.

The library name, 8 symbols, is on the next line. The correctness of this name is crucial for classifier performance, as well as the correctness of the next items – the number of library spectra (the size of spectral library), and the number of hits (the size of hitlist). Both numbers are given in **I8** format.

With the keyword “## CLASSLIM0 = ” begin (on the next line) ( $N_H + 1$ ) triples of the output variable, and the precision and recall as function of the former. This is for class 0. The same for class 1 is given after the keyword “## CLASSLIM1 = ”. The program uses only the first two values: the recall’s values are given as an additional information to the user about classifier’s performance.

**Table 3.** An excerpt from a KNN classifier file.

```
## CLASSNAME = p-benzene
## COMMENTS = para substituted benzene; KNN_CC 250/250/50
## CLASSTYPE = KNN_CC
## CLASSINIT = PL: 12/12/2001 4:01:30 PM
## CLASSFILE = L0/1 = 250; V0/1 = 250
## LIBNAME = IR13484
## LIBSIZE = 13484
## NUMHITS = 50
## CLASSLIM0 =
 0.000 100.00 0.00
 0.020 100.00 0.00
 0.040 100.00 0.00
 .....
 1.000 50.20 100.00
## CLASSLIM1 =
 0.000 50.00 100.00
 .....
 0.980 100.00 1.60
 1.000 100.00 0.80
## CLASSAFF =
  1 9999994999999999999594999999999999999299999999
 51 999999999999999999929999999994999599999999994999
 .....
 13401 9999999999999399949993493944999499999999999999999995
 13451 9995999999999999999199999999999999
## CLASSEND
```

The last classifier field is that of class affiliations. They begin after the keyword “## CLASSAFF = ”, and have the following representation. The first number

on the line (**I6** format) is not used by the program. It is given for the user, and shows the library spectrum's number for which the first class affiliation is (numbers 0 to 5 and 9). There follow four empty spaces, and 50 class affiliations for 50 library spectra. As can be seen from Table 3, the class affiliations for only 50 spectra (compounds) are given per line. The class affiliations for the next 50 spectra are given in the next line and so on. The numeric values of class affiliations mean the following:

0 - the corresponding spectrum (i.e. compound) is from class 0 and is included neither in the learning nor in the validation set;

1 - the corresponding spectrum (i.e. compound) is from class 1 and is included neither in the learning nor in the validation set;

2 - the corresponding spectrum (i.e. compound) is from class 0 and is from the learning set;

4 - the corresponding spectrum (i.e. compound) is from class 0 and is from the validation set;

3 - the corresponding spectrum (i.e. compound) is from class 1 and is from the learning set;

5 - the corresponding spectrum (i.e. compound) is from class 1 and is from the validation set;

9 - there is no information about the class affiliation of the spectrum.

The class affiliations designated with numbers 0, 1, 4, 5 and 9 do not determine how the classifier is operating: only the positions of numbers 2 and 3 are important. They determine the library numbers of spectra from the learning set: by the application of the classifier the unknown spectrum is searched in a library composed only from spectra from this learning set. Presence of 9 instead of 0 or 1 depends on the way the classifier is created (see the software section).

### 3. Sample Files

**3.1. Perkin-Elmer JCAMP-DX Files of mixtures.** All these spectra were recorded in Dept. Anal. Chem. At University of Plovdiv by the coworkers of Dr. Plamen Penchev. In brackets are given the v/v ratios.

◇ A1000011.JCM - p-xylene + m-xylene (1:1)

◇ A1000101.JCM - p-xylene + o-xylene (1:1)

- ◇ A1000110.JCM - m-xylene + o-xylene (1:1)
- ◇ A1010001.JCM - p-xylene + i-propylbenzene (1:1)
- ◇ A1010010.JCM - m-xylene + i-propylbenzene (1:1)
- ◇ A1010100.JCM - o-xylene + i-propylbenzene (1:1)
- ◇ FNB1IB1.JCM - n-butanol + i-butanol (1:1)
- ◇ FNB1IB4.JCM - n-butanol + i-butanol (1:4)
- ◇ FNB1IB9.JCM - n-butanol + i-butanol (1:9)
- ◇ FNB4IB1.JCM - n-butanol + i-butanol (4:1)
- ◇ FNB9IB1.JCM - n-butanol + i-butanol (9:1)
- ◇ 1V2R.JCM - benzylacetone
- ◇ 2V1R.JCM - ethyl-benzyl ketone
- ◇ AIPBEN.JCM - i-propylbenzene

### 3.2. Classifiers Files. These are simple text files.

- ◇ IR\_AL\_40.ICL - classifiers that use ANN and LDA methods [5].
- ◇ ir\_ann20.icl - classifiers that use ANN [5]: they are subset of the upper.
- ◇ ir\_knn20.icl classifiers that use KNN method: they can not be used with this version of the program because the Chemical Concept library of 13484 spectra [12] is not free, and we do not have rights to distribute it with this software.

## REFERENCES

1. H. Scsibrany, K. Varmuza; ToSiM. PC-Software for the Investigation of Topological Similarities, pp. 235-249 in: C. Jochum (Ed.); *Software Development in Chemistry*, Gesellschaft Deutscher Chemiker, Frankfurt am Main, 1994, Vol. 8.
2. J.T. Clerc; Automated Spectra Interpretation and Library Search Systems, pp. 145-162 in: H.L.C. Meuzelaar, T.L. Isenhour (Eds.); *Computer-Enhanced Analytical Spectroscopy*. Plenum Press, New York, (1987).

3. P.N. Penchev, A.N. Sohou, G.N. Andreev. Description and Performance Analysis of an Infrared Library Search System. *Spectroscopy Letters*, (1996), **29**, 1513-1522.
4. P.N. Penchev. *Application of Chemometric Methods for Identification of Organic Compounds from Their Infrared Spectra*. Ph. D. Thesis, (1998), Plovdiv, Bulgaria.
5. P.N. Penchev, G.N. Andreev and K. Varmuza. Automatic classification of infrared spectra using a set of improved expert-based features. *Anal. Chim. Acta*, (1999), **388**, 145 - 159.
6. K. Varmuza, W. Werther; Mass Spectral Classifiers for Supporting Systematic Structure Elucidation. *J. Chem. Inf. Comput. Sci.* (1996), **36**, 323-333.
7. J. Zupan, J. Gasteiger; *Neural Networks for Chemist: An Introduction*. Weinheim, Germany, VCH Publishers, (1993).
8. R.S. McDonald, P.A. Wilks Jr.; JCAMP-DX: A Standart Form for Exchange of Infrared Spectra in Computer Readable Form. *Appl. Spectrosc.*, (1988), **42**, 151-158.
9. *CDS-3 Applications Software for FT-IR Spectrophotometers*. Perkin-Elmer, Norwalk, Connecticut, USA, (1986)
10. The spectroscopic database SpecInfo is available from Chemical Concepts, PO Box 100202, D-69442 Weinheim, Germany.