MATLAB software for automated calculation of concentration of different compounds using extracted peak area from HPLC data file in pdf format

Wenfa Ng

Unaffiliated researcher, Singapore, Email: ngwenfa771@hotmail.com

Abstract

Chromatograms represent a class of data difficult to process expeditiously due to the large number of intermediary steps necessary to translate peak detection to a concentration reading of a specific compound. This problem is further exacerbated by the different output file format in which instrument manufacturers present chromatographic data. Steps necessary to convert a detected peak to a concentration reading include identification of compound using retention time, extraction of corresponding peak area, and calculation of concentration of compound by using a calibration curve. This work sought to develop a MATLAB software able to automatically extract peak area from chromatographic readout captured in pdf format and calculate the corresponding concentration values. Given manufacturer-specific formatting features in pdf file, the MATLAB software could only read and handle pdf files of HPLC readouts from Shimadzu's LabSolutions software. In processing the pdf file of each analyzed sample, entire content of the file was first read as a character string. Subsequently, specific delimiters were used to extract retention time and detected peak area for each compound. This information was subsequently processed to identify specific target compound of interest, where extracted peak area was used to calculate concentration of compound using a calibration plot. Overall, the program generates a database comprising filename, raw retention time and peak area data, as well as concentration values of each target compound in an easy to read format. Finally, to provide ease of access and a permanent file for storage, the program output the above database as an Excel file stored on the hard drive. One important advantage of this software is that it could process multiple pdf files simultaneously and there is no upper limit to the number of pdf files (or samples) that could be processed. Collectively, the MATLAB software capable of automatically extracting peak area and calculating concentration of different compounds would provide significant savings of time in handling large number of pdf files in a typical chromatographic run from a Shimadzu HPLC instrument.

Keywords: high performance liquid chromatography, peak area, concentration, calibration curve, data processing, MATLAB, Shimadzu, automated processing, pdf file, retention time,

Subject areas: biochemistry, biotechnology,

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27878v2 | CC BY 4.0 Open Access | rec: 3 Sep 2019, publ: 3 Sep 2019

Highlights

1) A MATLAB program able to handle pdf file of chromatogram output from Shimadzu's LabSolutions software was developed.

2) Able to automatically extract retention time and peak area information from the data, the program could identify specific compound and obtain the corresponding peak area.

3) Peak area subsequently forms the basis for the calculation of compound concentration using calibration information input by the user.

4) Extracted and calculated information was stored in a database that could be output to an Excel file.

5) Multiple pdf files could be processed simultaneously and there is no upper limit on the number of pdf files (or samples) that could be processed.

6) Overall, a MATLAB program was developed for automatic calculation of compound concentration from peak area extracted from chromatographic information.

Background

High performance liquid chromatography is a common tool in life sciences and biotechnology research for the analysis of fermentation samples where a variety of organic compounds could be sequentially detected by either refractive index detector or ultraviolet detector. Analytical readout appears in the form of a chromatogram defined by different peaks at different retention time. Area of each peak is directly proportional to concentration. Hence, a chromatogram helps illuminate the types of compounds present and their concentrations.

While presenting data as a neat package, chromatogram represents a type of data difficult to process. Why? Because multiple intermediary steps are needed to translate a peak area of a detected peak into concentration readings. Such translation involves the extraction of peak area associated with particular retention time (which identifies a compound), and processing it using a calibration curve of the compound to yield a concentration reading that could be used in downstream analysis. These steps could be completed with a modern spreadsheet, but nevertheless represents a significant investment of time for large number of samples.

Coupled with the difficulty of reading data from chromatogram is the differing types of output file format in which different instrument manufacturer package chromatographic data. Agilent, for example, outputs chromatographic data as Excel file in a defined format. On the other hand, Shimadzu uses pdf file format to output chromatographic data. Each file format presents particular challenges to the user in the extraction of peak area data from automatically

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27878v2 | CC BY 4.0 Open Access | rec: 3 Sep 2019, publ: 3 Sep 2019

characterized peaks. Irrespective of the output file format, significant amount of manual effort would be needed to process chromatographic data without the aid of automated software.

The solution is the development of a MATLAB software that automatically read and extract peak area associated with respective retention times belonging to particular compounds captured in a chromatogram. More importantly, the software automatically processes the peak area into a concentration reading by using a calibration coefficient available from the calibration curve of each target compound. The end result is a tabulation of filename, raw data and concentration values of each of the target compounds, each catalogued in a table that could be output as an Excel file for ease of access, storage and retrieval. Currently, the software is designed to process pdf files from Shimadzu's LabSolutions software.

Implementation

Specifically designed for processing the pdf file from Shimadzu's LabSolutions HPLC software, the MATLAB program first read the contents of the pdf file as a character array.[1] Using specific delimiters to extract retention time and peak area of different compounds, the software builds a structured array comprising retention time and peak area data and associate it with each filename in a larger database.

By comparing the retention time of specific compounds input by the user into the software, the MATLAB code would be able to automatically identify specific compound and extract the corresponding peak area. Given that retention time of the same compound may differ in different samples in a phenomenon known as retention time shift, detected retention time within a certain range from the retention time specified by the user would be used to identify a compound. The default error tolerance in retention time is 0.5 mins.

After the peak area of a compound is obtained, the next step is the division of peak area with a calibration coefficient of the compound to yield the final concentration reading. Calibration coefficients of different compounds are also input by the user into the software and comes from a linear regression of different points on a calibration plot. The above process is repeated for all target compounds and for all sample files, and the results are output into a table that could subsequently be output to an Excel file to help ease access, storage and retrieval.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27878v2 | CC BY 4.0 Open Access | rec: 3 Sep 2019, publ: 3 Sep 2019

datadb2

1x23 struct with 7 fields

Fields	Filename	Data	Glucose	Acetate	Ethanol	Butanone	Butanol
1	'0.5 gL Butanone std 08 ...	1x2 struct	[]	[]	[]	[]	[]
2	'0.5 gL glu ace eth 07 Ma...	1x6 struct	0.4789	0.4723	0.4489	[]	[]
3	'1 gL butanone std 08 Ma...	1x4 struct	[]	[]	[]	0.0682	0.0672
4	'1 gL glu acetate ethanol ...	1x6 struct	1.0081	1.0029	0.9877	[]	[]
5	'10 gL butanone std 08 M...	1x3 struct	[]	[]	[]	9.6543	9.7883
6	'10 gL glu acetate eth 07 ...	1x5 struct	10.0002	10.0008	10.0041	[]	[]
7	'20 gL butanone std 08 M...	1x3 struct	[]	[]	[]	20.1725	20.1057
8	'5 gL butanone std 08 Ma...	1x3 struct	[]	[]	[]	0.0525	[]
9	'5 gL glu acetate eth 07 ...	1x6 struct	5.7638	5.7451	1.6387	[]	[]
10	'Day 1 2 gL eth M9 R1 08 ...	1x8 struct	[]	1.3118	0.4938	[]	0.4902
11	'Day 1 2 gL eth M9 R2 08 ...	1x8 struct	[]	1.5821	0.5076	[]	0.8466
12	'Day 1 5 gL eth LB R2 08 ...	1x7 struct	[]	2.4509	2.4092	[]	1.1460
13	'Day 1 5 gL eth LB R3 08 ...	1x8 struct	0.0384	2.4766	2.3289	[]	1.0858
14	'Day 1 5 gL eth M9 R2 08 ...	1x9 struct	0.0344	2.0771	2.9107	[]	1.4132
15	'Time 0 phosphate R1 10 ...	1x4 struct	[]	[]	[]	7.5026	[]

Figure 1: Sample output from the MATLAB program illustrating concentrations (g/L) of different compounds in different HPLC sample denoted by different filename.

Key features

1) The software is capable of automatically identifying pertinent peaks of particular compounds based on each compound's retention time and retrieves the corresponding peak area from tabulated data.
2) Using peak area from specific compounds and calibration coefficient from a calibration curve, the software next calculates the compound's concentration.
3) Concentrations of different compounds are calculated for each sample, and the information is captured in a structured array, that could be written to an Excel file for ease of access and storage.
4) The software is able to simultaneously process multiple pdf files from different samples of an analysis set. There is no upper limit to the number of pdf files (or samples) that could be processed by the software.

Conclusions

Requiring multiple steps to identify relevant peaks, extract peak area of identified peaks and calculation of concentration, chromatograms represent a class of data difficult to process manually. To aid in expeditious processing of chromatographic data, a MATLAB program was developed to aid in automatic peak identification of compounds, extraction of peak area and calculation of compound concentration. First, information from the pdf file of each HPLC sample is read into a character array, where subsequently delimiters help identify relevant stretches of

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27878v2 | CC BY 4.0 Open Access | rec: 3 Sep 2019, publ: 3 Sep 2019

characters that contain retention time and peak area information. Next, pertinent peaks of specific compound are identified based on user specified retention time information, which allows the extraction of corresponding peak area. With peak area in hand and user specified calibration coefficient, compound concentration could be calculated. The process is repeated for each compound in the target list for all samples. Information is captured in a structured array which is finally output into an Excel file for ease of access and storage. Overall, the MATLAB software described herein should find use in enabling expedited calculation of concentrations of different compounds in HPLC sample data output via Shimadzu's LabSolutions program.

Source code

```
function process_HPLC_file

    error_rt = 0.5;


    fileinfo = dir('*.pdf');
    k1 = length(fileinfo);
    javaaddpath('iText-4.2.0-com.itextpdf.jar');
    for i=1:k1
        filename = fileinfo(i).name;
        cell = pdfRead(filename);
        rawdata = cell{1};
        data = process_rawdata(rawdata);
        datadb(i).Filename = filename;
        datadb(i).Data = data;

    end
       datadb = process_datadb(datadb, error_rt);
       datadb2 = calculate_conc(datadb);
       write_data_conc(datadb2);

end


function data = process_rawdata(rawdata)

        start1 = strfind(rawdata, 'Ret. Time');
        stop1 = strfind(rawdata, 'Area');
        k1 = length('Ret. Time');
        info1 = rawdata(start1+k1+1:stop1-2);
        info_cell = strsplit(info1, char(10));


        k2 = length(info_cell);
        for i = 1:k2
            content = info_cell{i};
            number = str2num(content);
            data(i).Ret_time = number;
        end
```


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27878v2 | CC BY 4.0 Open Access | rec: 3 Sep 2019, publ: 3 Sep 2019

```
        start2 = strfind(rawdata, 'Area');
        stop2 = strfind(rawdata, 'Height');
        k3 = length('Area');
        info2 = rawdata(start2+k3+1:stop2-2);
        info_cell2 = strsplit(info2, char(10));


        k4 = length(info_cell2);
        for i = 1:k4
            content = info_cell2{i};
            number = str2num(content);
            data(i).area = number;
        end

end


function datadb = process_datadb(datadb, error_rt)

    rt_glucose = 7.4;
    rt_acetate = 12.5;
    rt_ethanol = 18.4;
    rt_butanone = 23.3;
    rt_butanol = 26.3;


    k1 = length(datadb);

    for i = 1:k1
        datadb(i).Glucose = 0;
        datadb(i).Acetic_acid = 0;
        datadb(i).Ethanol = 0;
        datadb(i).Butanone = 0;
        datadb(i).Butanol = 0;
    end

    for i =1:k1
        data = datadb(i).Data;
        k3 = length(data);
        for j = 1:k3
            ret_time = data(j).Ret_time;


                index1 = ret_time-rt_glucose;
                index1 = abs(index1);

                if index1 < error_rt
                    datadb(i).Glucose = data(j).area;
                end

                index2 = ret_time-rt_acetate;
                index2 = abs(index2);
```


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27878v2 | CC BY 4.0 Open Access | rec: 3 Sep 2019, publ: 3 Sep 2019

```
                if index2 < error_rt
                    datadb(i).Acetic_acid = data(j).area;
                end

                index3 = ret_time-rt_ethanol;
                index3 = abs(index3);

                if index3 < error_rt
                    datadb(i).Ethanol = data(j).area;
                end

                index4 =(ret_time - rt_butanone);
                index4 = abs(index4);

                if index4 < error_rt
                     datadb(i).Butanone = data(j).area;
                end

                index5 =(ret_time - rt_butanol);
                index5 = abs(index5);

                if index5 < error_rt
                     datadb(i).Butanol = data(j).area;
                end

        end
    end

end


function datadb2 = calculate_conc(datadb)

    coefficient_glucose = 56177;
    coefficient_acetate = 18006;
    coefficient_ethanol = 16390;
    coefficient_butanone = 32731;
    coefficient_butanol = 38302;

    k1=length(datadb);

    for i=1:k1
        datadb2(i).Filename = datadb(i).Filename;
        datadb2(i).Data = datadb(i).Data;

        glucose_peak_area = datadb(i).Glucose;
        glucose_conc = glucose_peak_area/coefficient_glucose;
        datadb2(i).Glucose = glucose_conc;

        acetate_peak_area = datadb(i).Acetic_acid;
        acetate_conc = acetate_peak_area/coefficient_acetate;
        datadb2(i).Acetate = acetate_conc;

        ethanol_peak_area = datadb(i).Ethanol;
```


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27878v2 | CC BY 4.0 Open Access | rec: 3 Sep 2019, publ: 3 Sep 2019

```
        ethanol_conc = ethanol_peak_area/coefficient_ethanol;
        datadb2(i).Ethanol = ethanol_conc;

        butanone_peak_area = datadb(i).Butanone;
        butanone_conc = butanone_peak_area/coefficient_butanone;
        datadb2(i).Butanone = butanone_conc;

        butanol_peak_area = datadb(i).Butanol;
        butanol_conc = butanol_peak_area/coefficient_butanol;
        datadb2(i).Butanol = butanol_conc;
    end
end


function write_data_conc(datadb2)

    k1 =length(datadb2);

    A{1,1} = 'Filename';
    A{1,2} = 'Glucose conc.';
    A{1,3} = 'Acetate conc.';
    A{1,4} = 'Ethanol conc.';
    A{1,5} = 'Butanone conc.';
    A{1,6} = '2-Butanol conc.';

    for i=1:k1
        A{i+1,1} = datadb2(i).Filename;
        A{i+1,2} = datadb2(i).Glucose;
        A{i+1,3} = datadb2(i).Acetate;
        A{i+1,4} = datadb2(i).Ethanol;
        A{i+1,5} = datadb2(i).Butanone;
        A{i+1,6} = datadb2(i).Butanol;
    end

    xlswrite('Concentrations of compounds.xlsx', A);

end
```

Reference

1. Read text from a PDF document - File Exchange - MATLAB Central. Available at: https://www.mathworks.com/matlabcentral/fileexchange/63615-read-text-from-a-pdf-document. (Accessed: 18th July 2019)

New in this version

A software bug was corrected and the efficiency of the software was improved by removing one write data function in the MATLAB software.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27878v2 | CC BY 4.0 Open Access | rec: 3 Sep 2019, publ: 3 Sep 2019

Availability of software

MATLAB program files can be found at the following link:

https://figshare.com/articles/MATLAB_software_for_automatic_reading_of_Shimadzu_HPLC_pdf_files_and_calculating_concentration_of_different_compounds/9121760

Conflicts of interest

The author declares no conflicts of interest.

Funding

No funding was used in this work.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27878v2 | CC BY 4.0 Open Access | rec: 3 Sep 2019, publ: 3 Sep 2019