

# Requirements 2D-GC visualization

Baksteen, Sarah  
S3145034

Brink, Wouter  
S3862348

Gros, Oane  
S2972778

Jager, Maarten de  
S2957906

Jong, Michiel de  
S2550768

Strik, Oliver  
S3100693



**university of  
groningen**

## **Client**

Rohrbach, Léon  
Figueiredo, Monique

## **TA**

Argyrousis, George

Software Engineering  
University of Groningen  
April 1, 2019

Version 4.0

# 1 Introduction

This document elaborates the requirements for the project 2D-GC visualization. This project is commissioned by the Chemical Engineering department of the University of Groningen and is part of the course *Software Engineering*.

## Problem

2-Dimensional Gas Chromatography (2D-GC) is a chemical technique to accurately identify and quantify small traces of compounds in samples. This analytical technique is used in Groningen to aid in the search for new biofuels. For this research, accurate and interactive software for 3D-visualization and analysis is necessary.

The software which is currently being used by the Chemical Engineering department is limited in functionalities. This means that there is information which cannot be accessed yet with the current technology.

## Solution

We aim to build a desktop app to visualise 2D-GC data in one, two and three dimensional spaces, and try to work in comparative analysis, where the same analysis can be run smoothly on multiple samples.

# 2 Stakeholders

This section gives an overview of all stakeholders involved in this project.

There are two stakeholders in this project: Chemical analysts of the RUG Chemical Engineering department and the developers.

The chemical analysts are the clients of the project. These are the users of the program. The program is going to be a tool for their research. The chemical analysts determine the requirements of the project.

The second type of stakeholder are the developers. These are students following the course *Software Engineering*. The developers are responsible for writing the software and making sure that the desired functionalities are being implemented.

# 3 User Stories

- As a chemical analyst, I want to quickly view the outcome of a diagnostic test by visually assaying the 3D graph. I should be able to quickly load .txt files and intuitively control 3D and 2D plots of this data.
- As a chemical analyst, I want to find the quantity of a specific compound in multiple samples, and compare these quantities. These quantities should be found by integrating the graph, and shown inside the GUI.
- As a chemical analyst publishing research, I want to have a way to show the data in a visually appealing manner, so that a saved image would be appropriate for publication.
- As a chemical analyst, I want to be able to find the quantity of a larger chemical group in my sample by freeform selection of groups of peaks.
- As a developer, I want to be able to extend the program to work with 2-Dimensional Gas Chromatography Mass Spectrometry (GCXGC-MS), where an extra identification dimension is added on top of the 2D-GC data.

## 4 Functional requirements

### Critical Functional Requirements

- Showing 2D-GC output .txt files as interactive 3-dimensional graphs will be done by having a surface plot with mouse controls. The user should be able to zoom by scrolling, change tilt, yaw and pitch by dragging and move the eye by holding the middle mouse button and dragging.
- 2-dimensional graphs will be shown as heatmaps with colors defining the value of each point in the graph. The view should be controlled by scrolling to zoom, and dragging to move the eye. Showing the graphs side by side will be done by implementing a system of draggable tabs, where both graphs can be visible simultaneously, but one can also be collapsed.
- Finding the quantity of compounds will be done by drawing Regions of Interest in the 2D graph, over which will be integrated. Drawing the region of interest should be under a specific button or hotkey, after which a square region should be drawn at a preset location. After initial drawing, the region should stay customizable, by having the anchor points of the region draggable over the graph. The integrated value of all points of the graph that are within the region will be shown in a separate list under the graphs.

### Important Functional Requirements

- Be able to save the current state of the program, with the loaded data and any attached user inputs such as integration or labels in a specific file format from which the graph and any user inputs would be able to be reloaded by the program in a different session.
- Be able to select a freeform space to integrate over as a polyline region with multiple anchor points. The position of the anchor points at initial drawing should be specified by clicking in the graph in a 'drawing' mode. The integrations should be shown in an 'integration list' under the graph with a renamable label, absolute integrations and mean sums. The integration areas should be destructible via a button which is shown within the integration list.
- Changing the colors of the visualizations between multiple preset palettes. Switching between these should be done via a separate dialog window where all options are displayed.
- Convolution of the graph should be possible. Switching between raw data and convoluted data should be under a toggle button or action, and the convolution kernel should be able to switch between presets in a separate dialog. Optionally, these presets would have variables which could also be changed in this dialog.
- Showing the 1-dimensional plot of the first gas chromatography run through. This would be shown as a line plot of the integration over the x axis of the supplied 2D data. Zooming and tracking the eye would be handled by drawing a rectangle where the new plotting area would be, with a button to go back to the previous view.

### Useful Functional Requirements

- Having presets and editable user settings for the palettes, convolution kernels and drawing of regions of interest. These settings would be taken from text json files, so that users could write their own if the need later arises.

- Visualizing the integration areas in 3D, by highlighting the integration area in a different palette. Which integrations are shown should be toggled per integration within the integration list.
- Showing the x and y location of the mouse in scene coordinates in both the 2D and the 3D view would be very relevant. This would be shown in a status bar under the screen, where the x and y is also translated to the amount of minutes and seconds (the units of the x and y axes).
- Showing multiple chromatograms simultaneously, by having a 'compare with...' button, which would overlay the new chromatogram in the same plots as a previously opened one so the position and height of the peaks can be easily compared.
- Be able to export the 2D graph, 3D graph and integrations to publishable formats. This would be .png (or related) image files for the graphs, where any integration regions and such visible user inputs in the graph would be kept, and copied text to clipboard for the list of integrations.
- Be able to save peak data in a database via a button in the integration list, with any attached information, such as which sample it was from, integration value, and label. The user should be able to retrieve previously saved information in a look-up table while looking at a new sample.
- Compound recognition/matching with the internal database, where older labelled data (which was identified with mass spectrometry) would automatically be recognized if it's already in the internal database, and new data shows a peak at the same position. A relevant match would be shown in the integration list in a new column.

## Won't Do

- Linking to an outside Mass Spectrometry library. This is something which would be useful, as the client may switch from an Flame Ionization Detector (FID) detector to an MS detector in the near future. However, this would add a layer of complexity, as MS data adds an extra dimension to the data. We think it best to focus on a stable program for FID visualization, as this can also be used for GCxGC-MS data.

## 5 Non-functional requirements

- **Extensibility:** this is essential, as the client is preparing to switch to a different detection system, which alters the dimensionality of the data. We have to make sure that this extension would be easily incorporated in the program. This can be done by making sure the program architecture is clear and there is already a way to retrieve the mouse-scene coordinates inside the program.
- **Technology:** the program needs to be able to run on older versions of operating systems, as the measurement software itself is run on windows XP.
- **Maintainability:** the software has to be designed in such a way that the program is easily maintainable. After the course *Software Engineering* has ended, the developers will not provide further maintenance to the software.
- **Scalability:** The amount of information in the database will most likely grow with time. Therefore, the database has to be scalable to a certain extent.

## 6 Meeting Log

Date	Meeting log
25/02/2019	Talked about and discussed the requirements document.

## 7 Change Log

Date	Author	Change log
15/02/2019	Oane Gros	Initial version
25/02/2019	All	Introduction and MoSCoW
04/03/2019	Wouter Brink, Oliver Strik	Added meeting and change logs
04/03/2019	Wouter Brink	Added title page
08/03/2019	Oane Gros	Added basic user stories and non-functional requirements
10/03/2019	Oane Gros	Extended requirements to include more detailed description, and some alterations based on the first physical meeting with the client
31/03/2019	Wouter Brink	Added Terminology and Stakeholder sections.
31/03/2019	Wouter Brink	Restructured the document.
31/03/2019	Wouter Brink	Added Maintainability and Scalability to the non functional requirements.
01/04/2019	Maarten de Jager	Added glossary and corrected spelling
05/05/2019	Oane Gros	Concretized the functional requirements