Mondrian: Spreadsheet Layout Detection

Gerardo Vitagliano
Hasso Plattner Institute
University of Potsdam, Germany
gerardo.vitagliano@hpi.de

Lucas Reisener
Hasso Plattner Institute
University of Potsdam, Germany
lucas.reisener@student.hpi.de

Lan Jiang
Hasso Plattner Institute
University of Potsdam, Germany
lan.jiang@hpi.de

Mazhar Hameed
Hasso Plattner Institute
University of Potsdam, Germany
mazhar.hameed@hpi.de

Felix Naumann
Hasso Plattner Institute
University of Potsdam, Germany
felix.naumann@hpi.de

ABSTRACT

Spreadsheet datasets are valuable sources of data, but often ill-suited for machine consumption. Their unstructured nature allows users to arrange data and metadata freely in a human-readable format, often in canvas-like layouts. To extract their content, data practitioners need to resort to manual inspection and run cumbersome preparation pipelines. The Mondrian system assists users in identifying and handling multiregion layout templates: spreadsheet layouts composed of independent regions that appear repeatedly across different files. Mondrian comprises an automated approach to detect multiple regions within a single file and an algorithm that leverages mapping region layouts to graphs to compute layout similarity and identify templates [10]. Users interact with Mondrian through a web-based visual interface, that serves as a practical toolkit to handle collections of multiregion spreadsheets and enables their automated preparation.

CCS CONCEPTS

• **Information systems → Data cleaning**; *Extraction, transformation and loading*; *Document structure*.

KEYWORDS

data preparation, template recognition, multiregion layout

ACM Reference Format:
Gerardo Vitagliano, Lucas Reisener, Lan Jiang, Mazhar Hameed, and Felix Naumann. 2022. Mondrian: Spreadsheet Layout Detection. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3514221.3520152

1 MULTIREGION TEMPLATES

Spreadsheet files, ubiquitously used for data distribution and analysis, often suffer from data quality issues [1]. The Mondrian approach, algorithmically described in [10], assists users in exploring, preparing, and integrating data from large-scale spreadsheet corpora.

SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9249-5/22/06.
https://doi.org/10.1145/3514221.3520152

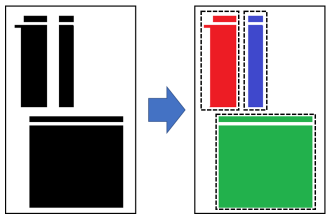


Figure 1: The layout of a multiregion file.

In particular, we address the challenges of layout template recognition and region extraction in "multiregion files", data files characterized by multiple, independent regions (Cf. Figure 1). In a dataset, the layout of files often follows a common "template", for instance, in files with related data from different sources, e.g., the same tables with different data points, or in files produced from the same source, e.g., different tables produced with the same export process.

The present work demonstrates an interactive interface for Mondrian, designed for users to visualize file layouts, explore templates, and automatically extract independent regions, either from a single multiregion file or from all files that belong to the same template.

Our research found multiregion layouts to be a common occurrence in real-world data sources: the Deco dataset [6], a collection of 854 files from the ENRON Excel corpus, contains 621 files with multiple regions and 71 unique layout templates; the Fuste dataset [10], a random sample of 886 spreadsheets from the FUSE web corpus, contains 391 files with multiple regions and 31 unique layout templates; Mitlöhner et al. reported in a survey of 141k open data csv files that 4.6% of the parsing errors were due to "too many tables" [9]. If users are interested in processing data contained in large spreadsheet collections, they must prepare the files by first recognizing their layout and then splitting the different regions into separate files to obtain machine-readable content.

The intuition of Mondrian is to transform data files into images to analyze their visual layout. Then, using density-based clustering, we automatically recognize independent spreadsheet regions as groups of pixels. For each file, we encode its region layout as a graph, and recognize templates with a graph similarity computation. With its unsupervised nature, as well as its independence from format-specific style features, our approach can generalize to a wide range of unseen files without the need for training data. The visual nature of Mondrian can be elegantly leveraged in a user-friendly graphical interface to interact with the results of the automated

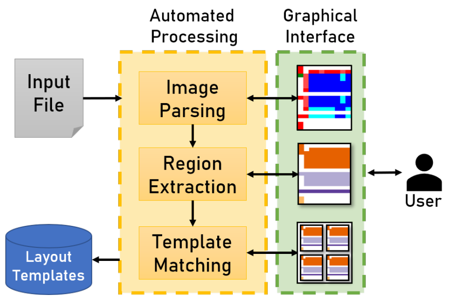


Figure 2: Mondrian pipeline.

components. Combining automated template recognition with a graphical interface, users can quickly navigate through files with the same layout and discover meaningful data relationships. Moreover, leveraging structural region detection across layouts, users can extract the same regions across multiple files with as simple as a click, irrespective of file-specific differences, such as different numbers of rows or different positions in the layout.

Detecting layout templates for sets of multiregion files is a novel research problem. However, the problem of recognizing tabular structures in single spreadsheets has seen some interest: WebSmatch [3], and TableSense [5], like Mondrian, leverage the intuition of analyzing spreadsheets by applying techniques from the computer vision domain combined with supervised machine learning. Supervised learning is also at the core of Pytheas [2], a line-based classifier to detect tables in CSV files, and in the genetic-based approach presented by Koci et al. [7], which uses a graph encoding of tabular layouts. Using supervised machine learning, all the aforementioned systems necessitate large quantities of training data, while the unsupervised nature of Mondrian makes it fit even for smaller datasets and more likely to generalize well to unseen data. The code and datasets used for developing Mondrian are publicly available [1].

Section 2 presents brief definitions of the concepts of region, layouts, and templates, an overview of Mondrian and its automated components. Section 3 demonstrates a concrete usage scenario of our system using its graphical interface. Section 4 provides brief concluding remarks.

2 AN OVERVIEW OF MONDRIAN

Mondrian leverages the information that lies in the visual distribution of spreadsheet cells to recognize and extract tabular data and layout templates. Figure 2 shows an overview of its pipeline.

Due to the short nature of this paper, and its focus on the user interaction with Mondrian, we discuss the automated steps of region detection and template inference only briefly. For a detailed description compounded with extensive experimental analysis on real-world datasets summing up to above 1,500 files, we refer to our prior work [10].

Mondrian operates on value-delimited files whose content is arranged in a grid-like fashion of cells. A group of non-empty, adjacent cells is an "element": often files contain multiple elements separated by empty cells, e.g., for visual purposes or due to missing data values. File regions are sets of elements that compose a semantically related unit. In the file of Figure 1, for example, there are three regions, each representing a different table and composed of two elements (the header and the data rows). The *layout* of a file is the set of all regions in it, along with their pairwise spatial relationship, expressed in terms of the distance and alignment of their cells. A *template* is a set of files with an "equivalent" layout: a layout with the same number and type of regions, arranged in the same positions relative to each other.

To extract file layouts and detect templates, the Mondrian approach follows three phases: image parsing, region extraction, and template matching. The *image parsing* phase creates a colored image using the data type of its cells. In the image, each pixel corresponds to a single file cell, and its color reflects the syntactic type of the cell content: white for empty cells, blue for numeric data, green for dates, and red for strings. Different shades of these colors reflect sub-types, e.g., light blue represents integers and dark blue represents floats. Using the web-based graphical interface, users of Mondrian can visualize the file content and the result of the image parsing. The *region extraction* phase segments images to obtain independent regions. In its first step, Mondrian identifies connected components: groups of 4-connected non-empty pixels adjacent to one another (e.g., the black polygons of Figure 1). Subsequently, we perform a *rectilinear* cut to partition them further, obtaining smaller elements. This step is needed because independent regions may still be adjacent, with their elements not visually separated by empty lines. The elements are then grouped to form regions using a density-based clustering algorithm. The clustering is based on three features: the distance between elements, their alignment, and their relative size. End-users can fine-tune the coefficients of these features as hyperparameters of the clustering distance. The boundary of a region corresponds to the maximum bounding box encompassing all elements clustered together. The resulting regions can be visually inspected in the graphical interface, and their boundaries can be interactively corrected if needed.

The *template matching* phase encodes each file layout as a graph, based on its extracted regions. The graphs are complete: each detected region is a node in the graph, and all of them are connected with pair-wise edges, labeled with a feature vector describing the spatial relationship for each pair of regions. This feature vector has three components: the *alignment* of the regions, the *magnitude* of their alignment, and their *distance*. The *alignment* measures whether two regions share some coordinates on either the x-axis or the y-axis, or if they overlap. The alignment *magnitude* of two regions corresponds to the number of pixels for which they are aligned. Regions that are not aligned have an alignment magnitude of 0. The *distance* of two regions is the distance of their two closest points. Overlapping regions have a distance of 0.

After encoding all the files' layouts in a dataset with such graphs, Mondrian measures layout similarity on pairs of layouts using a graph similarity measure based on the similarity-flooding algorithm [8]. To retain only interesting file pairs and prune the search space, Mondrian leverages the visual similarity of previously detected regions across files: each region is associated with a fingerprint calculated as the color histogram of its cells, which ultimately

[1]https://github.com/HPI-Information-Systems/Mondrian

reflects its syntactic structure. Since similar cell types have closer colors, regions with similar structures have closer color histograms.

Across different files, regions are compared using the cross-correlation of their fingerprints: although this operation is coarse-grained due to the unsophisticated color-encoding of cells, it is successful in discarding pairs of clearly different regions and retains only those with a high chance of belonging to the same template, i.e., with a cross-correlation above a given threshold.

If Mondrian finds two files to have a sufficiently similar pair of regions, it computes the graph similarity of their layouts. A detailed description of the full algorithm can be found in [10].

Once the similarity of all relevant file pairs is calculated, templates are defined as classes of file layouts with a similarity above a given threshold (configurable by end-users, 0.98 by default). Finally, Mondrian determines a transitive closure on layout templates such that if the pairs of files (f_a,f_b) and (f_b,f_c) have both a similarity above the threshold, a single template will contain all three files (F_a, F_b, F_c), regardless of the similarity of (f_a, f_c). The graphical interface presents the output of the template matching stage as a list of templates and their associated files, allowing users to interactively edit them and save the results.

Once templates are recognized, it is useful to match equivalent (data-)regions across different files with the same template. However, Mondrian recognizes only pairwise equivalence between regions of two given files. Matching a region across k files belonging to a template resolves to the "k-multicolored clique" problem, a problem proven to be of W[1] complexity [4]. Therefore, we address this problem heuristically: we order files in a template by their name and create a collection of regions, starting from one region and selecting the equivalent region for the next file, in a "daisy-chain" fashion.

In [10], we show that Mondrian is highly effective in dealing with multiregion spreadsheets. On different real-world spreadsheet corpora that contain an abundance of multiregion layouts, we outperform state-of-the-art approaches for detecting the region boundaries on CSV files. Moreover, our experimental results on layout similarity prove that Mondrian is successful in detecting recurring layout templates.

3 DEMONSTRATION SCENARIO

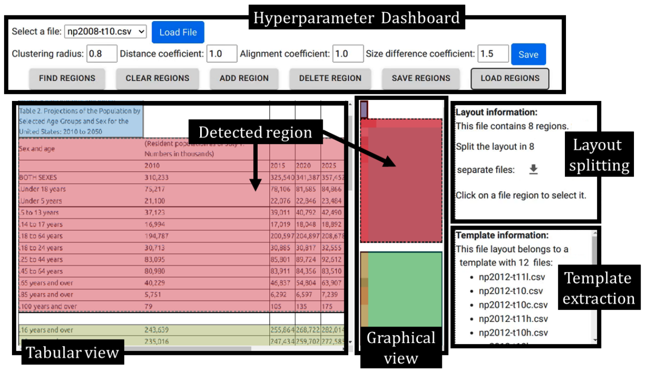
In this work, we demonstrate the interactive components and the graphical interface to Mondrian describing one demonstration scenario. A fully functional Mondrian demo with two additional scenarios is available online [2].

In this scenario, a data practitioner is interested in analyzing the historical United States population data, retrieved from the open data portal of the United States Census Bureau[3]. The summary tables for the censuses of several years are available in spreadsheet format, as a collection of CSV and XLS files.

The goal of the practitioner is to consolidate the information contained in different tables in a single source of truth, e.g., a relational database, to enable querying, analysis, and visualization. This task is tedious and not trivial: some of these files contain multiple data tables as well as metadata regions: preamble cells containing

[2]https://hpi.de/naumann/projects/data-preparation/mondrian

[3]https://www2.census.gov/programs-surveys/popproj/tables/ (accessed Jan. 03, 2022)



Select a file: np2008-t10.csv

Clustering radius: 0.8 Distance coefficient: 1.0 Alignment coefficient: 1.0 Size difference coefficient: 1.5

Table 2. Projections of the Population by Selected Age Groups and Sex for the United States: 2010 to 2050

Sex and age	(Resident population as of July 1. Numbers in thousands)			
	2010	2015	2020	2025
BOTH SEXES	310,233	325,540	341,387	357,45
Under 18 years	75,217	78,106	81,685	84,866
Under 5 years	21,100	22,076	22,846	23,484
5 to 13 years	37,123	39,011	40,792	42,490
14 to 17 years	16,994	17,019	18,048	18,892
18 to 64 years	194,787	200,597	204,897	208,67
18 to 24 years	30,713	30,885	30,817	32,555
25 to 44 years	83,095	85,801	89,724	92,612
45 to 64 years	80,980	83,911	84,356	83,510
65 years and over	40,229	46,837	54,804	63,907
85 years and over	5,751	6,292	6,597	7,239
100 years and over	79	105	135	175
16 years and over	243,639	255,864	268,722	282,01
	235,016	247,434	259,702	272,58

Layout information:
This file contains 8 regions.
Split the layout in 8 separate files:
Click on a file region to select it.

Template information:
This file layout belongs to a template with 12 files:
- np2012-t11l.csv
- np2012-t10.csv
- np2012-t10c.csv
- np2012-t11h.csv
- np2012-t10h.csv

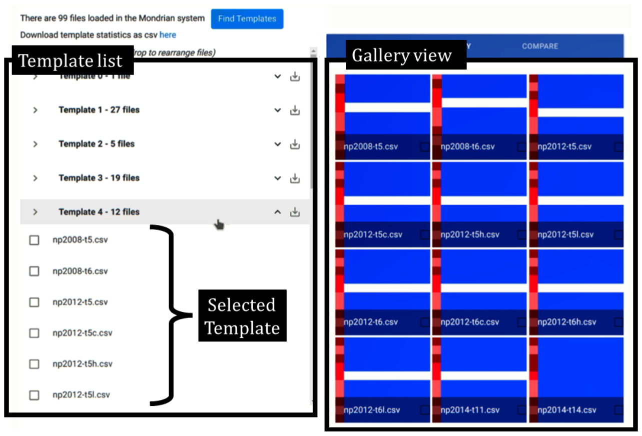
Figure 3: Region detection page with file layout regions.

the table name and structure, and footnote cells with additional metadata. To isolate the tables, the practitioner needs to inspect each file individually and split it according to its structure. Due to the nature of the dataset, some spreadsheet files share the same layout, both within a single year and across different years. To address these issues and leverage layout similarity to automate the preparation of these spreadsheets, the practitioner can exploit Mondrian's user-friendly web interface[4], based on our unsupervised layout and template detection approach [10]. At the time of file upload, Mondrian automatically performs the image parsing and region detection steps. The results of these processes can be viewed on the "Region Detection" page (Figure 3), which displays the content of a single file, both in tabular format and as a "Mondrian" image (Section 2). Colored rectangles represent the boundaries of the automatically detected regions both on the tabular and the graphical file view. Clicking on a rectangle prompts an "edit" mode, where the user can interactively change the detected regions' boundaries by performing drag-and-drop actions. Additionally, entire regions can be added and removed. For ease of use, all interactive actions can be performed either on the tabular view or on the compact graphical view. To leverage the results of the region detection phase and assist users in their further preparation of single multiregion spreadsheets, this page also offers a "split" feature to create a separate file for each of the detected regions. Moreover, if the layout of the file belongs to a template, users can select a given region and extract that region from all files of the same template, using the procedure described in Section 2.

To analyze an entire collection of uploaded files and exploit the automated Mondrian layout template matching, users can resort to the "Template Detection" page (Figure 4). The page shows two panels: the panel on the left contains a list populated with the various layout templates found in the dataset. Each entry contains the names of the files sharing the same template. The page allows interactive manipulation of the list: with drag-and-drop actions, users can move files from one template to another, as well as create new, empty templates. From the list, each template can be downloaded as a ZIP file containing all its files.

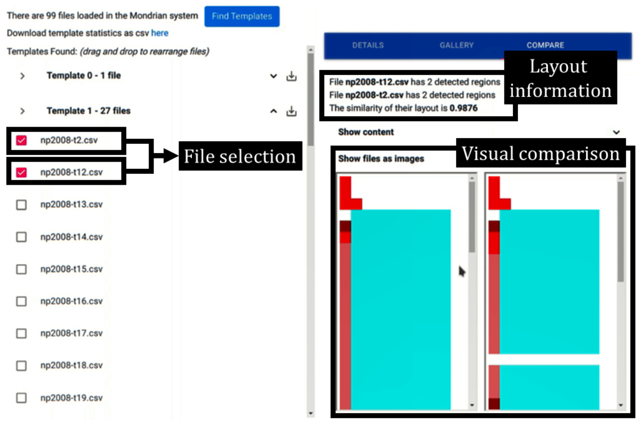
The panel on the right contains three tabs with additional visualizations for templates and their files: the "details" tab, the "gallery"

[4]Please also see the accompanying demo video file.


There are 99 files loaded in the Mondrian system

Download template statistics as csv here

Figure 4: Template detection page: gallery view.


There are 99 files loaded in the Mondrian system

Download template statistics as csv here

Templates Found: *(drag and drop to rearrange files)*

File **np2008-t12.csv** has 2 detected regions
File **np2008-t2.csv** has 2 detected regions
The similarity of their layout is **0.9876**

Figure 5: Table detection page: comparison view of two files.

tab, and the "compare" tab. The first tab shows the details of a given template selected from the list. Among other information, it contains the pairwise layout similarity of the files belonging to a selected template. By inspecting these scores, users can gain a more in-depth understanding of the reasons that led Mondrian to its clustering decisions. The second tab, selected in Figure 4, displays a gallery view of the renderings for all the files belonging to that template. The gallery shows the Mondrian images for all files that belong to a template. Even from a cursory glance, it is possible to validate the results of the template detection and spot possible "outliers" by visual inspection of the images shown in the gallery.

From the list on the left or from the gallery view, users can select pairs of files for further inspection. A detailed comparison for the selected pair of files then becomes available in the third tab of the right panel (Figure 5). Here, users can find information about the number of detected regions for each of them, as well as their layout similarity. Additionally, this tab presents a side-by-side comparison of two files' data content and their image rendering.

Finally, to leverage the information about the discovered layout templates, Mondrian can provide users with a downloadable csv file that contains the template statistics for the dataset: a list of files with their corresponding template, and the average layout similarity to every other file in the same template.

In our US Census Population scenario, out of 99 different spreadsheet files, the system identified 15 different layout templates. The biggest dimension for a template is 27 files, and the smallest is 1, for six different singleton templates. All non-singleton templates, except for one, contain files from different years. Upon further inspection, two pairs of singleton templates had files with the same structure, but one contained floating-point numbers and the other integer numbers. After interactively readjusting these templates, knowing which files share the same layout, the user can trivially extract the data tables and consolidate them in a relational database.

4 CONCLUSION

We presented Mondrian, a novel system designed to address data preparation on multiregion files. Thanks to its layout template detection, users can identify recurring simple or multiregion layouts occurring in a given collection of spreadsheet files, independent of their specific format. Leveraging the demonstrated user-friendly graphical interface, users can easily group different files according to their layout template as well as split them to retain only the interesting layout regions, for further automated preparation. In the demonstration, we showcase three separate scenarios.

ACKNOWLEDGEMENTS

This research was funded by the HPI research school on Data Science and Engineering.

REFERENCES

[1] Laura Chiticariu, Yunyao Li, Sriram Raghavan, and Frederick R. Reiss. 2010. Enterprise Information Extraction: Recent Developments and Open Challenges. In *Proceedings of the International Conference on Management of Data (SIGMOD)*. ACM, 1257–1258.
[2] Christina Christodoulakis, Eric Munson, Moshe Gabel, Angela Demke Brown, and Renée J. Miller. 2020. Pytheas: Pattern-based Table Discovery in CSV Files. *PVLDB* 13, 11 (2020), 2075–2089.
[3] Remi Coletta, Emmanuel Castanier, Patrick Valduriez, Christian Frisch, DuyHoa Ngo, and Zohra Bellahsene. 2012. Public data integration with WebSmatch. In *Proceedings of the International Workshop on Open Data (WOD)*. ACM, 5–12.
[4] Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. 2009. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science (TCS)* 410, 1 (2009), 53–61.
[5] Dong Haoyu, Liu Shijie, Han Shi, Fu Zhouyu, and Zhang Dongmei. 2019. TableSense: Spreadsheet Table Detection with Convolutional Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 69–76.
[6] Elvis Koci, Maik Thiele, Josephine Rehak, Oscar Romero, and Wolfgang Lehner. 2019. DECO: A Dataset of Annotated Spreadsheets for Layout and Table Recognition. In *Proceedings of the IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE Computer Society, Washington, DC, USA, 1280–1285.
[7] Elvis Koci, Maik Thiele, Oscar Romero, and Wolfgang Lehner. 2019. A Genetic-based Search for Adaptive Table Recognition in Spreadsheets. In *Proceedings of the IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE Computer Society, Washington, DC, USA, 1274–1279.
[8] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. 2002. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In *Proceedings of the International Conference on Data Engineering (ICDE)*. IEEE Computer Society, 117–128.
[9] Johann Mitlöhner, Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. 2016. Characteristics of open data CSV files. In *Proceedings of the Image Analysis and Processing Conference (ICIAP)*. IEEE Computer Society, Washington, DC, USA, 72–79.
[10] Gerardo Vitagliano, Lan Jiang, and Felix Naumann. 2022. Detecting Layout Templates in Complex Multiregion Files. In *PVLDB*, Vol. 15. ACM, New York, NY, USA, 646–658. Issue 3.