String Charts: Visual Transport Schedules

Trinish Nepal, Martin Rabensteiner, and Stephan Robinig

DAT.C406UF Information Visualisation 3VU SS 2025
Graz University of Technology

26 May 2025

Abstract

String charts have played a pivotal role in transportation planning for over a century, providing a visual means of representing temporal and spatial relationships in transit systems. This survey first explores the historical context of string charts, following the 1843 Versailles railway disaster. Key figures such as Robert Marie Édouard Petiet, Étienne-Jules Marey, and Charles Ibry laid the groundwork for graphical timetable representations, which evolved into modern string chart methodologies.

The survey then identifies and compares eleven contemporary tools used to create and manage string charts. Five of them are publicly available and/or open-source, while six are commercial enterprise solutions. Each tool is evaluated based on ownership, licensing, supported import/export formats and the used tech stack. The comparison highlights the diversity of current offerings, from lightweight tools to fully integrated enterprise platforms, reflecting a wide range of use cases and technical preferences.

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

String charts, also known as stringline charts, marey charts, time-location diagrams, or time-distance diagrams, are a widely used visualisation technique for planning and analysing schedules in transportation and logistics, particularly in railway and public transit systems. These diagrams plot the movement of trains or buses over time along a linear route, allowing for clear visualisation of scheduling conflicts, headways, and resource utilisation. The ability to generate and manipulate string charts digitally has become essential for planners and engineers, prompting the development of various specialised software tools.

This survey first provides a historical overview of the development of string charts. It then reviews eleven digital tools for generating string charts, supporting both practitioners and researchers in identifying tools that best suit their analytical and operational contexts, as well as tools for generating pleasing data visualisations for personal and academic use cases.

Chapter 2

Historical Background

The invention of string charts was closely tied to the need for improved safety when railroads became operational during the 19th century. A pivotal moment occurred on 08 May 1842, when a major accident happened on the Versailles railway near Paris. This exposed the importance of scheduling and coordinating trains properly. In April 1843, the French engineer Jules Petiet responded to the incident by publishing a memoir analysing the cause of the accident [Lalanne 1884].

In 1846 Charles Ibry presented his approach for the graphic timetable generation to the French public railway administration. His contribution helped standardizing the approach and laid foundation for a broader adoption across the European railway network [Lalanne 1884]. An early example of a string chart for the Northern Route between Calais and Paris from 1852 is shown in Figure 2.1. It is attributed to an L. Danel.

The origins of the invention remain somewhat uncertain, though it is widely recognised that Charles Ibry was responsible for the first documented design, specifically credited with inventing graphic charts to represent the movement of trains. Despite this, Étienne-Jules Marey has often been incorrectly associated with the invention due to the prominence of his publication [Marey 1878]. This influential work, which explores the intersection of psychology, graphic theory, and photography, serves as a general treatise on information visualisation and helped popularize the graphical method. One of Ibry's charts, shown in Figure 2.2, is contained on page 21, which perhaps inadvertently contributed to the confusion surrounding its origins.

Another key figure in the historical narrative is the Russian Lieutenant Serjev, of the Corps of Transportation Engineers, who is also speculated to have been the original inventor of string charts. However, it remains unclear whether Ibry or Serjev was the first to conceive the idea, as definitive evidence distinguishing their contributions has yet to emerge [Wainer et al. 2013].

The Gotthardbahn in Switzerland, opened in 1882, required precise scheduling due to its challenging Alpine terrain and single-track sections. String charts were used extensively to manage train movements through tunnels and steep gradients, ensuring efficient and safe operations. These charts became a vital part of day-to-day planning on one of Europe's most complex railway routes [von Röll 1914]. Figure 2.3 shows a graphical timetable from 1899.

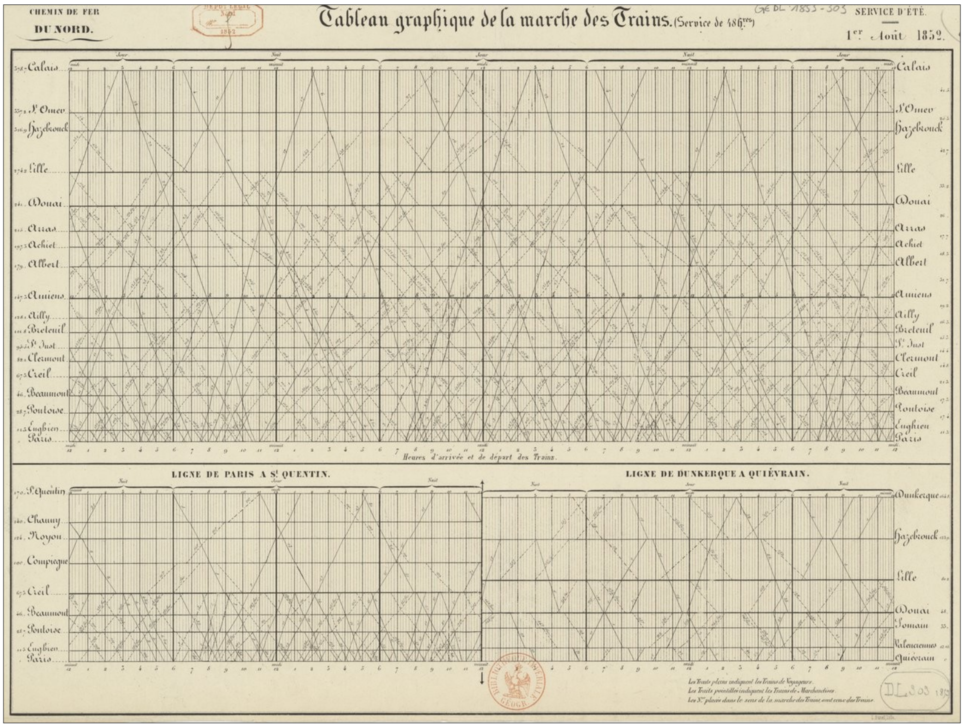


Figure 2.1: String chart of the Northern Route between Calais and Paris from 01 Aug 1852. [Extracted from gallica.bnf.fr / Bibliothèque nationale de France [Danel 1852]. Used under the Terms of Use of BnF.]

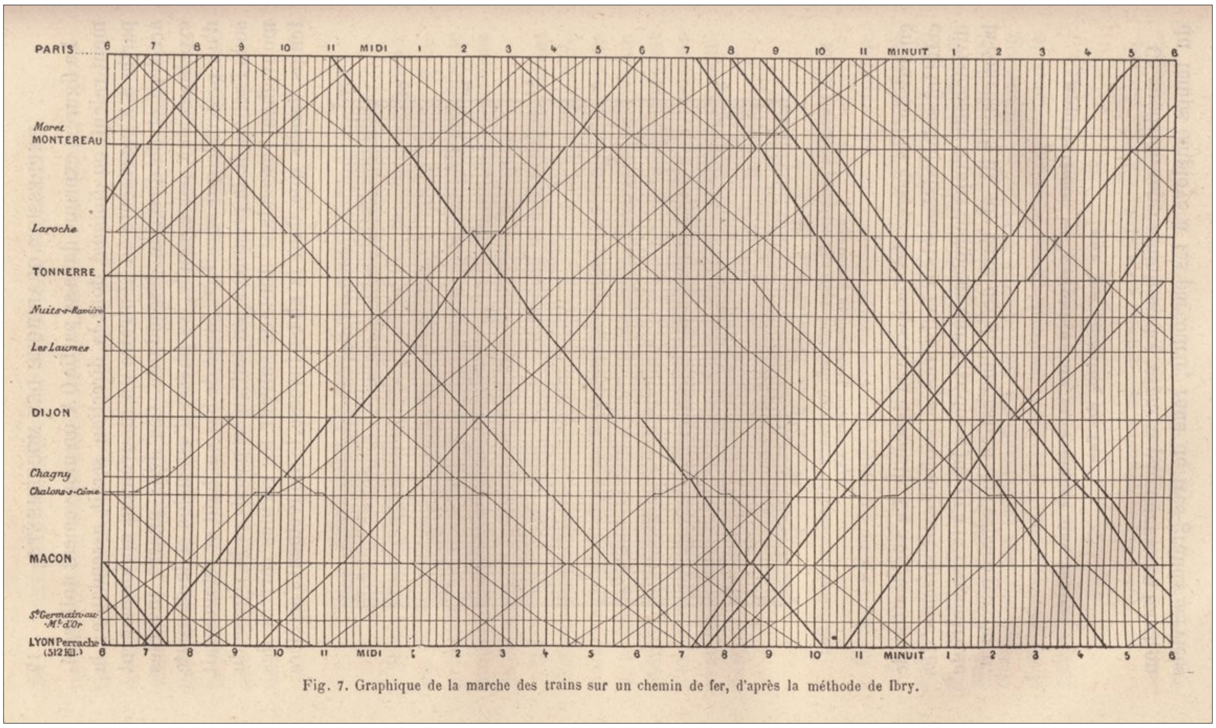


Figure 2.2: String chart created by Ibry, published By Étienne-Jules Marey in 1878. [Extracted from gallica.bnf.fr / Bibliothèque nationale de France [Marey 1878, page 21]. Used under the Terms of Use of BnF.]

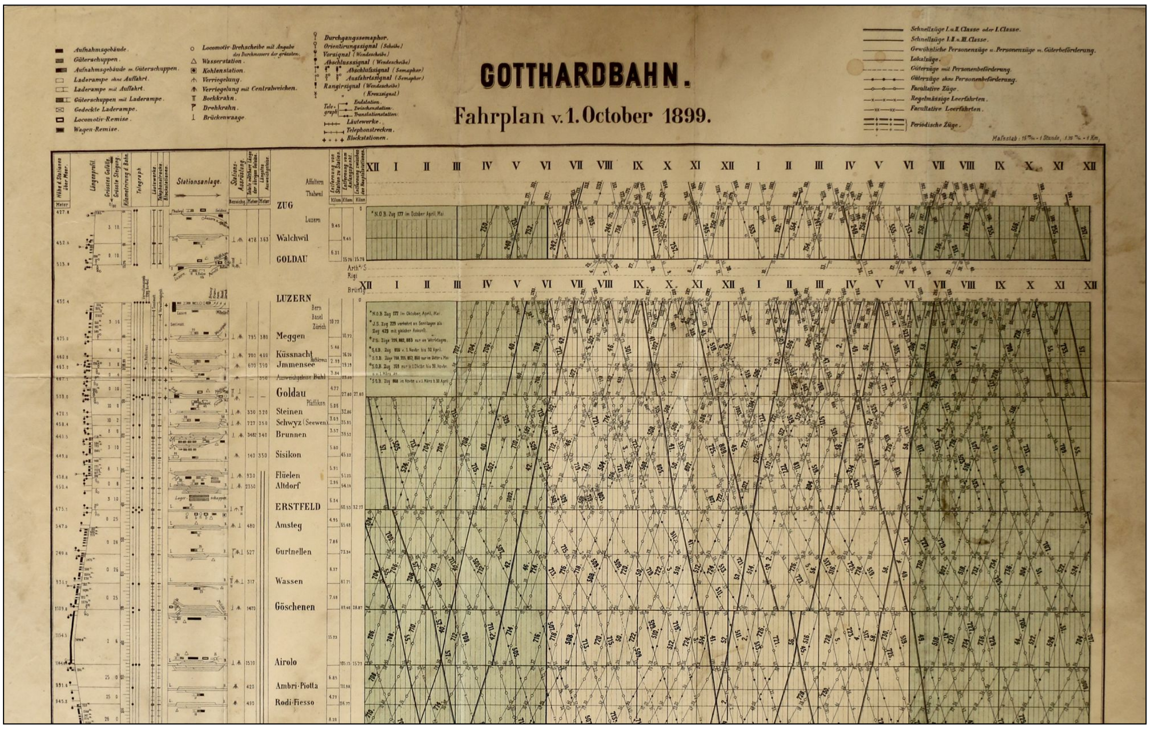


Figure 2.3: Part of the graphical timetable for the Gotthardbahn in Switzerland from 01 Oct 1899. [Image created by Hanspeter Baumeler [Baumeler 1899]. Used under the terms of a CC BY-SA 4.0 licence.]

Chapter 3

String Chart Tools

Many tools and applications are available for creating and interacting with string charts. Eleven such tools are reviewed in this chapter. They are grouped into two categories: five free tools and six commercial tools. The following criteria were used to compare the various tools:

- *Owner*: The organisation, company, person,or academic institution responsible for developing and maintaining the software.
- *Licence*: The licencing model under which the software is distributed (for example open-source, proprietary, or freeware).
- *Import Formats*: The formats supported for importing a dataset, for example, CSV, XML [W3C 2016], and GTFS (see Appendix A.
- *Export Formats*: The formats supported for exporting a string chart, for example, SVG [W3C 2025] or PNG.
- *Tech Stack*: The primary programming language(s), frameworks, and libraries used in the software's development.
- *Additional Features*: Notable capabilities such as real-time data integration, optimisation functions, user interface design, or integrations with other planning systems.

3.1 Free Tools

Five of the reviewed tools are publicly available and free to use: MBTA Viz, Marey Graph Generator, Marey's Trains, jTrainGraph, and String Charter 2. They are summarised in Table 3.1.

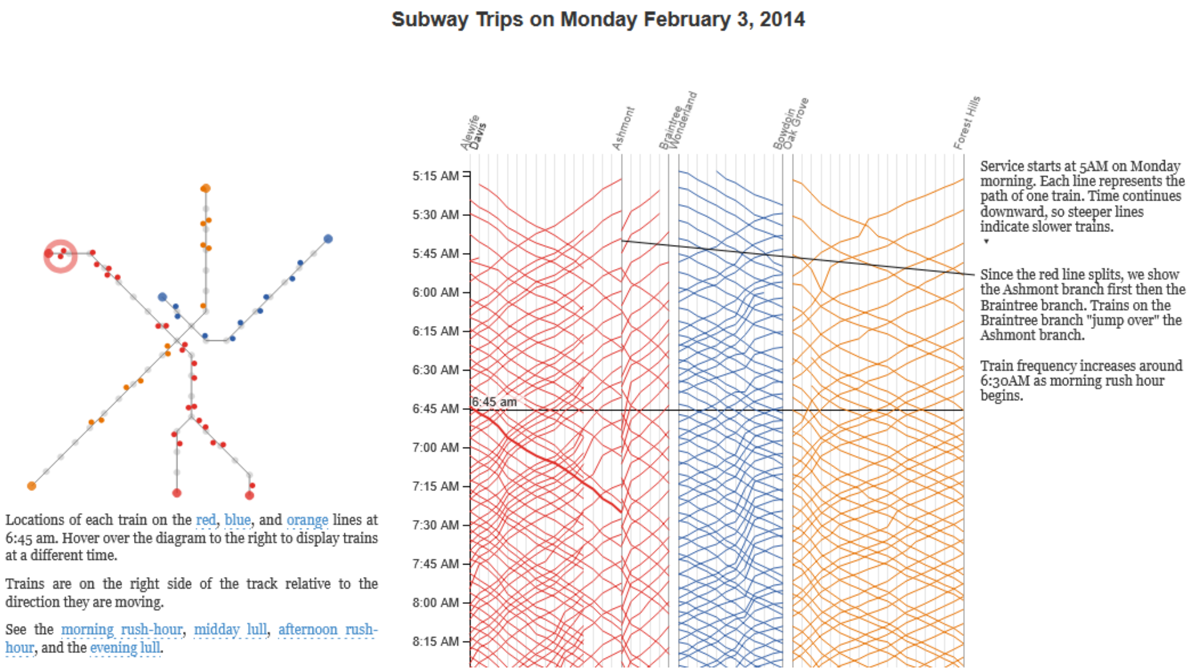
3.1.1 MBTA Viz

MBTA Viz is an interactive data visualisation developed by Mike Barry and Brian Card [Barry and Card 2014; Barry and Card 2023]. The application visualizes data provided by the Massachusetts Bay Transit Authority (MBTA) in Boston, USA. It has the following characteristics:

- *Owner*: The application was built by Mike Barry and Brian Card. The data was provided by MBTA and their Developer Relations Program.
- *Licence*: MIT licence [MIT 2025].
- *Import Options*: The application uses JSON format for its data input.
- *Export Options*: The application visualizes the data as a web page. There is no explicit facility to export the string chart as an image or vector graphic. The user can take a screenshot or print the

	Licence	*Import*	*Export*	*Tech Stack*	*Remarks*
MBTA Viz	MIT	JSON	Screenshot	HTML, JS, web server	
Marey Graph Generator	CC-BY-NC-SA 4.0	Manual input	Screenshot	HTML, JS	Cumbersome for large data.
Marey's Trains	ISC	TSV	SVG	JS, D3, web server	
JTrainGraph	Free, Pro, Commercial	FPL, CSV, XSL(C)	FPL, CSV, XSL(C), JPG, GIF, PNG, Print	Java	Collision detection.
String Charter 2	MIT	GTFS	SVG	Tauri, Node.js, TypeScript	

Table 3.1: Summary of the five reviewed tools which are publicly available and free to use.


Subway Trips on Monday February 3, 2014

Locations of each train on the red, blue, and orange lines at 6:45 am. Hover over the diagram to the right to display trains at a different time.

Trains are on the right side of the track relative to the direction they are moving.

See the morning rush-hour, midday lull, afternoon rush-hour, and the evening lull.

Service starts at 5AM on Monday morning. Each line represents the path of one train. Time continues downward, so steeper lines indicate slower trains.

Since the red line splits, we show the Ashmont branch first then the Braintree branch. Trains on the Braintree branch "jump over" the Ashmont branch.

Train frequency increases around 6:30AM as morning rush hour begins.

Figure 3.1: MBTA Viz: String chart with corresponding geographic train locations. [Screenshot taken by Stephan Robinig from MBTA Viz [Barry and Card 2014].]

web page.

- *Tech Stack*: The application is built with HTML and JavaScript and runs on a web server.

An example of a string chart created by MBTA Viz is shown in Figure 3.1. By hovering over a time on the string chart, the corresponding location of the trains is displayed in the route map on the left. Figure 3.2 shows the applications second use of string charts to visualize the delays on the same tracks over a longer period of time. Furthermore, the application provides many additional visualisations and info graphics displaying entrances and exits of people, delays on sections of track, and expected commute delays according to the time of day.

Since the application is specifically made for the MBTA dataset, using the code for a different transport

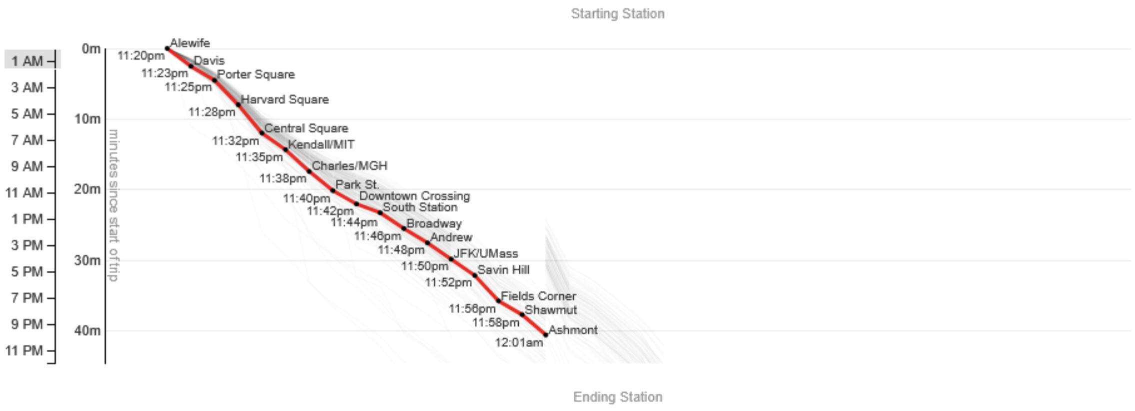


Figure 3.2: MBTA Viz: String chart showing delays on the same track. [Screenshot taken by Stephan Robinig from MBTA Viz [Barry and Card 2014].]

network will require some preparation of the data and tweaking of the code.

3.1.2 Marey Graph Generator

Marey Graph Generator is an open-source tool developed by CondensedChaos [CondensedChaos 2023a; CondensedChaos 2023b], as a complement for a game mod. It has the following characteristics:

- *Owner*: The application was developed by Github user CondensedChaos.
- *Licence*: CC-BY-NC-SA 4.0.
- *Import Options*: To create a string chart, the train stations, lines, and times have to be manually entered and cannot be loaded or imported.
- *Export Options*: The application visualizes the data as a web page. There is no explicit facility to export the string chart as an image or vector graphic. The user can take a screenshot or print the web page.
- *Tech Stack*: The application is built in JavaScript with jQuery. Highcharts is used to draw the string chart. To run the application, no HTTP Server is needed. It is sufficient to open the index.html file provided by the Github repository.

Figure 3.3 shows an example string chart generated by Marey Graph Generator. Due to the application's limitations, the string chart is only available in dark mode. Not being able to save and import previous data makes it quite cumbersome to use for large amounts of data.

3.1.3 Marey's Trains

Marey's Trains is a web application developed by Mike Bostock to generate string charts [Bostock 2021]. It has the following characteristics:

- *Owner*: Written by Mike Bostock and published on Observable.
- *Licence*: ISC licence [ISC 2025].
- *Import Options*: The application allows for data input in TSV format and needs information on the stations, their distances, and stop times.
- *Export Options*: The generated string chart can be exported as SVG.

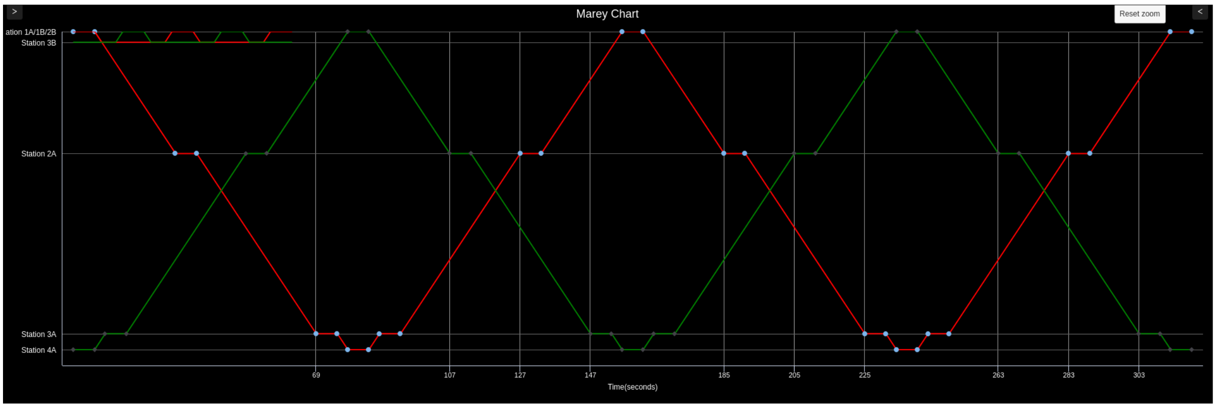
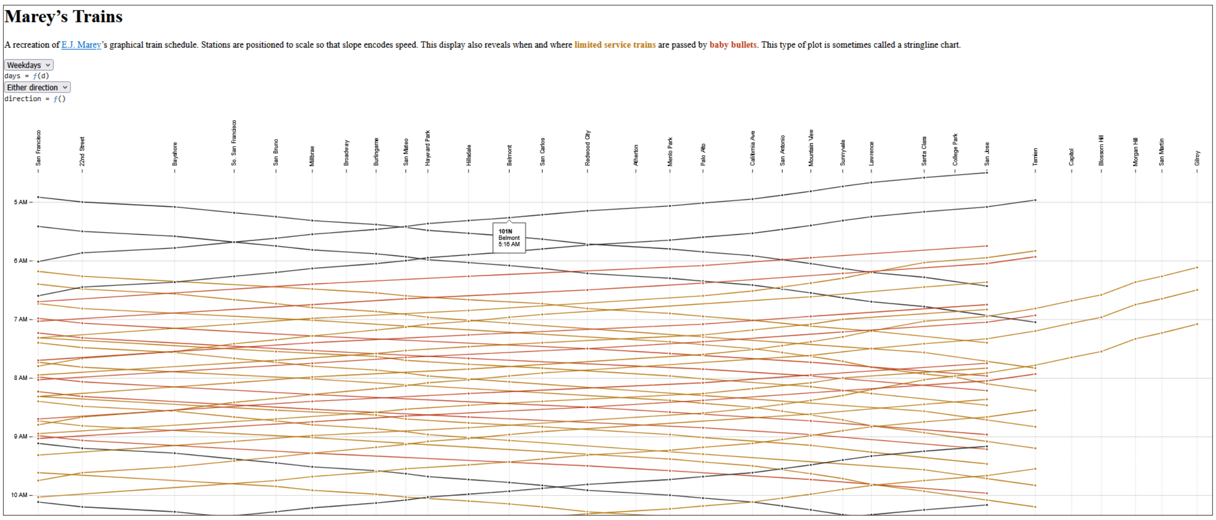


Figure 3.3: Marey Graph Generator: String chart of some simple example data. [Screenshot taken by Stephan Robinig using Marey Graph Generator [CondensedChaos 2023a].]



Marey's Trains

A recreation of E.J. Marey's graphical train schedule. Stations are positioned to scale so that slope encodes speed. This display also reveals when and where **limited service trains** are passed by **baby bullets**. This type of plot is sometimes called a stringline chart.

Weekdays
days = ƒ(d)
Either direction
direction = ƒ()

Figure 3.4: Marey's Train: String chart with provided data. [Screenshot taken by Stephan Robinig from Marey's Trains [Bostock 2021].]

- *Tech Stack*: The visualisation is written in JavaScript and uses the D3 package to generate the string chart. To run the application locally, an HTTP Server is required.

Figure 3.4 shows a string chart generated by a locally run instance of the application. Using the online implementation on Observable, one can modify the look and behaviour of the string chart on the fly, and immediately view the results.

To load custom data into the visualisation, it has to be formatted correctly. As can be seen in Listing 3.1, the TSV consists of time data. The header of each stop is encoded in the form:

```
stop|*name*|*distance*|*zone*
```

3.1.4 jTrainGraph

jTrainGraph is an application developed by Moritz Scherzinger, supporting string charts among other tools for railway planing, representation and supervision [Scherzinger 2022]. For this survey project, access to the to Pro version was kindly provided. It has the following characteristics:

- *Owner*: The application is developed by Moritz Scherzinger who programs, maintains, and distributes it as a private person.

```
number  type  direction stop|Graz|0|1 stop|Wien|40|1  stop|München|124|1
102 N S 4:55am  8:00am  16:05pm
104 N S 5:25am  9:30am  17:35pm
206 N S 7:11am  10:16am -
208 N S 7:24am  10:29am 18:34pm
210 N N 8:44am  11:49am -
312 N N 8:59am  11:04am -
314 N N 12:14pm 15:19pm -
216 N N 16:19pm - -
```

Listing 3.1: Marey's Train: Illustration of the TSV data format.

- *Licence*: jTrainGraph offers three licencing options. The free Basic version is available for non-commercial use, providing basic functionality for personal or educational projects. Users can upgrade to the Pro version for a one-time €25 fee, which unlocks additional import and export and train scheduling features including a collision detection. For commercial use, jTrainGraph offers a commercial licence that provides full functionality and professional support, with pricing tailored to specific use cases.
- *Import Options*: jTrainGraph offers not only to possibility to view data, but also to edit and schedule data about the route and trains. The main option to save and open data is the FPL format. FPL is short for Fahrplan (German for timetable) and is an XML format used among hobby rail roaders, for example also within the application FPLedit [Huber 2024].

  Within the paid version (jTrainGraph Pro), import and export is also possible to CSV and Excel (XLS, XLSX). To keep the data consistent, the spreadsheets follows a format (data template) similar to the data input interface within the application when exporting. For the import, the data template also needs to be set. To reach a better compatibility with arbitrary timetable spreadsheets, self-defined import templates are also supported.
- *Export Options*: It is possible to export a string chart as a raster graphic (JPG, GIF, PNG) or to print it (say to PDF).
- *Tech Stack*: The application is closed-source and is written in Java, therefore cross-platform compatible. In the context of this survey, it was tested on Ubuntu 24.10 and ran flawlessly. According to the manual, jTrainGraph 3.4.2 is compatible with Java 8 Update 101.

jTrainGraph allows users to create and edit string charts with customisable train paths and track layouts. Figure 3.5 shows the application's Main View. It supports multi-track configurations, timetable data import/export and graphical export. The software includes live mode for real-time updates, delay visualisation, and integration with external railway simulation tools such as Stellwerksim [Stellwerksim 2025].

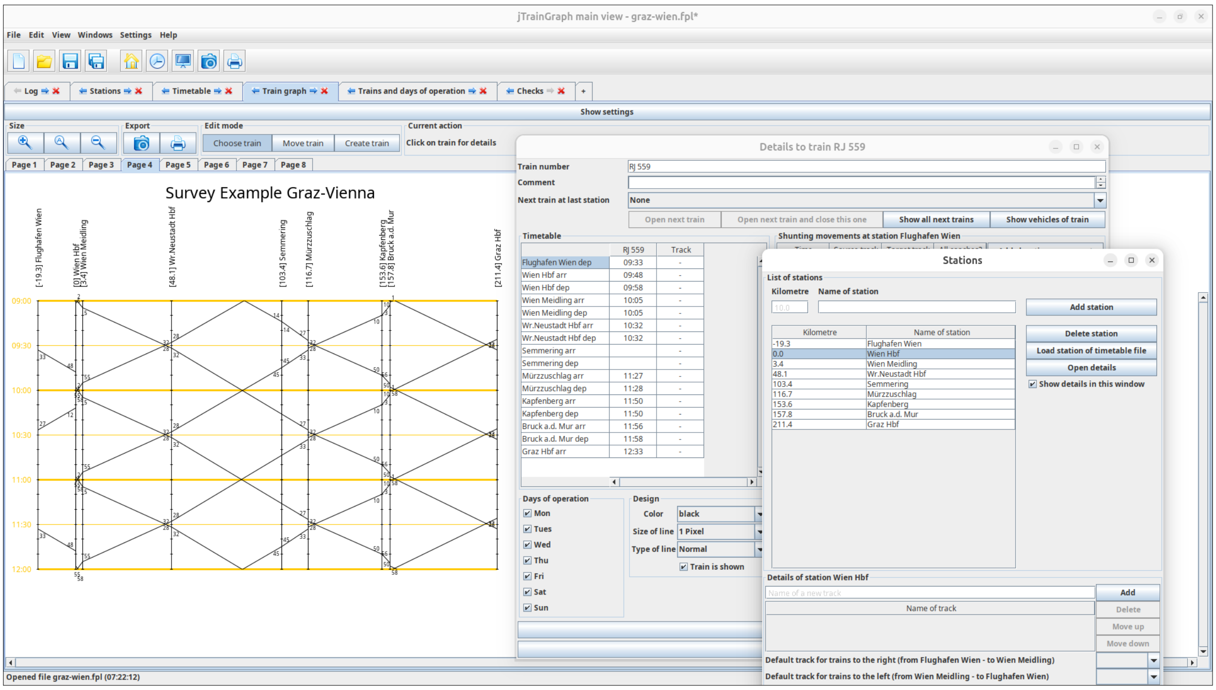
Regarding the string chart, it offers many format options for the different descriptions font properties, helplines, pagination, and the possibility to flip the x and y axes. Figure 3.6 shows a string chart exported from jTrainGraph as PNG.

3.1.5 String Charter 2

String Charter 2 was developed by three students in the scope of this course in 2023 [Gsellmann et al. 2023]. The main goal was to bring GTFS data into an SVG string charting tool.

It has the following characteristics:

- *Owner*: String Charter 2 was developed by Inge Gsellmann, Michael Hebesberger, and Danijela Lazarevic.



Current action

Click on train for details

Train number: RJ 559

Comment:

Next train at last station: None

Timetable

	RJ 559	Track
Flughafen Wien dep	09:33	-
Wien Hbf arr	09:48	-
Wien Hbf dep	09:58	-
Wien Meidling arr	10:05	-
Wien Meidling dep	10:05	-
Wr.Neustadt Hbf arr	10:32	-
Wr.Neustadt Hbf dep	10:32	-
Semmering arr		-
Semmering dep		-
Mürzzuschlag arr	11:27	-
Mürzzuschlag dep	11:28	-
Kapfenberg arr	11:50	-
Kapfenberg dep	11:50	-
Bruck a.d. Mur arr	11:56	-
Bruck a.d. Mur dep	11:58	-
Graz Hbf arr	12:33	-

Shunting movements at station Flughafen Wien

Days of operation

- ☑ Mon
- ☑ Tues
- ☑ Wed
- ☑ Thu
- ☑ Fri
- ☑ Sat
- ☑ Sun

Design

Color: black

Size of line: 1 Pixel

Type of line: Normal

☑ Train is shown

List of stations

Kilometre: 10.0

Name of station:

Kilometre	Name of station
-19.3	Flughafen Wien
0.0	Wien Hbf
3.4	Wien Meidling
48.1	Wr.Neustadt Hbf
103.4	Semmering
116.7	Mürzzuschlag
153.6	Kapfenberg
157.8	Bruck a.d. Mur
211.4	Graz Hbf

☑ Show details in this window

Details of station Wien Hbf

Name of a new track

Name of track

Default track for trains to the right (from Flughafen Wien - to Wien Meidling)

Default track for trains to the left (from Wien Meidling - to Flughafen Wien)

Opened file graz-wien.fpl (07:22:12)

Figure 3.5: jTrainGraph: The Main View showing the string chart and the edit windows for the timetable and the stations. [Screenshot taken by Martin Rabensteiner of jTrainGraph [Scherzinger 2022].]

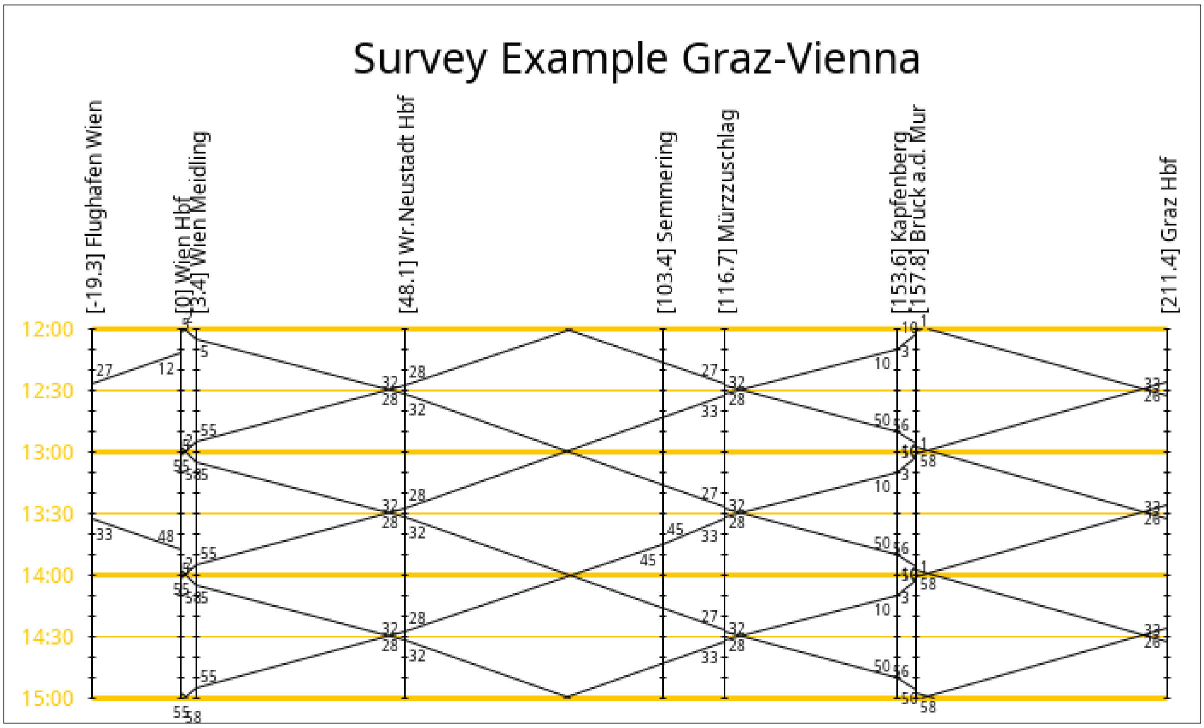


Figure 3.6: jTrainGraph: String chart exported as PNG.

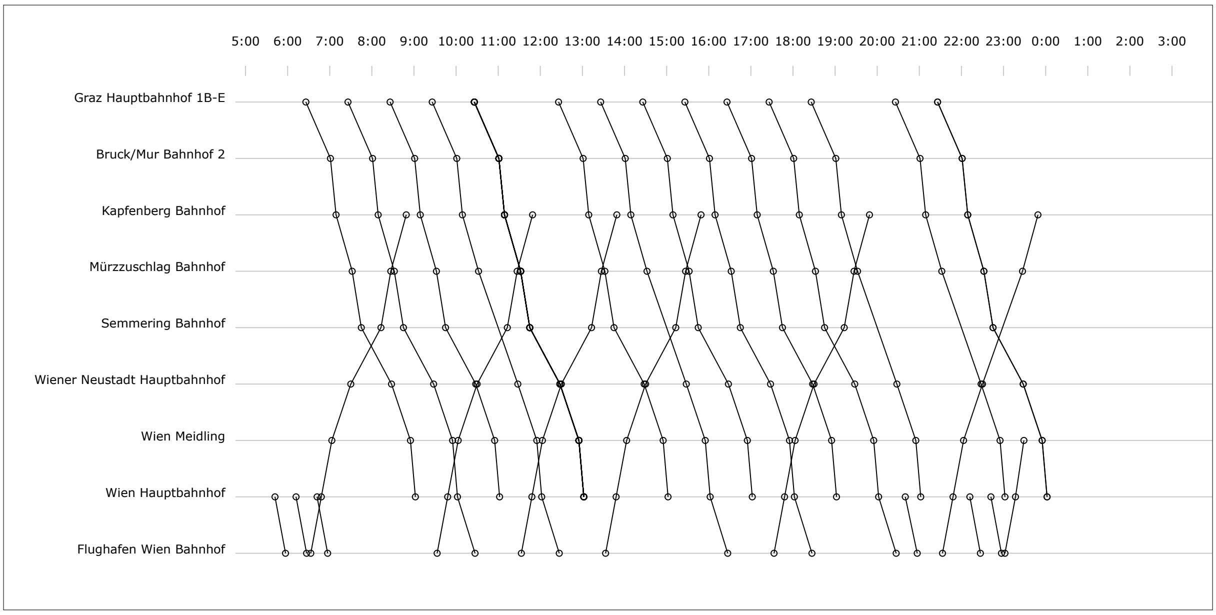


Figure 3.7: String Charter 2: Export from String Charter 2 with a route from Graz to Vienna from the ÖBB 2025 dataset.

- *Licence*: The tool was initially not licenced by the authors, but they subsequently agreed to place it under an MIT licence.
- *Import Options*: The only import option offered is the General Transit Feed Specification (GTFS, see Appendix A). However, many publically available GTFS datasets need a certain amount of cleaning and attention to produce good string chart results, especially when it comes to divergent stop patterns or where platform numbers are included in the station's name.
- *Export Options*: Generated string charts can be exported as SVG.
- *Tech Stack*: String Charter 2 is a web application written TypeScript. Native desktop applications are built using Tauri [Tauri 2025].

Figure 3.7 shows a string chart created by String Charter 2. Stations in the string chart are equally spaced, gGeographic distances between stations are not represented. In the application, the X and Y axis are flippable, making the representation flexible to what best fits the data. GTFS data is popular and widespread but can also be complicated to use. For simpler applications, for example a model railroad, supporting for a CSV or JSON import format would be helpful. Selection of routes can only be done via a sometimes very long selection box. This could be improved by a search box, where a user can search for routes or even stops and filter the routes by that.

3.2 Commercial Tools

Six commercial products for creating string charts were briefly explored: Viriato, OpenTrack, Treno, RailBase, ViziRail, and RailMP. They are summarised in Table 3.2. For the commercial tools, only limited information is publicly available and image permissions are sometimes restricted.

3.2.1 Viriato

Viriato is a commercial train schedule planning and optimisation tool, developed by SMA Group, a consulting and software company for railway systems [SMA 2025b]. It has the following characteristics:

- *Owner*: The software is owned and distributed by SMA and Partners.

	Licence	*Import*	*Export*	*Tech Stack*	*Remarks*
Viriato	Commercial, Educational	unknown	XML, PNG, etc.	Python and C# libraries	Collision Detection, API Libraries, Time Shift, Uncertainty analysis.
OpenTrack	Commercial, Educational	OpenML compatible Formats	PNG, OpenML compatible Formats	unknown	Uncertainty analysis, collision detection, data analysis stack.
Treno	Commercial	unknown	PDF, various raster formats	Java, SQL	Conflict detection, Hardware key required.
RailBase v2	Commercial	CSV, XSL(C), SQL	Printing	unknown	
ViziRail	Commercial	unknown	Printing	unknown	
Rail Movement Planner	Commercial	unknown	unknown	unknown	

Table 3.2: Summary of the six reviewed commercial tools.

- *Licence*: SMA provides its software and libraries under commercial licences as well as special educational contracts in cooperation with universities. Furthermore, SMA provides open-source implementations of different algorithms under the name #openviriato [SMA 2025a], although they need the commercial/educational libraries to work properly.
- *Import Options*: It is unknown which import formats Viriato supports. A programming interface in the form of #openviriato exists, which provides utilities for importing and exporting data.
- *Export Options*: Export is possible to XML and PNG. A programming interface in the form of #openviriato exists, which provides utilities for importing and exporting data.
- *Tech Stack*: It is not known which language Viriato is written in. A good guess mmight be C#, due to the existence of a C# API, as well as a Python API.

An exampkle of a string chart chart from Viriato can be seen in Figure 3.8.

3.2.2 OpenTrack

OpenTrack is a scientific software project created by the Institut für Verkehrsplanung und Transportsysteme of ETH Zürich for interactively simulating the train movement [OpenTrack 2025]. It has the following characteristics:

- *Owner*: OpenTrack Railway Technology GmbH.
- *Licence*: To use OpenTrack, a commercial or educational partnership has to be established.
- *Import and Export Options*: Wide support for RailML compatible formats such as XML, Excel, FPL, Viriato, and more.
- *Export Options*: String charts can be exported as PNG.

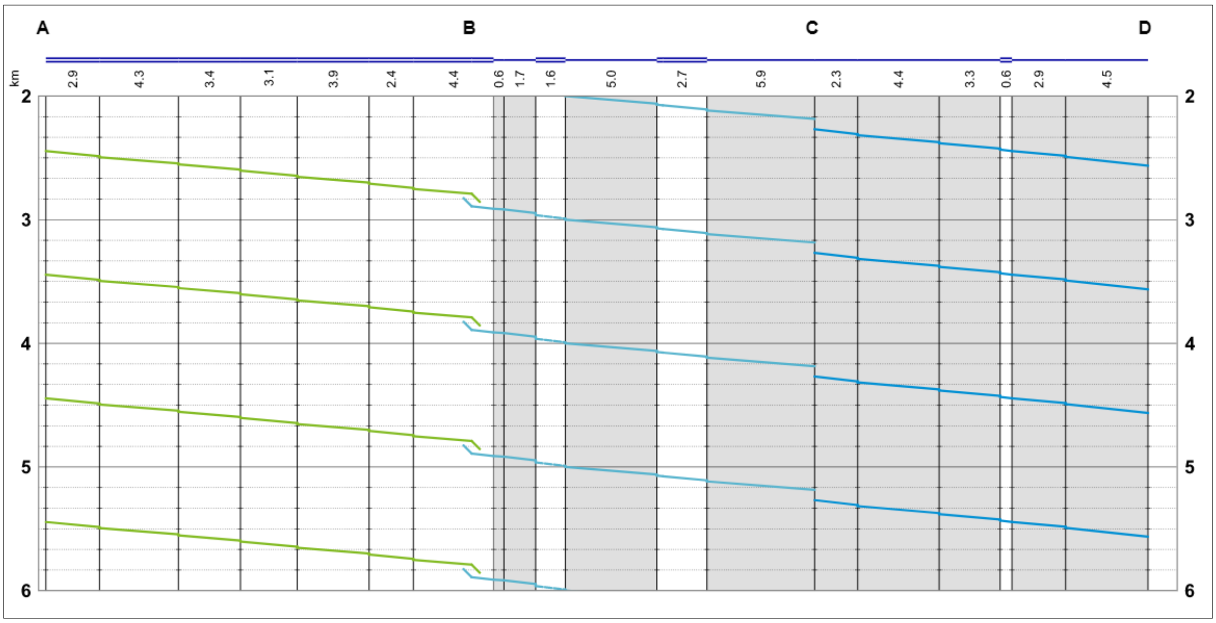


Figure 3.8: Viriato: String chart for an example with two transfers. [Viriato screenshot taken from SMA [2025a]. Used under the terms of an Apache 2.0 Licence.]

- *Tech Stack*: Unknown.

The project provides an impressive platform for many railway simulation solutions and a wide range of analysis functionality.

3.2.3 Treno

Treno is a suite of powerful timetable planning tools [trenolab 2025]. It has the following characteristics:

- *Owner*: trenolab.
- *Licence*: Commercially available software solution, which requires a hardware key to operate.
- *Import Options*: Unknown.
- *Export Options*: String charts can be exported as PDF or high resolution images in various formats.
- *Tech Stack*: The application is developed in Java and uses SQL.

3.2.4 RailBase

RailBase Scheduler is a software solution for model railroad tracking and management [Albion 2025]. It has the following characteristics:

- *Owner*: Albion Software.
- *Licence*: Commercial software.
- *Import Options*: The application supports loading data in various CSV formats and works with SQL.
- *Export Options*: String charts can be exported only by printing.
- *Tech Stack*: Unknown.

3.2.5 ViziRail

ViziRail is a graphical software tool for the planning, scheduling and monitoring of rail traffic [ViziRail 2025]. It has the following characteristics:

- *Owner*: ViziRail.
- *Licence*: Cmmercial product.
- *Import Options*: Unknown.
- *Export Options*: Graphics can be printed.
- *Tech Stack*: Unknown.

3.2.6 Rail Movement Planner

Rail Movement Planner is a real-time train scheduling and traffic planning system, built by RailMP to improve productivity and efficiency [RailMP 2025]. It has the following characteristics:

- *Owner*: RailMP.
- *Licence*: Commercial product.
- *Import and Export Options*: It is unknown which import and export formats the application supports.
- *Tech Stack*: It is unknown which programming language is used to implement the software.

Appendix A

General Transit Feed Specification (GTFS)

The General Transit Feed Specification (GTFS) is an open standard for describing public transport schedules [GTFS 2025]. It was initially developed through collaboration between Google and TriMet, the public transport agency in Portland, Oregon, with the purpose to have standardized API for schedules and geographical data. GTFS has been widely adopted by transport authorities, developers, and researchers worldwide.

GTFS consists of a structured collection of plain text files, typically packaged as a ZIP archive. Each file contains comma-separated values and represents a particular aspect of a transit system, such as agencies, stops, routes, trips, and service frequencies.

The default structure of GTFS Static has to contain at least the following text files:

- `agency.txt`
- `stops.txt`
- `routes.txt`
- `trips.txt`
- `stop_times.txt`
- `calendar.txt`

The widespread distribution of GTFS improved the accessibility and interoperability of transport data. A common format makes it possible for developers to integrate transit information into applications and services, for example in routing applications such as Google Maps. It is also used in traffic research for example in network coverage, transit needs, and multimodal connectivity [Google 2025].

An extension to the static format is GTFS Realtime, which allows transport companies to provide dynamic updates like delays, vehicle positions or deviations.

The popularity of GTFS shows the need for interoperability and connectivity through different transport agencies and enterprises and routing services and apps. Open data supports this trend toward a democratic and easy accessible solutions. In Austria, Mobilitätsverbünde Österreich provide many transport datasets in different formats, including GTFS [Mobilitätsverbünde Österreich 2025].

Bibliography

Albion [2025]. *RailBase*. 11 May 2025. http://albionsoftware.com/railbase.html (cited on page 15).

Barry, Mike and Brian Card [2014]. *Visualizing MBTA Data*. 10 Jun 2014. https://mbtaviz.github.io/ (cited on pages 7–9).

Barry, Mike and Brian Card [2023]. *MBTA Viz Repository*. 19 Jan 2023. https://github.com/mbtaviz/mbtaviz.github.io (cited on page 7).

Baumeler, Hanspeter [1899]. *Grafischer Fahrplan Gotthardbahn*. 01 Oct 1899. https://commons.wikimedia.org/wiki/File:Grafischer_Fahrplan_1899.jpg (cited on page 5).

Bostock, Mike [2021]. *Marey's Trains*. 07 Oct 2021. https://observablehq.com/@d3/mareys-trains (cited on pages 9–10).

CondensedChaos [2023a]. *Marey Chart Creator*. 20 Jul 2023. https://condensedchaos.github.io/ (cited on pages 9–10).

CondensedChaos [2023b]. *Marey Graph Generator Web App*. 20 Jul 2023. https://github.com/CondensedChaos/condensedchaos.github.io (cited on page 9).

Danel, L. [1852]. *Chemin de fer du Nord. Tableau graphique de la marche des Trains*. 01 Aug 1852. https://gallica.bnf.fr/ark:/12148/btv1b531363101 (cited on page 4).

Google [2025]. *GTFS Static Overview*. 12 May 2025. https://developers.google.com/transit/gtfs (cited on page 17).

Gsellmann, Inge, Michael Hebesberger and Danijela Lazarevic [2023]. *String Charter 2: Visual Transport Schedules*. 03 Jul 2023. https://courses.isds.tugraz.at/ivis/projects/ss2023/ivis-ss2023-g2-project-string-charter2.pdf (cited on page 11).

GTFS [2025]. *General Transit Feed Specification (GTFS)*. 12 May 2025. https://gtfs.org/ (cited on page 17).

Huber, Manuel [2024]. *FPLedit*. 03 Dec 2024. https://fahrplan.manuelhu.de/ (cited on page 11).

ISC [2025]. *ISC License*. ISC. https://isc.org/licenses/ (cited on page 9).

Lalanne, Léon [1884]. *Méthodes graphiques: Note sur un nouveau mode de représentation de la marche des trains sur une voie de communicstion*. In French. Comptes rendus hebdomadaires des séances de l'Académie des sciences (01 Jul 1884), pages 307–313. https://gallica.bnf.fr/ark:/12148/bpt6k3055h/f307 (cited on page 3).

Marey, Étienne-Jules [1878]. *La méthode graphique dans les sciences expérimentales et principalement en physiologie et en médecine*. 1878. https://gallica.bnf.fr/ark:/12148/bpt6k6211376f/ (cited on pages 3–4).

MIT [2025]. *MIT License*. MIT. https://mit-license.org/ (cited on page 7).

Mobilitätsverbünde Österreich [2025]. 12 May 2025. https://data.mobilitaetsverbuende.at/de/data-sets (cited on page 17).

OpenTrack [2025]. *OpenTrack Railway Technology*. 11 May 2025. https://opentrack.ch/ (cited on page 14).

RailMP [2025]. *Rail Movement Planner*. 11 May 2025. https://railmp.com/en/about-us/ (cited on page 16).

Scherzinger, Moritz [2022]. *jTrainGraph*. 30 Dec 2022. https://jtraingraph.de/ (cited on pages 10, 12).

SMA [2025a]. *#openviriato*. 11 May 2025. https://github.com/sma-software/openviriato.algorithm-platform.showcase.spot (cited on pages 14–15).

SMA [2025b]. *Timetable Planning with Viriato*. 11 May 2025. https://sma-partner.com/en/software/timetable-planning-with-viriato (cited on page 13).

Stellwerksim [2025]. 12 May 2025. https://stellwerksim.de/ (cited on page 11).

Tauri [2025]. *Tauri*. 11 May 2025. https://tauri.app/ (cited on page 13).

trenolab [2025]. *Treno*. 11 May 2025. https://trenolab.com/ (cited on page 15).

ViziRail [2025]. *ViziRail*. 11 May 2025. https://www.vizirail.com.au/ (cited on page 16).

Von Röll, Victor [1914]. *Gotthardbahn*. In: *Enzyklopädie des Eisenbahnwesens*. Volume 5. Enzyklopädie des Eisenbahnwesens. Urban & Schwarzenberg, 1914, pages 354–361. https://deutschestextarchiv.de/book/view/roell_eisenbahnwesen05_1914?p=364 (cited on page 3).

W3C [2016]. *Extensible Markup Language (XML)*. World-Wide Web Consortium, 11 Oct 2016. https://w3.org/XML/ (cited on page 7).

W3C [2025]. *Scalable Vector Graphics (SVG)*. World-Wide Web Consortium. https://w3.org/Graphics/SVG/ (cited on page 7).

Wainer, Howard, Polina Harik and John Neter [2013]. *Visual Revelations: Stigler's Law of Eponymy and Marey's Train Schedule: Did Serjev Do It Before Ibry, and What About Jules Petiet?* CHANCE 26 (06 Feb 2013), pages 53–56. doi:10.1080/09332480.2013.772394 (cited on page 3).