Deliverable 4.2

Open Repository of Demand and Supply Models and Algorithms for Emerging Mobility Solutions

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 815069.

Table of Contents

Summary sheet

Deliverable No.	4.2
Project Acronym	Momentum
Full Title	Modelling Emerging Solutions for Urban Mobility
Grant Agreement No.	815069
Responsible Author(s)	Santhanakrishnan Narayanan (TUM) Constantinos Antoniou (TUM) Athina Tympakianaki (Aimsun) Jordi Casas (Aimsun) Panagiotis Tzenos (CERTH) Josep Maria Salanova Grau (CERTH) Rodric Frederix (TML) Willem Himpe (TML) Kris Vanherle (TML) Joren Vanherck (TML) Ignacio Martín (Nommon Javier Burrieza (Nommon) Antonio D. Masegosa (UDEUSTO) Jenny Fajardo Calderín (UDEUSTO) Pablo Fernández Muga (UDEUSTO)
Peer Review	Tamara Djukic (Aimsun), Maria Chatziathanasiou (CERTH), Oliva García (Nommon),
Quality Assurance Committee Review	General Assembly
Date	29/04/2021
Status	Draft
Dissemination level	Public
Abstract	The objective of the present document is to describe the MOMENTUM open repository, which contains the models and algorithms for emerging mobility solutions. The open repository is hosted in the GitHub web platform. This

	document provides an overview of the structure of the repository and a summary of the model codes. Furthermore, it includes a brief guide to using GitHub, focusing on downloading and viewing the data. Thus, Deliverable 4.2 from MOMENTUM covers two contributions: 1) the open repository established in GitHub platform and 2) this complementary report.
Version	Issue 1 Draft 2
Work Package No.	4
Work Package Title	Modelling of Emerging Mobility Solutions
Programme	Horizon 2020
Coordinator	EMT Madrid
Website	www.h2020-momentum.eu
Starting date	01/05/2019
Number of months	36

Project partners

Organisation	Country	Abbreviation
EMPRESA MUNICIPAL DE TRANSPORTES DE MADRID SA	Spain	EMT
NOMMON SOLUTIONS AND TECHNOLOGIES SL	Spain	NOMMON
DIMOS THESSALONIKIS	Greece	THESS
ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS	Greece	CERTH
STAD LEUVEN	Belgium	LEUVEN
TRANSPORT & MOBILITY LEUVEN NV	Belgium	TML
STADT REGENSBURG	Germany	REGENSBURG
TECHNISCHE UNIVERSITAET MUENCHEN	Germany	TUM
AIMSUN SL	Spain	AIMSUN SL
POLIS – PROMOTION OF OPERATIONAL LINKS WITH INTEGRATED SERVICES, ASOCIATION INTERNATIONALE	Belgium	POLIS
UNION INTERNATIONALE DES TRANSPORTS PUBLICS	Belgium	UITP
UNIVERSIDAD DE LA IGLESIA DE DEUSTO ENTIDAD RELIGIOSA	Spain	UDEUSTO

Document history

Version	Date	Organisation	Main area of changes	Comments
Issue 1 Draft 1	29/04/2021	TUM	Initial version	
Issue 1 Draft 2	17/05/2021	TUM	All sections	Consortium internal review

Executive Summary

The overall goal of the MOMENTUM project is to develop a set of mobility data analysis and exploitation methods, transport models and planning and decision support tools able to capture the impact of new transport options and ICT-driven behavioural changes on urban mobility, to support local authorities in the task of designing the right policy mix to exploit the full potential of emerging mobility solutions.

The objective of the present document is to **describe the MOMENTUM open repository**, which contains the models and algorithms for emerging mobility solutions. The open repository is hosted in the GitHub web platform due to its capabilities to easily retrieve a project and track previous versions, along with the ability for multiple people to work simultaneously on the same component.

The present document provides an **overview of the structure of the repository and a summary of the model codes**. Furthermore, it includes a **brief guide to using GitHub**, focusing on downloading and viewing the data. Therefore, this report supports those who are interested in accessing the MOMENTUM models and algorithms, available at **https://github.com/h2020-momentum**.

1. Introduction

1.1 Scope and objective

The overall goal of the MOMENTUM project is to develop a set of mobility data analysis and exploitation methods, transport models, and planning and decision support tools, which can capture the impact of emerging transport modes (such as shared mobility services) and ICT-driven behavioural changes on urban mobility, to support local authorities in the task of designing the right policy mix, to exploit the full potential of emerging mobility solutions.

To achieve this general goal, one of the objectives set is to develop new modelling approaches that can ascertain the impact of emerging mobility concepts and solutions. This means that changes required to incorporate emerging mobility solutions into strategic transportation models, both in terms of supply and demand, had to be formulated, and explanatory and predictive models had to be constructed. Furthermore, methods to include inherent uncertainty in the developed models have to be designed. As a result, a set of demand and supply models and algorithms for modelling different aspects of the emerging mobility solutions have been developed and reported in Deliverable 4.1.

The models developed and reported in Deliverable 4.1 are stored in an open repository. This repository represents the main contribution of Deliverable 4.2, which consists of two parts: 1) **MOMENTUM open repository established in GitHub platform** and 2) this **complementary report containing an overview of the structure of the repository, a summary of the model codes and a brief guide to using GitHub**, focussing on downloading and viewing the models and algorithms.

1.2 Structure of the document

The remainder of this document is structured as follows:

- **Section 2** describes the repository and its structure.
- **Section 3** summarises the codes of the models.
- **Section 4** provides a guideline to using the repository.

1.3 Applicable documents

[I] MOMENTUM D4.1 Transport Modelling Approaches for Emerging Mobility Solutions, May 2021

2. Repository and its structure

The Open-source codes from the EU H2020 project MOMENTUM are made available in a GitHub repository at https://github.com/h2020-momentum. As explained in Deliverable 4.1 of the MOMENTUM project, the development of the models are based on a modular approach. The GitHub repository replicates this approach, and it is divided into four main *categories*: *Demand, Fleet management, Supply,* and *Sustainability*. Each of these categories is divided into *modules*, which contain the files for one or several models. The model scripts are developed in different programming languages, e.g., MATLAB, R and Python. More details on this are provided in Section 3. Besides the summary in Section 3, a readme file is provided for each model in the repository, in the respective folders. Technical details about the models are available in *Deliverable D4.1*. An overview of the structure of repository is shown in Figure 1.

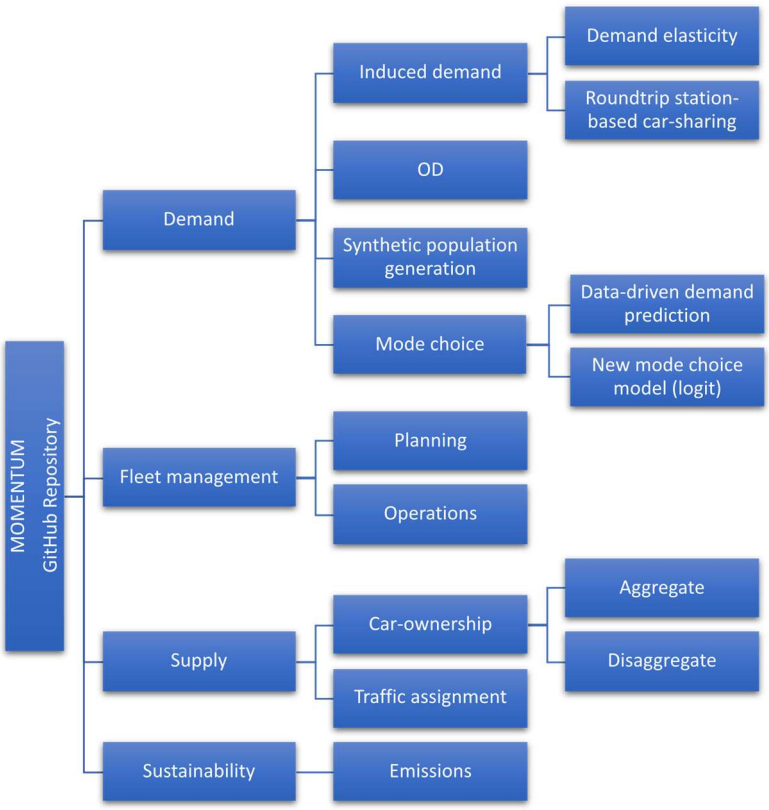

Figure 1 Structure of the MOMENTUM GitHub repository

The category *Demand* is divided into four modules, namely *Induced Demand*, *Mode choice*, *OD* and *Synthetic population generation.* Furthermore, the module *Induced demand* is subdivided into two elements: *Demand elasticity* and *Roundtrip station-based car-sharing; and t*he module *Mode choice* contains *Data-driven demand prediction* and *New mode choice models.* The category *Fleet management* is divided into *Planning* and *Operations* modules. The category *Supply* consists of *Car-ownership* and *Traffic assignment* modules. The former is subdivided into *Aggregate* and *Disaggregate* models. Finally, the category *Sustainability* contains the module *Emission*.

3. Summary of the model codes

This section provides a summary of the different models and algorithms for emerging mobility solutions, which are uploaded to the repository. The models are presented, following the structure shown in Section 2. For every model, its objective, programming language, input and output file types are presented. Furthermore, as some models might not be available in the public repository because of commercial reasons, a column indicating its availability in the repository is included. Finally, the column Upload *status* indicates whether the models are already available in the repository.

3.1 Demand

Model	Objective	Programming language	Input type	Output type	Availability	Upload status
Induced Demand module						
Demand elasticity	Calculation of demand elasticity	MATLAB	mat	mat	Yes	Yes
Demand for roundtrip station based carsharing	Estimation of demand (irregular and infrequent trips) for station-based round-trip sharing systems	R	CSV	CSV	Yes	Yes
Mode choice module						
Data driven demand prediction	Prediction of the number of trips captured by a shared mobility service	Python	CSV	CSV	Yes	Yes
New mode choice model	Estimation of modal split between conventional systems-as-a-whole, bike-sharing, car-sharing & ride-sharing	R	CSV	CSV	Yes	Yes
OD module						

OD matrix type classification	OD matrix classification and selection of the best matching matrix for the specific characteristics of the day or season of study	Python	CSV	CSV	Yes	Yes
OD matrix clustering	OD matrix clustering based on structural similarity measures and graph embedding techniques	R	CSV	CSV	Yes	Yes
Synthetic population generation module						
Synthetic population generation	Generation of a disaggregate representation of the actual population based on socio-demographic & other pertinent information[1].	Python	CSV, SHP	CSV	Yes	Yes (Pseudo code)

3.2 Fleet management

Model	Objective	Programming language	Input type	Output type	Availability	Upload status
Planning module						
Fleet management - planning	Plan demand responsive transport, car-sharing, ride-sharing, bike-sharing and micro-mobility systems	Python/C++	SHP, GTFS, JSON, External	JSON	Yes	Partial

[1] PopGen open source package (https://www.mobilityanalytics.org/popgen.html) is utilised as the basis. There may be relevant attributes (especially for the new mobility services) that are not available in both the census and the travel survey data. Hence, it would not be possible for PopGen to map them in the initial population synthesis. In such cases, statistical procedures will be used to enrich the synthetic population with additional attributes. This data-driven approach that enriches the synthetic population generation process is the one that will be hosted in the repository.

Operations Module						
Fleet controller	Algorithms for managing the fleet of shared vehicles	Python	CSV, JSON, XLSX	JSON	Yes	Partial
Dynamic Demand Simulation platform	Framework to simulate DRT, car-sharing, ride-sharing, bike-sharing and micro-mobility systems	C++	JSON	JSON	No (part of commercial software, Aimsun)	NA

3.3 Supply

Model	Objective	Programming language	Input type	Output type	Availability	Upload status
Car-ownership module						
Aggregate car-ownership model	Estimation of car-ownership at the level of traffic zones	R	CSV	CSV	Yes	Yes
Disaggregate car ownership model	Estimation of car-ownership at the household level	R	CSV	CSV	Yes	Yes
Traffic assignment module						
Traffic assignment method for urban environments	Data-driven method that considers traffic conditions in local roads, to circumvent problems that arise in classical assignment procedures due to centroid connector placement	MATLAB	mat	mat	Yes	Yes

Hybrid dynamic traffic assignment	Integration of mesoscopic and macroscopic models	C++	Dynamic scenario definition – internal format[2]	DB/CSV	No (part of commercial software, Aimsun Next)	NA

3.4 Sustainability

Model	Objective	Programming language	Input type	Output type	Availability	Upload status
Emission module						
Static emission model	Calculation of emissions based on a simplified COPERT-based EMEP approach	MATLAB or Python	CSV	CSV	Yes	Yes
Dynamic emission model	Calculation of emissions using dynamic speed profiles from a dynamic traffic assignment model	Python, C++	CSV	DB/ Exporting capability of the UI	No (part of commercial software, Aimsun Next)	NA

[2] The dynamic scenario definition in Aimsun Next has, as minimum input requirement, a base transport network and a traffic demand. Public transport plans as well as traffic signal plans can be also included.

4. Guide to Using the repository

This section provides a short guide to using GitHub, focusing on downloading and viewing the models and algorithms.

4.1 Overview and Concept

GitHub is a web platform that focuses on using the versioning tool called Git. It allows a team to easily retrieve a project, see previous versions and allows multiple people to work simultaneously on the same component.

A Git repository contains the project and assumes that it does not have any external dependencies. A repository then has "branches", which can be seen as personal repositories containing the edits of individual contributors. The most important branch is the "main" branch, which always contains the most updated working version of the project. On each branch, and especially the main branch, "commits" can be found, which are updates made by contributors.

4.2 Navigating Repository in a Web Browser

All the files in the directory can be easily accessed in a web browser. It works similarly to a normal file browser, where directories and files are directly shown when clicking on them. At any time, a file can be located in the repository by checking its folder path (see Figure 2).

Figure 2 Location of a file in the repository

4.3 Download an Individual File from Web Browser

To download an individual file, navigate to it in the web browser and then open it. In the top right corner, click "Raw" (see Step 1 in Figure 3). This opens the raw file, which can then be saved by right-clicking, choosing "Save as..." and then selecting the desired location in the computer (see Steps 2 to 4 in Figures 4 and 5). When saving the file, it is important to indicate the appropriate format based on the type of file (*e.g.,* .R, for R scripts; .m, for MATLAB scripts; etc. See Steps 5 and 6 in Figure 5).

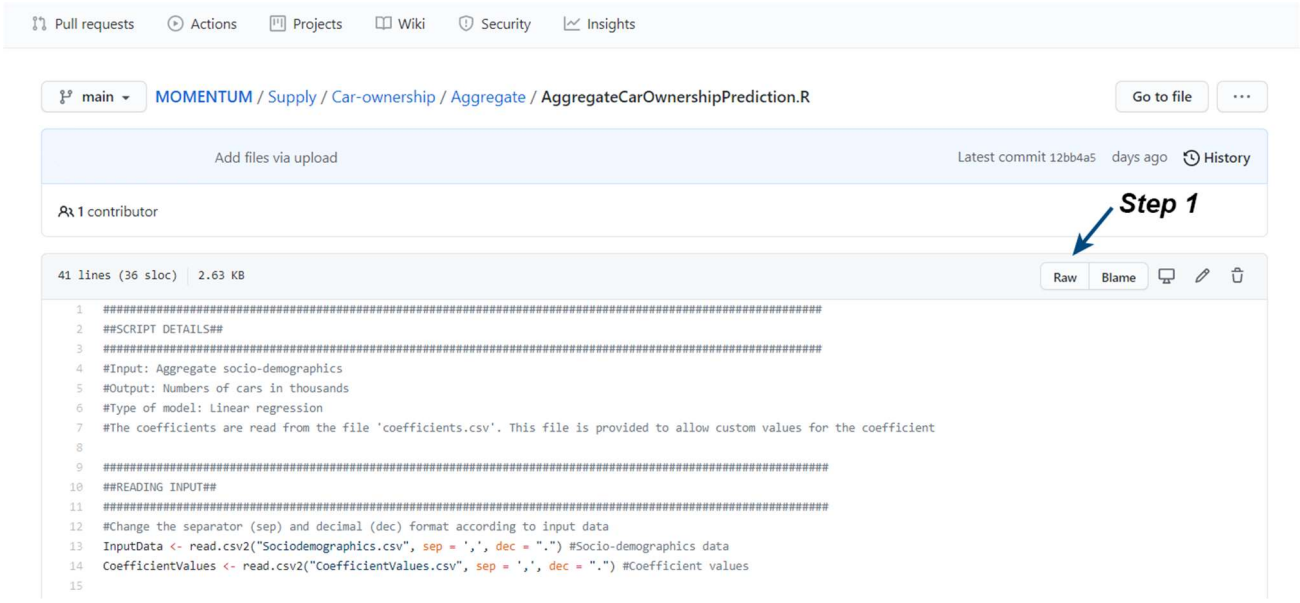

Figure 3 Opening raw file in web browser

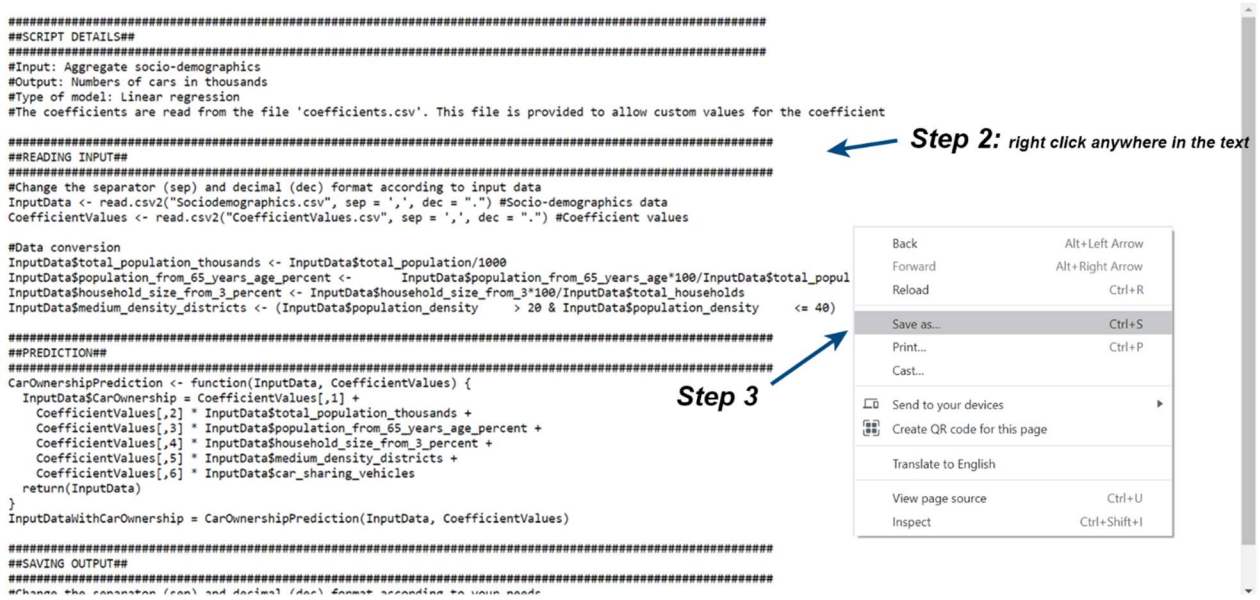

Figure 4 Saving raw file in the computer

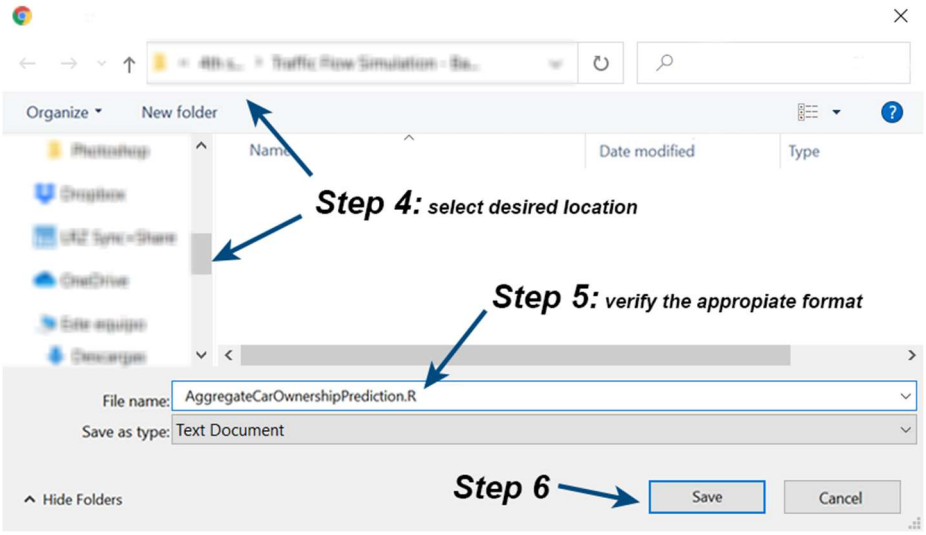

Figure 5 Selecting desired location in the computer and verifying file format

4.4 Quick Start to Download Entire Repository

The easiest way to download the entire repository is to click on the "Code" button, which creates a dropdown menu (see Step 1 in Figure 6). Then click "Download ZIP" to download a Zip file containing the entire repository (see Step 2 in Figure 6).

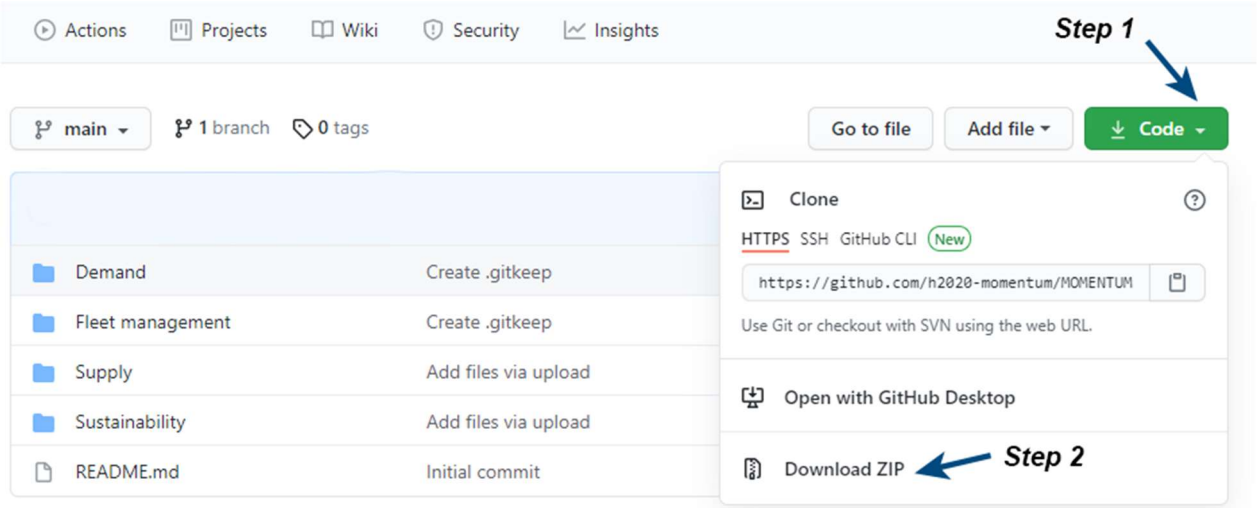

Figure 6 Steps to download the entire MOMENTUM GitHub repository

4.5 View Changes of Repository in Web Browser

Navigate to the repository and then click the "<N> commits" button in the top right corner, commits, with <N> being a number. This opens an overview of all changes over the entire period of the repository. When clicking on a commit, the changes made to all the files in that single update are shown. New lines in files are shown in green

and deletions in red. A copy of the entire repository at the time of that commit can be accessed by clicking on the "Browse files" button in the right side of the screen (see Figure 7).

Figure 7 Accessing the repository at the time of a certain commit

4.6 Navigate Repository in GitHub App

More advanced functions are available by opening the repository in the GitHub app. First, download "GitHub Desktop" at https://desktop.github.com/. Then navigate to the repository in the web browser, click "Code" and then "Open with GitHub Desktop" (see Steps 1 and 2 in Figure 8). This will automatically open the app on the computer, where a folder in the local system must be specified (see Figure 9). Then click "Clone" and wait for the download to finish (see Step 3 in Figure 9). This will create a local copy of the MOMENTUM repository on the computer and sync between the two locations.

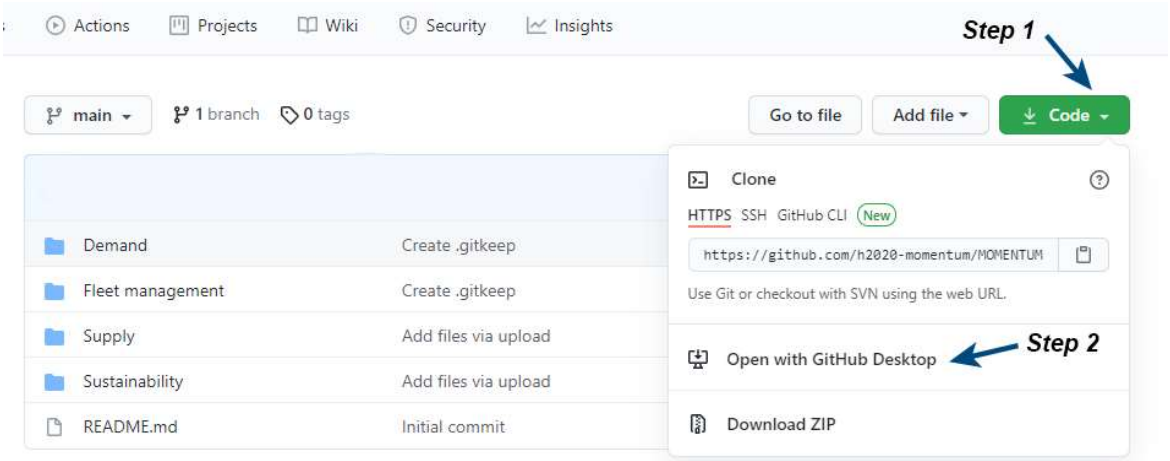

Figure 8 Opening repository in GitHub Desktop

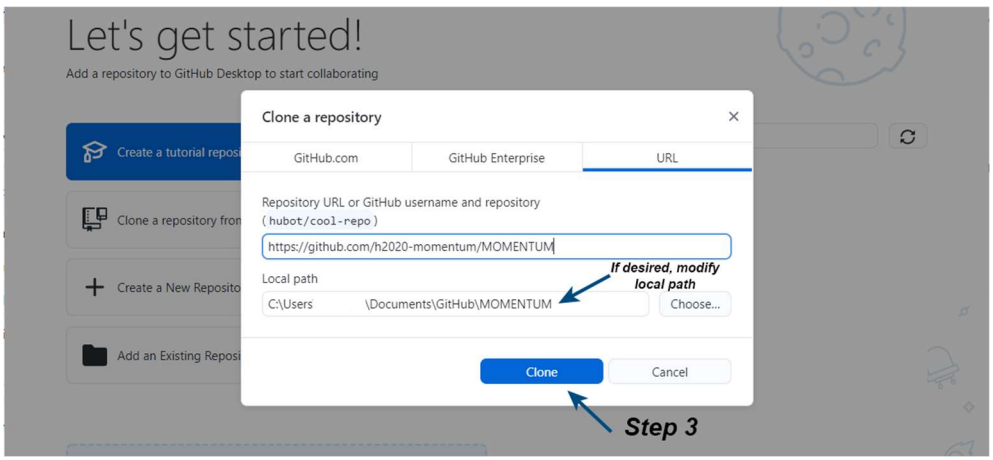

Figure 9 Cloning MOMENTUM repository into the computer using GitHub Desktop

Now the files can be easily navigated locally using the file explorer or in an external editor (*e.g.*, *RStudio, MATLAB, etc.*). By using the Desktop version, changes and uploads can be made by "committing" and then "pushing" (see Figure 10). Changes in the repository made by other contributors can be also checked using GitHub Desktop (see Figure 11). Further details on uploading changes can be found at https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/getting-started-with-github-desktop.

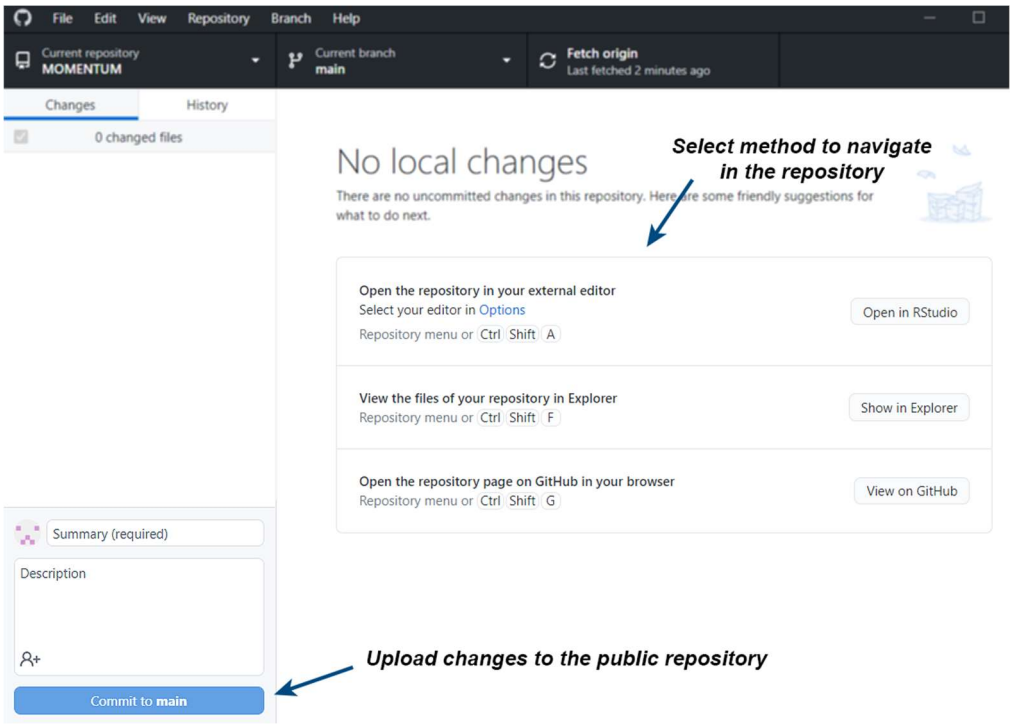

Figure 10 Main menu of GitHub Desktop

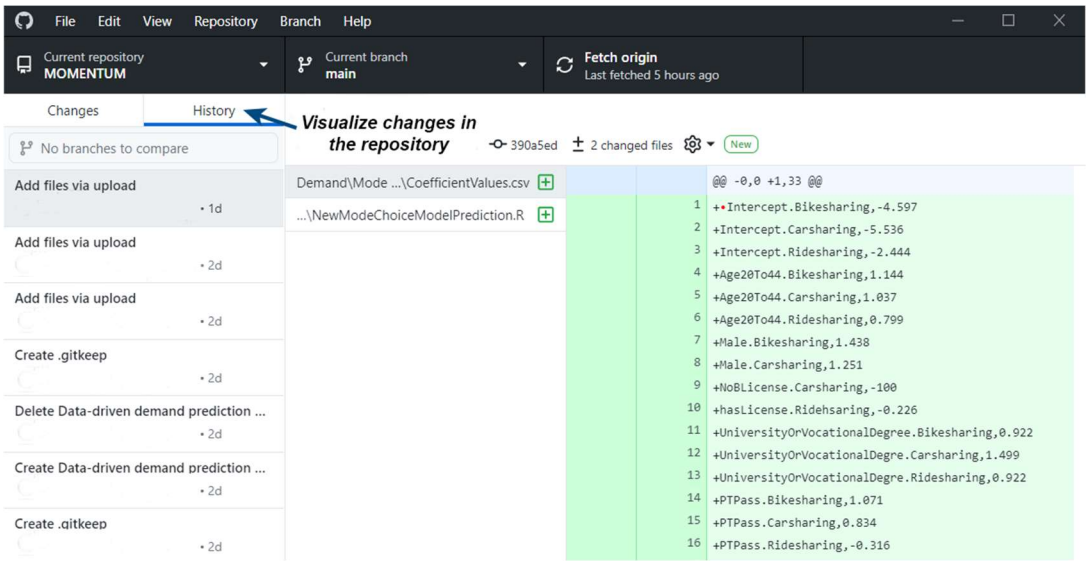

Figure 11 Checking changes in the repository using GitHub Desktop