BedloadR

https://www.bedloadweb.com/

Programmer's manual

DOI: 10.13140/RG

INRAE

OFB
OFFICE FRANÇAIS DE LA BIODIVERSITÉ

CONTENTS

1 INTRODUCTION

River managers often need to compute the river functioning, which generally requires several steps including the hydrology, the channel hydraulics, and finally the associated bedload transport, and many numerical tools are available online. These models are more or less complex, but most of them are not accessible to non-specialists in hydrology or in hydraulics. Meanwhile, many practitioners must have access to a (even rough) bedload estimate, and instead of getting involved in long and tedious trainings (by lack of time, energy or money), they prefer to implement by themselves some usual simple equations. This poses several problems: doing so it is a large source of error (these equations simple in appearance can hide some subtle details for non-specialist) and it is not optimum for large scale computation. The Bedload-*R* package was developed for covering this gap between complex models and simple hand calculation. This user friendly program not only supports users in all the successive steps required for a complete and serious sediment budget study (entering the data, choose between the panels of existing equations, edit the results and the graphics), it also permits to manipulate a large bedload database for educational purpose.

The Bedload-*R* program is already available online for a direct use at https://en.Bedloadweb.com/. This free access is comfortable for users because it does not require installing and configuring any program. However, a web access may also be limited when the web connection is of poor quality, and it reduces the high possibilities offered by the R environment to extend the existing functions to the user's own specific questions. This is why we thought it important to give access to the program in an open source way.

This document presents the Bedload-*R* package, the program algorithm and main functions and variables.

BedloadR can be installed and used by anyone, even without any knowledge in computer science. However for taking advantage of the full potentiality of the package, a basic knowledge of the R language is required (a lot of tutorials are available online).

2 OVERVIEW

2.1 Install

2.1.1 R-Shiny

The program was written in R and use Shiny (https://shiny.rstudio.com/) for the user interface.

RStudio must be installed https://rstudio.com/products/rstudio/.

Several libraries must be installed: **Shiny, knitr, DT, rhandsontable, shinyBS, shinyjs, stringr, zoo, shinycssloaders, fields**.

For instance the R command for installing the Shiny package is :

```
> install.packages("shiny")
```

2.1.2 The BedloadR package

The BedloadR package can be downloaded from https://github.com/inrae/BedloadR.git

It contains: 2 R files and 4 folders:

Nom	Modifié le	Type	Taille
data	26/03/2020 11:14	Dossier de fichiers	
user_db	26/03/2020 11:14	Dossier de fichiers	
user_projects	26/03/2020 11:16	Dossier de fichiers	
www	26/03/2020 11:14	Dossier de fichiers	
app	26/03/2020 11:17	Fichier R	607 Ko
bedloadR	25/03/2020 21:16	Fichier R	260 Ko

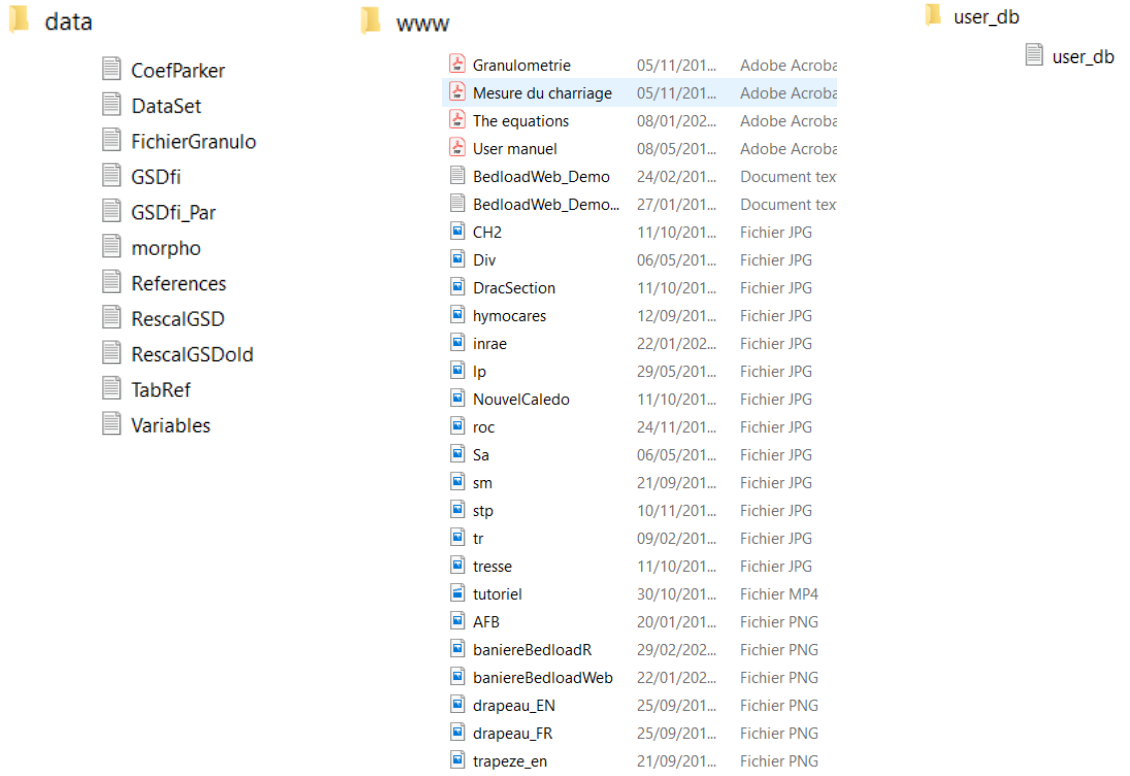
The **app.R** file contains the definition of the user interface (ui part) objects and the functions (server part) controlling these objects. The **BedloadR.R** file contains all functions needed for calculations.

The folder **/user_projects** is empty and will contain the user data backup (can be changed).

The folder **/user_db** contains the user_db.txt file where the program will record all users ID and password.

The folder /**www** contains all images used by the program (not the users' photos), and the files that can be downloaded from the help page: pdf documents and the example.txt file
The folder /**data** contains all data needed for the program to work:

- **CoefParker.txt** contains parameters used in the Parker90 bedload equation
- **dataset.txt** contains the bedload datasets
- **FichierGranulo.txt** contains all rivers' grain size distribution (dataset)
- **GSDfi.txt** is a post treatment of FichierGranulo.txt for fractional bedload calculation with some equations (Wilcock and Crowe, Parker 90)
- **GSDfi_Par.txt** is a variant of GSDfi.txt without the sand fraction (option Parker 90)
- **morpho.txt** contains information about the data set rivers' morphology
- **TabRef.txt** contains the bedload dataset metadata (average values)
- **References.txt** contains the data sets bibliography
- **Variables.txt** contains the label of each variable in both language, French and English.



The txt files are **tab-delimited** text files.

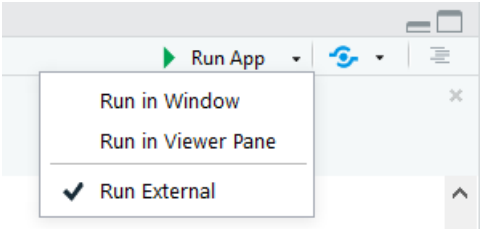
2.1.3 Get starting

Copy files app.R, BedloadR.R and folders /www, /data and /user_db in the working directory (for instance *C:/BedloadR/*). Open app.R in R studio and modify the three first lines:

```
8  ##############################################################################################################
9  L<-"E"#Choose"F" for the french version and 'E' for the english version
10 setwd("C:/BedloadR/") # Directory where is unstalled the package (can be changed)
11 rep<-"C:/BedloadR/user_projects"# Directory for users data back-up (can be changed)
12 ##############################################################################################################
```

You are free to choose the directories for 'setwd' and 'rep', but make sure these directories exist before launching the program.

The program can be launched from R Studio by clicking on the button RunApp (option 'run external') :



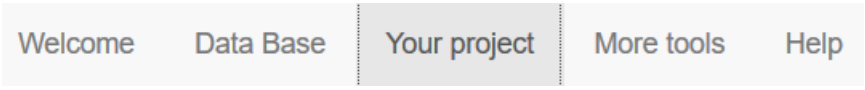
or with commands:

```
> library(shiny)
```

```
> runAPP("C:/BedloadR/")# Directory where is installed the BedloadR package
```

2.1.4 Menu

The main page presents 4 main tabs:



The "data base" page gives access to a large data set comprising more than 11000 values collected in the field or in the laboratory. Different entries permit to select a sub-data set corresponding to finite criteria (for instance corresponding to a given name or with slope in a given range). The data are displayed graphically with the equations selected by the user. It permits to check the ability of each equation to reproduce the data selection, but also to test the equation sensitivity to the input values (uncertainties). This comparison is educational and allows familiarizing neophytes with what can be expected from the calculation.

The "Your project" page gives access to 8 entries. The first entry 'Your project' is very important because it permits to create an account, which is required for saving and managing all the entries a user will do through his different sediment transport projects. The backup can be done locally (computer) or on the server if the online version is used. The account backup facilitates the projects management through a dropdown menu, and particularly the web account offers a big advantage: it permits to share the sediment projects with other users. For those who don't want to create an account, they have the possibility to create a simple text backup of their entries locally, in the computer (but this option does not save the photos). Note that the txt backup must be saved with ANSI encoding for local use (Bedload-*R*) and UFT8 encoding for web use (Bedload*Web*).

After the welcome page, different sub-tab are proposed each of which representing an essential step for a serious study. The tabs are presented in a logical order, each producing useful data to the following ones:

- the 'Granulometry' tab permits to define the granulometry of the sediments,

- the 'Cross Section' tab allows to define (topography) and to dress (roughness ..) the flow section,

- the 'Hydraulic' tab allows to calculate the main hydraulics quantities,

- the 'sediment transport' tab calculates (with the selected bedload equations) the sediment rating curve corresponding to the granulometry, the section and hydraulics previously defined,

- the 'Hydrology' tab permits to define an hydrology,

- the 'sediment budget' tab combines the sediment rating curve with the hydrology for computing sediment volumes,

- the 'Analysis' tab displays a synthesis of all calculations (for the different river cross sections and selected equations)

The "More Tools" page is the page dedicated to (your?) new developments.

Finally, the "Help" tab gives access to an example file as well as a series of documentation including a user manual.

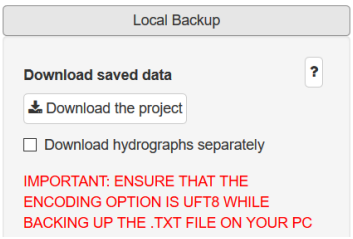
2.1.5 How to get started with Bedload-R?

The first step consists in downloading the user manual.pdf in the Help tab. Then, with the help of this manual, the best way to get familiar with the program is to spend time playing with the data and the equations. This step just consists in moving sliders and to activate dropdown

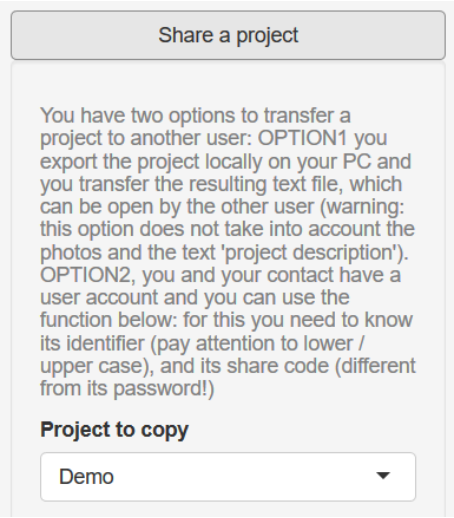
buttons, without entering any data. In a second time, for getting used with the 'Your project' menu, it is recommended to download the Example.txt file (available in the help tab) and save it in the computer; then in the "Your project" tab menu, upload the Example.txt file with the "Download the project" button in the "Local backup" panel. The program will read the Example.txt and display all data in each tab. The user can then play with different functions proposed in the menu.

2.1.6 BedloadR versus BedloadWeb

The programs are exactly the same. It is possible to transfer data created in one version to the other by exporting and importing a txt file of the project.



The transfer does not account for the photos and for the project description in page 'Your Project'. Be careful when saving the txt file because the website requires UFT8 format whereas the PC requires ANSI format. If these formats are not respected the program can stop, or some characters main not be displayed correctly.



Using the Web version presents several advantages: a project can be opened by several peoples when connecting to the same account. A project can also be copied to another user account (including photos), using the 'share a project' option. When using the Web version it is good to creates regularly a txt backup which can be opened at any time locally with the BedloadR program.

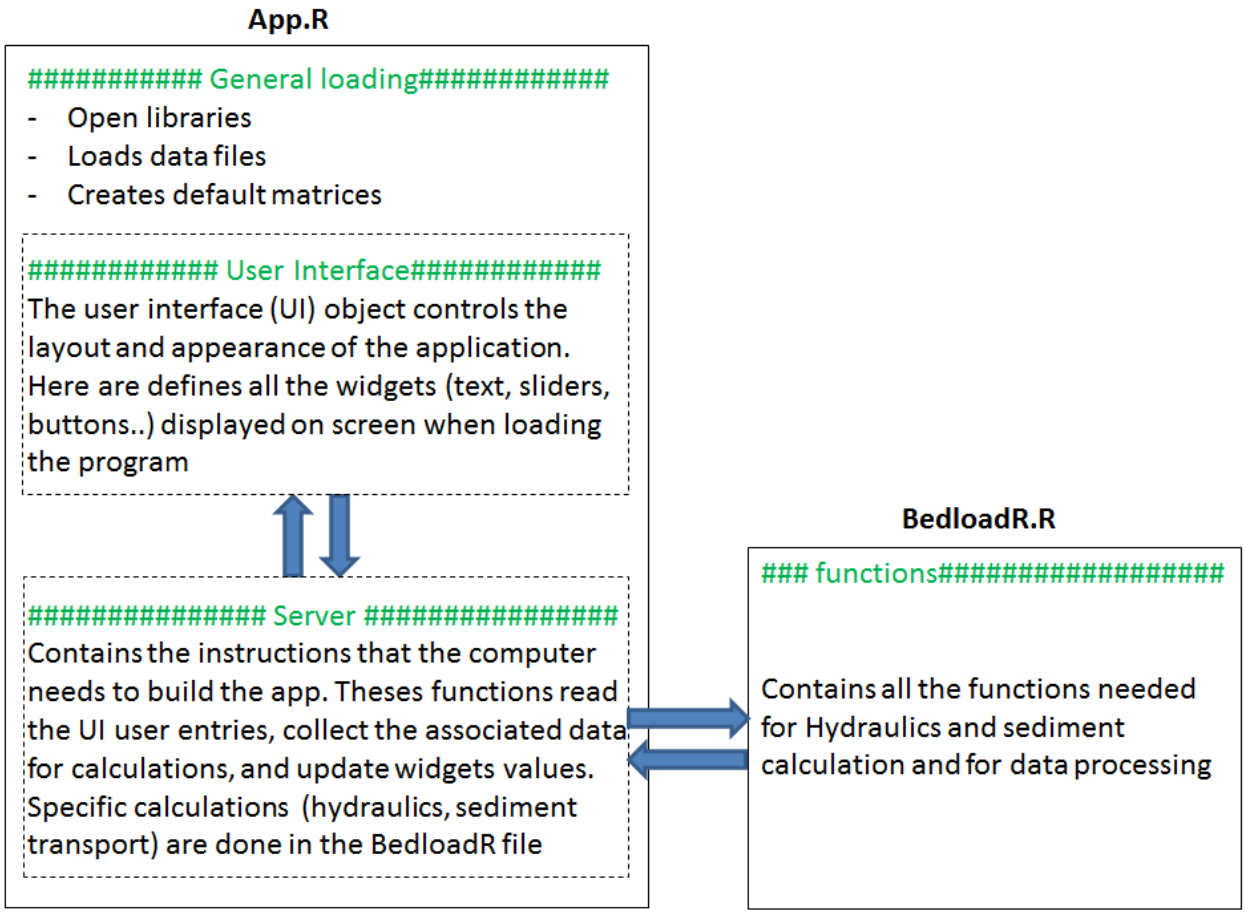
The R version permits to modify the program for specific objectives.

3 PROGRAM STRUCTURE

3.1Overview

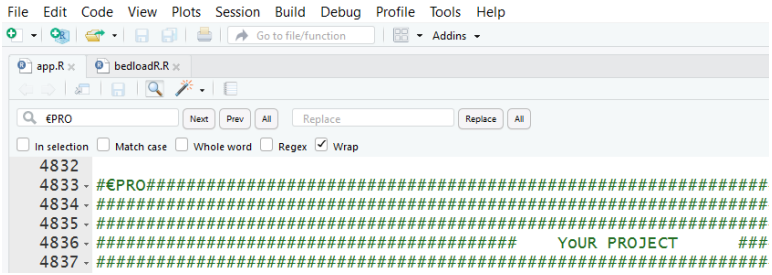
3.1.1 Program architecture (UI, Server and functions)

The **app.R** file contains the definition of the user interface (ui) objects and the server functions controlling these objects. The app.R file communicates with the **BedloadR.R** file comprising more than 190 R functions for hydraulic and bedload calculation, for data processing, or for graphics construction.



The App.R file comprises almost 15000 lines and the BedloadR.R file comprises almost 6500 lines.

Entering the following keywords in the 'Find' function of the R menu gives an easy access to the different part of the program.



Keyword	***Gives access to***
€GL	**General loading (top of App file)**
€UI	**User Interface**
€UIDB	User Interface Data Base
€UI1RIV	User Interface Data Base 1 River
€UIMULTI	User Interface Data Base Multicriteria selection
€UITB	User Interface 'Your Project'
€UIPRO	User Interface 'Your Project' Manage
€UIGSD	User Interface 'Your Project' Granulometry
€UISEC	User Interface 'Your Project' Section
€UIHYD	User Interface 'Your Project' Hydraulics
€UIQS	User Interface 'Your Project' Sediment transport
€UIHDL	User Interface 'Your Project' Hydrology
€UIBIL	User Interface 'Your Project' Sediment Budget
€UIANA	User Interface 'Your Project' Analysis
€UITB	User Interface 'More Tools'
€UIUNC	User Interface 'More Tools' Uncertainties
€UISUS	User Interface 'More Tools' Suspension
€UIDOC	User Interface Help
€SERVER	**SERVER** (same codes than above without 'UI')
€DB	Server Data Base
€TB	Server 'Your Project'
€PRO	Server Manage
€GSD	Server Grain Size Distribution
€SEC	Server Cross section
€HYDR	Server Hydraulics
€QS	Server Sediment Transport
€HDL	Server Hydrology
€BIL	Sever Sediment Budget
€ANA	Server Analysis
€TB	Server More Tools…

3.1.2 Libraries, Data Sets, Default matrices

When launching App.R, the program first loads libraries:

```
# Load packages ----
library(shiny)
library(knitr)#A General-Purpose Package for Dynamic Report Generation in R
library(DT)#An R interface to the DataTables library
library(rhandsontable)# data grid component with an Excel-like appearance
library(shinyBS)# update boutons color
library(shinyjs)# desactivate button
library(stringr)#work on string characters
library(zoo)#Infrastructure for Regular and Irregular Time Series (Z's Ordered Observations); interpolate missing values
library(shinycssloaders)#display icon when loading
```

In a second step, it reads all data files in the **/data** folder and performs a post treatment of these data by checking their integrity and computing some hydraulics parameters (shear stress, Shields stress, flow power….):

```
### LOAD DATA from directory ./data
DataSet<-read.delim("data/DataSet.txt", header=TRUE, sep="\t",dec=".",stringsAsFactors = F)
morpho<-read.delim("data/morpho.txt", header=TRUE, sep="\t",dec=".",stringsAsFactors = F)
Granulo<-read.delim("data/FichierGranulo.txt", header=TRUE, sep="\t",dec=".",stringsAsFactors = F)
TabRef<-read.delim("data/TabRef.txt", header=TRUE, sep="\t",dec=".",stringsAsFactors = F)
References<-read.delim("data/References.txt", header=TRUE, sep="\t",dec=".",stringsAsFactors = F)
CoefParker<-read.delim("data/CoefParker.txt", header=TRUE, sep="\t",dec=".",stringsAsFactors = F)
VAR<-read.delim("data/Variables.txt", header=TRUE, sep="\t",dec=".",stringsAsFactors = F)
GSDfi<-read.delim("data/GSDfi.txt", header=TRUE, sep="\t",dec=".",stringsAsFactors = F)
GSDfi_Par<-read.delim("data/GSDfi_Par.txt", header=TRUE, sep="\t",dec=".",stringsAsFactors = F)
if (L=="F") VAR<-VAR[,c(1,2)] else VAR<-VAR[,c(1,3)]##Choose the language
```

Then the program creates the 'Rescal' Matrix which contains all bedload equations (and associated parameters) and will play a central role for calculations in both the Data Base and the Your Project Tab.

```
####### DEFINITION OF MATRIX RESCAL FOR CALCULATIONS #########
### Matrix rescal is central for all hydraulics and bedload calculation
### when used with the dataset each line in Rescal corresponds to one line of the data set
### when used with Toolsbox each line corresponds to the flow depth discretization over the section from zero to Hmax
Rescal<-matrix(NA,ncol=106,nrow=600)
Rescal<-data.frame(Rescal)
names(Rescal)<-c("name","W","S","D50","D84","q","qsmeas",#C1_7
                 "qstar","p","d","R","U","f","taux","t50","t84",#C8_16
                 "t50MPM","ScorRick","t50Wilc",#C17_19
                 "tm","qscalReck","borneinfReck","bornesupReck",#Recking C20_23
                 "qscalMPM","borneinfMPM","bornesupMPM",#Meyer Peter & Muller#C24_26
                 "qcRick","qscalRick","borneinfRick","bornesupRick",#Rickenmann#C27_30
                 "qscalScho","borneinfScho","bornesupScho",#Schoklich#C31_33
                 "qscalEng","borneinfEng","bornesupEng",#Engelund#C34_36
                 "qscalPar","borneinfPar","bornesupPar",#parker#C37_39
                 "Eins","qscalEin","borneinfEin","bornesupEin",#Einstein#C40_43
                 "qscalBag","borneinfBag","bornesupBag",#Bagnold#C44_46
                 "Z","UuSmart","qscalSma","borneinfSma","bornesupSma",#Smart & Jaeggi#C47_51
                 "ustarVR","ParamT","qscalVR","borneinfVR","bornesupVR",#Van Rijn#C52_56
                 "ustar","t","qscalWilc","borneinfWilc","bornesupWilc",#Wilcock#C57_61
                 "tsg","phisg0","ParamOmega","sigma","sigma0","omega0","qscalPar90","borneinfPar90","bornesupPar90",#C62_70
                 "Abscisse","qcScho","omega","omegac",#C71_74
                  "hr","qr",#C74_76
                 "D16","Dm","GrLef","CLef","nLef","mLef","q0Lef","qstarLef","kskrLef","corLef","ZLef","MLef","FLef","CpLef",
                 "DstarCL","tcCL","qscalCL","borneinfCL","bornesupCL",#C94_98
                 "qscalWONG","borneinfWONG","bornesupWONG",#C99_101
                 "Fs","D16","D30","D75","D90")#
```

Finally, the program creates 18 matrices which will serve as default matrices for the 'Your Project' program. These matrices all have a base name starting by Ini_Sauvegarde attached to a 3 ID letters designing its content (see table later for IDs) : for instance for Grain Size Distribution it is Ini_SauvegardeGSD.

```
148 ###SauvegardeGSD contains backup of all entries for GSD
149 Ini_SauvegardeGSD<-matrix(nrow=1,ncol=10)
150 Ini_SauvegardeGSD<-as.data.frame(Ini_SauvegardeGSD)
151 names(Ini_SauvegardeGSD)<-c("NameGSD","NameECH","OptionSaisieGSD","OptionSaisiePhi","D50SAISIE","D84SAISIE","FsSAISIE","DmSAISIE"
152 Ini_SauvegardeGSD[1,1]<-c("    ")
```

3.1.3 Variables

Many variables are defined in the code, and to help their identification, they were assigned with a three letter code informing on which part of the code they are from. For instance "_GSD" refers to Grain Size Distribution, "_PRO" refers to project, "_SEC" refers to the river section, "_HYD" refers to hydraulics, "_QS" refers to sediment transport, "_Hydro" refers to hydrology, "_BIL" refers to sediment budget, and "_ANA" refers to analysis.

Each object is named by a variable, which can be assigned two values, in French or in English, what permits to load the program in both languages. For instance:

```
436         tabPanel(VAR$L[VAR$N=="STitre1"],icon = icon("image")
```

Where 'STitre1' has 2 values (French an English) in Variables.txt.

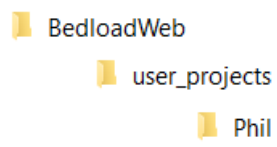
In the following the document focuses on the 'Your project' part of the program.

3.2 Users

When creating an account the program creates a new folder with the user name in the directory specified in the rep variable:

```
 8 ##############################################################################################################
 9 L<-"E"#Choose"F" for the french version and 'E' for the english version
10 setwd("C:/Bedloadweb/") # Directory where is unstalled the package
11 rep<-"C:/Bedloadweb/ToolBoxBProjects"# Directory for users data back-up (can be changed)
12 ##############################################################################################################
```

For example when creating the new user 'Phil', the program creates:



The user ID and password are saved in the **user_db.txt** file located in the **user_db** folder.

```
4882          user_db2<-Add_user(user_db,NewUser,Newpassword,Email,passeword2)
4883          write.table(user_db2, file="user_db/user_db.txt", row.names = FALSE, sep ="\t",append=FALSE, quote = FALSE)
4884          repUser<-paste0(rep,"/",input$NewUser)
4885          dir.create(path=repUser,showWarnings =FALSE)
```

Each time a user try to connect, the program checks if he is registered in the user-db file.

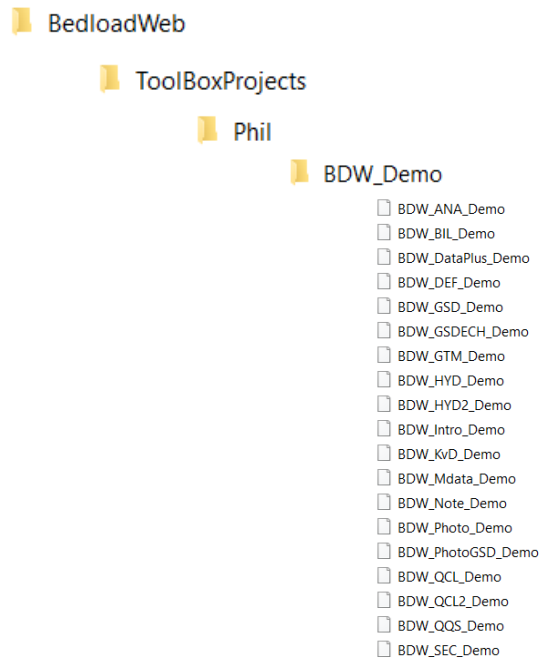
```
5061  observeEvent(input$button_login, {
5062    user_db<-read.delim("user_db/user_db.txt", header=TRUE, sep="\t",dec=".",stringsAsFactors = F)
5063      if(input$user!="" && input$password!="" && input$user %in% user_db$id && input$password == user_db$password[user_db$id == input$user])
5064        {
5065        current_user_status$logged <- TRUE
```

3.3Projects

When creating a new project the program:

1) creates a new folder with BDW_NameProject in the user folder
2) Creates a series of txt file where will be stored all the project data

For instance when saving the 'Demo' project, it creates:



For making these txt files, the program reads all matrices where the user data were stored (named 'Sauvegarde'+ 3ID letters designing the kind of data), copies the data in txt files and save these files with the name BDW_ 3 ID_NameProject.txt.

For instance when saving the grain Size distribution: Sauvegarde**GSD**() -> BDW_**GSD**_Demo.txt

If no data has been entered yet, the program uses the default matrices Ini_Sauvegarde created when launching the program (see description above).

```
6290    #SAVE GRANULO
6291    if(NROW(SauvegardeGSD())>0)
6292      SauvegardeGSD<-SauvegardeGSD()#this file contains GSD
6293    else
6294    {
6295      SauvegardeGSD<-na.omit(Ini_SauvegardeGSD)
6296    }
6297    write.table(SauvegardeGSD, file=paste0(rep,"/BDW_GSD_",NomProjet), row.names = FALSE, sep ="\t",append=FALSE, quote = FALSE)
```

When opening a project, it is the inverse: the program reads all txt files and converts these files into matrices useable by the functions.

For instance: BDW_**GSD**_Demo.txt -> Sauvegarde**GSD**()

Because these data must be kept available at any time for the program, but in the same time must be updated only when clicking on the SAVE buttons (and not each time a value is changed in a reactive function) Sauvegarde_() matrices are associated to ReactiveValues named **memory+IDcode**.

For instance the grain size distribution data in SauvegaredGSD() are always available by the program (for plotting, for hydraulics and bedload computation…). Its integrity is preserved until changes are ordered in the management functions (Save, Erase,..). Changes are done in memoryGSD$dat, which permits to work without altering the use SauvegaredGSD() in other functions, and SauvegaredGSD() is automatically updated in a reactive manner.

```
8488  # create the reactiveValue object memoryGSD
8489  memoryGSD <- reactiveValues(dat = NULL)
8490  # link the reactiveValue object to the reactive SauvegardeGSD matrix
8491  SauvegardeGSD<-reactive({
8492    SauvegardeGSD<-memoryGSD$dat
8493  })
8494  # each time memoryGSD$dat is modified in a function, SauvegardeGSD() is automatically updated
```

Here is the definition of ReactiveValues in R: "*This function returns an object for storing reactive values. It is similar to a list, but with special capabilities for reactive programming. When you read a value from it, the calling reactive expression takes a reactive dependency on that value, and when you write to it, it notifies any reactive functions that depend on that value. Note that values taken from the reactiveValues object are reactive, but* ***the reactiveValues object itself is not***."

In other words, the values stored in the Sauvegarde_() matrices are kept constant until data stored in memory_ are changed (used in the management functions: save, erase..).

All ID and their significance are given in the following table:

ID	Significance	Txt File	Matrix	Memory
ANA	Analysis	BDW_NameProj_ANA.txt	SauvegardeANA()	memoryANA
BIL	Sediment Budget	BDW_NameProj_BIL.txt	SauvegardeBIL()	memoryBIL
DataPlus	Other MetaData	BDW_NameProj_DataPlus.txt	SauvegardeDataPlus()	memoryDataPlus
DEF	Section Components	BDW_NameProj_DEF.txt	SauvegardeDEF()	memoryDEF
GSD	Grain Size Distribution (raw data)	BDW_NameProj_GSD.txt	SauvegardeGSD()	MemoryGSD
GSDECH	Grain Size Distribution (computed)	BDW_NameProj_GSDECH.txt	SauvegardeGSDECH()	memoryGSDECH
GTM	Model GTM	BDW_NameProj_GTM.txt	SauvegardeGTM()	memoryGTM
HYD	Hydrographs data	BDW_NameProj_HYD.txt	SauvegardeHYD()	memoryHYD
HYD2	Hydrographs Metadata	BDW_NameProj_HYD2.txt	SauvegardeHYD2()	memoryHY2
Intro	Project MetaData	BDW_NameProj_Intro.txt	SauvegardeIntro()	memoryIntro
KvsD	Roughness data	BDW_NameProj_KvsD.txt	SauvegardeKvsD()	memoryKvsD
Mdata	Section MetaData	BDW_NameProj_Mdata.txt	SauvegardeMdata()	memoryMdata
Note	Project description	BDW_NameProj_Note.txt	SauvegardeNote()	memoryNote
Photo	Link to Section photos	BDW_NameProj_Photo.txt	SauvegardePhoto()	memoryPhoto
PhotoGSD	Link to GSD photos	BDW_NameProj_PhotoGSD	SauvegardePhotoGSD()	memoryPhotoGSD
QCL	Flow duration curves data	BDW_NameProj_QCL.txt	SauvegardeQCL()	memoryQCL
QCL2	Flow duration curves Metadata	BDW_NameProj_QCL2.txt	SauvegardeQCL2()	memoryQCL2
QQS	Validation data Q(H) and Qs(Q)	BDW_NameProj_QQs.txt	SauvegardeQQs()	memoryQQS
SEC	Section	BDW_NameProj_SEC.txt	SauvegardeSEC()	memorySEC

When downloading a project in a txt file (Local Backup) all Sauvegarde_ matrices are pasted in the list **ListSauvPro()**, which is saved as a txt file on the computer.

```
5699   ListSauvPro<-list(SauvegardeGSD,SauvegardeGSDECH,SauvegardeSEC,SauvegardeQS,SauvegardeGTM,SauvegardeDEF,
5700                     SauvegardeMdata,SauvegardeKvsD,SauvegardePhoto,SauvegardePhotoGSD,SauvegardeQCL,SauvegardeHYD,
5701                     Intro,SauvegardeDataPlus,SauvegardeHYD2,SauvegardeQCL2,ExportBIL,SauvegardeANA)
```

```
5704 #write backup in a txt file
5705 output$downloadPROJ <- downloadHandler(
5706   filename = function() {
5707     "Export_BedloadWeb.txt"
5708   },
5709   content = function(file) {
5710     ListSauvPro<-ListSauvPro()
5711     write.table(ListSauvPro, file, sep="\t",col.names=TRUE, row.names=FALSE,fileEncoding = "UTF-8")
5712 })
```

3.4 Grain Size Distribution

When opening a project, the program opens the project folder and reads several files containing all information about all Grain Size Distributions already saved in this project. These files are then transformed into Sauvegarde_ matrices directly useable by the program:

File	Matrix	Content
BDW_GSD_NameProject.txt	SauvegardeGSD()	Raw data for all samples (measured data or model)
BDW_GSDECH_NameProject.txt	SauvegardeGSDECH()	Post-treated data for all samples
BDW_PhotoGSD_NameProject.txt	SauvegardePhotoGSD()	MetaData and link to photos

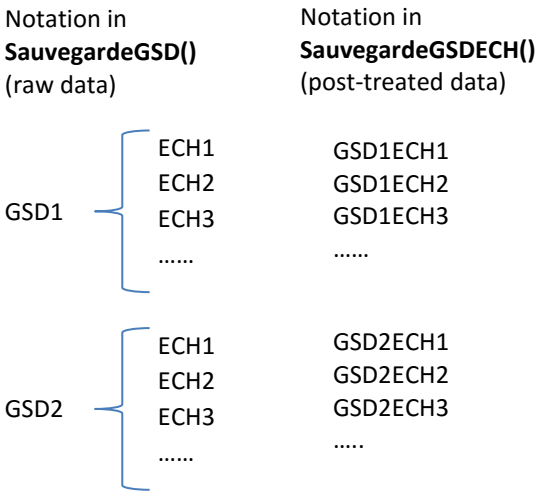
Inversely, when saving a project all matrices are copied in the txt files in the project floder.

IMPORTANT REMINDER ABOUT NOTATION:

ECH = 1 sample composing the GSD

GSD = averaged of several samples

An incremental code is assigned to each new sample ECH and new GSD:



3.4.1 Create a grain size distribution

2 options permit to create a grain size distribution

Option	Samples entered by user	Model	Destination
Data	Di, fi(%)	D_{50}	Sample [D,%]
Entries and function	### Excel like table Option Phi: DFPhi<-reactiveValues(mat=NULL) Option 0.5Phi: DFDemiPhi<-reactiveValues(mat=NULL) Option free: DFLibrePhi<-reactiveValues(mat=NULL)	SaisieGSDmODEL<-reactive({ D50River<-as.numeric(input$D50SAISIE) D84River<-as.numeric(input$D84SAISIE)	**SauvegardeECH**() stores raw data for all new sample ECH created in the active GSD
Matrix	7686 SaisieGSD<-reactive({ 7687 if(input$OptionSaisie_PHI==1) DF <- DFDemiPhi$mat 7688 if(input$OptionSaisie_PHI==2) DF <- DFPhi$mat 7689 if(input$OptionSaisie_PHI==3) DF <- DFLibrePhi$mat 7690 OptionSaisie_GSD<-input$OptionSaisie_GSD 7691 D50SAISIE<-as.numeric(input$D50SAISIE) 7692 if (OptionSaisie_GSD==1) NewSaisieGSD<-Saisie(DF) else NewSaisieGSD<-SaisieGS 7693 SaisieGSD<-FormatageSaisie(NewSaisieGSD,OptionSaisie_GSD,D50SAISIE) 7694 })		**SauvegardeGSDECH**() stores all post-treated data created in the Project

The final matrix is **SaisieGSD()** whatever the entries option used.

The SaisieGSD() is used to compute the matrix **PerSaisie()** which contains standard percentiles "D5","D10","D16","D25","D50","D75","D84","D90","D95","D100".

SaisieGSD() and PerSaisie() are updated when the user enters new data OR when opening an existing sample (ECH)

Each time a project is opened, the program reconstructs the **MeanGSD()** matrix by averaging (post-treated) data saved in SauvegardeGSDECH().

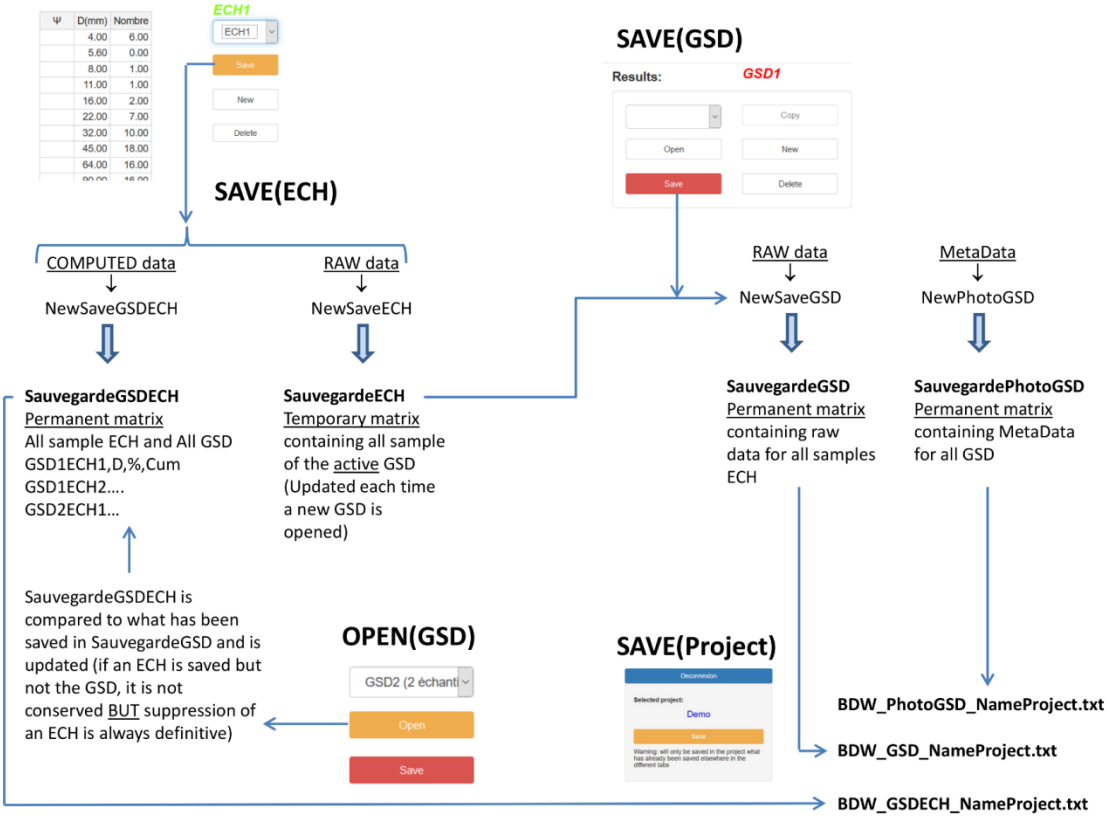
```
7726 ## Creates Mean GSD from several samples
7727 MeanGSD<-reactive({##compute the mean GSD matrix by averaging all samples
7728   SauvegardeGSDECH<-SauvegardeGSDECH()
7729   listSauvGSD<-listSauvGSD()
7730   MeanGSD<-MeanGSD_f(listSauvGSD,SauvegardeGSDECH)
7731 })
```

The **MatrixPlotGSD()** contains all post-treated GSD (1 by column) and is used for plotting the average GSD and its samples ECH.

3.4.2 Save a grain size distribution

'NewSaveXXX' are matrices created only when clicking on the save button; they contain all raw and computed data for the Sample and GSD to be saved.

The saving process is a bit tricky because it concerns saving of samples in a given GSD and the saving of the GSD itself.



Only **SauvegardeGSD()** (containing RAW DATA for all samples ECH and all GSD) and **SauvegardeGSDECH()** (containing COMPUTED DATA for l samples ECH and all GSD) will be saved in the project files BDW_GSD_NameProject.txt and BDW_GSDECH_NameProject.txt. SauvegardeECH is a temporary file containing all RAW DATA for the active GSD.

An additional file **SauvegardePhotoGSD()** conserves all the GSD MetaData and will be saved in BDW_PhotoGSD_NameProject.txt

When clicking the SAVE (ECH) button:

1) The program creates 2 matrices

- **NewSaveECH** contains RAW data for the new sample ECH to be saved
- **NewSaveGSDECH** contains COMPUTED data for the new sample ECH to be saved

2) These matrices are copied in the save matrices:

- NewSaveECH is added to the save file **SauvegardeECH**: this file is a temporary file containing all sample of the active GSD. This file is updated each time a new GSD is opened; it is used to udpdate the **SauvegardeGSD** table when saving GSD.
- NewSaveGSDECH is added to **SauvegardeGSDECH** : this save file contains post-treated data for all GSD and all samples ECH in the project.

Before copying NewSaveECH or NewSaveGSDECH in the save files, the program checks if for the given sample ECH a previous version has already been saved. If YES it is deleted (suppGSDECH_f) before copying the updated data.

If it is the first sample, Ini_SauvegardeECH and Ini_SauvegardeGSDECH are used by default.

```
7929  ################
7930  selectGSDECH<-NameGSDECH()
7931  isolate(dat <- memoryGSDECH$dat)
7932  memoryGSDECH$dat <-suppGSDECH_f(dat,selectGSDECH)
7933  isolate(dat <- memoryGSDECH$dat)
7934  if (is.null(dat))
7935  {
7936    Ini_SauvegardeGSDECH<-na.omit(Ini_SauvegardeGSDECH)
7937    memoryGSDECH$dat <- rbind(memoryGSDECH=Ini_SauvegardeGSDECH,NewSaveGSDECH())
7938  } else memoryGSDECH$dat <- rbind(dat,NewSaveGSDECH())
7939  ####################
```

When clicking the SAVE (GSD) button:

1) The program suppresses all previous backup concerning this GSD
2) The program creates a default GSD1 and ECH1 if it is a new project.

```
8178  ######################################## PREPARE DEFAULT GSD IF NEW PROJECT
8179  # If new project, create first sample GSD1 + ECH1
8180  selectECH<-NameECH()
8181  verif<-str_detect(selectECH, "ECH")
8182  if (verif==FALSE & numGSD()==1)
8183  {...}
```

3) The program copies the RAW DATA (NewSaveGSD) in the SauvegardeGSD file

```
8226  # Copy the new raw data (NewSaveGSD) in the raw data save file (SauvegardeGSD)
8227  isolate(dat <- memoryGSD$dat)
8228  if (is.null(dat))
8229  {
8230    Ini_SauvegardeGSD<-na.omit(Ini_SauvegardeGSD)
8231    memoryGSD$dat <- rbind(memoryGSD=Ini_SauvegardeGSD,NewSaveGSD)#### NewSaveGSD = SauvegardeECH
8232  } else memoryGSD$dat <- rbind(dat,NewSaveGSD)
8233  SauvegardeGSD<-memoryGSD$dat
```

NOTE: The COMPUTED data (NewSaveGSDECH) are not saved in the SauvegardeGSDECH file when clicking in the SAVE(GSD) button. These data are saved only when saving a sample first. This is why a sample which has not been saved with SAVE(ECH) will not be conserved.

WARNING: if a new sample is saved <SAVE(ECH)> but the GSD is not saved <SAVE(GSD)>, **SauvegardeGSD** is not updated (raw data are not saved). When opening again this GSD the program compares **SauvegardeGSD** and **SauvegardeGSDECH**, and update **SauvegardeGSDECH**. The sample will not appear even if it has been saved as a sample. HOWEVER, when erasing an existing sample ECH, the data are definitely removed from the **SavegardeGSDECH** file even if the GSD has not been saved, and will not appear when opening this GSD again.

```
8394    ############
8395    SauvegardeGSDECH<-SauvegardeGSDECH()
8396    SauvegardeGSD<-SauvegardeGSD()
8397    SauvegardeGSDECH<-updateGSDECH_f(SauvegardeGSD,SauvegardeGSDECH)
8398    isolate(dat <- memoryGSDECH$dat)
8399    memoryGSDECH$dat <- SauvegardeGSDECH
```

3.5 River cross section

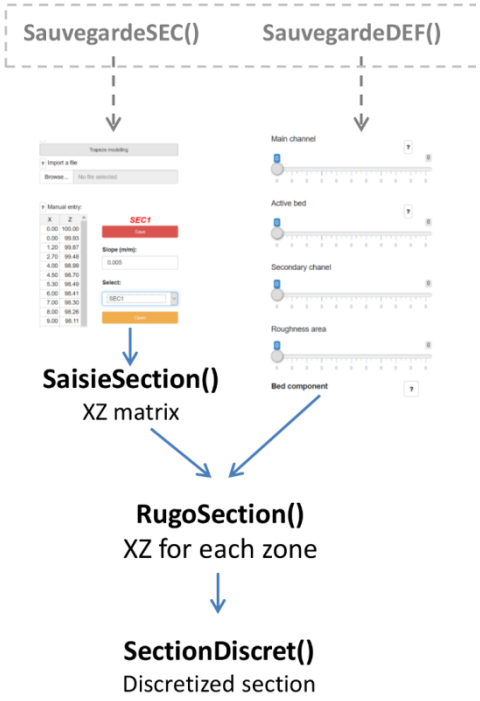
When opening a project, the program reads several files in the project folder, containing all information about all sections already saved in this project. These files are then transformed into matrices directly useable for screen display and calculation.

Inversely, when saving a project all matrices are copied in the txt files in the project folder.

File	Matrix	Content (raw data only)
BDW_SEC_NameProject.txt	SauvegardeSEC()	The sections XZ
BDW_DEF_NameProject.txt	SauvegardeDEF()	Definition of the bed components (main channel, secondary channel..)
BDW_KvsD_NameProject.txt	SauvegardeKvsD()	Roughness value (D84 or K) for each part of the bed
BDW_Mdata_NameProject.txt	SauvegardeMData ()	Metadata for all sections (morphology..)
BDW_DataPlus_NameProj.txt	SauvegardeDataPlus()	Other Metadata
BDW_QS_NameProject.txt	SauvegardeQS()	H(Q) and Qs(Q) validation data for the sections
BDW_Photo_NameProject.txt	SauvegardePhoto()	Links for photos

As was the case for the GSD, the ID code is conserved from the txt file name to the matrix name.

3.5.1 Opening or creating a section



Values are entered in an (excel like) XZ editing table, by 3 possible methods:
1) by hand
2) by importing a txt file
3) by modeling trapezes

When opening an existing section this editing table is updated automatically from the raw data saved in SauvegardeSEC().

Whatever the entries method, the final matrix is **SaisieSection()** which is a 2 column matrix (X,Z)

This section is used with the user entries defining the bed components (sliders for Main channel, Secondary channel, Roughness zones…). When opening an existing section sliders are updated automatically from raw data saved in **SauvegardeDEF()**.

A new matrix **RugoSection()** is created. This matrix columns ("X","Z","Actif","FixeRD","FixeRG","Rugo2","CH2") are filled with appropriate the Z value if X belongs to this definition zone, or NA if not.

Notations:
The code uses some French notations assigned to each variable associated with the bed components; the correspondence is given in the following table:

Code	Definition
LitMin	MAIN CHANNEL
LitActif	ACTIVE BED
CH2	SECONDARY CHANNEL
Rugo2	ROUGHNESS ZONE
LitFixRG	LEFT BANK
LitFixRD	RIGHT BANK

3.5.2 Save a section

The saving process uses reactive values which keep in memory all the information concerning the active section.

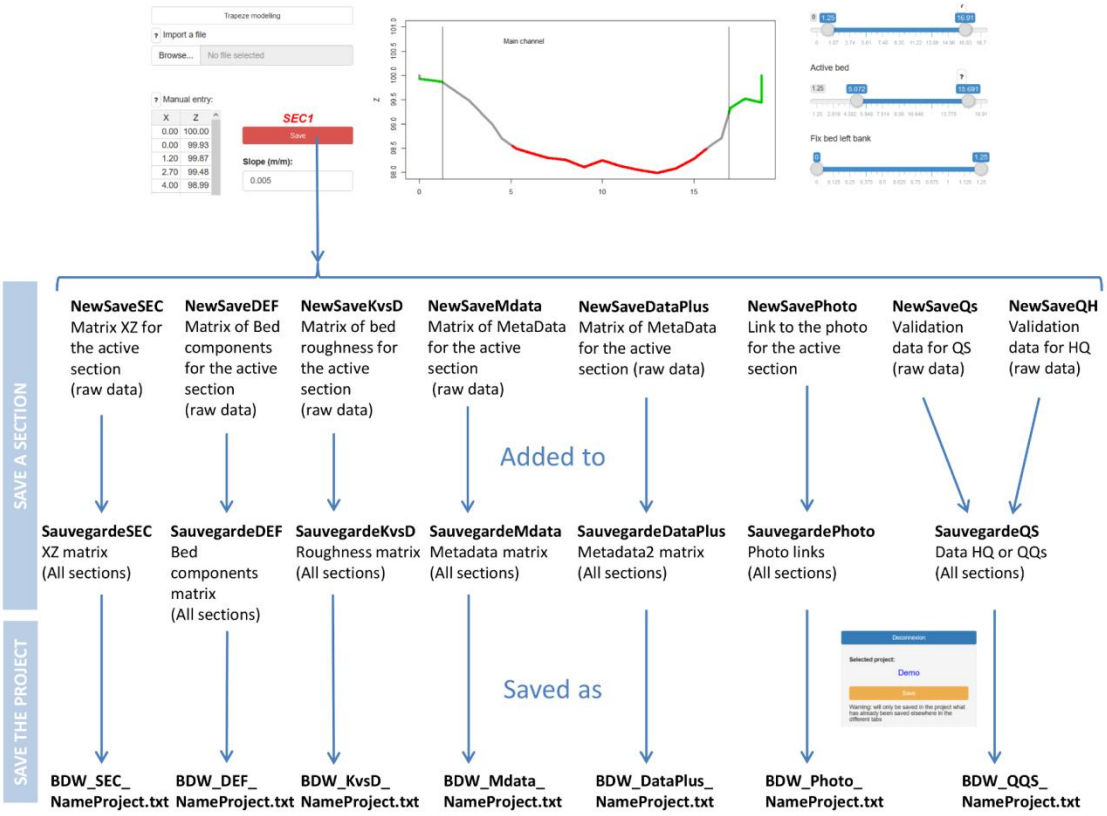
When saving a section, NewSave_ matrices are created for collecting and formatting all the raw data related to the section to be saved. The procedure is quite similar to what has been presented for the grain size distribution. In a first step, the program check if the section already exists, and if yes, erases the associated data before saving.

Then the NewSave_ matrices are copied to Sauvegarde_ matrices which contain information about all the project sections.

When saving the project, the Sauvegarde_ matrices are copied in DBW_.txt files.

```
## ____________DATA_____________
memorySEC <- reactiveValues(dat = NULL)# store XZ values
SauvegardeSEC<-reactive({
  SauvegardeSEC<-memorySEC$dat
})
memoryMdata <- reactiveValues(dat = NULL)# Store Metadata
SauvegardeMdata<-reactive({
  SauvegardeMdata<-memoryMdata$dat
})
memoryDataPlus <- reactiveValues(dat = NULL)# Store Metadata
SauvegardeDataPlus<-reactive({
  SauvegardeDataPlus<-memoryDataPlus$dat
})
memoryDEF <- reactiveValues(dat = NULL)# store bed components definition
SauvegardeDEF<-reactive({
  SauvegardeDEF<-memoryDEF$dat
})
memoryKvsD <- reactiveValues(dat = NULL)# store bed roughness values
SauvegardeKvsD<-reactive({
  SauvegardeKvsD<-memoryKvsD$dat
})
memoryQS <- reactiveValues(dat = NULL)# store validation data QH and QQs
SauvegardeQS<-reactive({
  SauvegardeQS<-memoryQS$dat
})
memoryPhoto <- reactiveValues(dat = NULL)# store link to photo
SauvegardePhoto<-reactive({
  SauvegardePhoto<-memoryPhoto$dat
})
## ____________NAMES_____________
memorynameSEC <- reactiveValues(dat = "")## NameSEC is the code (ex: SEC1)
NameSEC<-reactive({
  NameSEC<-memorynameSEC$dat
})
output$NameSEC<-renderText({
  NameSEC<-NameSEC()
})
output$NameSEC2<-renderText({
  NameSEC2<-NameSEC()
})
memorynomSEC <- reactiveValues(dat = "")## NomSEC is the real name (ex: Below the bridge)
NomSEC<-reactive({
```

```
    ##supress the data if already registered, before new saving with the updated values
    numSEC<-substr(selectSEC,4,6)
    isolate(dat <- memorySEC$dat)
    memorySEC$dat <-suppSEC_f(dat,selectSEC)
    isolate(dat <- memoryDEF$dat)
    memoryDEF$dat <-suppDEF_f(dat,selectSEC)
    isolate(dat <- memoryKvsD$dat)
    memoryKvsD$dat <-suppKvsD_f(dat,selectSEC)
    isolate(dat <- memoryQS$dat)
    memoryQS$dat <-suppQS_f(dat,selectSEC)
    isolate(dat <- memoryQS$dat)
    memoryQS$dat <-suppQH_f(dat,selectSEC)
    isolate(dat <- memoryPhoto$dat)
    memoryPhoto$dat <-suppPhoto_f(dat,selectSEC)
    isolate(dat <- memoryMdata$dat)
    memoryMdata$dat <-suppMdata_f(dat,selectSEC)
    isolate(dat <- memoryDataPlus$dat)
    memoryDataPlus$dat <-suppDataPlus_f(dat,selectSEC)
  } else numSEC<-numSEC()
```



Some of these data are entered in the Hydraulivs (SauvegardeKvsD) and in the sediment transport (SauvegardeQs) pages.

The matrix SauvegardeDataPlus has been created for saving additional parameters which were not saved in the SauvegardeMdata matrix in a first version of the program. This is a 30 columns matrix. For the moment only column1 (name of the section) and columns 11 to 23 are used.

We will use column 24 in an example (add an equation) presented later in this document.

The other columns are available for future developments.

```
### This function prepare the second Metadata file for the section to be saved
### will be added to SauvegardeDataPlus<-c("nameSEC","V2","V3","V4","V5",....
NewSaveDataPlus_f<-function(numSEC,Omegac,TcCamenen,q0Lefort,TcMPM,TcParker,TrParker,TmRecking,
{
  if (numSEC>0)
  {
    nameSEC<-paste0("SEC",numSEC)
    NewSaveDataPlus<-matrix(0,1,30)
    NewSaveDataPlus<-as.data.frame(NewSaveDataPlus)
    names(NewSaveDataPlus)<-c("nameSEC","V2","V3","V4","V5","V6","V7","V8","V9","V10",
                              "V11","V12","V13","V14","V15","V16","V17","V18","V19","V20",
                              "V21","V22","V23","V24","V25","V26","V27","V28","V29","V30")
    NewSaveDataPlus[1,1]<-nameSEC
    NewSaveDataPlus[1,11]<-Omegac
    NewSaveDataPlus[1,12]<-TcCamenen
    NewSaveDataPlus[1,13]<-q0Lefort
    NewSaveDataPlus[1,14]<-TcMPM
    NewSaveDataPlus[1,15]<-TcParker
    NewSaveDataPlus[1,16]<-TrParker
    NewSaveDataPlus[1,17]<-TmRecking
    NewSaveDataPlus[1,18]<-qcRickenmann
    NewSaveDataPlus[1,19]<-qcSchock
    NewSaveDataPlus[1,20]<-TcSmart
    NewSaveDataPlus[1,21]<-ucVR
    NewSaveDataPlus[1,22]<-TrWC
    NewSaveDataPlus[1,23]<-TcWong
  } else {}
  return(NewSaveDataPlus)
}
```

3.5.3 Open a section

When opening a section, the program:

1) reads the section name selected in the dropdown list
2) extracts all data related to this section the Sauvegarde_ matrices
3) update all screen concerned by these data

```
### OPEN A SECTION#################
observeEvent(input$OpenSEC,{#OPEN A SECTION
  selectSEC<-selectSEC()
  verif<-str_detect(selectSEC, "SEC")
    if (verif==TRUE)
    {
      SauvegardeSEC<-SauvegardeSEC()
      SaisieSEC<-selectSEC_f(selectSEC,SauvegardeSEC)
```

……

```
       RUG1<-SaisieDEF[1,10]
       RUG2<-SaisieDEF[1,11]
       CH21<-SaisieDEF[1,12]
       CH22<-SaisieDEF[1,13]
       Slope<-SaisieDEF[1,14]
       coteCH2<-SaisieDEF[1,15]
       minL<-min(SaisieSEC[,1])
       maxL<-max(SaisieSEC[,1])
       updateSliderInput(session, "LitMineur", value = c(LM1,LM2),min=minL,max=maxL)
       updateSliderInput(session, "LitActif", value = c(LA1,LA2),min=LM1, max=LM2)
       updateSliderInput(session, "LitFixeRD", value = c(LFRD1,LFRD2),min=minL,max=LM1)
```

3.5.4 Erase a section

When erasing a section, the program:

1) reads the section name selected in the dropdown list
2) erase all information related to this section the Sauvegarde_ matrices
3) update all screen displays, with zero entries

3.6 Hydraulics

The program reads the matrix MeanGSD(), creates a list of existing (saved) Grain Size Distributions, and updates all dropdown menus in the Hydraulics page. From this menu the user choose which GSD must be associated with each bed component, and when clicking on the SAVE (HYDRAULICS) button, these user entries are saved in the bed roughness matrix KvsD.

```
10480 #CREATION of the ROUGHNESS MATRIX KVSD: Type","K","D84","KD","Message","GSD"
10481 #Contains all information for the hydraulics page/To be modify with caution
10482 KvsD<-reactive({
10483   LFRD<-input$DefLitFixeRD
10484   LFRG<-input$DefLitFixeRG
10485   CH2<-input$DefinitionCH2
10486   Rugo2<-input$AjoutRugo2
```

When opening an existing section these entries are updated directly from SauvegardeKvsD()

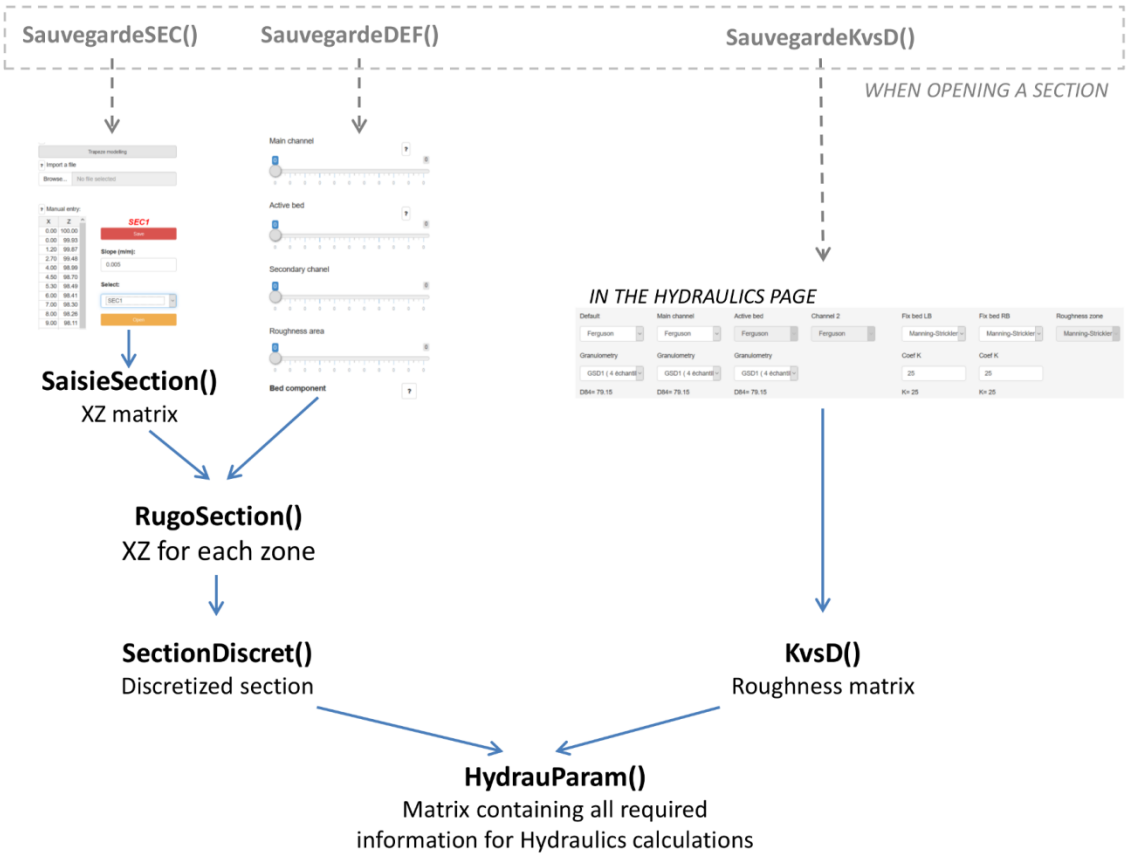
```
9571       SauvegardeKvsD<-SauvegardeKvsD()
9572       SaisieKvsD<-selectKvsD_f(selectSEC,SauvegardeKvsD)
9573       SaisieKvsD<-selectKvsD_f2(SaisieKvsD,SauvegardePhotoGSD)
9574       updateSelectInput(session, "Granulo_Defaut", choices = listSauvGSD, selected=SaisieKvsD[1,7])
9575       updateSelectInput(session, "Granulo_LitMin", choices = listSauvGSD, selected=SaisieKvsD[2,7])
9576       updateSelectInput(session, "Granulo_LitActif", choices = listSauvGSD, selected=SaisieKvsD[3,7])
9577       updateSelectInput(session, "Granulo_CH2", choices = listSauvGSD, selected=SaisieKvsD[4,7])
9578       updateSelectInput(session, "Granulo_LitFixRD", choices = listSauvGSD, selected=SaisieKvsD[5,7])
9579       updateSelectInput(session, "Granulo_LitFixRG", choices = listSauvGSD, selected=SaisieKvsD[6,7])
```

The RugoSection() matrix created in the 'Section' page is discretized (1000 space step are used but it can be changed if necessary) which gives the matrix called **SectionDiscret().**

```
10641 #MATRIX SectionDiscret:"X","Z","XdX","ZdZ","H","HdH","dR","dA","Type","KvsD","K","D84","U"
10642 ##discretization of the section; created only once for a given section
10643 SectionDiscret<-reactive({
10644     SaisieSection<-SaisieSection()
10645     murSEC<-input$murSEC
10646     Hmur<-Hmur()
10647     N_Z<-1000
10648     if (input$murSEC) SaisieSection<-SaisieSection_f2(SaisieSection,Hmur)
10649     discretisationSection_f(SaisieSection,N_Z)
10650 })
```

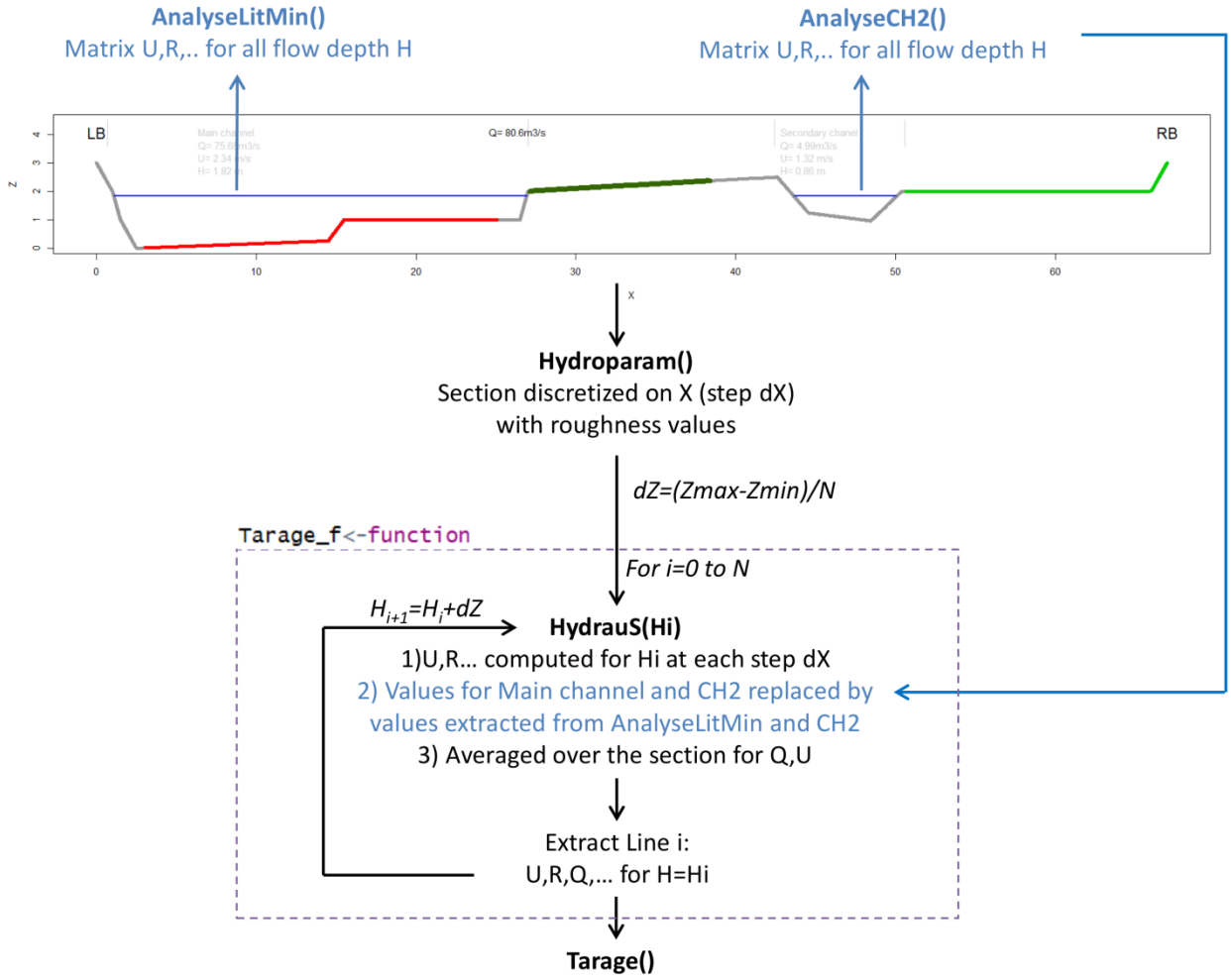
The SectionDiscret() is mixed with the roughness matrix **KvsD()** and gives the final matrix **HydrauParam()** which contains all information required for hydraulics.

X	Z	XdX	ZdZ	H	HdH	dP	dA	Type	KvsD	K	D84	U
0.0000	102.0000	0.0000	101.9520	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.9520	0.0000	101.9039	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.9039	0.0000	101.8559	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.8559	0.0000	101.8078	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA



The program considers separately hydraulic calculations for the main channel and the secondary channel (where the mean velocity is computed for the whole section with the hydraulic radius), and the rest of the section (where velocity is computed with the local flow depth at each space step dX).

First, for the main channel and for the secondary channel, hydraulics parameters (wetted perimeter P, wetted area A, hydraulic radius R, and mean velocity U) are computed for each flow depth H (incremented by the space step dZ after each calculation). The resulting matrices are **AnalyseLitMin()** and **AnalyseCH2().**

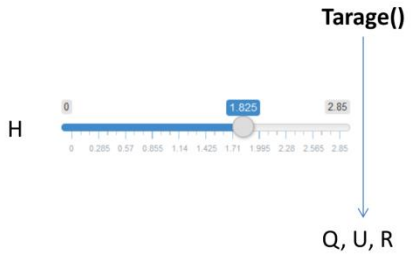


They are used to build the **Tarage()** matrix which gives the hydraulics results for each flow depth. In the **Tarage_f** function, the maximum flow depth Zmax is discretized with space step dZ (calculated for a number of steps N=100). For each Z+dZ in the interval [0, Hmax] the program:

1) Calculates the main channel hydraulics parameters **HydrauResultLitMin=**(P, A, R,U) by extrapolating AnalyseLitMin().
2) Calculates the secondary channel parameters **HydrauResultCH2=**(P, A, R, U) by extrapolating AnalyseCH2().
3) Computes hydraulics parameters for each space step dX in HydrauParam(), and everywhere the 'Type' column has the value 'LitMin' or 'CH2', replace these values by values available in HydrauResultLitMin and HydrauResultCH2. It gives the matrix **HydrauS.**
4) Averages values in HydrauS and report hydraulics results (Q,U,R…) in **Tarage()** for the considered flow depth Hi**.**
5) Repeat the calculations until Hi=Hmax

The matrix Tarage() is computed again each time a section is opened.

Finally when the user specifies a given flow depth **Heau()** with the slider, the hydraulics parameters (U,R,Q..) are extrapolated from Tarage().



3.7 Sediment transport

3.7.1 Calculation

First the program computes the hydraulics condition to be considered in the matrix Rescal:

1) The flow depth is discretized from 0 to Hmax=Zmax-Zmin considering the section geometry
2) For each depth, all hydraulics parameters are extrapolated from Tarage()
3) Additional parameters required for bedload are computed (shear stress, Shield stress…)
4) Bedload is computed for each equation

The final matrix is renamed **RescalGraphPro()** and contains Qs(Q) for all equations.

RescalGraphPro() is transformed with respect to the units specified by the user into **RescalGraphUnit()**.

RescalGraphUnit() permits to draw the bedload models for the active section.

When the user enters a flow depth H (with slider), the corresponding Q_S values are extrapolated from RescalGraphUnit in matrix **RescalQs()**.

3.7.2 Save

When saving the Sediment transport page, critical shear stress and discharge entered for bedload equations are saved in matrix SauvegardeDataPlus, columns 11 to 23 .

```
### This function prepare the second Metadata file for the section to be saved
### will be added to SauvegardeDataPlus<-c("nameSEC","V2","V3","V4","V5",....
NewSaveDataPlus_f<-function(numSEC,Omegac,TcCamenen,q0Lefort,TcMPM,TcParker,TrParker,TmRecking,
{
  if (numSEC>0)
  {
    nameSEC<-paste0("SEC",numSEC)
    NewSaveDataPlus<-matrix(0,1,30)
    NewSaveDataPlus<-as.data.frame(NewSaveDataPlus)
    names(NewSaveDataPlus)<-c("nameSEC","V2","V3","V4","V5","V6","V7","V8","V9","V10",
                              "V11","V12","V13","V14","V15","V16","V17","V18","V19","V20",
                              "V21","V22","V23","V24","V25","V26","V27","V28","V29","V30")
    NewSaveDataPlus[1,1]<-nameSEC
    NewSaveDataPlus[1,11]<-Omegac
    NewSaveDataPlus[1,12]<-TcCamenen
    NewSaveDataPlus[1,13]<-q0Lefort
    NewSaveDataPlus[1,14]<-TcMPM
    NewSaveDataPlus[1,15]<-TcParker
    NewSaveDataPlus[1,16]<-TrParker
    NewSaveDataPlus[1,17]<-TmRecking
    NewSaveDataPlus[1,18]<-qcRickenmann
    NewSaveDataPlus[1,19]<-qcSchock
    NewSaveDataPlus[1,20]<-TcSmart
    NewSaveDataPlus[1,21]<-ucVR
    NewSaveDataPlus[1,22]<-TrWC
    NewSaveDataPlus[1,23]<-TcWong
  } else {}
  return(NewSaveDataPlus)
}
```

The user can also associate validation data Qs(Q) to the section. These data are saved in **SauvegardeQs()**.

When saving the Sediment Transport page, SauvegardeMdata() and SauvegardeDataPlus() are also updated.

Validation data for bedload Grain Size Distribution can also be saved. Because several data sets can be entered (one GSD per discharge Q) the saving process is identical to the one used for Grain Size Distribution or the Sections: a counter counts the number of samples already saved and ranks the new samples with name GTM+SamplerNumber. The new data formatted in **NewsaveGTM()** are added to **SauvegaredGTM()**.

3.8 Hydrology

Two kind of hydrological data are created:

- Hydrographs HYD: discharge as a function of time Q(t)
- Flow duration curves QCL: discharge as a function of return period Q(T)

The algorithm is repeated 2 times almost exactly the same way for HYD and QCL.

3.8.1 Creation

As for SECTION, an excel-like table permits to enter the data. Data can also be uploaded from txt files.

Whatever the entries option, the raw data matrices are **SaisieHydro()** and **SaisieQCL()**. They are discretized to produce the final matrix **RescalHYD()**.

```
13341 ### This function discretizes the hydrology with a given time step correponding to the unit time (s, mn , h or d)
13342 RescalHYD<-reactive({
13343   OptHydro<-input$OptHydro
13344   if (OptHydro==1) SaisieHydro<-SaisieHydro() else SaisieHydro<-SaisieQCL()
13345   UnitQ<-UnitQ()
13346   N_HYD<-1000
13347   RescalHYD<-RescalHYD_f(OptHydro,SaisieHydro,UnitQ,N_HYD)
13348 })
```

3.8.2 Hydrological data management

The procedure is exactly as for GSD and sections.

A counter counts the number of existing HYD and QCL and rank the new one. A name is crated with HYD+Counter or QCL+Counter. Newsave matrices are created and contain the name, raw data and Metadata:

- **NewsaveHYD**: Hydrographs raw data
- **NewSaveHYD2**: Hydrographs MetaData
- **NewSaveQCL**: Flow Dration Curves raw data
- **NewSaveQCL2**: Flow Dration Curves MetaData

When saving the NewSave matrices are added to the save matrices:

- NewsaveHYD added to **SauvegardeHYD()**
- NewSaveHYD2 added to **SauvegardeHYD2()**
- NewSaveQCL: added to **SauvegardeQCL()**
- NewSaveQCL2: added to **SauvegardeQCL2()**

3.9 Sediment budget

3.9.1 Select hydrology

The sediment budget is computed for the active section (which is open in the cross section page):

```
#Display name of the active section ('SEC' code)
output$NameSECBIL<-renderText({
  NameSECBIL<-NameSEC()
})
#Display full name of the Active Section
output$NomSECBIL<-renderText({
  if (is.na(nchar(NomSEC()))) NomSEC<-"" else NomSEC<-NomSEC()
})
```

For computing a sediment budget, the program must associate a hydrology signal to this section, which is selected by the user in the dropdown menu (each time the program modify and save Hydrology data, the lists are updated in the Hydrology page and in the Sediment Budget page).

```
#SELECT an hydrograph HYD or Flow Duration Curve QCL in the dorpdown menu
selectHYDBIL<-reactive({#select hydrograph in the dorpdown menu
  if (nchar(as.character(input$ListHYDBIL))>0)
  {
    selectHYDBIL<-as.character(input$ListHYDBIL)
    selectHYDBIL<-strsplit(selectHYDBIL,split=' (', fixed=TRUE)
    selectHYDBIL<-unlist(selectHYDBIL)
    selectHYDBIL<-selectHYDBIL[[1]]
  } else selectHYDBIL<-"XXX"
})
```

Then the program keeps in memory the selected data.

```
#____________________Memory_____________________
# Keep in memory the hydrological data selected by user and imported from SauvegardeHYD or SauvegardeQCL
memoryHYDBIL <- reactiveValues(dat = NULL)
memoryQCLBIL <- reactiveValues(dat = NULL)
#OPEN file HYD or QCL selected in the dropdown menu and affect data to the memory
observeEvent(c(input$OpenHydroBIL),{
  OptHydroBIL<-input$OptHydroBIL
  listSauvHYD<-listSauvHYDFullName()
  listSauvQCL<-listSauvQCLFullName()
  if(OptHydroBIL==1)
  {
```

The program reads the units and data are post-treated taking into account the watershed surface:

```
14377 ## ADAPT THE HYDROGRAH TO THE SIZE OF THE WATERSHED WITH THE MYER FORMULA
14378 SaisieHydroBIL<-reactive({## Adjust the hydrograph to the watershed area with the Myer equation
14379   SaisieHydro<-memoryHYDBIL$dat
14380   BVREF<-as.numeric(input$BVREF)
14381   BVSEC<-as.numeric(input$BVSEC)
14382   MYER<-as.numeric(input$MYER)
14383   SaisieHydroBIL<-SaisieHydro_f2(SaisieHydro,BVREF,BVSEC,MYER)
14384 })
```

The final data (Hydrograph or Flow Duration Curve) are conserved in matrix **SaisieHydroBIL()**.

3.9.2 Sediment budget calculation

The program uses the bedload models computed for the active section in **RescalGraphPRO()**. These models are adapted accounting for units specified by the user; the new matrix is **RescalGraphUnitHYD()**.

For each time step in SaisieHydroBIL(), the program extrapolates Qs(Q) in RescalGraphUnitHYD() and associates a Qs (t) value to each Q(t) value. The results are in **RescalHYDBIL()**.

Finally, the transported sediment budget is obtained by summing Qs(t)*dt for each equation in RescalHYDBIL(). The results are in **RescalBIL(),** which comprises 2 columns: results in m3 and m3/year (this later has NA values if the hydrograph duration is less than 1 year).

3.9.3 Saving a sediment budget

A **NewSaveBIL()** matrix is created with the sediment budget data to be saved. 2 columns are created and named with the section name SEC _num+ M3333T ('T' for a Time period volume, in m3) or M3333A ('A' for a mean Annual volume in m3/year). When a hydrograph is used and its total duration is less than 1 year the value in column M3333A is NA.

```
14559 ## Matrix of new data to save
14560 ## for each section 2 columns are created
14561 ## they are named M3333T (T for a Time period volume with HYD) and M3333A (A for a mean Annual volume with QCL)
14562 ## M3333T is always assigned a value, M3333T only if computation with QCL (otherwise NA)
14563 ## This will be used later for selecting budgets computed with HYD or with QCL
14564 NewSaveBIL<-reactive({
14565   NewSaveBIL<-RescalBIL()
14566   NameSEC<-NameSEC()
14567   colnames(NewSaveBIL)<-c("Equation",paste0(NameSEC,"M3333T"),paste0(NameSEC,"M3333A"))
14568   NewSaveBIL
14569 })
14570
14571 ## SAVE the sediment budget
14572 ## New columns (M333T and M3333A) are added for each new sections
14573 observeEvent(input$SaveBIL,{
14574   selectSEC<-NameSEC()
14575   verif<-str_detect(selectSEC, "SEC")
14576   if (verif==TRUE)
14577   {
14578     ###########
14579     isolate(dat <- memoryMdata$dat)
14580     memoryMdata$dat <-suppMdata_f(dat,selectSEC)# supress the data if aready registered, before new saving with the updated values
```

MetaData are also saved in SauvegardeMdata and Sauvegarde DataPlus when saving a Sediment budget.

3.10 Analysis

This page does not perform any calculation. It is only displays the previous results.

Only Sections for which a sediment budget has been saved ca be considered in this page. The list (dropdown menu) is updated when saving a sediment budget in the 'Sediment Budget' page:

```
14611       SauvegardeBIL<-SauvegardeBIL[,(str_detect(names(SauvegardeBIL),"SEC")==TRUE)]
14612       ListSECANA<-as.character(colnames(SauvegardeBIL))
14613       ListSECANA<-as.character(ListSECANA)
14614       ListSECANA<-strsplit(ListSECANA,split='M3333', fixed=TRUE)#coupe le nom avant @(
14615       ListSECANA<- matrix(unlist(ListSECANA), ncol=2, byrow=TRUE)
14616       ListSECANA<-as.data.frame(ListSECANA)
14617       namesBIL<-unique(ListSECANA[,1])
14618       ListSECANA<-listSauvSEC_f1(RappelPhoto,namesBIL)
14619       ListSECANA<-ListSECANA[,3]
14620     } else ListSECANA<-""
14621     updateSelectInput(session, "ListSECANA", choices = c("",as.character(ListSECANA)), selected=NULL)
```

For the concerned sections, the program built a MetaData matrix **ListSECBIL()** containing the section name SEC2, the section full name SEC2 (near the bridge) and names SEC2M3333T and SEC2M3333A already used in SauvegardeBIL() and referring to the Sediment budget results in m3 and m3/yr.

In a second step the program considers the Sections selected by user for the analysis in **SelectANA()**

```
14739 ### CREATES LIST of sections selected for the analysis
14740 selectSECANA<-reactive({
14741   selectSEC<-as.character(input$ListSECANA)
14742   selectSEC<-strsplit(selectSEC,split=' (', fixed=TRUE)#coupe le nom avant @(
14743   selectSEC <- matrix(unlist(selectSEC), ncol=2, byrow=TRUE)
14744   selectSECANA <-as.data.frame(selectSEC)
14745 })
```

A new result matrix is built: for the sections in SelectANA, ListSECBIL permits to extract the concerned columns M3333T and M3333A in SauvegardeBIL(). Only columns M3333T are kept if the selected unit is M3 or M3333A if M3/year. Only lines corresponding to the selected equations are kept. Columns name are adapted (Code or Full name) according to the user request. The final matrix is called **RescalGraphANA().**

This matrix is displayed on screen and is used to build the figures.

To extract the statistics, the program use the R boxplot function

```
6179 #### This function creates a table with the statistical data extrated from bowplot function (Median, percentiles...)
6180 statsANA_f<-function(RescalGraphANA)
6181 {
6182   L<-length(RescalGraphANA[1,])
6183   MIN<-min(RescalGraphANA[,2:L])
6184   MIN<-max(MIN,10-3)
6185   MAX<-max(RescalGraphANA[,2:L])
6186   bp<-boxplot(RescalGraphANA[,2:L]## use the boxplot function to create the stats
6187               , use.cols = TRUE
6188               ,log="y"
6189               ,ylim=c(MIN,MAX)
6190               ,xlab=""
6191               ,range = 0# range=0 accounts for outliers!
6192   )
6193   mytable <- round(bp$stats,1)### extract stats values
6194   mytable <-data.frame(mytable)
6195   q<-rep(NA,5)
6196   mytable<-data.frame(q,mytable)
6197   names(mytable)<-names(RescalGraphANA)
6198   mytable[,1]<-c('min','lower quartile','median','upper quartile','max')
6199   return(mytable)
6200 }
```

4 MODIFYING THE PACKAGE

For taking advantage of the full potentiality of the package, a basic knowledge of the R (a lot of tutorials are available online) and Shiny language (https://shiny.rstudio.com/tutorial/) is required.

The package can be modified by different ways. The simplest way consists to change only the data files (keeping the code intact). A more complex change (but still relatively simple operation) consists in adding a new equation: in that case it is sufficient to repeat rigorously what has already been written for e other equations. Finally, the user can propose a new and original piece of code. These three examples are presented in this chapter.

4.1Add a new data set

For adding a data set the following files located in the folder **/data** must be modified:

- **DataSet.txt** contains the bedload datasets
- **FichierGranulo.txt** contains all rivers' grain size distribution (dataset)
- **GSDfi.txt** is a post treatment of FichierGranulo.txt for fractional bedload calculation with some equations (Wilcock and Crowe, Parker 90)
- **GSDfi_Par.txt** is a variant of GSDfi.txt without the sand fraction (option Parker 90)
- **morpho.txt** contains information about the data set rivers' morphology
- **TabRef.txt** contains the bedload dataset metadata (average values)
- **References.txt** contains the data sets bibliography

Adding a new data set does not present any difficulty as long as the following rules are respected :

- The name of a river must be written exactly the same way everywhere
- all different keywords already used in the txt files must be written exactly the same way (for instance the words used for describing 'Morphology' or 'measurement Technique')
- values are mandatory for some of the parameters
- respect the txt format when saving (ANSI should be appropriate when installing the package in personal computer)
- The txt files must be saved as **tab-delimited** text files.

Let add, as an example, the data collected in the Severaisse River (Misset et al 2020[1]). The corresponding name in the data set will be "Severaisse Villar Loubiere".

Open the file **Dataset.txt** in a tab sheet (excel for instance) and select the line where the new data will be inserted.

1	Nø	River	ID	Date	W	S	Q	U	H	R	D16	D50	D84	D90	qs	Morphology
6273	6263	Saskatchewar	53		3.05	0.00745	4.71	NA	0.73	NA	NA	0.0019	0.0565	NA	382	Riffle-pool
6274	6264	Saskatchewar	54		3.05	0.00745	4.71	NA	0.73	NA	NA	0.0016	0.0569	NA	593	Riffle-pool
6275	6265	Saskatchewar	55		3.05	0.00745	4.95	NA	0.79	NA	NA	0.0075	0.1144	NA	56.6	Riffle-pool
6276	6266	Squaw Creek	1	05/05/1994	6.8	0.01	0.59	0.36	0.24	0.23	0.02	0.043	0.072	0.082	0.0099	Plane Bed
6277	6267	Squaw Creek	2	05/05/1994	6.8	0.01	0.59	0.36	0.24	0.23	0.02	0.043	0.072	0.082	0.147	Plane Bed
6278	6268	Squaw Creek	1	05/07/1994	6.89	0.01	1.08	0.53	0.3	0.27	0.02	0.043	0.072	0.082	1.25	Plane Bed
6279	6269	Squaw Creek	2	05/07/1994	6.89	0.01	1.08	0.53	0.3	0.27	0.02	0.043	0.072	0.082	0.467	Plane Bed
6280	6270	Squaw Creek	1	05/08/1994	7.44	0.01	1.2	0.57	0.28	0.26	0.02	0.043	0.072	0.082	0.523	Plane Bed
6281	6271	Squaw Creek	2	05/08/1994	7.44	0.01	1.2	0.57	0.28	0.26	0.02	0.043	0.072	0.082	1.26	Plane Bed

The best consists to preserve the river name alphabetical order because this order is automatically preserved in the option 'Select a River', but not in the Multicriteria option Table.

The following data are mandatory: Name, W(m), S(m/m), Q (m3/s) or H(m), D50(m), q_s(g/s/m). By default the program considers D84=2.1D50 and Morphology=Riffle-Pool.

1	Nø	River	ID	Date	W	S	Q	U	H	R	D16	D50	D84	D90	qs	Morphology
6271	6270	Saskatchewar	51		3.05	0.007	5.71	NA	0.88	NA	NA	0.002	0.0829	NA	162	Riffle-pool
6272	6271	Saskatchewar	52		3.05	0.007	5.25	NA	0.76	NA	NA	0.0021	0.0685	NA	227	Riffle-pool
6273	6272	Saskatchewar	53		3.05	0.007	4.71	NA	0.73	NA	NA	0.0019	0.0565	NA	382	Riffle-pool
6274	6273	Saskatchewar	54		3.05	0.007	4.71	NA	0.73	NA	NA	0.0016	0.0569	NA	593	Riffle-pool
6275	6274	Saskatchewar	55		3.05	0.007	4.95	NA	0.79	NA	NA	0.0075	0.1144	NA	56.6	Riffle-pool
6276	6275	Severaisse Vill	1	30/04/2018	13	0.011	13.3	NA	0.87	NA	0.003	0.049	0.123	0.16	99	Riffle-pool
6277	6276	Severaisse Vill	2	02/05/2018	13	0.011	9.03	NA	0.72	NA	0.003	0.049	0.123	0.16	20	Riffle-pool
6278	6277	Severaisse Vill	3	02/05/2018	13	0.011	8.96	NA	0.72	NA	0.003	0.049	0.123	0.16	4	Riffle-pool
6279	6278	Severaisse Vill	4	02/05/2018	13	0.011	9.08	NA	0.72	NA	0.003	0.049	0.123	0.16	0	Riffle-pool
6280	6279	Severaisse Vill	5	06/05/2018	13	0.011	17.56	NA	1.03	NA	0.003	0.049	0.123	0.16	192	Riffle-pool
6281	6280	Severaisse Vill	6	07/05/2018	13	0.011	17.79	NA	1.04	NA	0.003	0.049	0.123	0.16	307	Riffle-pool
6282	6281	Severaisse Vill	7	07/05/2018	13	0.011	18.19	NA	1.07	NA	0.003	0.049	0.123	0.16	256	Riffle-pool
6283	6282	Severaisse Vill	8	09/05/2018	13	0.011	16.94	NA	1.02	NA	0.003	0.049	0.123	0.16	592	Riffle-pool
6284	6283	Severaisse Vill	9	09/05/2018	13	0.011	18.79	NA	1.08	NA	0.003	0.049	0.123	0.16	334	Riffle-pool
6285	6284	Severaisse Vill	10	14/05/2018	13	0.011	13.32	NA	0.92	NA	0.003	0.049	0.123	0.16	12	Riffle-pool
6286	6285	Severaisse Vill	11	14/05/2018	13	0.011	13.38	NA	0.92	NA	0.003	0.049	0.123	0.16	28	Riffle-pool

If a grain size distribution is available, the data must be inserted in **FichierGranulo.txt** (the position is not important).

[1] Misset, C., A. Recking, C. Legout, M. Bakker, N. Bodereau, L. Borgniet, M. Cassel, T. Geay, F. Gimbert, O. Navratil, H. Piégay, N. Valsangkar, M. Cazilhac, A. Poirel, and S. Zanker (2020), Combining multi-physical measurements to quantify bedload transport and morphodynamic interactions in an Alpine braiding river reach, *Geomorphology*, doi:https://doi.org/10.1016/j.geomorph.2019.106877

1	River	D	Phi	Percent	Cumul
693	Yampa River a	4	2	4.11111111	96.7037037
694	Yampa River a	8	3	1.8562963	98.56
695	Yampa River a	16	4	0.79294118	99.3529412
696	Yampa River a	32	5	0.64705882	100
697	Severaisse Vill	4	2	21.3930348	21.3930348
698	Severaisse Vill	8	3	2.98507463	24.3781095
699	Severaisse Vill	11	3.5	2.73631841	27.1144279
700	Severaisse Vill	16	4	3.48258706	30.5970149
701	Severaisse Vill	22	4.5	3.48258706	34.079602
702	Severaisse Vill	32	5	5.2238806	39.3034826
703	Severaisse Vill	45	5.5	7.960199	47.2636816
704	Severaisse Vill	64	6	12.4378109	59.7014925
705	Severaisse Vill	91	6.5	13.1840796	72.8855721
706	Severaisse Vill	128	7	12.6865672	85.5721393
707	Severaisse Vill	191	7.5	8.70646766	94.278607
708	Severaisse Vill	280	8.1	4.72636816	99.0049751
709	Severaisse Vill	340	8.4	0.49751244	99.5024876
710	Severaisse Vill	560	9.1	0.49751244	100

The river morphology must be inserted in **morpho.txt.** If unknown, put 'NA' and the program will consider 'Riffle-Pool' by default.

1	River	morpho
132	Wind river	Riffle-pool
133	Wisconsin River at Muscoda	Sand bed
134	Yampa River at Deerlodge Park	Sand bed
135	Fivemile Creek	Riffle-pool
136	Severaisse Villar Loubiere	Riffle-pool

References associated with the data set must be inserted in the file **TabRef.txt.** The first column is assigned a number which will be used to read this reference in the TabRef file. Be careful to **save with ANSI format** for the local use otherwise the program will return an empty text.

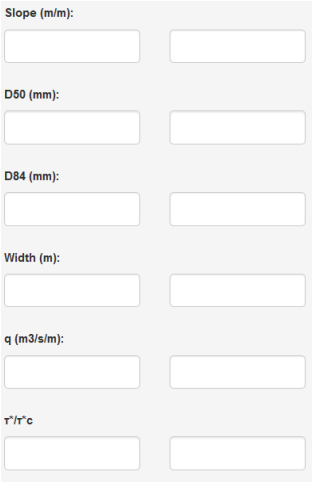
1	N	Reference
62	61	Yu, G.-A., Wang, Z.-Y., Zhang, K., Chang, T.-C., Liu, H., 2009. Effect of incoming sediment on the transport rate of bed load in mountains streams. International Journal of Sediment Research 24, 260–273.
63	62	Hinton, D., R. H. Hotchkiss, and M. Cope (2018), Comparison of Calibrated Empirical and Semi-Empirical Methods for Bedload Transport Rate Prediction in Gravel Bed Streams, Journal of Hydraulic Engineering, 144(7), 17.
64	63	Hinton, D., R. Hotchkiss, and D. P. Ames (2017), Comprehensive and Quality-Controlled Bedload Transport Database, *Journal of Hydraulic Engineering* , *143* (2), 6p, doi:10.1061/(ASCE)HY.1943-7900.0001221.
65	64	Misset, C., A. Recking, C. Legout, M. Bakker, N. Bodereau, L. Borgniet, M. Cassel, T. Geay, F. Gimbert, O. Navratil, H. Piégay, N. Valsangkar, M. Cazilhac, A. Poirel, and S. Zanker (2020), Combining multi-physical measurements

The **TabRef.txt** comprises the data set MetaData:

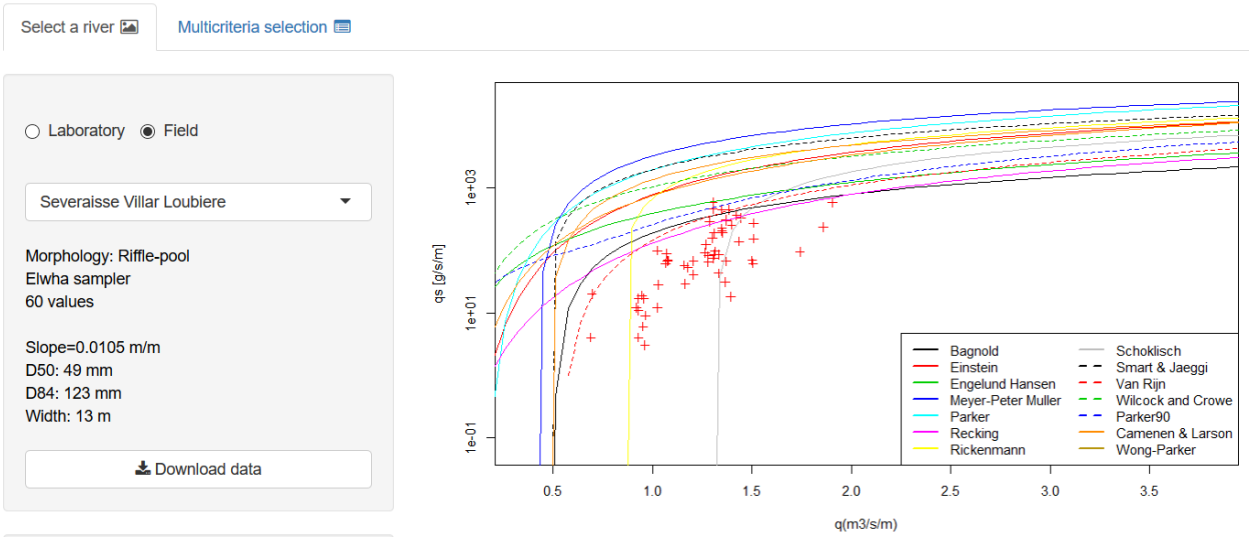
1	CaseAcocher	River	Pente	D50	D84	Largeur	q	ttc	Morphologie	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Technique	T2	T3	T4	T5
281	NA	Smart19	0.07	10.5	12.6	0.2	0.15	3.33607654	Canal											Laboratoire				
282	NA	Smart20	0.1	10.5	12.6	0.2	0.15	3.86839672	Canal											Laboratoire				
283	NA	Smart21	0.15	10.5	12.6	0.2	0.15	4.64142294	Canal											Laboratoire				
284	NA	Smart22	0.2	10.5	12.6	0.2	0.125	4.8893237	Canal											Laboratoire				
285	NA	Severaisse Villar	0.0105	49	123	13			Riffle-pool	64										Elwha sampler				

River, Pente (slope in mm), D50(mm) and Largeur (width in m) are mandatory. By default D84=2.1D50. Columns R1 to R10 must recall the numbers associated to the reference in the TabRef.txt file (up to 10 references). Columns Technique, T2 to T5 must recall the measurement technics. Columns q and ttc represents the unit discharge q (the maximum unit discharge in

m3/s/m) and transport stage τ^*/τ_c^* characterizing the data. By default q=1m3/s/m and ttc is computed from q. These values are used when selecting data in the Multicriteria page. It is clearly not optimized because for the moment the search considers only one characteristic values in the TabRef.txt files (q max).



When launching the code the Severaisse data are displayed with the equations:



By default the program considers q=1 m3/s/m. Such wrong value would not impact the calculations with the bedload equations, but it will impact the Multicriteria search.

Appropriate values can be deduced from the data (by identifying the median value); an easy way to determine roughly theses values consists in changing the display option in the 'Select a River' page, and to read graphically (and approximately) the qmax value for the data set:

The work is not finished. When testing the equations in the Multicriteria page, program returns an error message:

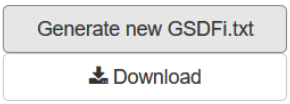
A last step is still required: updating files **GSDfi.txt** and **GSDfi_Par.txt**. These files transform the GSD in a format required for fractional calculation with equations Wilcock and Crowe and Parker90.

	name	D50	D84	fi1	fi2	fi3	fi4	fi6
2	D	D	D	0.5	1.41421	2.4495	3.46410162	4.89897949
3	Phi	Phi	Phi	0.5	0.5	1.2925	1.79248125	2.29248125
6276	Severaisse	0.049	0.123	0	0	0	0.21393035	0.01492537
6277	Severaisse	0.049	0.123	0	0	0	0.21393035	0.01492537
6278	Severaisse	0.049	0.123	0	0	0	0.21393035	0.01492537
6279	Severaisse	0.049	0.123	0	0	0	0.21393035	0.01492537
6280	Severaisse	0.049	0.123	0	0	0	0.21393035	0.01492537

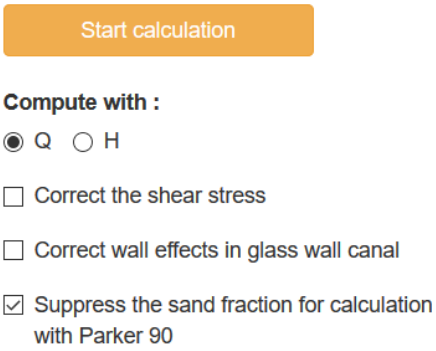
fi1024	Fs	D16	D30	Dm	Dg	D75	D90
861.031939	1	1	1	0.60046502	1	1	1
9.74992294	1	1	1	0.29829794	1	1	1
0	0	0.09142586	0.01514286	0.07817553	0.037	0.09716667	0.16004
0	0	0.09142586	0.01514286	0.07817553	0.037	0.09716667	0.16004
0	0	0.09142586	0.01514286	0.07817553	0.037	0.09716667	0.16004
0	0	0.09142586	0.01514286	0.07817553	0.037	0.09716667	0.16004
0	0	0.09142586	0.01514286	0.07817553	0.037	0.09716667	0.16004

Columns f1 to f1024 contain the fraction of the total grain size distribution comprised in each size class between D=1mm and D=1024 mm. The last columns give sand fraction Fs and characteristic diameters (in meter) Dm, Dg, D16, D30, D75 and D90 required for some equations.

You can modify this file by yourself, but a function (**RescalGSD_f2** in BedloadR.R) permits to update the GSDFi.txt file automatically. You can get access to it with the button 'Generate new GSDFi.txt' in the Help page. For activating this button you must connect (in the 'Your Project' user tab) with ID= '**admin_addnewdata**' and password='**admin_addnewdata**' (you will get a message login failed but the button is activated in the Help page).



When clicking on this button a Download button appears which permits to download the GSDfi.txt file (caution: **it takes several minutes!**). You must save this file in the **/data** folder. Start again the operation for creating the **GSDfi_Par.txt** file after having clicked on "Suppress the sand fraction for calculation with Parker 90" in the 'Muticriteria Selection' page.



Again, save this file in the /data folder, and now it's done the package was updated with the new data. Hen testing to compute with the equations in the Multicriteria page it should work.

4.2 Add an equation

In this econd example we present how to add a new bedload equation to the program.

4.2.1 The equation

Let's add for instance the Ackers and White equation (1973)[2]. This equation is written:

$$q_b = \rho_s 0.025 \frac{q D_{35}}{d}\left[\frac{F_{gr}}{A_{gr}} - 1\right]^{1.5} \quad (1)$$

With $$F_{gr} = \frac{1}{\sqrt{g(s-1)D_{35}}}\left[\frac{U}{\sqrt{32}log\left(\frac{10d}{D_{35}}\right)}\right]$$

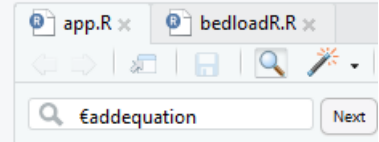
where:

- q_b is bedload transport per unit width in g/s/m (Q_b/W where W is the flow width) ,
- d is the flow depth (m)
- A_{gr}=0.17 is the threshold for mobility
- ρ_s ≈2650000 g/m^3 is the sediment density,
- s=ρ_s/ρ is the relative density (ρ≈1000 Kg/m^3 is water density),
- g is acceleration of gravity (9.81 ms^{-2}),
- D_{84} is the bed grain size dstributionD_{84}.
- D_{35} is the bed grain size dstributionD_{35}

[2] Ackers, P., and W. R. White (1973), Sediment transport; new approach and analysis, *Journal of the Hydraulics Division*, *99*(11), 2041-2060.

4.2.2 Make the change

A lot of changes are required in the Server part for insuring a good calculation and results display. One simple way to add an equation is to make a search with "**€addequation**" which will drive you to each part of the code where equations are defined or called, and then you just have to reproduce the existing commands for the new equation.



The first step consists to list all the variables required for the calculation and complete the Rescal matrix:

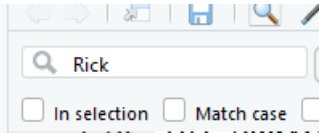
```
117 Rescal<-matrix(NA,ncol=106,nrow=600)
118 Rescal<-data.frame(Rescal)
119 names(Rescal)<-c("name","W","S","D50","D84","q","qsmeas",#C1_7
120                  "qstar","p","d","R","U","f","taux","t50","t84",#C8_16
121                  "t50MPM","ScorRick","t50Wilc",#C17_19
122                  "tm","qscalReck","borneinfReck","bornesupReck",#Recking C20_23
123                  "qscalMPM","borneinfMPM","bornesupMPM",#Meyer Peter & Muller#C24_26
124                  "qcRick","qscalRick","borneinfRick","bornesupRick",#Rickenmann#C27_30
125                  "qscalScho","borneinfScho","bornesupScho",#Schoklich#C31_33
126                  "qscalEng","borneinfEng","bornesupEng",#Engelund#C34_36
127                  "qscalPar","borneinfPar","bornesupPar",#parker#C37_39
128                  "Eins","qscalEin","borneinfEin","bornesupEin",#Einstein#C40_43
129                  "qscalBag","borneinfBag","bornesupBag",#Bagnold#C44_46
130                  "Z","UuSmart","qscalSma","borneinfSma","bornesupSma",#Smart & Jaeggi#C47_51
131                  "ustarVR","ParamT","qscalVR","borneinfVR","bornesupVR",#Van Rijn#C52_56
132                  "ustar","t","qscalWilc","borneinfWilc","bornesupWilc",#Wilcock#C57_61
133                  "tsg","phisg0","ParamOmega","sigma","sigma0","omega0","qscalPar90","borneinfPar90","bornesupPar90",#C62_70
134                  "Abscisse","qcScho","omega","omegac",#C71_74
135                   "hr","qr",#C74_76
136                  "D16","Dm","GrLef","CLef","nLef","mLef","q0Lef","qstarLef","kskrLef","corLef","ZLef","MLef","FLef","CpLef",
137                  "DstarCL","tcCL","qscalCL","borneinfCL","bornesupCL",#C94_98
138                  "qscalWONG","borneinfWONG","bornesupWONG",#C99_101
139                  "Fs","D16","D30","D75","D90")#
```

All changes in a matrix may cause a problem if somewhere in the code the variable is called by its rank in the matrix. This should not occur because variable are usually called by their name and not by their position. It is however important to keep the matrix intact because the rank of bedload model calculation (numbers indicated on the right of each line indicate the matrix columns) is used for editing the txt report.

We can see that all the variables listed above for the new equation already exist in the Rescal matrix (we will use D_{30} instead of D_{35}). So we need to define only the columns for the intermediate value Fgr, the computed q_s values (qsAW) and its associated bounds borneinfAW and bornesupAW delimiting plus or minus one order of magnitude values.

The matrix size must be changed by adding 4 columns, and columns are added at the end:

```
Rescal<-matrix(NA,ncol=110,nrow=600)
Rescal<-data.frame(Rescal)
names(Rescal)<-c("name","W","S","D50","D84","q","qsmeas",#C1_7
                 "qstar","p","d","R","U","f","taux","t50","t84",#C8_16
                 "t50MPM","ScorRick","t50Wilc",#C17_19
                 "tm","qscalReck","borneinfReck","bornesupReck",#Recking C20_23
                 "qscalMPM","borneinfMPM","bornesupMPM",#Meyer Peter & Muller#C24_26
                 "qcRick","qscalRick","borneinfRick","bornesupRick",#Rickenmann#C27_30
                 "qscalScho","borneinfScho","bornesupScho",#Schoklich#C31_33
                 "qscalEng","borneinfEng","bornesupEng",#Engelund#C34_36
                 "qscalPar","borneinfPar","bornesupPar",#parker#C37_39
                 "Eins","qscalEin","borneinfEin","bornesupEin",#Einstein#C40_43
                 "qscalBag","borneinfBag","bornesupBag",#Bagnold#C44_46
                 "Z","UuSmart","qscalSma","borneinfSma","bornesupSma",#Smart & Jaeggi#C47_51
                 "ustarVR","ParamT","qscalVR","borneinfVR","bornesupVR",#Van Rijn#C52_56
                 "ustar","t","qscalWilc","borneinfWilc","bornesupWilc",#Wilcock#C57_61
                 "tsg","phisg0","ParamOmega","sigma","sigma0","omega0","qscalPar90","borneinfPar90","bornesupPar90",#C62_70
                 "Abscisse","qcScho","omega","omegac",#C71_74
                  "hr","qr",#C74_76
                 "D16","Dm","GrLef","CLef","nLef","mLef","q0Lef","qstarLef","kskrLef","corLef","ZLef","MLef","FLef","CpLef",'
                 "DstarCL","tcCL","qscalCL","borneinfCL","bornesupCL",#C94_98
                 "qscalWONG","borneinfWONG","bornesupWONG",#C99_101
                 "Fgr","qscalAW","borneinfAW","bornesupAW",#C102_105
                  "Fs","D16","D50","D75","D90")#
```



Ssearch on '€addequation' drive us now to the definition of Ini_SauvegardeBIL where we add the name of the equation Ackers-White (keep the same writing everywhere):

```
Ini_SauvegardeBIL<-matrix(NA,ncol=3,nrow=16)#€addequation
Ini_SauvegardeBIL<-data.frame(Ini_SauvegardeBIL)
names(Ini_SauvegardeBIL)<-c("Equation","m3","m3/an")
Ini_SauvegardeBIL[,1]<-c("Ackers-White","Bagnold","Camenen Larson","Einstein-Brown","Engelund",
                         "Lefort","Meyer-P&M","Parker79","Parker90","Recking","Rickenmann","
                        Schocklitch","Smart Jaeggi","Van Rijn","Wilcock-Crowe","Wong-Parker")
```

Next we change the matrix 'Equations':

```
Equations<-matrix(0,16,2)
Equations<- as.data.frame(Equations)
names(Equations)<-c("CaseAcocher","Equation")
Equations$Equation<-c("Bagnold", "Einstein-Brown", "Engelund-Hansen","Meyer-Peter & Muller","Parker79", "Recking",
                      "Rickenmann","Schoklitsch","Smart and Jaeggi", "Van-Rijn","Wilcock Crowe","Parker90","Lefort",
                      "Camenen","Wong-Parker","Ackers-White")
```

Next we add the equation name in the list to be displayed (be careful, the French and English versions do not have the same list):

```
ChoixEquationsE<-c(#equation Lefort is not in the english version
 "Ackers-White"=16,
  "Bagnold" = 1,
  "Camenen-Larson"=14,
  "Einstein-Brown" = 2,
  "Engelund"=3,
  "Meyer-Peter Muller"=4,
  "Parker79"=5,
  "Parker90"=12,
  "Recking"=6,
  "Rickenmann"=7,
  "Schocklitch"=8,
  "Smart Jaeggi"=9,
  "Van Rijn"=10,
  "Wilcock Crowe"=11,
  "Wong-Parker"=15
)

#€equation
SelectEquationsF<-c(16,1,14,2,3,13,4,5,12,6,7,8,9,10,11,15)
SelectEquationsE<-c(16,1,14,2,3,4,5,12,6,7,8,9,10,11,15)
```

We now move to the UI part of the App.R file.

The next search '€addequation' drives us to the Qs page where the threshold values can be entered by the user. For the Ackers White equation is concerns A_{gr}:

```
                conditionalPanel(
                  condition = "input.DtcPRO==1",
                  checkboxInput("CortPRO", label = VAR$L[VAR$N=="BLD21"]),
                  h5(strong(VAR$L[VAR$N=="BLD22"])),
                  #€addequation
                  div(style="display:inline-block", HTML("Ackers-White: Agr=")),
                  div(style="display:inline-block", textInput("Agr",label=NULL, value=NULL,width = "70px")),
                  div(style="display:inline-block", textOutput("ValueAgr")),
                  br(),
                  div(style="display:inline-block", HTML("Bagnold: &omega;c=")),
                  div(style="display:inline-block", textInput("Omegac",label=NULL, value=NULL,width = "70px")),
                  div(style="display:inline-block", textOutput("ValueOmegac")),
```

It's done for the UI part of the code. Next changes concern **the Server part**.

```
RescalGraph<-reactive({
```

Defines the defaut values for Agr:

```
    Agr<-NA
    Omegac<-NA
    TcCamenen<-NA
    q0Lefort<-NA
    TcMPM<-NA
    TcParker<-NA
```

Then we call the function ACKERSWHITE (to be created in BedloadR.R):

```
    Rescal<-FICHIERSRescal(DataQs, ParHydrau, NomRiver, S, D50, D84,W,TypeData, Rescal,scaleX)
    Rescal<-HYDRAULIQUE(S,W,D50,D84,Rescal,ParHydrau,TypeData,FigplotMULTI,NbRiver,WallEffect,Cort,RhoEau,RhoSed,s)
    Rescal$morpho<-detectMorpho_f(NomRiver,morpho,TypeData)
    Rescal<-morcDIAMETER(Rescal,dataGSD,GSD,NomRiver,nomGSD)
    Rescal<-ACKERSWHITE(Rescal,CoefTc,RhoEau,RhoSed,s,Agr)
    Rescal<-BAGNOLD(Rescal,CoefTc,RhoEau,RhoSed,s,Omegac)
```

And so on… all the Server functions designed by €addequation must be updated.

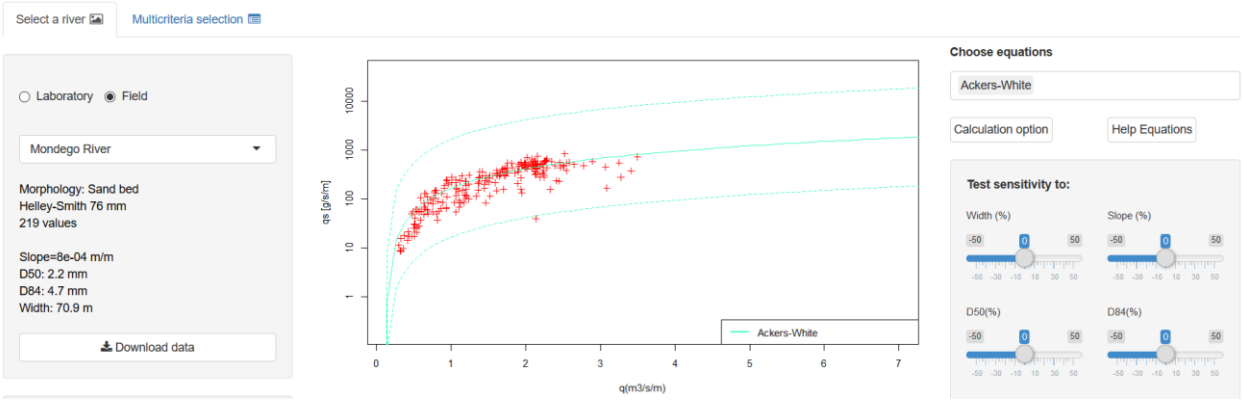
Now we move to the BedloadR.R file.

The ACKERSWHITE function is defined first:

```
### Ackers and White equation
### Ackers, P., and W. R. White (1973), Sediment transport; new approach and analysis, Journal of the Hydraulics
ACKERSWHITE<-function(Rescal,CoefTc,RhoEau,RhoSed,s,AgrAW)
{
  if (is.na(AgrAW)|AgrAW==0)Agr<-0.17*CoefTc else Agr<-AgrAW*CoefTc
    Rescal$Fgr<-1/sqrt(9.81*(s-1)*Rescal$D30)*(Rescal$U/(sqrt(32)*log10(10*Rescal$d/Rescal$D30)))
    Rescal$qscalAW<-1000*RhoSed*0.025*Rescal$q*Rescal$D30/Rescal$d*(Rescal$Fgr/Agr-1)^1.5+.0000000000001
    Rescal$qscalAW[Rescal$Fgr<=Agr]<-0.0000000000001
  return(Rescal)
}
```

Continue the search on **€addequation** until the end of the BedloadR.R file and make all the changes by copying and modifying the lines already written for the existing equations. A new color must be assigned to this equation for graphical display; here we choose the HTML code "#b7950b" which corresponds to a blue cyan.

If the changes are all done scrupulously it should work at the end! Now when launching the program the Ackers and White equation is available:



4.2.3 Accounting for existing projects

When modifying the code you must keep in mind that the new code must accept the files already saved by the user (existing projects).

For instance, in the above example, after having added the equation the code could not recognize the file SauvegardeBIL saved with the old version because this file lists the equations used for computing the sediment budget:

Equation	m3	m3/an	SEC3M3333T	SEC3M3333A	SEC2M3333T	SEC2M3333A	SEC1M3333T	SEC1M3333A	SEC4M3333T	SEC4M3333A
Bagnold	0	0	94.1557	0	1789	0	9978.8975	9978.8975	119.4319	0
Camenen Larson	0	0	457.4711	0	13682	0	52357.3768	52357.3768	812.8267	0
Einstein-Brown	0	0	471.9715	0	14464	0	56362.1757	56362.1757	859.2521	0
Engelund	0	0	445.7155	0	3841	0	25564.6408	25564.6408	634.3828	0
Lefort	0	0	502.5724	0	5658	0	17125.8505	17125.8505	401.4029	0
Meyer-P&M	0	0	1374.0415	0	24159	0	157926.3049	157926.3049	2664.5822	0
Parker79	0	0	1023.7778	0	22103	0	115901.7397	115901.7397	1869.8799	0
Parker90	0	0	152.4124	0	3993	0	12988.9067	12988.9067	297.7498	0
Recking	0	0	172.5736	0	2556	0	18284.1183	18284.1183	349.6244	0
Rickenmann	0	0	529.7249	0	17742	0	83453.9113	83453.9113	1252.7923	0
Schocklitch	0	0	492.7008	0	7895	0	32053.1761	32053.1761	401.3563	0
Smart Jaeggi	0	0	419.2401	0	11754	0	70133.1820	70133.1820	1446.0390	0
Van Rijn	0	0	63.8689	0	2907	0	8200.6199	8200.6199	85.7437	0
Wilcok-Crowe	0	0	1188.8756	0	10176	0	82596.6135	82596.6135	4060.7139	0
Wong-Parker	0	0	604.7044	0	11193	0	71626.5054	71626.5054	1174.5350	0

When trying to open this old file, the new code fails in trying to affect this 15 lines matrix (15 Equations) to the new 16 lines RescalBIL() matrix (16 Equations), and stops.

Two options are possible to correct this problem:

1) **Option1**: Erase the old saved files by replace the SauvegardeBIL() matrix by Ini_ SauvegardeBIL(): all the project data are conserved, but the user will have to save again the sediment budgets for accounting for the new equation. In particular, the analysis already saved are not available anymore:

```
7407      RappelBIL<-read.delim(paste0(rep,"/BDW_BIL_",NomProjet), header=TRUE, sep="\t",dec=".",check.names = FALSE)
7408      ## €addequation
7409      ## when adding a new equation this 3 lines are required for concistancy with old projects
7410      ## => add the new equation in the existing SauvegardeBIL files
7411      L1<-length(RappelBIL[,1])
7412      L2<-length(Ini_SauvegardeBIL[,1])
7413      if (L1!=L2) RappelBIL<-na.omit(Ini_SauvegardeBIL)##This option suppress all saved sed budget
```

2) **Option2:** Conserve the existing file, check for its content, and update the structure by adding the new equation:

```
7407      RappelBIL<-read.delim(paste0(rep,"/BDW_BIL_",NomProjet), header=TRUE, sep="\t",dec=".",check.names = FALSE)
7408      ## €addequation
7409      ## when adding a new equation this 3 lines are required for concistancy with old projects
7410      ## => add the new equation in the existing SauvegardeBIL files
7411      L1<-length(RappelBIL[,1])
7412      L2<-length(Ini_SauvegardeBIL[,1])
7413      if (L1!=L2) RappelBIL<-checkBILforNewEquation_f(RappelBIL,Ini_SauvegardeBIL)## add equation with budget=0
```

This second option calls the function CheckBILforNewEquation in BedloadR.R:

```
#This function check consistency with existing SauvegardeANA files when addding a new equation
checkBILforNewEquation_f<-function(RappelBIL,Ini_SauvegardeBIL)
{
  nameEquations<-c(as.character(RappelBIL[,1]))
  RappelBIL[,1]<-nameEquations
  ###############
  Ll<-length(RappelBIL[1,])
  Lc<-length(Ini_SauvegardeBIL[,1])
  NewMatrix<-matrix(0,nrow=Lc,ncol=Ll)
  NewMatrix<-data.frame(NewMatrix)
  colnames(NewMatrix)<-colnames(RappelBIL)#€addequation
  NewMatrix[,1]<-c("Ackers-White","Bagnold","Camenen Larson","Einstein-Brown","Engelund","Lefort",
                   "Meyer-P&M","Parker79","Parker90","Recking","Rickenmann","Schocklitch",
                   "Smart Jaeggi","Van Rijn","Wilcock-Crowe","Wong-Parker")
  ###############
  L2<-length(RappelBIL[,1])
  Names<-RappelBIL[,1]
  for (i in (1:L2))
  {
    j<-which(match(NewMatrix$Equation,Names[i])==TRUE)#
    if (!is.na(j) & j>0) NewMatrix[j,]<-RappelBIL[i,]
  }
  RappelBIL<-NewMatrix
return(RappelBIL)
}
```

This modification must be used each time an old project is opened; finally the program uses the two options:

- Option 1 is used when uploading a project saved in txt format (local backup)
- Option 2 is used when opening a project from the save menu

Another problem occurred when opening an old project: the new variable Agr imported from column 24 of SauvegardeDataPlus() was not read as 'NA' but as '0'. Because the ACKERSWHTE function computes Fgr/Agr the code stopped. The function had to be modified for accounting for this unexpected numerical zero value.

```
ACKERSWHITE<-function(Rescal,CoefTc,RhoEau,RhoSed,s,AgrAW)
{
  if (is.na(AgrAW)|AgrAW==0)Agr<-0.17*CoefTc else Agr<-AgrAW*CoefTc
```

4.2.4 Useful trick

Despite adding an equation is not so complicated, it is rarely works the first time, because format issues are common in R. In that case it can be very useful to visualize the table causing trouble, which can be done by searching "TESTPAGE" with the search tool:

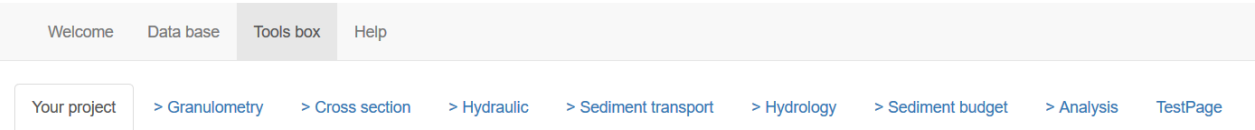
And activate the few lines in the UI page:

```
3226 ###€UITESTPAGE#############################################################################################
3227 #### ADD THESE LINES TO DISPLAY A TABLE FOR CHECKING ITS CONTENT (usefull in a debbug process for instance)
3228 ,
3229 tabPanel("TestPage",
3230          textOutput("Verif_Text"),
3231          tableOutput("Verif_Table1"),
3232          tableOutput("Verif_Table2")
3233 )
```

Clicking on the NEXT button drives you the SERVER part where you must specify the table to visualize:

```
15257 ####€TESTPAGE#############################################################################################
15258 ################################   EDIT A VERIFICATION PAGE ##########################################
15259 ##############
15260 output$Verif_Text<-renderText({### visualize a text, a list, a vector
15261     # variabe<-
15262 })
15263 #############
15264 output$Verif_Table1 <- renderTable(#visualize a table in UI
15265   digits=4,
15266   {
15267     # RescalHYD<-RescalGraph()
15268   }
15269 )
15270 ############
15271 output$Verif_Table2 <- renderTable(#visualize a table in UI
15272   digits=4,
15273   {
15274   # rescal<-Ini_SauvegardeBIL
15275   }
15276 )
```

Now when launching the code a new page "TestPage" is displayed where you can check the consistency of the tables and other variables:



This is very helpful for solving problems.

4.3 Add a new piece of code

The two above examples conserved the code integrity. Here we present how to extend the existing code for investigating new questions.

The interactions between the different parts of the code are tricky, and it is highly recommended when adding a new piece of code to not modify what already exists. **The best and safer way would consist in building the new tools by using only the code outputs:**

- **1) Read the data in the Sauvegarde() matrices**.
- **2) use these data as input for the new module**

4.3.1 Example: add a page for suspension load

In this example, we add a page for computing suspension associated with bedload transport following the methodology proposed by Misset et al (2019)[3]. They observed a strong correlation between the suspended load and bedload which was explained by the fact that the coarse bed contains a lot of fine sediments, and the more it is mobilized by the flow, the more it releases fine sediments in the main flow. The equation is:

$$SSC^* = \begin{cases} 4.04 \times 10^{-5} q_b^{*0.187} & if\ q_b^* < 8.49 \times 10^{-6} \\ 9.12 \times 10^{-2} q_b^{*0.849} & otherwise \end{cases} \quad (2)$$

Where SSC* is the dimensionless Suspended load concentration and q_b^* is the dimensionless bedload transport defined by:

$$SSC^* = \frac{SSC}{\rho_s} \quad \text{and} \quad q_b^* = \frac{q_b}{\sqrt{g(s-1)D_{84}^3}}$$

And where:

[3] Misset, C., A. Recking, O. Navratil, C. Legout, A. Poirel, M. Cazihlac, V. Briguet, and M. Esteves (2019), Quantifying bed-related suspended load in gravel bed rivers through an analysis of the bedload-suspended load relationship, Earth Surface Processes and Landforms.

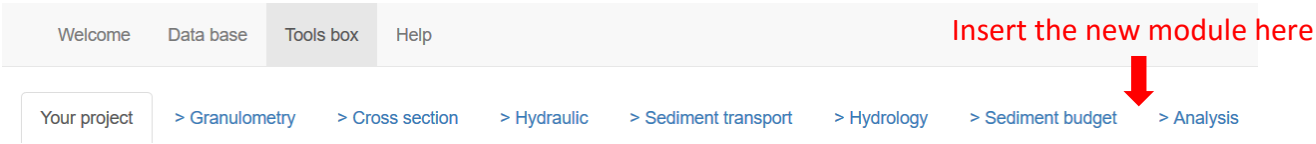
- SSC [mg/l] is the suspended load concentration,
- q_b is bedload transport per unit width (Q_b/W where W is the flow width) ,
- $\rho_s \approx 2650 Kg/m^3$ is the sediment density,
- $s=\rho_s/\rho$ is the relative density ($\rho \approx 1000\ Kg/m^3$ is water density),
- g is acceleration of gravity (9.81 ms^{-2}),
- D_{84} is the bed grain size dstributionD_{84}.

SSC is what we want to calculate and all other data are already computed by BedloadR for a given section and Grain Size Distribution.

The code presented in the following is already inserted in the package, but as comment (#). To make it active you must select the corresponding part of the code in UI (enter €UISUS in the find option), Server (enter €SUS in the find option) and BedloadWeb (at the end) and remove '#' by selecting the lines and typing **Ctrl+Shift+C** on keyboard.

4.3.2 The User Interface

Let try to insert the suspension page just after the Sediment Budget page.



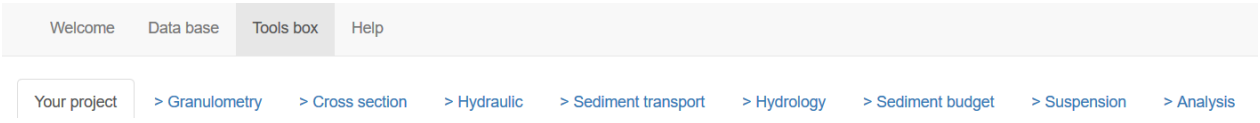
Entering "€UIANA" in the find option of the R menu, gives access directly to the part of the UI code at the transition between the page 'Sediment Budget' and the page 'Analysis', where we want to install the new page.

```
3125          )
3126       ),#END UI SEDIMENT BUDGET
3127 #€UIANA#############################################################################################################
3128 #|||||||||||||||||||||||||||||||||||||||||||||||||   UI ANALYSIS  |||||||||||||||||||||||||||||||||||||||||||||||||
3129 ######################################################################################################################
3130     tabPanel(VAR$L[VAR$N=="STitre9"],#icon = icon("calculator"),
```

The command for creating a new page at the same level than the Sediment Budget page is 'tabPanel()' to be inserted just after the Sediment Budget UI code:

```
3126       ),#END UI SEDIMENT BUDGET
3127     tabPanel(VAR$L[VAR$N=="STitre11"]
3128
3129
3130     ),
3131 #€UIANA#############################################################################################################
3132 #|||||||||||||||||||||||||||||||||||||||||||||||||   UI ANALYSIS  |||||||||||||||||||||||||||||||||||||||||||||||||
3133 ######################################################################################################################
```

The title of the panel must be defined in the variable.txt file where we create the value STitre11= Suspension. When launching the program we get:



Name	French	English
Titre1	Bienvenue	Welcome
Titre2	La base de données	Data base
Titre3	Boite à outils	Tools box
Titre4	Aide	Help
STitre1	Sélectionner une rivière	Select a river
STitre2	Sélection multicritère	Multicriteria selection
STitre3	Votre projet	Your project
STitre4	> Granulométrie	> Granulometry
STitre5	> Section	> Cross section
STitre6	> Hydraulique	> Hydraulic
STitre7	> Transport solide	> Sediment transport
STitre8	> Hydrologie	> Hydrology
STitre9	> Analyse	> Analysis
STitre10	> Bilan sédimentaire	> Sediment budget
STitre11	> Suspension	> Suspension

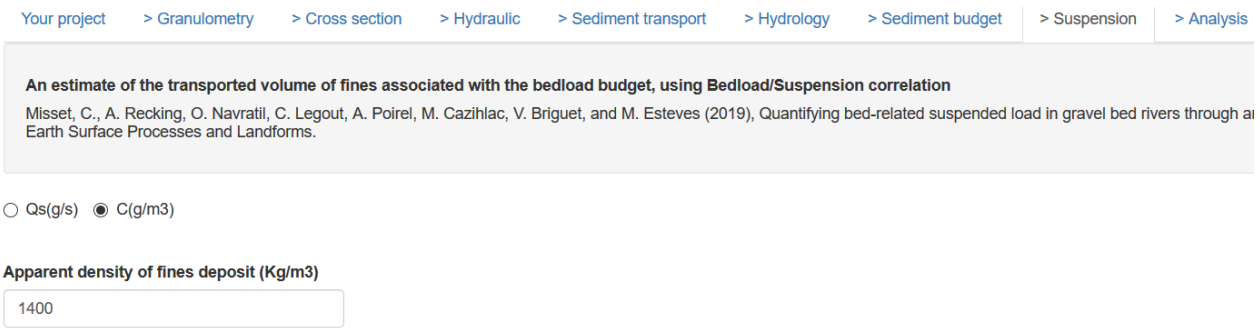
Now we must write the UI code defining the content of the page.

```
3127 S#########################################################################################################################
3128 ||||||||||||||||||||||||||||||||||   UI SUSPENSION|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
3129 #########################################################################################################################
3130 bPanel(VAR$L[VAR$N=="STitre11"],
3131        wellPanel(
3132          tags$style(type="text/css", "#titreSusp {text-align:center;}"),
3133          h5(strong(("An estimate of the transported volume of fines associated with the bedload budget,
3134                     using Bedload/Suspension correlation"))),
3135          h5(("Misset, C., A. Recking, O. Navratil, C. Legout, A. Poirel, M. Cazihlac, V. Briguet, and
3136              M. Esteves (2019), Quantifying bed-related suspended load in gravel bed rivers through an
3137              analysis of the bedload-suspended load relationship, Earth Surface Processes and Landforms. "))
3138        ),
3139        radioButtons("OptPlotSUS", label = NULL,
3140                     choiceNames = list("Qs(g/s)","C(g/m3)"),
3141                     choiceValues = list("1","2"),
3142                     selected = "2",
3143                     inline = TRUE
3144        ),
3145        br(),
3146        textInput("RhoSUS",label="Apparent density of fines deposit (Kg/m3)",value=1400),
3147        plotOutput('plotSUS') %>% withSpinner(color="#0dc5c1"),
3148        h4(textOutput("VolumeSusp")),
3149        h4(textOutput("VolumeSuspPerYear")),
3150        downloadButton("downloadDataSUS","Download results",width="80px",style ="color:blue")
```

In this code we include:

- A title "Correlation bedload Suspension"
- A widget radioButtons giving access the unit options in g/s or in g/m3
- A graphical display of the result with plotOutput
- A text giving the result of total volume transported
- A download button for downloading he results

When launching the code we get:



With no figure yet, because we must define the calculation in the Server function.

4.3.3 The server function

The server must contain all the functions controlling the objects defined in the UI page. These function can be inserted anywhere in the Server part.

A suspension budget will be computed for the active section, and the different steps are:

- Reading the hydrographs associated with the sediment budget for the section
- Computing bedload for each time step of the hydrograph
- Transforming bedload to suspension with Equation 2
- Summing the volume transported at each time step for a total volume

The needed data already exist:

- The hydrograph discretized in the Sediment Budget page is RescalHYDBIL()
- The name of this Hydrograph is NameHYDBIL()
- Additional iinformation about this hydrograph are in SauvegardeHYD2()
- Bedload models are available in RescalGraphPro() for the active section
- The bed D_{84} in D84_Lit_Actif()

The program reads the value entered for the apparent fine sediment density:

```
14999  ## APPARENT DENSITY FOR FINE SEDIMENT IN PLACE
15000  ## (ACCCOUNTING FOR INTERGRAINS SPACE)
15001  RhoSUS<-reactive({
15002    RhoSUS<-as.numeric(input$RhoSUS)
15003    if (is.na(RhoSUS)) RhoSUS<-1400
15004    RhoSUS
15005  })
```

Then it reads the name of the Hydrograph NameHYDBIL() associated with the active section, and extracts the associated unit time in hydrology MetaData save file SauvegardeHYD2()

```
15007 ## Extract the unit time of the hydrograph used in the Sediment Budget page
15008 UnitTempsSUS<-reactive({
15009   selectHYD<-NameHYDBIL()
15010   verif<-str_detect(selectHYD, "HYD")
15011   if (verif==TRUE & nchar(selectHYD)>0 & !is.null(selectHYD) & !is.na(selectHYD))
15012   {
15013     SauvegardeHYD2<-SauvegardeHYD2()
15014     SaisieHYD2<-selectHYD2_f(selectHYD,SauvegardeHYD2)
15015     UnitTempsBIL<-SaisieHYD2[1,3]
15016     if (nchar(UnitTempsBIL)>0 & !is.null(UnitTempsBIL) & !is.na(UnitTempsBIL)) UnitTempsBIL<-UnitTempsBIL
15017     else UnitTempsBIL<-3
15018   } else UnitTempsBIL<-3
15019   UnitTempsSUS<-UnitTempsBIL
15020 })
```

In a second step we prepare the data and start the step by step computation; results are in the RescalSUS() matrix. Some data such as D50, slope and KvsD are used to check the consistency (calculation stops if these data do not exist). The RescalSUS_f function is in BedloadR.R.

```
15022 ## Calculate Suspension for each time step of the Hydrograph
15023 RescalSUS<-reactive({
15024   RhoEau<-RhoEau()
15025   RhoSed<-RhoSed()
15026   s<-RhoSed/RhoEau
15027   Tarage<-Tarage()
15028   KvsD<-KvsD()
15029   Slope<-Slope()
15030   D50<-D50_LitActif()/1000
15031   D84<-D84_LitActif()/1000
15032   OptHydro<-input$OptHydroBIL #1=HYD, 2=QCL
15033   UnitTemps<-UnitTempsSUS()#1=s, 2=mn, 3=h, 4=d
15034   RescalGraph<-RescalGraphPro()#Bedload transport in g/m/s
15035   RescalHYD<-RescalHYDBIL()# Bedload for each time step of the hydrograph
15036   RescalSUS<-RescalSUS_f(OptHydro,RescalGraph,RescalHYD,UnitTemps,Tarage,KvsD,Slope,D50,D84,RhoSed,s)
```

The next functions use the results **RescalSUS()** for computing and displaying the total and annual transported volumes:

```
15039 ## Calculate and display the total volume
15040 output$VolumeSusp<-renderText({
15041   RescalSUS<-RescalSUS()
15042   RhoSed<-RhoSed()
15043   RhoSUS<-RhoSUS()
15044   VolumeSusp<-VolumeSusp_f(RescalSUS,RhoSed,RhoSUS)
15045 })
15046
15047 ## Calculate and display he total annual volume
15048 output$VolumeSuspPerYear<-renderText({
15049   RescalSUS<-RescalSUS()
15050   RhoSed<-RhoSed()
15051   RhoSUS<-RhoSUS()
15052   UnitTemps<-UnitTempsSUS()#1=s, 2=mn, 3=h, 4=j
15053   VolumeSuspPerYear<-VolumeSuspPerYear_f(RescalSUS,UnitTemps,RhoSed,RhoSUS)
15054 })
```

A function is also needed for graphical display:

```
15056 ## GRAPHICAL OUTPUT
15057 output$plotSUS <- renderPlot({
15058   OptPlotSUS<-input$OptPlotSUS
15059   Legende<-input$LegendeHYD
15060   SaisieHydro<-SaisieHydroBIL()
15061   UnitTemps<-UnitTempsBIL()
15062   UnitTemps<-paste0(VAR$L[VAR$N=="BIL58"]," (", UnitTemps,")")
15063   UnitQ<-UnitQBIL()
15064   UnitMasse<-input$UnitMasse
15065   ReelApp<-input$ReelApp
15066   RhoSed<-RhoSed()
15067   Legende<-"TRUE"
15068   if (!is.null(RescalSUS())) RescalSUS<-RescalSUS()
15069   plotSUS_f(UnitTemps,Legende,UnitMasse,UnitQ,RescalSUS,SaisieHydro,OptPlotSUS,RhoSed)
15070 })
```

Finally a last function allows downloading results in txt format.

```
15072 output$downloadDataSUS <- downloadHandler(
15073     filename = function() {
15074       paste("Suspension.txt")
15075     },
15076     content = function(file) {
15077       RescalSUS<-RescalSUS()
15078       write.table(RescalSUS, file, row.names = FALSE, sep ="\t",append=FALSE, quote = FALSE)
15079     }
15080   )
```

The server functions call several functions (RescalSUS_f, VolumeSusp_f,…) which are specifically dedicated to calculation, and hosted in the BedloadR.R file.

4.3.4 The BedloadR.R functions

In this example we consider only calculation with Hydrographs (HYD). If the active hydrology is a flow duration curve (option QCL, OpyHydro=2), the program returns an empty matrix:

```
6371 RescalSUS_f<-function(OptHydro,RescalGraph,RescalHYD,UnitTemps,Tarage, KvsD,Slope,D50,D84,RhoSed,s)
6372 {
6373   if(OptHydro==1)
6374   {}
6404   else
6404   {
6405     RescalSUS<-matrix(0,5,7)
6406     RescalSUS<-data.frame(RescalSUS)
6407     colnames(RescalSUS)<-c("Time", "dt","Q","Reck","C","MES","Vdt")
6408     RescalSUS$Reck<-0
6409     RescalSUS$MES<-0
6410     RescalSUS$C<-0
6411     RescalSUS$Vdt<-0
6412   }
6413   return(RescalSUS)
6414 }
```

When OpyHydro=1 (option HYD) the program computes suspension:

```
6373  if(OptHydro==1)
6374  {
6375    RescalSUS<-subset(RescalHYD, select=c("Time", "dt","Q","Reck"))## select
6376    L1<-length(RescalSUS[,1])#compte le nombre de lignes
6377    matSus<-matrix(NA,L1,3)
6378    matSus<-data.frame(matSus)
6379    colnames(matSus)<-c("C","MES","Vdt")
6380    RescalSUS<-cbind(RescalSUS,matSus)
6381    if (length(Tarage[,1])>1 & (KvsD[2,2]>0|KvsD[2,3]>0) & D50>0 & D84>0 & Slope>0 & !is.na(Slope))
6382    {
6383      if (length(na.omit(RescalGraph$qscalReck))>1) RescalSUS$Reck<-approx(x=RescalGraph$Q,
6384                                                                           y=RescalGraph$qscalReck,
6385                                                                           xout=RescalSUS$Q)$y
6386      else RescalSUS$Reck<-0
6387      RescalSUS$Reck<-RescalSUS$Reck/(RhoSed*1000*sqrt(9.81*(s-1)*D84^3))
6388      RescalSUS$C[RescalSUS$Reck<8.49e-6]<-4.04e-5*RescalSUS$Reck[RescalSUS$Reck<8.49e-6]^1.87e-1
6389      RescalSUS$C[RescalSUS$Reck>=8.49e-6]<-9.12e-2*RescalSUS$Reck[RescalSUS$Reck>=8.49e-6]^8.49e-1
6390      RescalSUS$MES<-RescalSUS$C*RescalSUS$Q*RhoSed*1000# en g/s
6391      RescalSUS$Vdt<-RescalSUS$MES*RescalSUS$dt
6392      if (UnitTemps==2) RescalSUS$Vdt<-RescalSUS$Vdt*60
6393      if (UnitTemps==3) RescalSUS$Vdt<-RescalSUS$Vdt*3600
6394      if (UnitTemps==4) RescalSUS$Vdt<-RescalSUS$Vdt*3600*24
6395    }
6396    else
6397    {
6398      RescalSUS$Reck<-0
6399      RescalSUS$MES<-0
6400      RescalSUS$C<-0
6401      RescalSUS$Vdt<-0
6402    }
6403  } else
```

This part of the code is detailed hereafter:

- Lines 6375-6380: we create a matrix for exporting the results:
- Lines 6381 : we check for consistancy (returns zero values if some data are missing)
- Lines 6383-6386: we prepare the Bedload data. Here only the Recking (Reck) equation is considered for the Bedload/Suspension correlation (because this equation was used in Misset et al 2019 presenting the method), but any other equation could be used (according the confidence the user has in the capacity of the equation to reproduce a realistic bedload). For computing suspension with the above equation, bedload must be given in g/s/m. Bedload values in RescalHYDBIL() are given for total section (Qs=qs*W) and in the unit selected by the user (can be g, Kg, tons or m3). We could choose to convert this column into the required unit, but a simplest way used here consist to extract bedload from RescalGraphPRO() where the unit is g/s/m.
- Lines 6387-6390: once we have the bedload values we compute suspension with Eq.2
- Lines 6391-6394: Finally the suspended load computed at each time step is transformed in a volume for the time step duration

The next function calculates the total volume by summing the column in the RescalSUS() matrix (result is sent to the server function):

```
6417 VolumeSusp_f<-function(RescalSUS,RhoSed,RhoSUS)
6418 {
6419   VolumeSusp<-round(sum(na.omit(RescalSUS$Vdt))/(RhoSed*1000),0)
6420   VolumeSuspApp<-round(VolumeSusp*RhoSed/RhoSUS,0)
6421   VolumeSusp<-paste0("Volume= ",VolumeSusp," m3 ,Volume apparent= ",VolumeSuspApp," m3")
6422   return(VolumeSusp)
6423 }
```

The next function computes the volume per year when it's possible (result is sent to the server function):

```
6425 VolumeSuspPerYear_f<-function(RescalSUS,UnitTemps,RhoSed,RhoSUS)
6426 {
6427   VolumeSusp<-sum(na.omit(RescalSUS$Vdt))/(RhoSed*1000)
6428   D<-sum(RescalSUS$dt)
6429   if ((UnitTemps==4) && D>365)
6430   {
6431     VolumeSuspPerYear<-round(VolumeSusp/(D/365),0)
6432   } else VolumeSuspPerYear<-0
6433   VolumeSuspPerYearApp<-round(VolumeSuspPerYear*RhoSed/RhoSUS,0)
6434   VolumeSuspPerYear<-paste0("Volume= ",VolumeSuspPerYear," m3/an ,Volume apparent= ",
6435                             VolumeSuspPerYearApp," m3/an")
6436   return(VolumeSuspPerYear)
6437 }
```

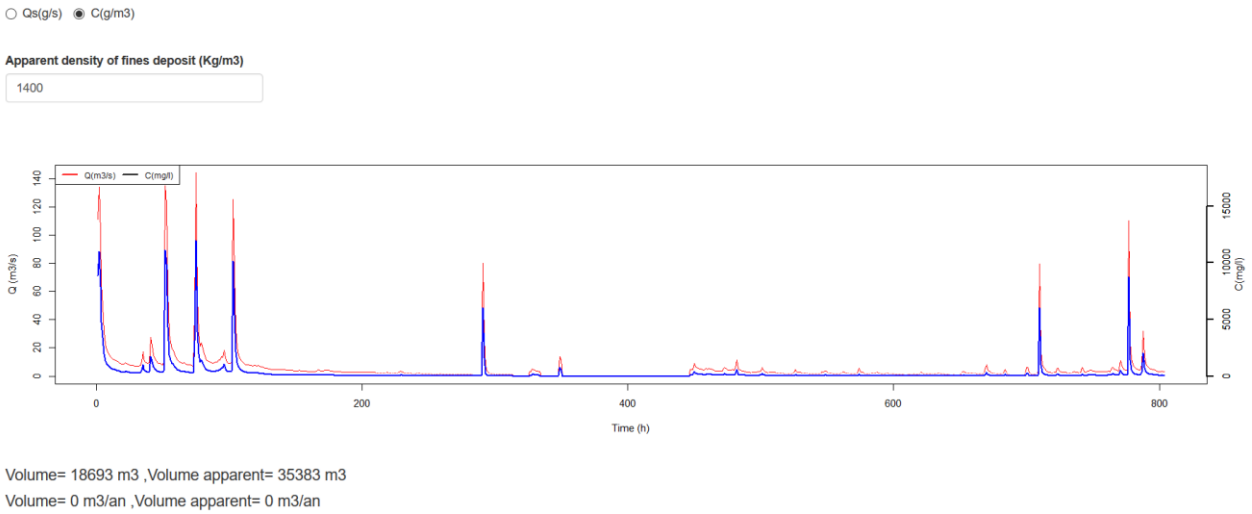
The next function prepares the figure:

```
6439 plotSUS_f<-function(UnitTemps,Legende,UnitMasse,UnitQ,RescalSUS,SaisieHydro,OptPlotSUS,RhoSed)
6440 {
6441   UnitQ<-paste0("Q (",UnitQ,")")
6442   par(mar = c(6, 4, 4, 4))#PLOT A (empty) DEFAUT FIGURE
6443   plot(x=SaisieHydro[,1],y=SaisieHydro[,2],pch=20,col="red",type="l",lwd=1
6444        ,xlab=UnitTemps,ylab=UnitQ,showWarnings =FALSE
6445   )
6446   if(OptPlotSUS==1)# COLLECT THE DATA IN THE APPROPRIATE UNITS
6447   {
6448     RescalSUS$C<-RescalSUS$MES
6449     col=1
6450   }else{
6451     RescalSUS$C<-RescalSUS$C*RhoSed*1000
6452     col=4
6453   }
6454   Cmax<-max(RescalSUS$C)
6455   par(new=TRUE)# REPLACE THE DEFAULT FIGURE
6456   plot(x=RescalSUS$Time,y=RescalSUS$Q,pch=3,col="white",ylim=c(0,1.5*Cmax),# PLOT HYDROLOGY
6457        xlab="",ylab="",showWarnings =FALSE,axes=FALSE)
6458   axis(side=4)
6459   points(RescalSUS$Time, RescalSUS$C,type="l",col=col,lwd=2,# PLOT SUSPENSION
6460          showWarnings =FALSE,ylab="",axes=FALSE)
6461   if(OptPlotSUS==1)
6462   {
6463     leg<-c("Q(m3/s)","Qs(g/s)")# BUILT LEGEND
6464     Ylabel<-"Qs(g/s)"
6465   } else  {
6466     leg<-c("Q(m3/s)","C(mg/l)")
6467     Ylabel<-"C(mg/l)"
6468   }
6469   axis(4, ylim=c(0,1.5*Cmax),lwd=2,col="black")
6470   mtext(Ylabel, side=4, line=2)
6471   if (Legende==TRUE) legend("topleft",legend=leg,ncol=2,lty=c(1,1),# DISPLAY LEGEND
6472                             lwd=2,cex=0.8,col=c(2,1),pt.cex = .8)
6473 }
```

4.3.5 Test with the BedloadWeb_Demo.txt

To activate the suspension module you must select the corresponding part of the code (by searching for '€SUS') in UI, Server and BedloadWeb, and remove # at the beginning of each line with Ctrl+Shift+C.

Download the BedloadWeb_Demo.txt file available in the help tab and creates a new project. Open SEC2, which sediment budget was computed with a hydrograph. Check the results in the suspension page.



Go to the hydrology page, open HYD1, change the hydrograph unit to 'days' and save. Now results are also given in m3/year.

4.3.6 Adding a menu

It is possible to add a menu for managing the new module (saving..).

The algorithm (for counting, ranking, saving with NewSave() matrices ...) can be copied and pasted from the other pages.

The data can be saved in the columns still available in the matrix SauvegardeDataPlus(), as what has been done when adding an equation in the above example, or new Sauvegarde .txt files can be defined.

In all cases, when the new code allows saving new data, what is important is to make sure that it will also accept to read the old data (project already saved).

5 APPENDIX: FILES FORMAT

5.1Grain Size Distribution

SauvegardeGSDECH()

Contains post-treated data for all samples ECH and all GSD

NameGSDECH	D(mm)	%	Cumul
GSD1ECH1	0.0312	0.0446	0.0446
GSD1ECH1	0.0442	0.0185	0.0631
GSD1ECH1	0.0625	0.0262	0.0893
GSD1ECH1	0.0884	0.0370	0.1263
GSD1ECH1	0.1250	0.0523	0.1786
GSD1ECH1	0.1768	0.0740	0.2525
GSD1ECH1	0.2500	0.1046	0.3571
GSD1ECH1	0.3536	0.1479	0.5051
GSD1ECH1	0.5000	0.2092	0.7143
GSD1ECH1	0.7071	0.2959	1.0102

…

GSD4ECH1	45.2548	4.0000	40.0000
GSD4ECH1	64.0000	7.0000	47.0000
GSD4ECH1	90.5097	11.0000	58.0000
GSD4ECH1	128.0000	15.0000	73.0000
GSD4ECH1	181.0193	11.0000	84.0000
GSD4ECH1	256.0000	16.0000	100.0000
GSD4ECH1	362.0387	0.0000	100.0000
GSD4ECH1	512.0000	0.0000	100.0000
GSD4ECH1	724.0773	0.0000	100.0000
GSD4ECH1	1024.0000	0.0000	100.0000
GSD4ECH1	1448.1547	0.0000	100.0000
GSD4ECH1	2048.0000	0.0000	100.0000
GSD4ECH1	2896.3094	0.0000	100.0000

SauvegardeECH()

Contains raw data of all samples ECH for the active GSD.

It is a temporary file which is updated each time a new GSD is open (not saved as txt ith the project files).

NameGSD	NameECH	OptionSaisieGSD	OptionSaisiePhi	D50SAISIE	D84SAISIE	FsSAISIE	DmSAISIE	D	Nbre
GSD1	ECH1	1	3	NA	NA	NA	NA	4.0000	6
GSD1	ECH1	NA	NA	NA	NA	NA	NA	5.6000	0
GSD1	ECH1	NA	NA	NA	NA	NA	NA	8.0000	1
GSD1	ECH1	NA	NA	NA	NA	NA	NA	11.0000	1
GSD1	ECH1	NA	NA	NA	NA	NA	NA	16.0000	2
GSD1	ECH1	NA	NA	NA	NA	NA	NA	22.0000	7
GSD1	ECH1	NA	NA	NA	NA	NA	NA	32.0000	10
GSD1	ECH1	NA	NA	NA	NA	NA	NA	45.0000	18
…									
GSD1	ECH2	1	3	NA	NA	NA	NA	4.0000	11
GSD1	ECH2	NA	NA	NA	NA	NA	NA	5.6000	0
GSD1	ECH2	NA	NA	NA	NA	NA	NA	8.0000	0
GSD1	ECH2	NA	NA	NA	NA	NA	NA	11.0000	4
GSD1	ECH2	NA	NA	NA	NA	NA	NA	16.0000	10
GSD1	ECH2	NA	NA	NA	NA	NA	NA	22.0000	7
GSD1	ECH2	NA	NA	NA	NA	NA	NA	32.0000	11
GSD1	ECH2	NA	NA	NA	NA	NA	NA	45.0000	12
…									
GSD1	ECH3	2	2	25	NA	NA	NA	2.0000	NA
GSD1	ECH3	NA	NA	NA	NA	NA	NA	4.0000	NA
GSD1	ECH3	NA	NA	NA	NA	NA	NA	8.0000	NA
GSD1	ECH3	NA	NA	NA	NA	NA	NA	16.0000	NA
GSD1	ECH3	NA	NA	NA	NA	NA	NA	32.0000	NA
GSD1	ECH3	NA	NA	NA	NA	NA	NA	64.0000	NA
….									
GSD1	ECH4	1	2	NA	NA	NA	NA	2.0000	11
GSD1	ECH4	NA	NA	NA	NA	NA	NA	4.0000	4
GSD1	ECH4	NA	NA	NA	NA	NA	NA	8.0000	16
GSD1	ECH4	NA	NA	NA	NA	NA	NA	16.0000	14
GSD1	ECH4	NA	NA	NA	NA	NA	NA	32.0000	12
GSD1	ECH4	NA	NA	NA	NA	NA	NA	64.0000	8

SauvegardeGSD()

Contains the raw data for all samples ECH for all GSD saved in the project.

NameGSD	NameECH	OptionSaisieGSD	OptionSaisiePhi	D50SAISIE	D84SAISIE	FsSAISIE	DmSAISIE	D	Nbre
GSD1	ECH1	1	3	NA	NA	NA	NA	4.0000	6
GSD1	ECH1	NA	NA	NA	NA	NA	NA	5.6000	0
GSD1	ECH1	NA	NA	NA	NA	NA	NA	8.0000	1
GSD1	ECH1	NA	NA	NA	NA	NA	NA	11.0000	1
GSD1	ECH1	NA	NA	NA	NA	NA	NA	16.0000	2
GSD1	ECH1	NA	NA	NA	NA	NA	NA	22.0000	7
.....									
GSD1	ECH3	NA	NA	NA	NA	NA	NA	2048.0000	NA
GSD1	ECH4	1	2	NA	NA	NA	NA	2.0000	11
GSD1	ECH4	NA	NA	NA	NA	NA	NA	4.0000	4
GSD1	ECH4	NA	NA	NA	NA	NA	NA	8.0000	16
GSD1	ECH4	NA	NA	NA	NA	NA	NA	16.0000	14
GSD1	ECH4	NA	NA	NA	NA	NA	NA	32.0000	12
GSD1	ECH4	NA	NA	NA	NA	NA	NA	64.0000	8
GSD1	ECH4	NA	NA	NA	NA	NA	NA	128.0000	NA
GSD1	ECH4	NA	NA	NA	NA	NA	NA	256.0000	NA
GSD1	ECH4	NA	NA	NA	NA	NA	NA	512.0000	NA
GSD1	ECH4	NA	NA	NA	NA	NA	NA	1024.0000	NA
GSD1	ECH4	NA	NA	NA	NA	NA	NA	2048.0000	NA
GSD2	ECH2	1	3	NA	NA	NA	NA	4.0000	8
GSD2	ECH2	NA	NA	NA	NA	NA	NA	5.6000	1
GSD2	ECH2	NA	NA	NA	NA	NA	NA	8.0000	4
GSD2	ECH2	NA	NA	NA	NA	NA	NA	11.0000	4
GSD2	ECH2	NA	NA	NA	NA	NA	NA	16.0000	7
.....									
GSD4	ECH1	NA	NA	NA	NA	NA	NA	181.0193	11
GSD4	ECH1	NA	NA	NA	NA	NA	NA	256.0000	16
GSD4	ECH1	NA	NA	NA	NA	NA	NA	362.0387	NA
GSD4	ECH1	NA	NA	NA	NA	NA	NA	512.0000	NA
GSD4	ECH1	NA	NA	NA	NA	NA	NA	724.0773	NA
GSD4	ECH1	NA	NA	NA	NA	NA	NA	1024.0000	NA
GSD4	ECH1	NA	NA	NA	NA	NA	NA	1448.1547	NA
GSD4	ECH1	NA	NA	NA	NA	NA	NA	2048.0000	NA
GSD4	ECH1	NA	NA	NA	NA	NA	NA	2896.3094	NA

SauvegardePhotoGSD()

NameGSD	LinkPhotoGSD	CommentGSD	LongGSD	LatGSD	METHODMEAS_GSD
GSD0	XXXX	test	44Â°58'40.29	5Â°44'2.29	1
GSD1	XXXX	4 échantillons sur le même site			1
GSD2	XXXX	2 échantillons			1
GSD3	XXXX	3 échantillons			1
GSD4	XXXX	1 échantillon			1

NewSaveECH ()

Raw data for the active sample to be saved (built when saving a sample)

NameGSD	NameECH	OptionSaisieGSD	OptionSaisiePhi	D50SAISIE	D84SAISIE	FsSAISIE	DmSAISIE	D	Nbre
GSD1	ECH2	1	3					4.0000	11.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	5.6000	0.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	8.0000	0.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	11.0000	4.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	16.0000	10.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	22.0000	7.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	32.0000	11.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	45.0000	12.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	64.0000	16.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	90.0000	11.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	128.0000	6.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	180.0000	6.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	256.0000	3.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	360.0000	2.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	512.0000	1.0000
GSD1	ECH2	NA	NA	NA	NA	NA	NA	720.0000	0.0000

OptionSaisieGSD= data or model
OptionSaisiePhi= options Phi, 0.5Phi or free for the editing table
D50SAISIE= D50 entered for the GSD model
D84SAISIE= D84 entered for the GSD model
FsSAISIE= Sand Fraction Fs entered for the GSD model
DmSAISIE= Minimum diameter entered for the GSD model

NewSaveGSDECH ()

Computed data for the active sample to be saved

NameGSDECH	D(mm)	%	Cumul
GSD1ECH2	0.0312	0.0859	0.0859
GSD1ECH2	0.0442	0.0356	0.1215
GSD1ECH2	0.0625	0.0503	0.1719
GSD1ECH2	0.0884	0.0712	0.2431
GSD1ECH2	0.1250	0.1007	0.3438
GSD1ECH2	0.1768	0.1424	0.4861
GSD1ECH2	0.2500	0.2014	0.6875
GSD1ECH2	0.3536	0.2848	0.9723
GSD1ECH2	0.5000	0.4027	1.3750
GSD1ECH2	0.7071	0.5695	1.9445
GSD1ECH2	1.0000	0.8055	2.7500
GSD1ECH2	1.4142	1.1391	3.8891
GSD1ECH2	2.0000	1.6109	5.5000
GSD1ECH2	2.8284	2.2782	7.7782
GSD1ECH2	4.0000	3.2218	11.0000
GSD1ECH2	5.6569	0.0000	11.0000
GSD1ECH2	8.0000	0.0000	11.0000
GSD1ECH2	11.3137	4.6274	15.6274
GSD1ECH2	16.0000	9.3726	25.0000
GSD1ECH2	22.6274	7.6902	32.6902
GSD1ECH2	32.0000	10.3098	43.0000
GSD1ECH2	45.2548	12.2146	55.2146
GSD1ECH2	64.0000	15.7854	71.0000

NewSaveGSD ()

NewSaveGSD is SauvegardeECH() (see above)

NameGSD	NameECH	OptionSaisieGSD	OptionSaisiePhi	D50SAISIE	D84SAISIE	FsSAISIE	DmSAISIE	D	Nbre
GSD1	ECH1	1	3					4.0000	6.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	5.6000	0.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	8.0000	1.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	11.0000	1.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	16.0000	2.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	22.0000	7.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	32.0000	10.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	45.0000	18.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	64.0000	16.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	90.0000	16.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	128.0000	12.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	180.0000	10.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	256.0000	4.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	360.0000	1.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	512.0000	1.0000
GSD1	ECH1	NA	NA	NA	NA	NA	NA	720.0000	0.0000
GSD1	ECH2	2	2	22	56			2.0000	NA
GSD1	ECH2	NA	NA	NA	NA	NA	NA	4.0000	NA
GSD1	ECH2	NA	NA	NA	NA	NA	NA	8.0000	NA
GSD1	ECH2	NA	NA	NA	NA	NA	NA	16.0000	NA
GSD1	ECH2	NA	NA	NA	NA	NA	NA	32.0000	NA
GSD1	ECH2	NA	NA	NA	NA	NA	NA	64.0000	NA
GSD1	ECH2	NA	NA	NA	NA	NA	NA	128.0000	NA
GSD1	ECH2	NA	NA	NA	NA	NA	NA	256.0000	NA
GSD1	ECH2	NA	NA	NA	NA	NA	NA	512.0000	NA
GSD1	ECH2	NA	NA	NA	NA	NA	NA	1024.0000	NA
GSD1	ECH2	NA	NA	NA	NA	NA	NA	2048.0000	NA

MeanGSD()

Each time a project is opened, the program reconstructs the MeanGSD matrix by averaging (post-treated) data saved in SauvegardeGSDECH()

D	GSD1	GSD2	GSD3	GSD4
0.0312	0.1378	0.1080	0.1562	0.1719
0.0442	0.1949	0.1527	0.2210	0.2431
0.0625	0.2756	0.2160	0.3125	0.3437
0.0884	0.3898	0.3054	0.4419	0.4861
0.1250	0.5513	0.4319	0.6250	0.6875
0.1768	0.7796	0.6108	0.8839	0.9723
0.2500	1.1025	0.8638	1.2500	1.3750
0.3536	1.5592	1.2217	1.7678	1.9445
0.5000	2.2050	1.7277	2.5000	2.7500
0.7071	3.1184	2.4433	3.5355	3.8891
1.0000	4.4100	3.4554	5.0000	5.5000

…..

181.0193	96.8183	97.5268	98.2224	84.0000
256.0000	98.7738	99.5000	99.4424	100.0000
362.0387	99.5185	100.0000	99.7638	100.0000
512.0000	100.0000	100.0000	100.0000	100.0000
724.0773	100.0000	100.0000	100.0000	100.0000
1024.0000	100.0000	100.0000	100.0000	100.0000
1448.1547	100.0000	100.0000	100.0000	100.0000
2048.0000	100.0000	100.0000	100.0000	100.0000
2896.3094	100.0000	100.0000	100.0000	100.0000

5.2Sections

SauvegardeSEC()

Contains all the sections defined in the project (RAW DATA)

```
8980 memorySEC <- reactiveValues(dat = NULL)
8981 SauvegardeSEC<-reactive({
8982   SauvegardeSEC<-memorySEC$dat
8983 })
```

nameSEC	X	Z
SEC1	0.0000	100.0000
SEC1	0.0000	99.9300
SEC1	1.2000	99.8700
SEC1	2.7000	99.4800
SEC1	4.0000	98.9900
SEC1	4.5000	98.7000
SEC1	5.3000	98.4900

.....

SEC3	113.5000	12.0100
SEC3	113.0000	12.4600
SEC3	116.0000	12.5000
SEC3	116.0000	12.5700
SEC4	0.0000	3.0000
SEC4	0.0000	0.0000
SEC4	14.0000	0.0000
SEC4	14.0000	3.0000

SauvegardeDEF()

Contains definition of the sections components (RAW DATA, limits of sliders values)

```
8995 memoryDEF <- reactiveValues(dat = NULL)
8996 SauvegardeDEF<-reactive({
8997   SauvegardeDEF<-memoryDEF$dat
8998 })
```

nameSEC	LM1	LM2	LA1	LA2	LFRD1	LFRD2	LFRG1	LFRG2	RUG1	RUG2	CH1	CH2	Slope	coteCH2
SEC1	1.2500	16.9100	5.0720	15.6910	0.0000	1.2500	16.9100	18.7000	0.0000	0.0000	0.0000	0.0000	0.005	0
SEC2	0.0000	24.3000	1.3970	22.4880	0.0000	0.0000	24.3000	24.3000	0.0000	0.0000	0.0000	0.0000	0.007	0
SEC3	58.8300	95.1400	62.7580	86.7810	0.0000	20.3150	104.3200	104.3200	0.0000	63.8400	98.4700	115.1600	0.005	0
SEC4	0.0000	14.0000	0.0500	13.9500	0.0000	0.0000	14.0000	14.0000	0.0000	0.0000	0.0000	0.0000	0.006	0

LM= Main channel
LA= active bed
LFRD= Right bank
LFRG= left bank
RUG= roughness zone
CH= Secondary channel

SauvegardeMdata()
Contains MetaData for all sections (RAW DATA)

nameSEC	CortPRO	WallEffectPRO	OptionParkerPRO	betaPRO	gamma2PRO	CortHYD	WallEffectHYD	OptionParkerHYD	UnitQ	UnitTemps	UnitMasse
SEC3	0	1	0	2	20	0	1	0	2	3	3
SEC2	0	1	0	2	20	0	1	0	2	3	3
SEC1	0	1	0	2	20	0	1	0	2	3	3
SEC4	0	0	1	2	20	0	0	1	2	3	3

UnitMasse	HYD	QCL	FitGTMPRO	OptHydro	BVREF	BVSEC	MYER	MUR	HMUR	UnitQs	RhoEau	RhoSed	RhoApp
3	HYD1		2	1	122	85	0.8000	0	2	1	1000	2650	2000
3	HYD1		2	1	NA	NA	0.8000	1	2	1	1000	2650	2000
3	HYD2	QCL1	2	2	NA	NA	0.8000	1	2	1	1000	2650	2000
3	HYD1	QCL1	2	1	NA	NA	0.8000	0	2	1	1000	2650	2000

CortPRO= Shear stress correction in Sediment Transport page
WallEffectPRO=wall correction if flume in Sediment Transport page
OtionParkerPRO= suppress the sand fraction in GSD in Sediment Transport page
betaPRO= Parameter of the GTM model (in Sediment Transport page)
gamma2PRO= Parameter of the GTM model (in Sediment Transport page)
CortHYD= Shear stress correction in Sediment budget page (=CortPRO)
WallEffectHYD=wall correction if flume in Sediment budget page (=WallEffectPRO)
OtionParkerHYD= suppress the sand fraction in Sediment budget page (=OtionParkerPRO)
UnitQ= unit for discharge (Hydrology page)
UnitTemps= time unit (Hydrology Page)
UnitMasse= unit for bedload budget (Sediment Budget page)
HYD= Hydrograph associated with the section (Sediment Budget page)
QCL= Flow Duration Curve associated with the section (Sediment Budget page)
FitGTMPRO= Option for the GTM model (in Sediment Transport Page)
BVREF= Size of the Referent watershed (Sediment Budget page)
BVSEC= Size of the Section Watershed (Sediment Budget page)
MYER= MYER exponent (Sediment Budget page)
MUR= Add vertical walls to the section (Sediment Budget page)
HMUR= Height of the Walls
UnitQs= Unit for Qs plot (var ParHydrauGraphYPRO in Sediment Transport page)
RhoEau=Water density (RhoEauPRO in in Sediment Transport page)
RhoSed=Sediment density (RhoSedPRO in in Sediment Transport page)
RhoApp= apparent sediment density (RhoAppPRO in in Sediment Transport page)

SauvegardeDataPlus()

Contains additional MetaData for all sections (RAW DATA)

```
8990 memoryDataPlus <- reactiveValues(dat = NULL)
8991 SauvegardeDataPlus<-reactive({
8992   SauvegardeDataPlus<-memoryDataPlus$dat
8993 })
```

nameSEC	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	V29	V30
SEC3	0	0	0	0	0	0	0	0	0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0	0	0	0	0	0
SEC2	0	0	0	0	0	0	0	0	0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0	0	0	0	0	0
SEC1	0	0	0	0	0	0	0	0	0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0	0	0	0	0	0
SEC4	0	0	0	0	0	0	0	0	0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0	0	0	0	0	0

Only columns 1 and 11 to 23 are used. Other columns are availables for new data.

The values stored are the threshold values retained for each equation.

```
2969     NewSaveDataPlus[1,1]<-nameSEC
2970     NewSaveDataPlus[1,11]<-Omegac
2971     NewSaveDataPlus[1,12]<-TcCamenen
2972     NewSaveDataPlus[1,13]<-q0Lefort
2973     NewSaveDataPlus[1,14]<-TcMPM
2974     NewSaveDataPlus[1,15]<-TcParker
2975     NewSaveDataPlus[1,16]<-TrParker
2976     NewSaveDataPlus[1,17]<-TmRecking
2977     NewSaveDataPlus[1,18]<-qcRickenmann
2978     NewSaveDataPlus[1,19]<-qcSchock
2979     NewSaveDataPlus[1,20]<-TcSmart
2980     NewSaveDataPlus[1,21]<-ucVR
2981     NewSaveDataPlus[1,22]<-TrWC
2982     NewSaveDataPlus[1,23]<-TcWong
```

SauvegardePhoto()

Contains link to photo and some Metadata

nameSEC	LinkPhoto	Comment	LongSEC	LatSEC	MorphoSEC	NomSection
SEC0	XXXX	test	44Â°58'40.29	5Â°44'2.29	6	XXXX
SEC2	XXXX				6	ex avec chronique de Q
SEC3	XXXX				4	ex avec section complexe
SEC4	XXXX				6	ex avec débits classés
SEC1	XXXX				3	

LinkPhoto= address for the photo (page SECTION)
MorphoSEC= code for the section morphology (Sediment Transport page)
Nomsection= full name of the section (page SECTION)

RugoSection()

Assign the beds component definition to each X value(NA everywhere and Z value repeated in the concerned column)

X	Z	Actif	FixeRD	FixeRG	Rugo2	CH2
0.0000	102.0000	NA	NA	NA	NA	102.0000
0.0000	101.9520	NA	NA	NA	NA	101.9520
0.0000	101.9039	NA	NA	NA	NA	101.9039
0.0000	101.8559	NA	NA	NA	NA	101.8559
0.0000	101.8078	NA	NA	NA	NA	101.8078
0.0000	101.7598	NA	NA	NA	NA	101.7598
0.0000	101.7117	NA	NA	NA	NA	101.7117
0.0000	101.6637	NA	NA	NA	NA	101.6637
0.0000	101.6156	NA	NA	NA	NA	101.6156
0.0000	101.5676	NA	NA	NA	NA	101.5676

5.3Hydraulics

SectionDiscret()

Discretize the section with a space step (1000 values)

X	Z	XdX	ZdZ	H	HdH	dP	dA	Type	KvsD	K	D84	U
0.0000	102.0000	0.0000	101.9520	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.9520	0.0000	101.9039	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.9039	0.0000	101.8559	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.8559	0.0000	101.8078	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.8078	0.0000	101.7598	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.7598	0.0000	101.7117	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.7117	0.0000	101.6637	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.6637	0.0000	101.6156	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.6156	0.0000	101.5676	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.5676	0.0000	101.5195	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.5195	0.0000	101.4715	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA
0.0000	101.4715	0.0000	101.4234	1.0000	0.0000	0.0480	0.0000	NA	NA	NA	NA	NA

HydrauParam()

Mix of RugoSection() , SectionDiscret() and KvsD

X	Z	XdX	ZdZ	H	HdH	dP	dA	Type	KvsD	K	D84	U
0.0000	102.0000	0.0000	101.9520	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.9520	0.0000	101.9039	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.9039	0.0000	101.8559	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.8559	0.0000	101.8078	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.8078	0.0000	101.7598	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.7598	0.0000	101.7117	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.7117	0.0000	101.6637	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.6637	0.0000	101.6156	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.6156	0.0000	101.5676	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA
0.0000	101.5676	0.0000	101.5195	1.0000	0.0000	0.0480	0.0000	CH2	2.0000	25.0000	0.0792	NA

SauvegardeKvsD()

Contains roughness values for all sections (RAW DATA)

nameSEC	Type	K	D84	KD	Message	GSD
SEC1	Defaut	25.0000	0.0792	2.0000	0.0000	GSD1
SEC1	LitMineur	25.0000	0.0792	2.0000	0.0000	GSD1
SEC1	LitActif	25.0000	0.0792	2.0000	0.0000	GSD1
SEC1	CH2	25.0000	0.0792	2.0000	0.0000	GSD1
SEC1	LFRD	25.0000	0.0792	1.0000	0.0000	GSD1
SEC1	LFRG	25.0000	0.0792	1.0000	0.0000	GSD1
SEC1	Rugo2	25.0000	0.0792	1.0000	0.0000	GSD1
SEC2	Defaut	25.0000	0.0781	2.0000	0.0000	GSD2
SEC2	LitMineur	25.0000	0.0781	2.0000	0.0000	GSD2
SEC2	LitActif	25.0000	0.0781	2.0000	0.0000	GSD2

……..

SEC3	LFRG	25.0000	0.0792	1.0000	0.0000	GSD1
SEC3	Rugo2	25.0000	0.0792	1.0000	0.0000	GSD1
SEC4	Defaut	25.0000	0.1810	2.0000	0.0000	GSD4
SEC4	LitMineur	25.0000	0.1810	2.0000	0.0000	GSD4
SEC4	LitActif	25.0000	0.1810	2.0000	0.0000	GSD4
SEC4	CH2	25.0000	0.0792	2.0000	0.0000	GSD1
SEC4	LFRD	25.0000	0.0792	1.0000	0.0000	GSD1
SEC4	LFRG	25.0000	0.0792	1.0000	0.0000	GSD1
SEC4	Rugo2	25.0000	0.0792	1.0000	0.0000	GSD1

Type= Main Channel, Secondary Channel, Roughness zone, bank…
K= Strickler value
D84= D84 of the selected grain size distribution
GSD= selected GSD
Message= a code signaling a problem in the roughness selction (not used)

AnalyseLitMin()

Matrix of hydraulics resulst for the main channel (calculation for each H)

H	P	A	R	U
0.0000	0.0000	0.0000	0.0000	0.0000
0.0400	1.1322	0.0228	0.0202	0.0199
0.0800	2.1443	0.0892	0.0416	0.0579
0.1200	2.9426	0.1913	0.0650	0.1102
0.1600	4.1071	0.3322	0.0809	0.1497
0.2000	5.2004	0.5172	0.0995	0.1987
0.2400	6.2935	0.7457	0.1185	0.2508
0.2800	7.2632	1.0154	0.1398	0.3105
0.3200	8.2547	1.3264	0.1607	0.3693
0.3600	8.7677	1.6642	0.1898	0.4511
0.4000	9.2556	2.0220	0.2185	0.5302
0.4400	9.7687	2.3996	0.2456	0.6036
0.4800	10.2747	2.7966	0.2722	0.6735
0.5200	10.6469	3.2120	0.3017	0.7490

HydrauResultLitMin()

Main channel hydraulics values (wetted perimeter P, wetted area A, hydraulic radius R, and mean velocity U) for water depth specified by the user (slider). Extrapolated from from AnalyseLitMin()

data
12.9764
8.4170
0.6486
1.4923

HydrauSection() or HydrauS (in function Tarage_f)

For a given water level Heau, and for each part of the section X+dX, U is computed for local flow depth H=Heau-Z in HydrauParam(), and secondly, for part of the section corresponding to 'Main channel' and 'secondary channel', U values are replaced by U values computed in HydrauResultLitMin() and HydrauResultCH2().

HydrauSection() is computed for graphical display

HydrauS is computed at each space step in function Tarage_f for the construction of Tarage().

X	Z	XdX	ZdZ	H	HdH	dP	dA	Type	KvsD	K	D84	U
0.0000	2.0095	0.0000	99.9985	0.0000	0.0000	0.0015	0.0000	CH2	2.0000	25.0000	0.0792	0.0000
0.0000	2.0079	0.0000	99.9969	0.0000	0.0000	0.0015	0.0000	CH2	2.0000	25.0000	0.0792	0.0000
0.0000	2.0064	0.0000	99.9954	0.0000	0.0000	0.0015	0.0000	CH2	2.0000	25.0000	0.0792	0.0000
0.0000	2.0048	0.0000	99.9938	0.0000	0.0000	0.0015	0.0000	CH2	2.0000	25.0000	0.0792	0.0000
0.0000	2.0033	0.0000	99.9923	0.0000	0.0000	0.0015	0.0000	CH2	2.0000	25.0000	0.0792	0.0000
....												
4.1251	0.9269	4.1361	98.9110	0.0000	0.0000	0.0127	0.0000	LitMin	2.0000	25.0000	0.0792	1.4290
4.1361	0.9205	4.1471	98.9047	0.0000	0.0029	0.0127	0.0000	LitMin	2.0000	25.0000	0.0792	1.4290
4.1471	0.9141	4.1582	98.8983	0.0029	0.0093	0.0127	0.0001	LitMin	2.0000	25.0000	0.0792	1.4290
4.1582	0.9077	4.1692	98.8919	0.0093	0.0157	0.0127	0.0001	LitMin	2.0000	25.0000	0.0792	1.4290
4.1692	0.9013	4.1802	98.8855	0.0157	0.0220	0.0127	0.0002	LitMin	2.0000	25.0000	0.0792	1.4290
4.1802	0.8950	4.1912	98.8791	0.0220	0.0284	0.0127	0.0003	LitMin	2.0000	25.0000	0.0792	1.4290
4.1912	0.8886	4.2022	98.8727	0.0284	0.0348	0.0127	0.0003	LitMin	2.0000	25.0000	0.0792	1.4290

Tarage()

H	Q	RLitMin	ALitMin	ULitMin	RCH2	ACH2	UCH2	U	A	P	L	La	Lm
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0401	0.0005	0.0202	0.0230	0.0200	0.0000	0.0000	0.0000	0.0200	0.0229	1.1322	1.1291	1.1291	1.1291
0.0802	0.0052	0.0417	0.0897	0.0581	0.0000	0.0000	0.0000	0.0581	0.0896	2.1684	2.1622	2.1622	2.1622
0.1203	0.0212	0.0651	0.1923	0.1105	0.0000	0.0000	0.0000	0.1105	0.1921	2.9426	2.9309	2.9309	2.9309
0.1604	0.0501	0.0811	0.3339	0.1501	0.0000	0.0000	0.0000	0.1502	0.3337	4.1313	4.1081	4.1081	4.1081
0.2005	0.1036	0.0997	0.5199	0.1993	0.0000	0.0000	0.0000	0.1993	0.5196	5.2246	5.1892	5.1892	5.1892
0.2406	0.1886	0.1188	0.7496	0.2517	0.0000	0.0000	0.0000	0.2517	0.7492	6.2935	6.2462	6.2462	6.2462
0.2807	0.3177	0.1401	1.0206	0.3114	0.0000	0.0000	0.0000	0.3114	1.0202	7.2872	7.2312	7.2312	7.2312
0.3208	0.4942	0.1612	1.3328	0.3709	0.0000	0.0000	0.0000	0.3709	1.3326	8.2547	8.1922	8.1922	8.1922
0.3609	0.7569	0.1904	1.6719	0.4528	0.0000	0.0000	0.0000	0.4528	1.6716	8.7677	8.6967	8.6967	8.6967
0.4009	1.0802	0.2191	2.0309	0.5320	0.0000	0.0000	0.0000	0.5320	2.0306	9.2807	9.2012	9.2012	9.2012
0.4410	1.4589	0.2463	2.4099	0.6054	0.0000	0.0000	0.0000	0.6054	2.4096	9.7857	9.6985	9.6985	9.6985

RLitMin= Hydraulics radius main channel
ALitMin= wetted area main channel
ULitMin=mean flow velocity main channel
RCH2=Hydraulic radius secondary Channel
ACH2=wetted area secondary channel
UCH2=mean velocity secondary channel
U= Section mean flow velocity
A= section wetted area
P= Section wetted perimeter
L=Water surface width
La=Water surface width (active section)
Lm= water surface width (man channel)

5.4Sediment transport

SauvegardeQS()

nameSEC	Q	QS	Type
SEC1	0.0000	0.0000	QH
SEC1	0.0000	0.0000	QS
SEC1	2.0000	1.0000	QS
SEC1	5.0000	250.0000	QS
SEC1	12.0000	1500.0000	QS
SEC2	0.0000	0.0000	QS
SEC2	0.0000	0.0000	QH
SEC3	0.0000	0.0000	QS
SEC3	0.0000	0.0000	QH
SEC4	0.0000	0.0000	QS
SEC4	0.0000	0.0000	QH
SEC5	0.0000	0.0000	QS
SEC5	0.0000	0.0000	QH

This matrix is actually used for both Q(H) and or Qs(Q)

Type indicates which kind of data are concerned .

Column QS can therefore have H values or QS values.

SauvegardeGTM()

All validation data for bedload Grain Size Distribution

nameSEC	nameGTM	beta	gamma2	TfTc	Tc	Dref	Q	phi0	gamma0	gamma1	D	%
SEC1	GTM1	2	20	2	NA	84	12	0.0100	0.1000	0.5000	2	4
SEC1	GTM1	2	20	2	NA	84	12	0.0100	0.1000	0.5000	4	5
SEC1	GTM1	2	20	2	NA	84	12	0.0100	0.1000	0.5000	6	12
SEC1	GTM1	2	20	2	NA	84	12	0.0100	0.1000	0.5000	8	12
SEC1	GTM1	2	20	2	NA	84	12	0.0100	0.1000	0.5000	12	5
SEC1	GTM1	2	20	2	NA	84	12	0.0100	0.1000	0.5000	34	6
SEC1	GTM2	2	20	2	NA	84	12	0.0100	0.1000	0.5000	2	2
SEC1	GTM2	2	20	2	NA	84	12	0.0100	0.1000	0.5000	4	4
SEC1	GTM2	2	20	2	NA	84	12	0.0100	0.1000	0.5000	6	12
SEC1	GTM2	2	20	2	NA	84	12	0.0100	0.1000	0.5000	8	23
SEC1	GTM2	2	20	2	NA	84	12	0.0100	0.1000	0.5000	12	12

5.5 Hydrology

SauvegardeHYD()

Hydraugraphs raw data

nameHYD	T	Q
HYD1	1	111.0000
HYD1	2	133.9020
HYD1	3	100.7330
HYD1	4	65.0770
HYD1	5	49.7860
HYD1	6	31.5820
HYD1	7	22.5410
HYD1	8	18.3640
HYD1	9	16.6760
HYD1	10	14.8810
HYD1	11	13.4830

SauvegardeHYD2()

Hydraugraphs MetaData

nameHYD	UnitQ	UnitTime	NomHydrogramme	NomStation	LongStation	LatStation	NoteHydrogramme
HYD1	2	3	Event 1989	Diren			
HYD2	2	3	Short event	Model			

SauvegardeQCL()

Flow Duration Curves raw data

nameQCL	T	Q
QCL1	99.9945	120.0000
QCL1	99.9863	100.0000
QCL1	99.9726	90.0000
QCL1	99.9452	79.0000
QCL1	99.8630	62.0000
QCL1	99.0000	51.1000
QCL1	98.0000	43.7000
QCL1	95.0000	35.8000
QCL1	90.0000	29.0000
QCL1	80.0000	20.2000
QCL1	70.0000	13.7000
QCL1	60.0000	8.9000
QCL1	50.0000	5.5600
QCL1	40.0000	3.8900
QCL1	30.0000	2.9400
QCL1	20.0000	2.3500
QCL1	10.0000	1.9000
QCL1	5.0000	1.6100
QCL1	2.0000	1.3300
QCL1	1.0000	1.1400

SauvegardeQCL2()

Flow Duration Curves MetaData

nameQCL	NomQCL	NomStationQCL	LongStationQCL	LatStationQCL	NoteQCL
QCL1	Data stat	Diren			

5.6 Sediment Budget

SauvegardeBIL()

Equation	m3	m3/an	SEC3M3333T	SEC3M3333A	SEC2M3333T	SEC2M3333A	SEC1M3333T	SEC1M3333A	SEC4M3333T	SEC4M3333A
Bagnold	0	0	94.1557	0	1789	0	9978.8975	9978.8975	119.4319	0
Camenen Larson	0	0	457.4711	0	13682	0	52357.3768	52357.3768	812.8267	0
Einstein-Brown	0	0	471.9715	0	14464	0	56362.1757	56362.1757	859.2521	0
Engelund	0	0	445.7155	0	3841	0	25564.6408	25564.6408	634.3828	0
Lefort	0	0	502.5724	0	5658	0	17125.8505	17125.8505	401.4029	0
Meyer-P&M	0	0	1374.0415	0	24159	0	157926.3049	157926.3049	2664.5822	0
Parker79	0	0	1023.7778	0	22103	0	115901.7397	115901.7397	1869.8799	0
Parker90	0	0	152.4124	0	3993	0	12988.9067	12988.9067	297.7498	0
Recking	0	0	172.5736	0	2556	0	18284.1183	18284.1183	349.6244	0
Rickenmann	0	0	529.7249	0	17742	0	83453.9113	83453.9113	1252.7923	0
Schocklitch	0	0	492.7008	0	7895	0	32053.1761	32053.1761	401.3563	0
Smart Jaeggi	0	0	419.2401	0	11754	0	70133.1820	70133.1820	1446.0390	0
Van Rijn	0	0	63.8689	0	2907	0	8200.6199	8200.6199	85.7437	0
Wilcok-Crowe	0	0	1188.8756	0	10176	0	82596.6135	82596.6135	4060.7139	0
Wong-Parker	0	0	604.7044	0	11193	0	71626.5054	71626.5054	1174.5350	0

2 columns for each sections named SEC _num+ M3333T ('T' for a Time period volume, in m3) or M3333A ('A' for a mean Annual volume in m3/year). When a hydrograph is used and that the total hydrograph duration is less than 1 year the value in column M3333A is 0.

RescalHYDBIL()

Bedload computed for each of the hydrology time step and each equation

Time	dt	W	Q	d	taux	t50	t84	Bag	CL	Ein	Eng	Lef	MPM	Par	Par90	Reck	Rick	Scho	Sma	VR	Wilc	WONG
1.0000	1.0000	18.7000	111.0000	2.6377	88.3625	0.1949	0.0690	0.0067	0.0576	0.0484	0.0191	0.0253	0.0912	0.0910	0.0243	0.0162	0.0425	0.0248	0.0400	0.0098	0.0505	0.0441
2.0000	1.0000	18.7000	133.9020	2.8984	96.4818	0.2128	0.0753	0.0081	0.0732	0.0630	0.0247	0.0326	0.1082	0.1112	0.0331	0.0203	0.0504	0.0298	0.0478	0.0124	0.0621	0.0525
3.0000	1.0000	18.7000	100.7330	2.5151	84.3836	0.1861	0.0659	0.0060	0.0506	0.0422	0.0167	0.0221	0.0832	0.0817	0.0205	0.0144	0.0389	0.0226	0.0364	0.0086	0.0453	0.0402
4.0000	1.0000	18.7000	65.0770	2.0497	68.2514	0.1505	0.0533	0.0037	0.0268	0.0262	0.0090	0.0117	0.0534	0.0482	0.0089	0.0081	0.0259	0.0144	0.0231	0.0046	0.0269	0.0255
5.0000	1.0000	17.2448	49.7860	1.8147	60.0188	0.1324	0.0468	0.0027	0.0176	0.0183	0.0062	0.0077	0.0400	0.0340	0.0051	0.0056	0.0200	0.0107	0.0173	0.0031	0.0193	0.0190
6.0000	1.0000	14.7764	31.5820	1.4539	49.4891	0.1091	0.0386	0.0016	0.0089	0.0098	0.0035	0.0032	0.0248	0.0191	0.0021	0.0030	0.0125	0.0060	0.0108	0.0015	0.0115	0.0116
7.0000	1.0000	13.5593	22.5410	1.2419	42.1810	0.0930	0.0329	0.0010	0.0047	0.0053	0.0022	0.0014	0.0158	0.0110	0.0009	0.0017	0.0083	0.0034	0.0070	0.0008	0.0073	0.0072
8.0000	1.0000	13.1793	18.3640	1.1313	38.2555	0.0844	0.0299	0.0007	0.0031	0.0034	0.0016	0.0007	0.0116	0.0076	0.0006	0.0011	0.0063	0.0021	0.0052	0.0005	0.0055	0.0052
9.0000	1.0000	12.9967	16.6760	1.0835	36.5439	0.0806	0.0285	0.0006	0.0025	0.0027	0.0014	0.0005	0.0099	0.0063	0.0004	0.0009	0.0055	0.0016	0.0045	0.0004	0.0048	0.0044
10.0000	1.0000	12.8281	14.8810	1.0304	34.5850	0.0763	0.0270	0.0005	0.0019	0.0021	0.0012	0.0003	0.0080	0.0050	0.0004	0.0007	0.0046	0.0010	0.0037	0.0002	0.0040	0.0035

ResclaBIL()

Sediment budget computed for the active section

Columns m3/an is NA if the total hydrograph duration is less than 1 year

Equation	m3 (reel)	m3/an
Bagnold	10553.7284	4797.1493
Camenen Larson	76303.8461	34683.5664
Einstein-Brown	70413.0711	32005.9414
Engelund	34453.1706	15660.5321
Lefort	30456.3044	13843.7747
Meyer-P&M	157483.9497	71583.6135
Parker79	136499.4443	62045.2019
Parker90	29696.3419	13498.3372
Recking	23396.0078	10634.5490
Rickenmann	79922.4546	36328.3885
Schocklitch	34259.9010	15572.6823
Smart Jaeggi	70133.3567	31878.7985

5.7 Analysis

ListSECBIL()

nameSEC	LinkPhoto	Comment	LongSEC	LatSEC	MorphoSEC	NomSection	Compo	M3333T	M3333A
SEC2	XXXX				6	ex avec chronique de Q	SEC2 (ex avec chronique de Q)	SEC2M3333T	SEC2M3333A
SEC3	XXXX				4	ex avec section complexe	SEC3 (ex avec section complexe)	SEC3M3333T	SEC3M3333A
SEC4	XXXX				6	ex avec débits classés	SEC4 (ex avec débits classés)	SEC4M3333T	SEC4M3333A
SEC1	XXXX				3		SEC1 (SEC1)	SEC1M3333T	SEC1M3333A

When the user select Sections SEC to be analyzed, the two last columns re used to extract appropriate columns in SauvegardeBIL().

SauvegardeANA()

SECTION	Equation
4	6
91	1
75	14
1	2
TRUE	3
FALSE	4
FALSE	5
XXXX	12
XXXX	7
XXXX	8
XXXX	9
XXXX	10
XXXX	11
XXXX	15
XXXX	NA
SEC4	NA
SEC1	NA
SEC2	NA
SEC3	NA

Column1 (top): sliders values and other display options

Column1(Bottom): Sections retained by user for the analysis

Column2: The equations retained by user for the analysis