

10. Tuning of autoISF settings for Full Closed Loop aided by the emulator

V.3.8

Please note that with autoISF you are in an early-dev. environment, where the user interface is **not optimized for safety** of users who stray away from intended ways to use. Good safety features exist, but these are only as good as the development-oriented user understands and implements them. This is not a medical product, refer to disclaimer in [section 0](#)



10.1 Installing the Emulator on your PC

- 10.1.1 File structure on your PC
- 10.1.2 Load config and py files
- 10.1.3 Desktop button "Emulation_start"
- 10.1.4 Other software requirements

10.2 Analyzing **loop decisions** in logfiles

- 10.2.1 **noChange.vdf**
- 10.2.2/3 Locate logfiles / prepare Emulator
- 10.2.4 Run emulation and inspect results
 - 10.2.4.1 .txt (all SMB tab infos)
 - 10.2.4.2 Tabular (.csv) presentation of all loop decisions
 - 10.2.4.3 Manual extraction of key data into .xls or .odc
 - 10.2.4.4 .pdf chart
 - 10.2.4.5 delta table

10.3 What-if analysis

- 10.3.1 Define (**yourChange**).vdf
- 10.3.2 Run emulation
- 10.3.3 Inspect results
 - 10.3.3.1 Logs (all SMB tab infos)
 - 10.3.3.2 Tabular (.csv) presentation of all loop decisions
 - 10.3.3.3 Semi-automated extraction of key data
 - 10.3.3.4 .pdf chart
 - 10.3.3.5 delta table

[Available related case studies:](#)

Case studies **still missing:**

Based on older autoISF and older emulator versions, examples from emulator use can be found in [case study 6.2](#), in [case study 4.1](#) (last pages there), and [case study 8.2](#)

You can set up and tune the system for Full Closed Loop as described in previous sections. Doing this by frequently analyzing screenshots that must be taken in real-time of the AAPS **SMB tab** is tedious, however.

More elegant and precise tuning can be done with a special evaluation software for the AAPS logfiles, by using the **emulator**. It is described here: <https://github.com/ga-zelle/APS-what-if> /

40 Documentation-in-English. There (under / Software) you find the files needed to download on your
41 PC, and the primary instructions:

42

43

44 In the emulator, you can see in tabular and graphical form, which autoISF component, and other
45 settings, contributed to SMB values that determined the glucose curve.

46 In the following, we look into how you create your relevant data.

47 Application examples for tuning are given in associated case studies (we need newer ones).

48

49 Note that the iOS based variants of autoISF for Trio or iAPS (oref loops for i-Phone) can
50 not use the emulator. Refer to [section 11.3](#).

51

52 Join <https://discord.gg/n3tD5eXExC> for seeking (and giving) help with the emulator set-up or use,
53 and to exchange experience.

54

55

56

57

58

59

60

10.1 Installation of the emulator on your PC

Installation is a one-time process, and you best refer to the installation guide of the developer, here: https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/Installation%20Guide.pdf

Below, I attempt to spell out some additional details “for IT dummies” (like myself)

10.1.1 Create your PC folder structure

The suggested folder names and structure shown below is of course not mandatory, but only a suggestion.

On your PC, create a folder “**Logfiles_Emulator**” with 3 sub-folders: “AAPS_logs”, “Emulator” and “Emulator_Studies”

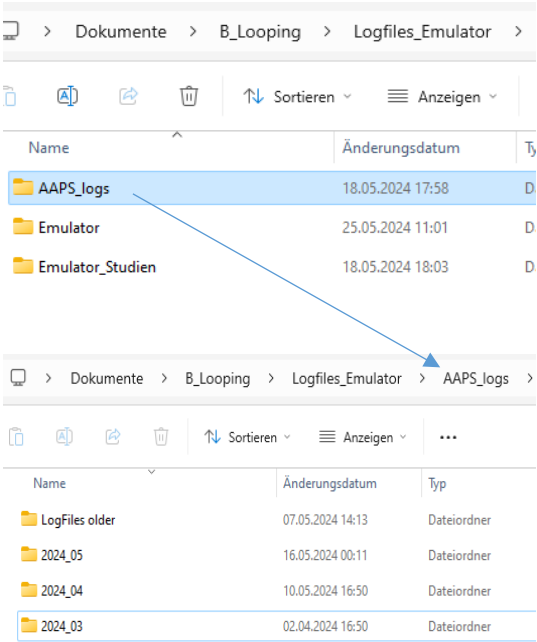
AAPS_logs: Put all your stored AAPS logfiles into that sub-folder. My folder structure for Logfiles and Emulation on the PC has 3 monthly folders, plus one folder with data from previous months and years (which I am less likely to analyze).

The logfiles you ALWAYS must copy-in from your phone before they get automatically erased there after x days (about 2 weeks, much shorter for 1-minute Libre3).

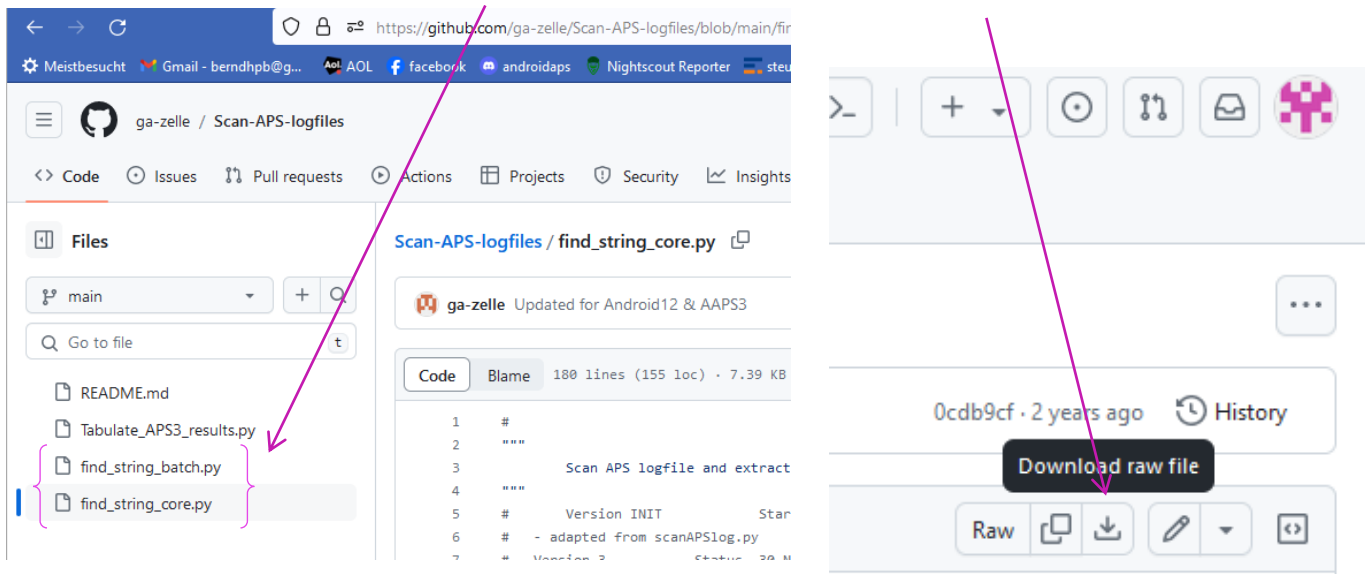
It is advisable to additionally store a pdf from **Nightscout Reporter** in the file for every month, with daily glucose charts, 24h scatter graph, etc. From it, you can much easier find which days and times are of high interest to analyze with the emulator.

Emulator: Neighboring “AAPS_logs” is the “Emulator” folder into which most downloads from the developer’s repo will go in [section 10.1.2](#)

Emulator_studies is a folder, where, for now, you should provide some **sub-folders** “Study_1”, “Study_2” ... **Study_n**. Later, when you use the emulator, you will use these “addresses” for the program to dump results from the emulation into. Additionally you will probably put related AAPS screenshots and Nightscout.Reporter or xDrip/Statistics charts into each project folder to support analysis.



115 4). From another section in Github, “Scan-APS-logfiles”, fetch two more .py files by
116 repeat steps 1)-3). for these two. They are from: <https://github.com/ga-zelle/Scan-APS-logfiles/blob/main>



117
118
119
120 5)-Retrieve these two .py files in your PC’s downloads folder, and move them into your emulator file (as
121 already was included two pictures higher up).

123 10.1.3 Create an “emulation start button” on your desktop

124 One of the files in your “Emulator” folder is “**emulator_GUI.py**“

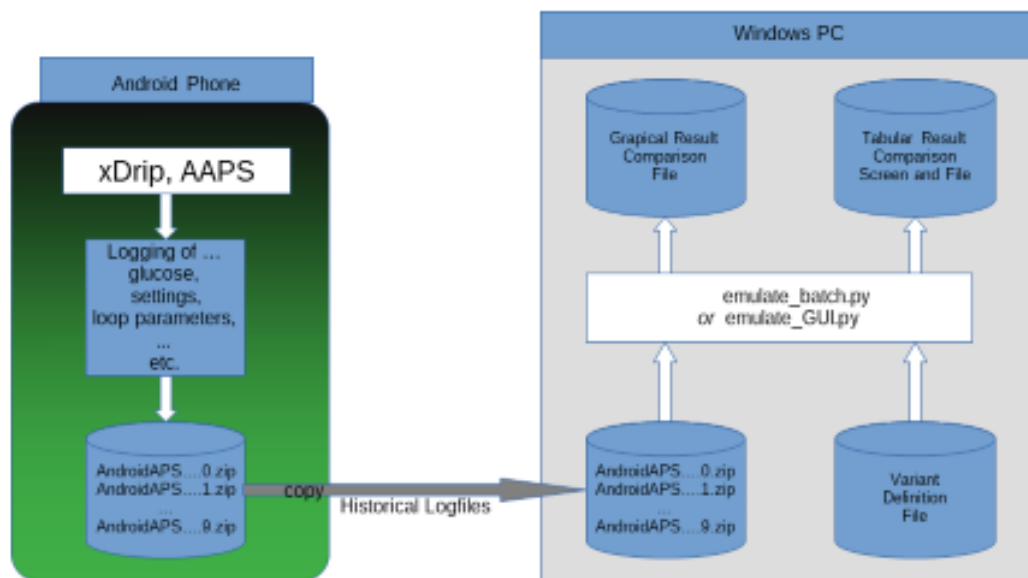
- 125 • Create, in your Emulator folder, a **link to it**
- 126 • Drag **that link** onto your **desktop**
- 127 • Name it something like “Emulator_start”: This is your **start button** for emulations on the PC

129 10.1.4 Other software requirements

130 Make sure you have **Notepad++** on your PC (see [section 10.2.1](#)).
131 QPython 3L will be needed on the smartphone, later (see [section 11](#)).

134 10.2 Analyzing loop decisions in logfiles

135 Instead of making many screenshots every 5 (or, w/ Libre3, every 1) minutes after meals, and analyzing them
136 later, a much more elegant and powerful way to analyze your loop decisions (and how you might want to
137 influence them with different settings, see [section 10.3](#) for this), is to use the emulator.



Sketch of Running the Emulator on a Windows PC

Github/ga-zelle / APS-what-if

10.2.1 Set up a “no change” .vdf file.

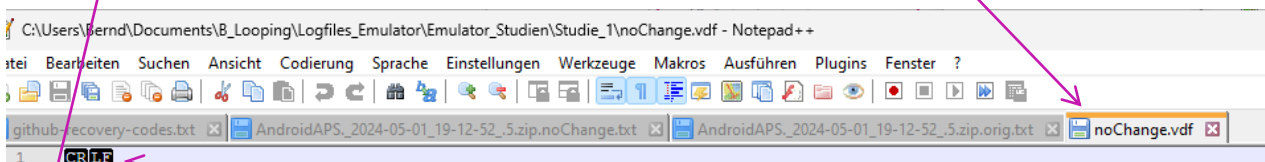
1). To do this, just open **Notepad++** (from list of all programs on your PC).

2). Name your file “noChange.vdf”.

It is just empty in the lines that would define any change to be investigated.

Note: for “what-if” analysis, entries will be made (in a second .vdf later, see [section 10.3](#))

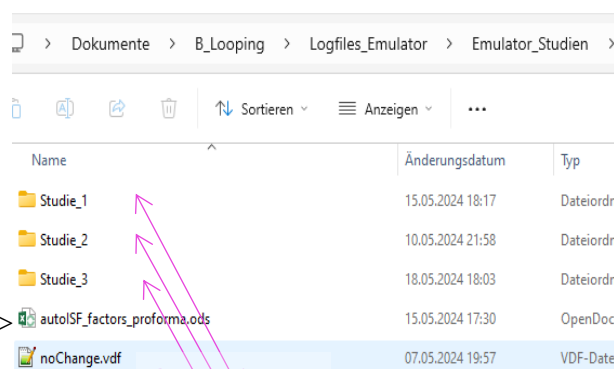
The no change .vdf should look like something like this:



Lines 1 and following should all be free (CR LF might show, depending on your settings in your WORD program)

3). Store that “noChange.vdf” in your “Emulator studies” folder, on the top level, besides the single studies folders

4). From that position, you always make a copy, and paste *into each* Study_1 ...n :



See [section 10.3.3.3](#), regarding this ->

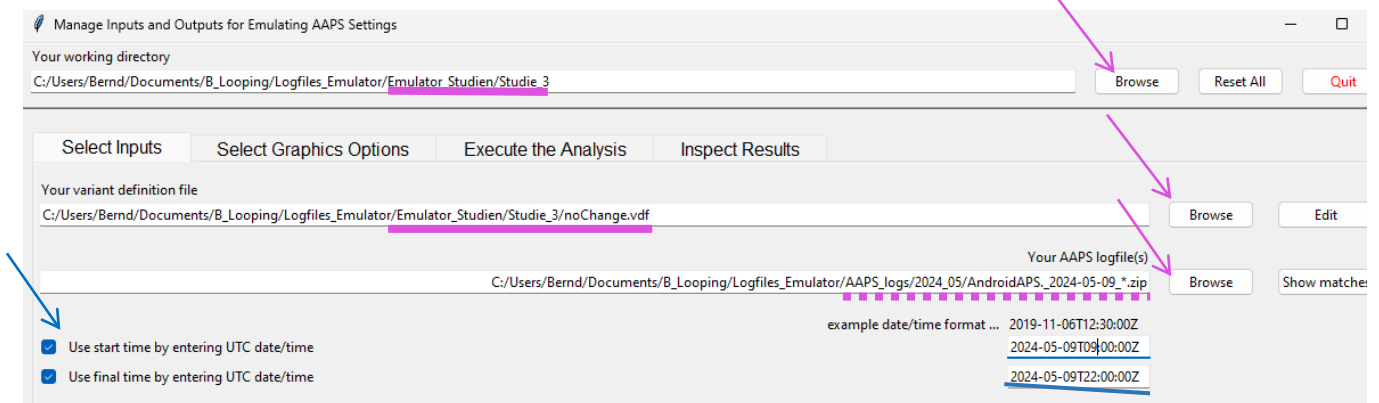
10.2.2 Locate relevant logfiles and prepare the Study_n folder

- 1). Make sure you have the AAPS logfiles that you want to analyze in your “AAPS_logs folder”
- 2). In your “Emulator_Studies” folder, create (or use a prepared) “Study_n” sub-folder, with a copied-in (not: moved!) noChange.vdf (It must be *in all* Study_n files).

10.2.3 Prepare your emulator run for Study_n

Now go to your PC **desktop**, and start the emulator by **just pressing the button “emulator start”** that you installed in step [10.1.3](#)

This opens a big dialogue box with 3 fields that you must fill in with the applicable path (*without* any quotation marks “.”) from your Windows Explorer file system, best done via (3x) Browse button:



- a) The top box marks the path to your current emulator project (“Studie_3” is my “Study_n” where I want to store results)
- b) The middle box marks the path to your current vdf (what kind of analysis; here: “...noChange.vdf” = *read-only*. (For *what-if*, see [section 10.3](#))
- c) The third box marks the path to your AAPS logfiles you wish to look into. A good way to do this is:
 - Browse in your Windows Explorer to any logfile from the desired day (2024-05-09 in above example)
 - Replace the time with an asterix * (this means you look at **all-day** data, in UTZ time). Check whether this will work by pressing Show matches . You should see all logfiles from that day in a pop-up info box.
 - As I wanted to look at 11 am –midnight (for lunch and dinner related data), I :
 - clicked the bottom left two boxes
 - copied the date 2024-05-09 over the default date in the bottom right two data fields

- after T (for time), I entered the desired time of analysis AFTER conversion into my local time (Central EU summer time minus 2 hours = UTZ; so to look at 11 to midnight of my AAPS screen, I must enter here 09.00:00Z, and below it 22:00:00Z).

Entries at the bottom are not mandatory, but when clicking these little boxes (bottom left) you can define a start and/or an end-point for analyzing, within the logfiles specified in the field above.

10.2.4 Run emulation

Now we are ready to go: Press “Run emulation”

This produces sometimes an error message (e.g. if you have a syntax error, or incompatible software versions: => seek help, in the Github materials provided by ga-zelle, or in Discord/Full-Closed-Looping/emulate-aaps here: <https://discord.gg/n3tD5eXExC>)

After a short moment results should show up, which you can look into in a couple of ways.

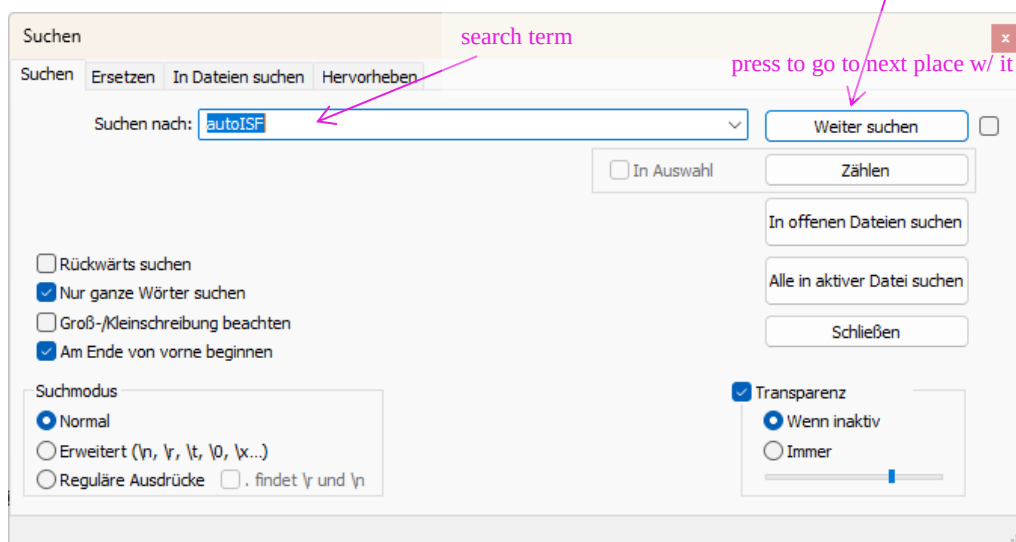
Optional: You could have a quick look into the **.log** file to see whether the run had errors (see [section 3.8](#))

10.2.4.1 SMB tab contents in (date..) **noChange.txt** result file

This ...txt file basically gives you “all the SMB tab” infos, in a super long list (but without needing to make screenshots in real-time, every 5 minutes.)

Search options help find what lines are of interest to your analysis:

By using the **search function** you can jump, in that super long list, to all places that e.g. have „autoISF“ in it or „script debug“, or „SMB disabled“ (if you want to analyze when that happened). Precise spelling, as in this .txt (or in SMB tab) is of course important.



216 10.2.4.2 Table of results (...noChange.csv file)

217

218 The .csv file in your project folder gives a tabular presentation of how parameters like bg, iob,
219 iobTH, the various ISF contributors, bg target, insulinRequired etc. develop every 5 minutes, and
220 what SMB size and %TBR resulted.

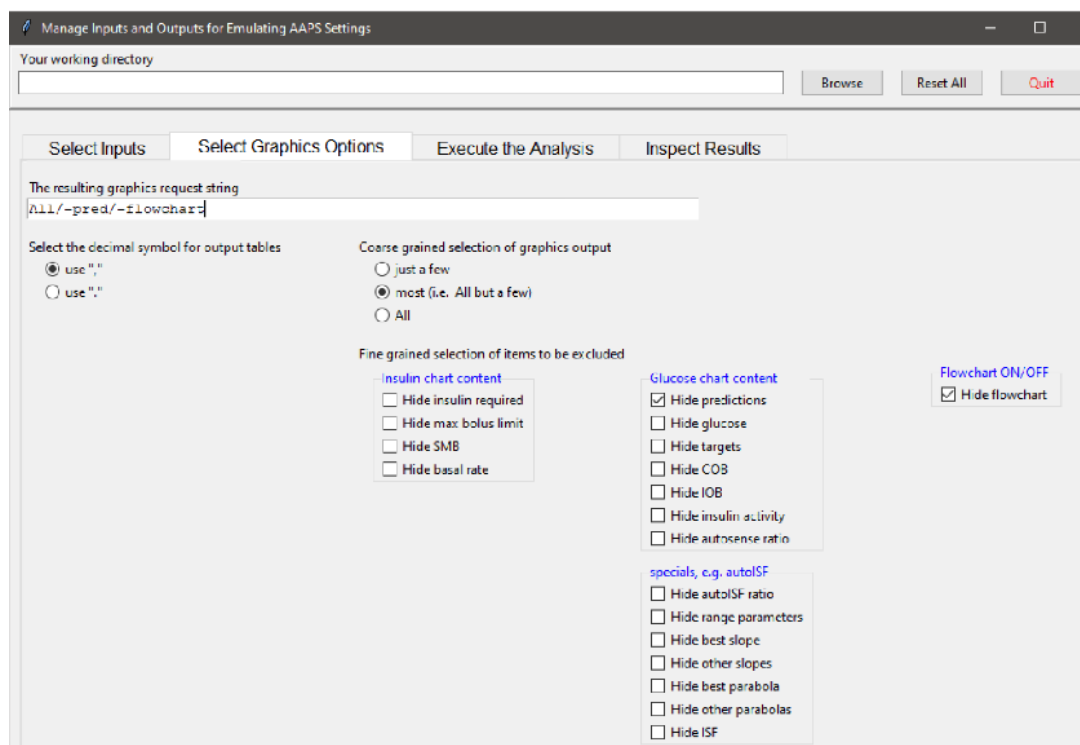
221 It is a vast table, so you may want to reduce it to something more “digestible”, either after transfer
222 to your standard calculation program (next [section 10.2.4.3](#)). Or you can also make settings to suppress
223 information you are usually not interested in (or do not know how to interpret, anyways) under “Select
224 Graphics Options” when you open the emulator, before executing any analysis:

225 First, select your preferred way of outputting decimals (point or comma).

226 Then select whether you want “All” possible outputs in the graph, or “Most” = all except those you tick “off”
227 in the boxes for each output parameter.

228 In case you would use “Some/just a few”, you would have to tick those few you that do want to see, by ticking
229 the corresponding boxes.

230 Recommendation is to look at (nearly) everything offered (as your default setting that you can leave
231 untouched in most of your emulator runs):



232

233 It might be easier, to not deal with customizing the csv file, and rather copy the data into your
234 favorite calculation program:

235

236 10.2.4.3 Analysis of the noChange.csv table in Excel or LibreOffice calc.

237

238 Best copy the entire table into a new .xls or .ods sheet, where you can:

- 239
- add right next to the UTC (Unix Time Code) your corresponding “AAPS time”

240 For instance, adding +2/24 translates the UTC column into central European summer time column
 241 next to it (where currently a row of Z stands). Likewise, subtract like -5/24 from UTC for an US East
 242 Coast time scale.

243 *(Fun fact: Our oref loop stubbornly works on UTC, un-impressed by our folly to jump twice a year*
 244 *into or out of a local summer time, or to travel across time zones. If some data get lost in translation*
 245 *there, it is only to us, with our stupid time change. For the loop, its database (e.g. on insulin activity)*
 246 *remains unambiguously intact).*

247 Highlight all time fields (both entire columns), and switch from hh:mm:ss format to hh:mm.
 248 *(While the seconds are important for the loop's calculations, for our comparison with Nightscout or*
 249 *other charts and data, it is much easier without the seconds attached)*

250 • **hide** any column you find less important to look at for your intended analysis

251 That way, “boxes” (data fields) retain their original position in tables

252 Also, in case later you want to look into additional info, you can simply un-hide the relevant columns
 253 (or lines:.)

254 • **hide** lines (time segments) you find less important to look at for your intended analysis

255

256 Usually you will color mark where relevant SMBs were given, which of the ISFs (and underlying
 257 weights) was strongly contributing (note that this can be good or not good). Also where iobTH was
 258 exceeded, whether an Automation kicked in e.g. setting a TT, or when there were periods with zero
 259 insulinRequired.

260 In [section 10.3.4](#) we present an extra tool that does a standardized table reduction and color marking
 261 for you!

262 You may be able to formulate a hypothesis or two, what settings (...ISF_weights, iobTH%,
 263 SMB_range_extention, autoISFmax ...) should be changed for improvement (then go to [10.3](#))

264

265 [10.2.4.4.. Graph noChange.pdf](#)

266

267 After your emulation run, under Inspect Results, you can open the pdf file that is last in the results list
 268 offered.

269

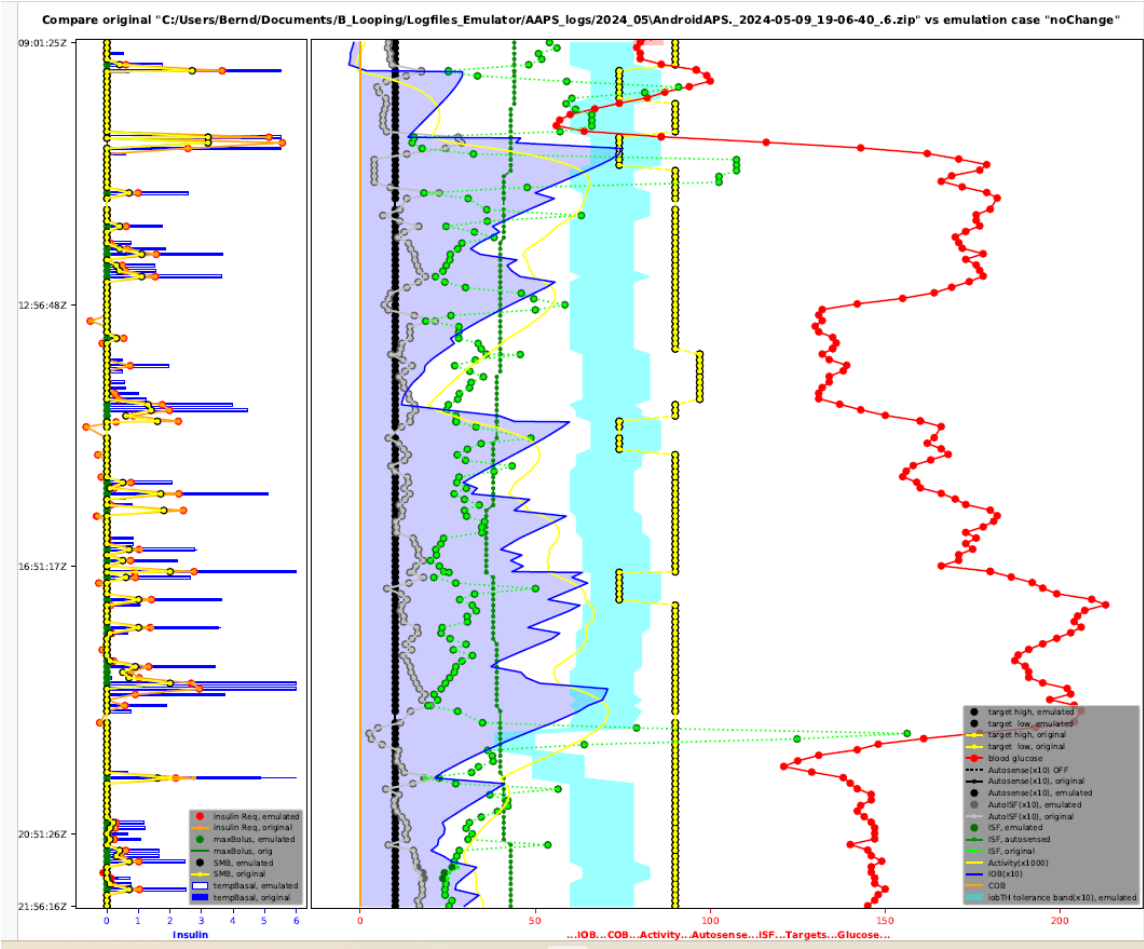
270 This **noChange.pdf** is a chart that shows along the time axis (down), from right to left:

271 • Red: the bg curve

272 • Yellow: the bg target (note that I do no manual “EatingSoonTT” but for bg rises over +10 mg/dl
 273 I have an Automation that sets low TT for a couple of minutes)

274 • Light blue corridor: Left edge is set iobTH, and bandwidth +30% (would be +20% at elevated
 275 TT)

- Dark blue line: iob (exceeding twice the iobTH, with temp. SMB shut-off)
- As bg did not convincingly come down enough, one could hypothesize that iobTH should be elevated. ((But, again, this would have to be confirmed also with other kinds of meals)).
- Thin yellow line: Insulin activity
 - Green dotted line: ISF as would result from AAPS w/Autosens
 - Green scatter points: autoISF ISF no Chage (lighter points) or what-if (darker points)
 - Black line: Profile ISF
 - Gray scatter points: ISF weakened (to the left of black line) or strengthened (to the right)
 - Orange line: cob=0 at all times (in FCL)



More see discussed together with (yourChanges).pdf in [section 10.3.3.4](#)

10.2.4.5 delta table

In case you want to analyze delta, short and long average deltas:

- you could do some of that just using your .xls extract from .csv (see section 10.2.4.3).

- There is also an extra sheet provided, “delta” (That works only if you have your **emulator_core.py** updated to the one from 02June2024 or newer).
- This is definitely core to FCL using Automations ([section 13.1](#)).
- When using acceleration detection via autoISF, the deltas are mainly meaningful to define personal Automations that involve Conditions using these glucose curve characteristics. An example would be to use the delta table to identify cases of compression low

10.3 “What-if” analysis using the emulator

In the following you see an example how you can analyze a day of logfiles, and selecting the time span of interest, for instance 11-24 h to look at how autoISF managed lunch and dinner.

You will go through the emulator exactly as you already did in [section 10.2](#), where you exclusively had the noChange.vdf on board.

However, this time you focus on (yourChange).vdf, see below, [10.3.1](#).

Repeat, if you have two or more such vdf defined.

(Just clear old results before pressing “execute analysis” each time.)

All results are automatically captured for all runs, all **in your selected “Study_n” folder**, together with the noChange results

- Results files with noChange in their name are always your actual loop data ...
- as opposed to results on “what-if”, that contain name of the (yourChange).vdf in their file name see e.g. at around [line 380](#)

How to proceed, step by step:

10.3.1 Define your investigated changes in (yourChange).vdf (one, or several)

1). Define for which one (I suggest max three) parameter(s) in your current profile settings you want to look into a different setting. Recommendation is to use a factor, like for example: current setting * 0.9, or current setting * 1.2, and use that in your naming for this vdf file, too.

You may want to consult [APS-what-if /Documentation in English/Guide to VDF Files for the AAPS Emulator.pdf](#) Access directly, or via [section 3.8](#)

327 Within one study, you can make several emulator runs with several (yourChange).vdf files (all based on what
328 really happened, as captured with the noChange.vdf).

329 All results, like the csv results table, will appear then *several times* in your study file, only *with different*
330 *name endings* as in the underlying vdf.

331

332 Example: I like to check in my actual data (they are in my noChange.vdf emulator run), **in which time**
333 **points the following parameter changes would make a** (how) big **difference** in the loop's decision:

- 334 • 20% higher bgAccel_ISF_weight to boost the first SMBs stronger: How would that tend to ramp
335 up early iob; and might that get too strong in other parts of the data? Or does it bounce into a
336 restriction (maxSMB size; autoISFmax; iobTH...) that I might need to widen?
- 337 • Doubling my cautiously set bgBrake_ISF_weight shall give me insight into the workings of that
338 parameter (and whether using a much smaller weight than for bgAccel_ISF_weight is really
339 what I should keep doing)
- 340 • As my bg came down from a persistent high quite slowly, I elevate the dura_ISF by 20%

341

342 *Tuning advice:* Actually, it would make more sense to first find my “optimal”, maybe indeed
343 elevated, bgAccel_ISF_weight. *Then*, in a *new* project_n+1, do (automatically) a noChange run
344 **with that**, plus a (yourChange) run with the stronger dura weight, investigated *on that* basis.
345 Reason: 1) As we always say, better do only one change at a time. 2) A better job with bg control
346 via bgAccel_ISF will reduce the peak height and provide a different (easier) scenario for
347 dura_ISF to manage.

348

349 2).Now, to **write** your **(yourChange). vdf for the emulator** (this is same procedure as you did in section
350 10.2.1 for the noChange.vdf):

351

- 352 • just open Notepad++ (from list of all programs on your PC) to create a new vdf:.

353 Alternatively you can also take another pre-existing vdf file, copy it into your current
354 project , edit as desired, and give it a new name (re-name it)

355 **Caution:** Make absolutely sure (best by looking it up in the SMB tab, down in the profile set
356 section) to **spell each term exactly** as your loop uses it (probably w/ decimal points, not comma)

357

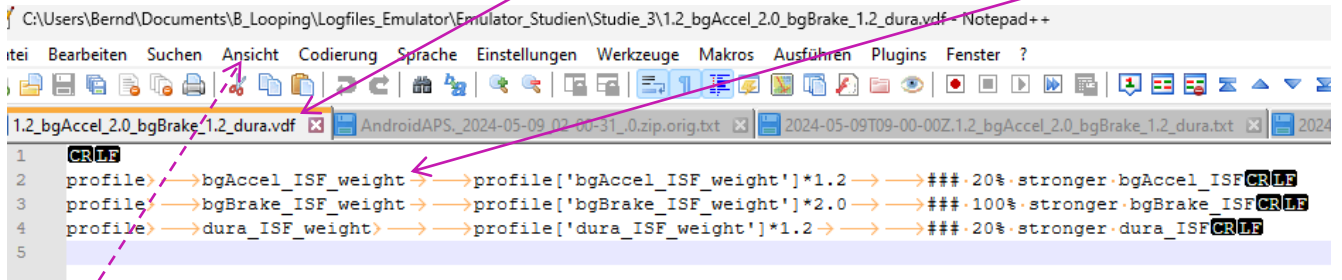
- 358 • ...when you make one line per parameter (separating entries with spacers->):

359 profile->(parameter) ->->profile['(parameter)']*(factor)->->###(comment as you like)

360

361

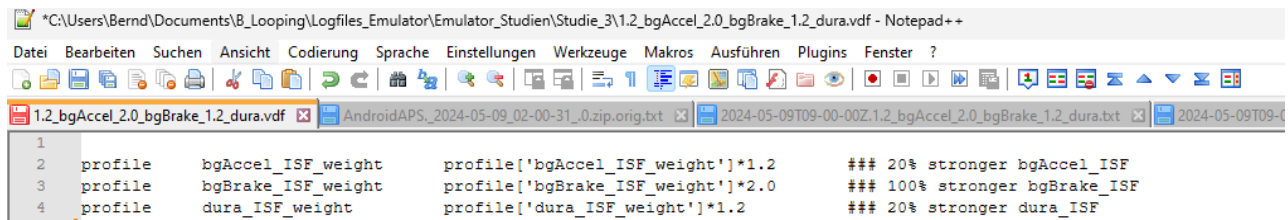
The (yourChange) .vdf should look like something like this:



```
C:\Users\Bernd\Documents\B_Looping\Logfiles_Emulator\Emulator_Studien\Studie_3\1.2_bgAccel_2.0_bgBrake_1.2_dura.vdf - Notepad++
Datei Bearbeiten Suchen Ansicht Codierung Sprache Einstellungen Werkzeuge Makros Ausführen Plugins Fenster ?
1.2_bgAccel_2.0_bgBrake_1.2_dura.vdf AndroidAPS_2024-05-09_02-00-31_0.zip.orig.txt 2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.txt 2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.txt
1 CR LF
2 profile>→bgAccel_ISF_weight→→profile['bgAccel_ISF_weight']*1.2→→### 20% stronger bgAccel_ISF CR LF
3 profile>→bgBrake_ISF_weight→→profile['bgBrake_ISF_weight']*2.0→→### 100% stronger bgBrake_ISF CR LF
4 profile>→dura_ISF_weight→→profile['dura_ISF_weight']*1.2→→### 20% stronger dura_ISF CR LF
5
```

Via view/ show/hide symbols (CR , LF, tabs ...), you can have different looks, see other example two pages down.

Or, with settings/view/remove symbols (for line feed, tab etc) it can also look like this:



```
*C:\Users\Bernd\Documents\B_Looping\Logfiles_Emulator\Emulator_Studien\Studie_3\1.2_bgAccel_2.0_bgBrake_1.2_dura.vdf - Notepad++
Datei Bearbeiten Suchen Ansicht Codierung Sprache Einstellungen Werkzeuge Makros Ausführen Plugins Fenster ?
1.2_bgAccel_2.0_bgBrake_1.2_dura.vdf AndroidAPS_2024-05-09_02-00-31_0.zip.orig.txt 2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.txt 2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.txt
1
2 profile bgAccel_ISF_weight profile['bgAccel_ISF_weight']*1.2 ### 20% stronger bgAccel_ISF
3 profile bgBrake_ISF_weight profile['bgBrake_ISF_weight']*2.0 ### 100% stronger bgBrake_ISF
4 profile dura_ISF_weight profile['dura_ISF_weight']*1.2 ### 20% stronger dura_ISF
```

3). Name your vdf (in example below: 1.2_bgAccel_2.0_bgBrake_1.2_dura.vdf) ...

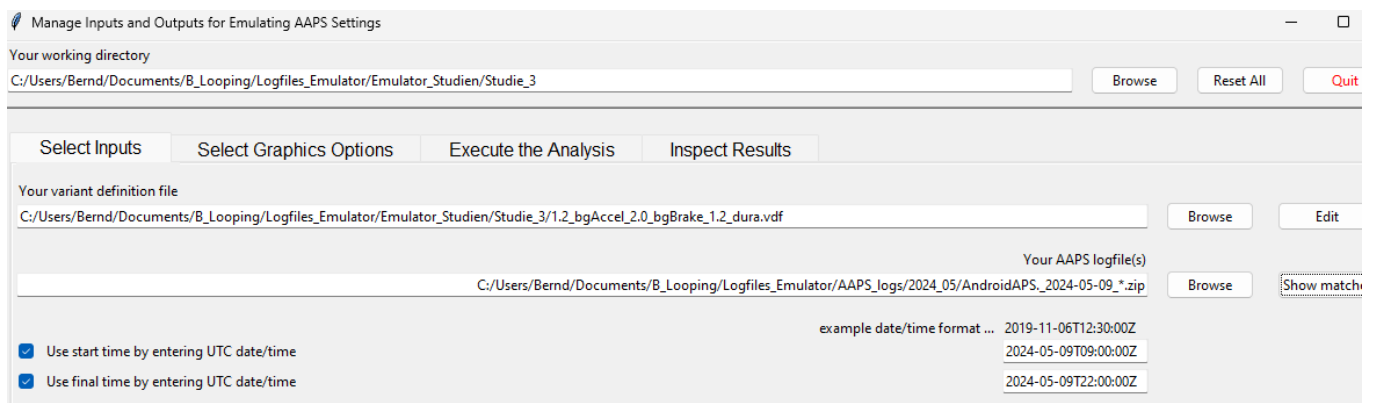
4). Store this (yourChange).vdf in the folder for your current Study_n you are about to start (see my storage path C: Studie3.....vdf – Notepad++ in the top line:)

10.3.2 Run the emulator with (yourChange).vdf

The “what-if” emulator run is done the same way as you did the noChange.vdf run (section 10.2), which had no (yourChange).vdf on bord

=> No surprise, running an emulation with only that noChange.vdf, yields same result in emul columns as is orig columns. - However, now :

The (yourChange).vdf must be loaded into the 2nd input field, where formerly you had the noChange.vdf.:



Manage Inputs and Outputs for Emulating AAPS Settings

Your working directory
C:/Users/Bernd/Documents/B_Looping/Logfiles_Emulator/Emulator_Studien/Studie_3 [Browse] [Reset All] [Quit]

Select Inputs | Select Graphics Options | Execute the Analysis | Inspect Results

Your variant definition file
C:/Users/Bernd/Documents/B_Looping/Logfiles_Emulator/Emulator_Studien/Studie_3/1.2_bgAccel_2.0_bgBrake_1.2_dura.vdf [Browse] [Edit]

Your AAPS logfile(s)
C:/Users/Bernd/Documents/B_Looping/Logfiles_Emulator/AAPS_logs/2024_05/AndroidAPS_2024-05-09_*.zip [Browse] [Show matches]

example date/time format ... 2019-11-06T12:30:00Z
2024-05-09T09:00:00Z
2024-05-09T22:00:00Z

☒ Use start time by entering UTC date/time
☒ Use final time by entering UTC date/time

In the 3rd input field, give the path to your stored logfiles. A good way to do this is:

- 386 • Browse in your Windows Explorer to any logfile from the desired day (2024-05-09 in above ex-
- 387 ample)
- 388 • Replace the time with an asterix * (this means you look at all-day data, in UTZ time). Check
- 389 whether this will work by pressing Show matches . You should see all logfiles from that day in a
- 390 pop-up info box.
- 391 • As I wanted to look at 11 am –midnight for lunch and dinner related data, I :
- 392 ○ clicked the bottom left two boxes
- 393 ○ copied the date 2024-05-09 over the default date in the bottom right two data fields
- 394 ○ after T (for time), I entered the desired time of analysis AFTER conversion into my local
- 395 time (Central EU summer time minus 2 hours = UTZ; so to look at 11 to midnight of
- 396 my AAPS screen, I must enter here 09.00:00Z, and below it 22:00:00Z).

397 After making these entries, press Execute the Analysis, (evtl also Clear old Data) and then press Run

398 Emulation, you can look the results up under “Inspect Results”. First you could have a quick look into

399 the .log file to see whether the run had errors (see [section 3.](#))

400

401 10.3.3 Emulation results

402

Your working directory
C:/Users/Bernd/Documents/B_Looping/Logfiles_Emulator/Emulator_Studien/Studie_3

Select Inputs Select Graphics Options Execute the Analysis Inspect Results

*.log - Your file showing edits from the variant assignments	C:/Users/Bernd/Documents/B_Looping/Logfiles_Emulator/Emulator_Studien/Studie_3/2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.log	Browse	Show
*.csv - Your table comparing key values of original vs emulation	C:/Users/Bernd/Documents/B_Looping/Logfiles_Emulator/Emulator_Studien/Studie_3/2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.csv	Browse	Show
*.delta - Your table comparing bg deltas of original vs emulation	C:/Users/Bernd/Documents/B_Looping/Logfiles_Emulator/Emulator_Studien/Studie_3/2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.delta	Browse	Show
*.orig.txt - Your short log of original analysis	C:/Users/Bernd/Documents/B_Looping/Logfiles_Emulator/Emulator_Studien/Studie_3/2024-05-09T09-00-00Z.orig.txt	Browse	Show
*.txt - Your short log of emulated analysis	C:/Users/Bernd/Documents/B_Looping/Logfiles_Emulator/Emulator_Studien/Studie_3/2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.txt	Browse	Show
*.pdf etc. - Your graphic file comparing key values of original vs emulation	C:/Users/Bernd/Documents/B_Looping/Logfiles_Emulator/Emulator_Studien/Studie_3/2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.pdf	Browse	Show

403

404 All results from your (yourChanges).vdf emulator go automatically where the noChange.vdf results are

405 already stored, in our example into the “Studie 3” file, below:

406

407 Besides the 1.2_bgAccel_2.0_bgBrake_1.2_dura.vdf case which I like to look into for the present high carb

408 meal, I also prepared another vdf that investigates a factor 1.2 stronger pp_ISF and a weaker, factor 0.8,

409 bgAccel_ISF (with the intention to test this, and a noChange (that ideally would already contain the conclusion

410 on adapting the bgAccel_ISF_weight*), on a low carb meal later.

411 * Note the challenge here is to iterate between the typical meals of your personal spectrum to find **one** set

412 of settings that work good-enough **for all** of them.

B_Looping > Logfiles_Emulator > Emulator_Studien > Studie_3				Studie_3 durchsuchen
Sortieren ▾ Anzeigen ▾ ...				
Name	Änderungsdatum	Typ	Größe	
2024-05-09T09-00-00Z .noChange.pdf	15.05.2024 17:17	Adobe Acrobat-D...	77 KB	
2024-05-09T09-00-00Z .noChange.csv	15.05.2024 17:17	Microsoft Excel C...	51 KB	
2024-05-09T09-00-00Z .noChange.log	15.05.2024 17:17	Notepad++ Docu...	35 KB	
2024-05-09T09-00-00Z .noChange.txt	15.05.2024 17:17	Notepad++ Docu...	281 KB	
2024-05-09T09-00-00Z .orig.txt	15.05.2024 17:17	Notepad++ Docu...	281 KB	
2024-05-09T09-00-00Z .1.2_pp_0.8_bgAccel.pdf	15.05.2024 17:16	Adobe Acrobat-D...	78 KB	
2024-05-09T09-00-00Z .1.2_pp_0.8_bgAccel.csv	15.05.2024 17:16	Microsoft Excel C...	51 KB	
2024-05-09T09-00-00Z .1.2_pp_0.8_bgAccel.log	15.05.2024 17:16	Notepad++ Docu...	57 KB	
2024-05-09T09-00-00Z .1.2_pp_0.8_bgAccel.txt	15.05.2024 17:16	Notepad++ Docu...	281 KB	413
2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.csv	17.05.2024 21:29	Microsoft Excel C...	51 KB	
2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.log	17.05.2024 21:29	Notepad++ Docu...	66 KB	
2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.pdf	17.05.2024 20:40	Adobe Acrobat-D...	78 KB	
2024-05-09T09-00-00Z.1.2_bgAccel_2.0_bgBrake_1.2_dura.txt	17.05.2024 21:29	Notepad++ Docu...	282 KB	414
1.2_bgAccel_2.0_bgBrake_1.2_dura.vdf	17.05.2024 20:38	VDF-Datei	1 KB	415
1.2_pp_0.8_bgAccel.vdf	10.05.2024 21:55	VDF-Datei	1 KB	
noChange.vdf	07.05.2024 19:57	VDF-Datei	1 KB	

10.3.4(yourChange).txt: “what-if” impact on loop decisions (as in SMB tab)

The **noChange.txt** has all the info your series of SMB tabs had that day.

How to search in this vast list is shown elsewhere (see [section 10.2.4.3](#)).

Likewise, the **(yourChange).txt** gives *for each loop decision* in all detail how and why each single decision *would have* changed with the different parameter inputs you are checking out here

In the two (yourChange) examples here, , it was a check on the difference

- a 20% stronger pp_weight and 20% weaker bgAccel_weight
- a 20% stronger weight for both, bgAccel_ and dura_ISF, and a doubling of bgBrake_weight

would make.

Note that all these “what if” data can only give rough hints, notably about **the first** greater change that you would see with the investigated changed setting. So it works quite well for our main problem in FCL, investigating how to ramp up iob quickly after detection of acceleration.

433 Note that any relevant change would put your bg curve on a different trajectory, so that would influence *all*
434 *following* results. Therefore, what you get here is **not** a complete modelling how your bg would have
435 developed in the alternative scenario.
436 But you can investigate in which stages the parameter(s) you are looking at in your current “what-if” had big
437 influence, and in which direction the changes would go. (see also charts shown in [section 10.3.3.4](#)).
438 Analyzing how to safely come down from a high glucose plateau while limiting hypo danger towards the end
439 of digestion is also to some extent possible.

440

441 A good other way to employ the what-if analysis is real time, on your smartphone, using speech synthesis
442 (see [section 11](#)): Then you get real-time info, as to exactly when a significantly different proposal would
443 emerge, and can decide (and watch!) real-time whether to follow the new idea and not was probably better.

444

445 Observe that a setting change must work well for you

446

- not just in one point of time, and

447

- not just for one kind of meal,

448 but you must look at all time slots in the investigated meal, plus analyze with the same tool a totally different
449 meal within your usual spectrum, how things work out there

450

451 [10.3.3.2 Tabular results](#)

452

453 [A\) .csv results table and spreadsheet copies of it](#)

454

455 The **noChange.csv** table gives all relevant data. Besides development of bg and iob you see the calculated
456 insulinRequired in each loop decision, and how each of the autoISF categories contributed to the decision
457 (notably regarding SMB size).

458

459 Note that the “**acce_ISF**” results are only in case of positive acceleration (that is our main focus)
460 driven by the bgAccel_ISF_weight setting. (These are all positions > 1.0 in the “acce ISF” columns).

461

462 **In case of negative acceleration** (decelerating rise, positions < 1.0 in the “**acce ISF**” columns),
463 **bgBrake_ISF_weight** is applied. As discussed in [section 4.4](#), bgBrake_ISF might be most
464 important (and interesting to analyze) in slowly resorbing meals.

465

466 Note: maxBolus=0 means in this table that SMBs were not capped by maxBolus.

467

468

469 The **(your change).csv** shows in detail how **every single** loop **decision** would be influenced by the different
470 settings you are investigating.

471 To inspect that huge table, click on the Z behind the start UTC time entry (see black box in the Z column of
472 the table, next page).

473 If you like to see the bg in each screen, too, go 3 or 4 columns farther to the right with your black
474 box.

475 Then, go to window/fix. Now you can scroll through the data and always see headline and time (or time and
476 bg level).

477 To further ease analysis, feel free to temporarily erase (hide) any columns that you (think you) do not
478 need for the intended analysis. More suggestions see in [section 10.2.4.2](#)

The screenshot shows the LibreOffice Calc interface with a spreadsheet titled "2024-05-09T09-00-00Z.1.2_pp_0.8_bgAccel.csv". A context menu is open over the "Z" column header, with options: "Neues Fenster", "Fenster schließen", "Teilen", and "Fixieren". The "Fixieren" option is highlighted. The spreadsheet contains a table with columns: id, UTC time, Z, UNIX time, bg accel, bg brake, target low, orig, orig, emul, emul, cob, job, emul, emul, act, orig, orig, utes. The table has 37 rows of data. The status bar at the bottom shows "Tabelle 1 / 1", "Standard", "Summe=0", and "100%".

id	UTC time	Z	UNIX time	bg accel	bg brake	target low	orig	orig	emul	emul	cob	job	emul	emul	act	orig	orig	utes
0	0,3759837963	Z	1715245285,9	80			90	90	90	90	0	-0,18	6	7,8	0,001	1	0,81	10
1	0,3793981481	Z	1715245580,3	79	79		90	90	90	90	0	-0,23	6	7,8	0,001	1	0,78	15
2	0,3829166667	Z	1715245884,2	80			90	90	90	90	0	-0,28	6	7,8	0	1	0,86	20
3	0,386412037	Z	1715246186,2	80			90	90	90	90	0	-0,28	6	7,8	-0,001	1	0,85	25
4	0,3898263889	Z	1715246482	86			90	90	90	90	0	-0,33	6	7,8	-0,001	1	0,91	0
5	0,393587963	Z	1715246806,8	96			74	74	74	74	0	0,18	6,6	8,58	-0,001	1	1,75	0
6	0,3942013889	Z	1715246859,5	96			74	74	74	74	0	2,92	6,6	8,58	0	1	1,75	0
7	0,3968055556	Z	1715247084,8	99			74	74	74	74	0	2,92	6,6	8,58	0,005	1	1,31	5
8	0,4002430556	Z	1715247381,7	100	100		74	74	74	74	0	2,83	6,6	8,58	0,011	1	0,75	10
9	0,4037384259	Z	1715247683,2	94	94		74	74	74	74	0	2,72	6,6	8,58	0,015	1	0,48	0
10	0,4071643519	Z	1715247979,7	87	87		74	74	74	74	0	2,59	6,6	8,58	0,018	1	0,54	0
11	0,4107407407	Z	1715248288,8	82	82		74	74	74	74	0	2,45	6,6	8,58	0,02	1	0,73	0
12	0,4141435185	Z	1715248583	74	74		90	90	90	90	0	2,3	6	7,8	0,022	1	0,75	0
13	0,4176273148	Z	1715248883,1	67	67		90	90	90	90	0	2,14	6	7,8	0,022	1	0,7	0
14	0,4210300926	Z	1715249177,9	60	60		90	90	90	90	0	1,99	6	7,8	0,023	1	0,65	0
15	0,4245949074	Z	1715249485,3	57	57		90	90	90	90	0	1,82	6	7,8	0,023	1	0,65	0
16	0,4280439815	Z	1715249783,8	56	56		90	90	90	90	0	1,67	6	7,8	0,022	1	0,65	5
17	0,4315277778	Z	1715250084,6	64			90	90	90	90	0	1,51	6	7,8	0,021	1	0,75	0
18	0,435	Z	1715250384,3	86			74	74	74	74	0	1,36	6,6	8,58	0,02	1	2,8	0
19	0,4355787037	Z	1715250434,4	86			74	74	74	74	0	4,58	6,6	8,58	0,021	1	2,8	0
20	0,4384953704	Z	1715250686,1	116			74	74	74	74	0	4,45	6,6	8,58	0,026	1	2,9	0
21	0,4419675926	Z	1715250987	143			74	74	74	74	0	7,44	6,6	8,58	0,038	1	2,43	0
22	0,4454513889	Z	1715251287,7	162	162		74	74	74	74	0	7,4	6,6	8,58	0,048	1	1,33	0
23	0,4490046296	Z	1715251594,5	171	171		74	74	74	74	0	7,15	6,6	8,58	0,055	1	0,4	0
24	0,452349537	Z	1715251883,2	179	179		74	74	74	74	0	6,84	6,6	8,58	0,06	1	0,4	5
25	0,4558217593	Z	1715252183,5	177	177		90	90	90	90	0	6,48	6	7,8	0,063	1	0,4	10
26	0,4594328704	Z	1715252495,4	169	169		90	90	90	90	0	6,1	6	7,8	0,065	1	0,4	15
27	0,4630208333	Z	1715252805,5	166	166		90	90	90	90	0	5,72	6	7,8	0,066	1	0,4	5
28	0,4662847222	Z	1715253087,3	172	172		90	90	90	90	0	5,37	6	7,8	0,065	1	0,86	25
29	0,4697453704	Z	1715253386,2	179			90	90	90	90	0	5	6,36	8,27	0,064	1	2,26	5
30	0,4731828704	Z	1715253683,7	182			90	90	90	90	0	5,54	6,36	8,27	0,064	1	1,33	10
31	0,480162037	Z	1715254286,4	180			90	90	90	90	0	4,81	6,36	8,27	0,063	1	1,13	0
32	0,4837731481	Z	1715254598,6	176	176		90	90	90	90	0	4,44	6,36	8,27	0,061	1	0,65	5
33	0,4873032407	Z	1715254903,9	176			90	90	90	90	0	4,09	6	7,8	0,059	1	1,13	10
34	0,4905555556	Z	1715255184,4	177			90	90	90	90	0	3,78	6	7,8	0,056	1	1,67	15
35	0,4940625	Z	1715255487,9	173			90	90	90	90	0	3,98	6	7,8	0,054	1	1,26	20
36	0,4974884259	Z	1715255783,9	170	170		90	90	90	90	0	3,69	6	7,8	0,053	1	1,07	25
37	0,5009722222	Z	1715256084,7	171			90	90	90	90	0	3,39	6	7,8	0,05	1	1,38	30

479
480
481 Still, the csv tables are overwhelming. You could proceed in either of two directions now:
482

A) Convert both (or all 3) csv files into one table in Excel or into Libre office calculator. Hide columns (and eventually also lines) that are of no particular interest for your analysis. Mark differences between noChange and (yourChanges) column data with color, add extra columns with additional calculations ...

This route is good to compare quantitative impacts of autoISF categories in critical time points.

B) For the core data relevant to assessing your autoISF settings, there is an extra tool for convenient analysis - see the following [section 10.3.3.3](#)

That tool is quite a bit of work to set up. Decide for yourself whether you do it, or whether you rather work with extracting the csv table into Excel (A), and work freely from there.

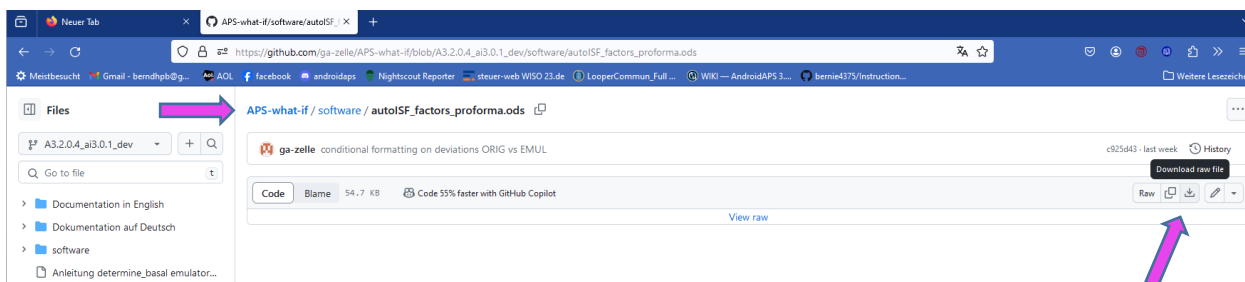
10.3.3.3 Automated extraction from tabular results *(optional add-on)*

Decide for yourself, whether you rather go from the csv results table into .xls and produce what you want to see there for yourself.

With a bit of extra set-up effort (steps 1)-7), next 4 pages) *you can install an adjunct tool* that will always produce the nice graph for you as shown on the end of this [section 10.3.3.3, in step 7](#).

Realizing that going through the following steps 1) – 7) is currently quite complicated, the developer is working at radically simplifying it in an update to be expected this summer (2024). If you feel a bit challenged, we suggest to just wait for this update, which will largely be embedded in a newer version of the autoISF_factors_proforma.ods, see next section.

1).autoISF_factors_proforma.ods is provided as an **extra tool** that you download from here:



Put that file on your PC one level above the single files for all your studies:

Emulator_Studien durch

Sortieren Anzeigen

Name	Änderungsdatum	Typ	Größe
Studie_1	15.05.2024 18:17	Dateiordner	
Studie_2	10.05.2024 21:58	Dateiordner	
Studie_3	15.05.2024 18:15	Dateiordner	
autoISF_factors_proforma.ods	15.05.2024 17:30	OpenDocument S...	55 KB
noChange.vdf	07.05.2024 19:57	VDF-Datei	1 KB

2).Now, if we want to use this tool on the two csv files of our Studie_3 file, we must proceed as follows (for each of the two .csv files, separately):

- Click on the .csv file and open in Libre office calculator.
- Make sure the time column is set to US_English:

Textimport - [2024-05-09T09-00-00Z%20.1.2_pp_0.8_bgAccel.csv]

Importieren

Zeichensatz: Unicode (UTF-8)

Sprache: Standard - Deutsch (Deutschland)

Ab Zeile: 1

Trennoptionen

☐ Feste Breite ☒ Getrennt

☒ Tabulator ☐ Komma ☒ Semikolon ☐ Leerzeichen ☐ Andere

☐ Feldtrenner zusammenfassen Texttrenner: "

Weitere Optionen

☐ Werte in Hochkomma als Text ☒ Erweiterte Zahlenerkennung

Feldbefehle

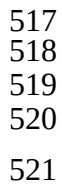
Spaltentyp: US-Englisch

	Standard	US-Englisch	Standard	Standard	Standard	Standard	Standard	S
1					bg	bg	target	
2		UTC		UNIX	accel	brake	low	
3	id	time	Z	time			orig	
4	0	09:01:25	Z	1715245285,9	80		90	
5	1	09:06:20	Z	1715245580,3	79	79	90	
6	2	09:11:24	Z	1715245884,2	80		90	
7	3	09:16:26	Z	1715246186,2	80		90	
8	4	09:21:21	Z	1715246482,0	86		90	

Hilfe OK Abbrechen

3).Now start, in Libre office calculator, the autoISF_factors_proforma.ods ...

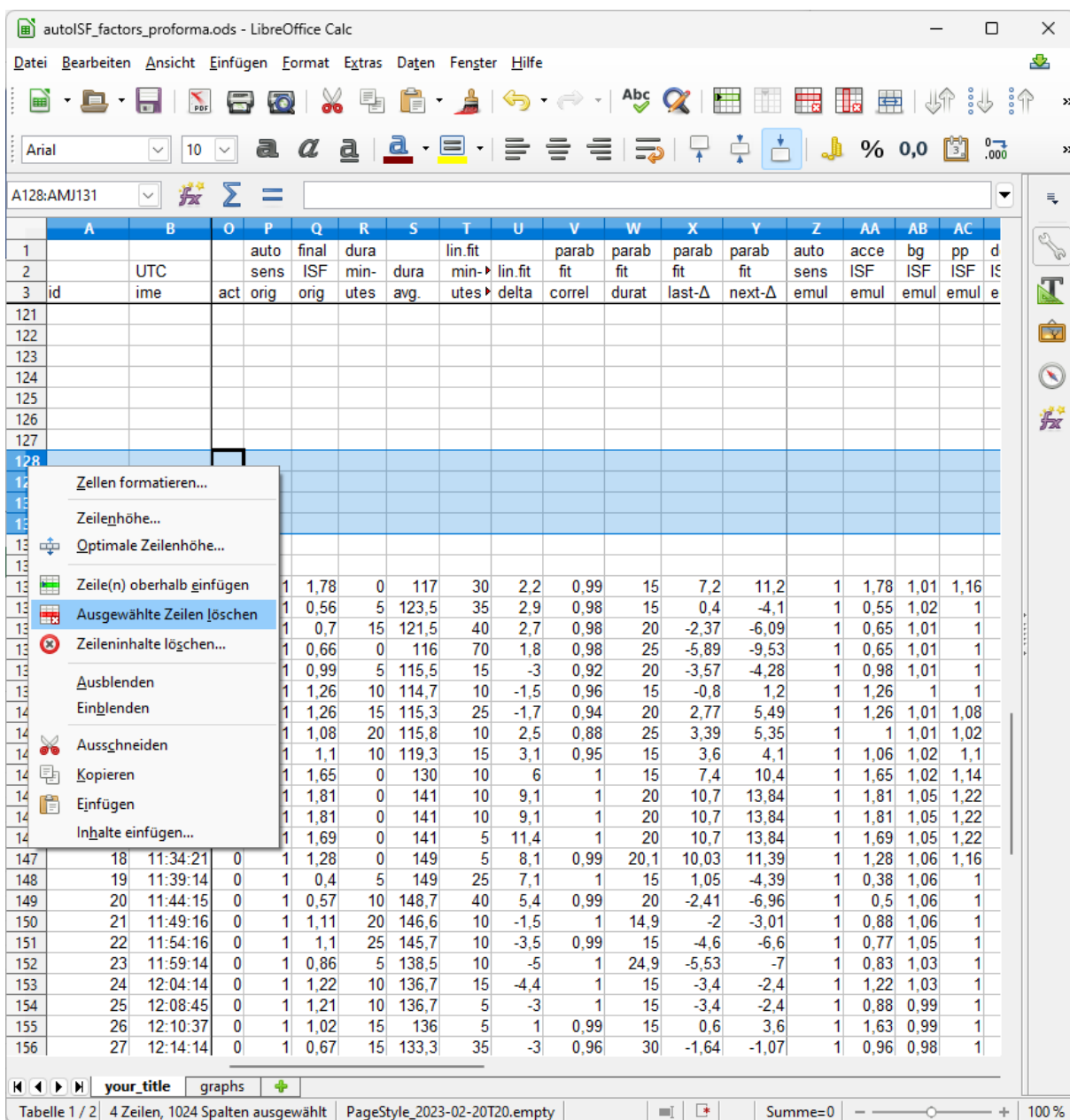
This turns the first 30-some lines of your csv table (left side) into a form in which important effects are highlighted in color, and formatting is improved:



4). In the autoISF_factors_proforma table, highlight 20 or more lines (not including the first or last), and mouse right hand/insert above ...



524 Do this as often as you need to create the number of lines that your emulated csv file comes with.
 525 If you ended up with too many lines, erase the superfluous number (any four, in the example):



526
527

528 5).Then just copy it in, by selecting all data lines in the emulated csv, and pasting (paste special, values only)
 529 into box A4 of your “elonged” autoISF_factors_performa.ods.

530

531 6).The bottom tab “your_title” should be re-named by you, best with day of log you analyze, and your what-
 532 if parameters (so, the name of your csv file could be put in here)

533

534 Now you have a table with optimized lay-out that incorporates key data from both your no change
 535 AND of your investigated changes.csv files.

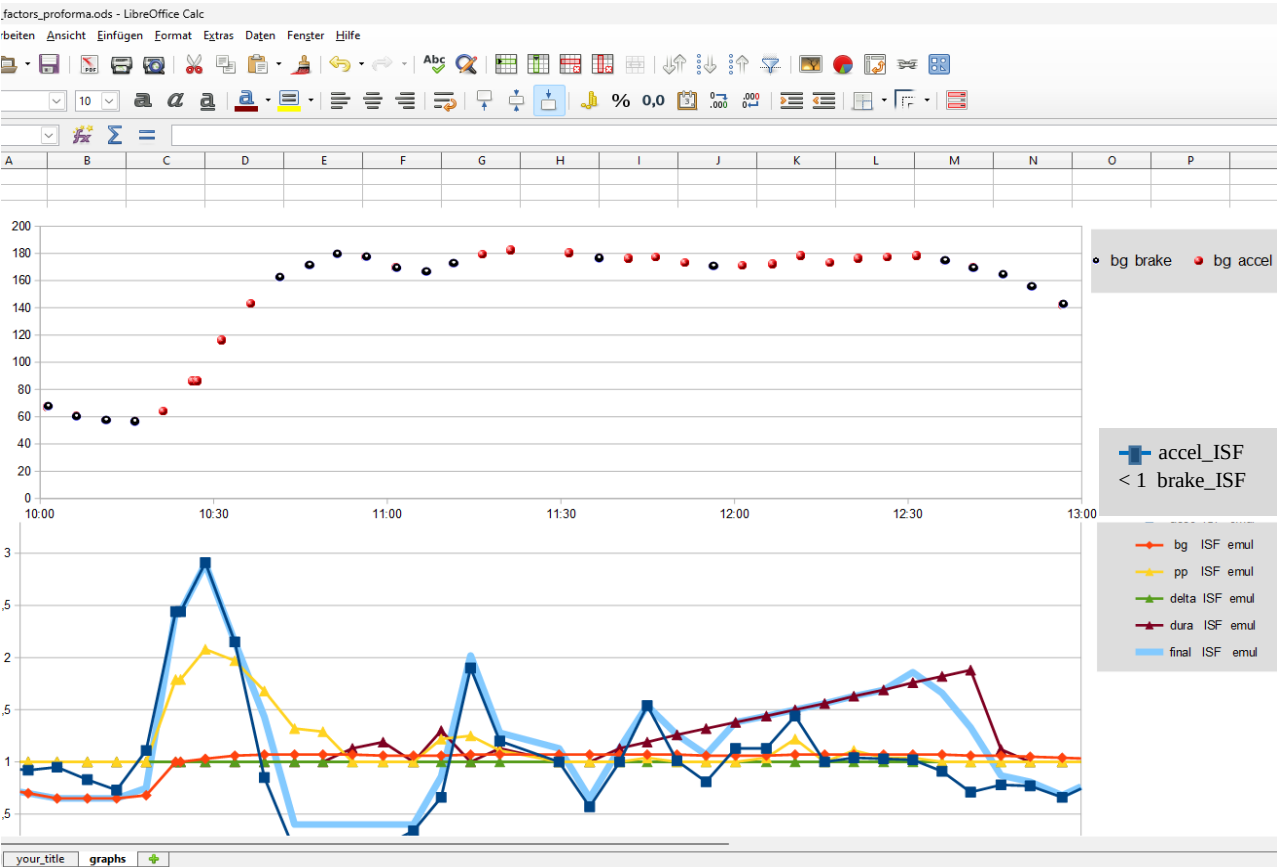
536

A similar table is available on the (i-)phone if you use the autoISF dev variant of iAPS or Trio (see [section 11.3](#))

7).A super neat extra feature is already pre-programmed, which you can see if you click on the bottom **tab “graphs”**.

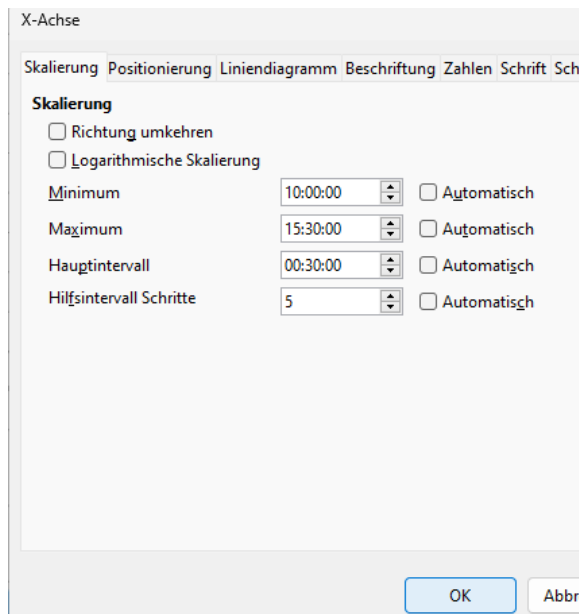
The top graph is the bg curve (the actually seen bg).

Note that for the what-if no bg development over the time range is available. (The noChange one is also given there).



The bottom graph (do one for each, the noChange or the (yourChanges) case) shows the amplification factors coming from each autoISF category, and the overall resulting ISF amplification.

You probably have to widen the time scale (double click on the time axis, and type the desired time span (min and max UTC)(and spacing of data points, 00:30:00 or 00:15:00) into this box:



In the given example above, the 2.5 hours were not enough yet to analyze this 10:30 UTC (12:30 AAPS) lunch; we need to look until bg is near target (hopefully before dinner starts).

10.3.3.4 Chart coming with the emulator

In case you find the extra steps described in the preceding section “too much”: Also the emulator offers one chart (the pdf offered at the bottom of the screen as shown below the “[10.3.3](#) Emulaton results” headline).

First look at the initial bg rise in the noChange.pdf chart (emulation results from your noChange.vdf run), and see how bgAccel_ISF and pp_ISF acted, or could have acted in improved ways.

Then look into in (yourChange).pdf to see potential effects (or what other change to try). (Actually, you probably will have to go into a detailed analysis of several lines and columns of the tables as discussed in sections [10.3.3.2](#) and [10.3.3.3](#)).

Note that ideally we want FCL coverage of our entire “normal day” meal spectrum by **one** set of settings.

So, not much is gained if you put a lot of effort in optimizing FCL settings for one meal.

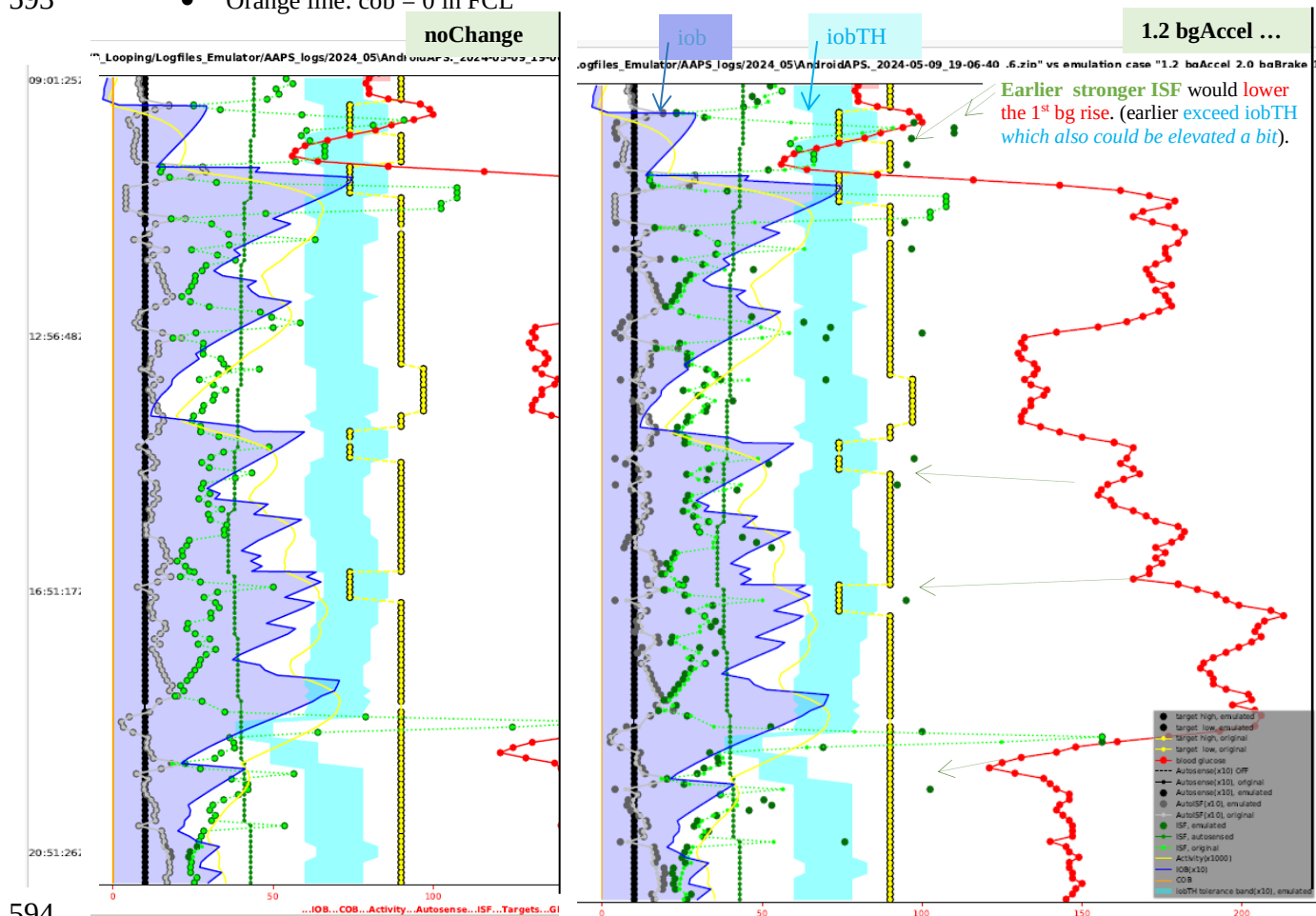
You will need iterations. Do such analysis for **two or three very different meals** that you wish the algorithm to automatically handle. See [section 4.2/4.3](#) on how meals with very different carb loads might benefit or also suffer from aggressive or mild (category)_ISF_weights you could set.

The initial iob received might be limited by allowed SMB sizes, autoISFmax, or the (dynamic!) iobTH. You will have to look into the data table to find out about this (a quick orientation - notably regarding the light blue iobTH band, see next page - is also possible in the pdf result files you have in your project file (project file example “Studie 3” in 2nd chart under the [10.3.3](#). headline).

580 Only once you found OK weights for bgAccel- and pp_ISF_weights, does it make sense to go tune the
 581 dura_ISF_weight. 12:00 – 12:45 UTC in above graph, the resulting effective ISF is dominated by dura_ISF.
 582 Just judging from the picture, a stronger weight might be worth trying. However, we really need to see the
 583 insulinRequired calculation and the further development because impatience about bringing bg values down
 584 faster too often results in hypoglycemia later.

585
 586 The **noChange.pdf** is a chart that shows along the time axis (down), from right to left:

- 587
- Red: the bg curve
 - Yellow: the bg target (note that I do no manual “EatingSoonTT” but for bg rises over +10 mg/dl I have an Automation that sets low TT for a couple of minutes)
 - Light blue corridor: Left edge is set iobTH, and bandwidth +30% (would be +20% at elevated TT)
 - Dark blue line: iob (exceeding twice the iobTH, with temp. SMB shut-off)
 - Orange line: cob = 0 in FCL
- 593



- As bg did not convincingly come down enough, one could hypothesize that iobTH should be elevated. ((But, again, this would have to be confirmed also with other kinds of meals)).
- Thin yellow line: Insulin activity
- Green dotted line: ISF as would result from AAPS w/Autosens

600 • **Green scatter points:** autoISF ISF no Change (lighter points) or **what-if (darker points)**

601 Foreseeably, this is the strongest difference between our noChange (left) and 120% bgAcel_ISF_weight
602 (right) in the picture below. (Note the red bg curve is *both times* the really seen bg, because the what-if
603 case only looks at each single loop decision). The first (->) time the dark green dot is far to the right, this
604 would get the bg down, we would start to see a (<-) bg lowering effect, shifting the red curve to the left

605 • Black line: Profile ISF

606 • Gray scatter points: ISF weakened (to the left of black line) or strengthened (to the right)

607
608 Regarding the other changed parameters: Stronger dura_ISF would suggest more insulin towards the end of
609 plateaus; this should have helped in the 1st plateau (red curve, top right quadrant of the picture). However,
610 same setting would have to work also on 2nd plateau; the chart cuts off there, so too early to see whether a
611 hypo danger might result.

612 Effect from doubling the bgBrake_ISF effect are hard to evaluate. Better probably to look in .csv tables, or
613 run a separate emulation for that change only.

614

615 Always check for 2 or 3 kinds of your meals whether the “new” parameter settings really are on average
616 better. (See negative example in [case study 8.2!](#))

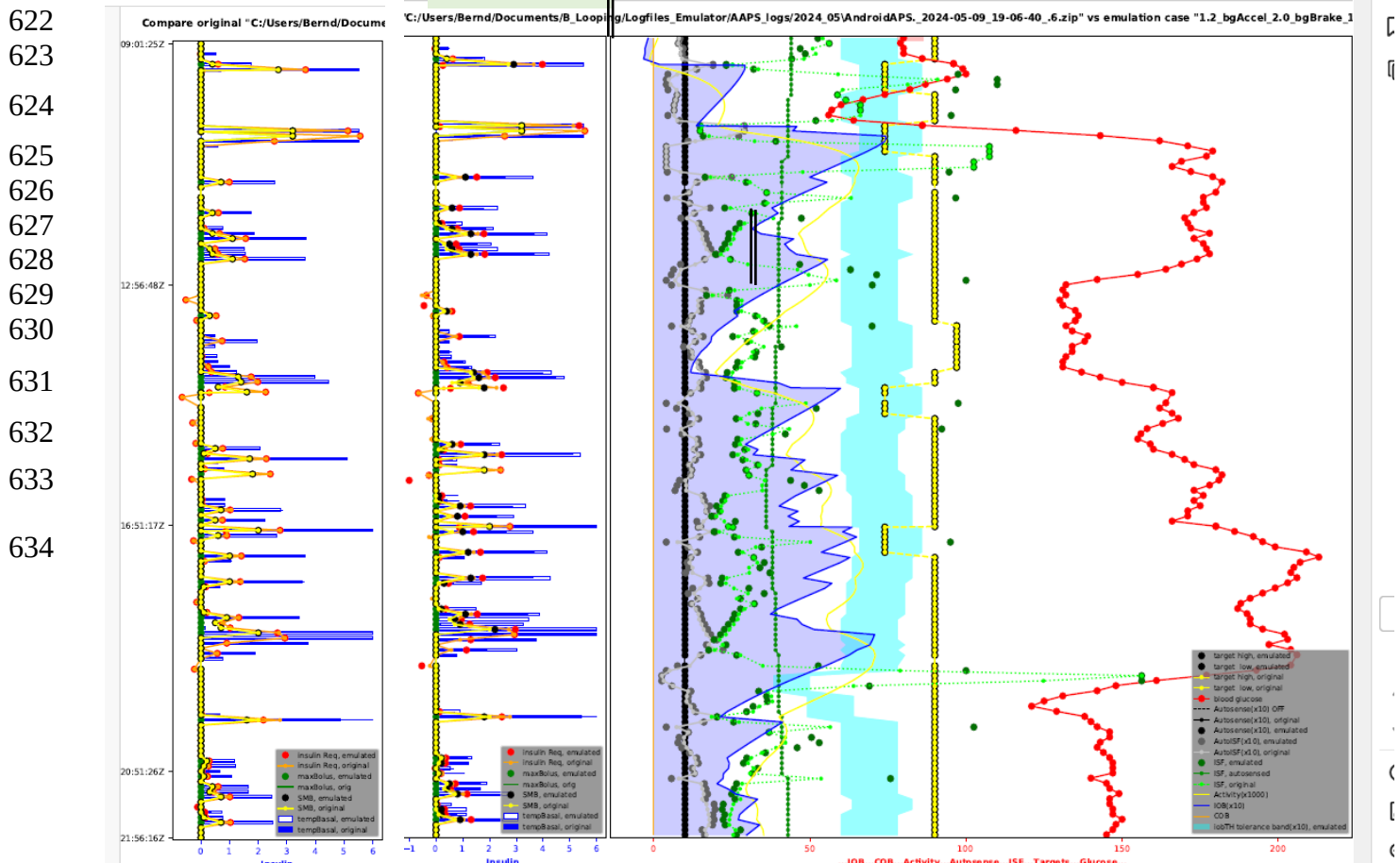
617

618 Part of both above shown charts (left side of each, with blue peaks) was cut out.....

619 (unfinished / to be explained later) (Here: yourChange = 1.2_bgAcel_2.0_bgBrake_1.2_dura)

620

621 ←



635 10.3.3.5 delta table from “what-if” run with (yourChange).vdf

636

637 In case you want to analyze delta, short and long average deltas, see [section 10.2.4.5](#)

638

639 To analyze deltas in a “what-if” scenario really does not make much sense, because effects from

640 each single change ripples through many subsequent situations, and it is impossible to predict how

641 glucose curve, and therefore also how deltas, would develop in the what-if case.

642

643