10. Tuning of autoISF settings for Full Closed Loop aided by the Emulator
V.3.3

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
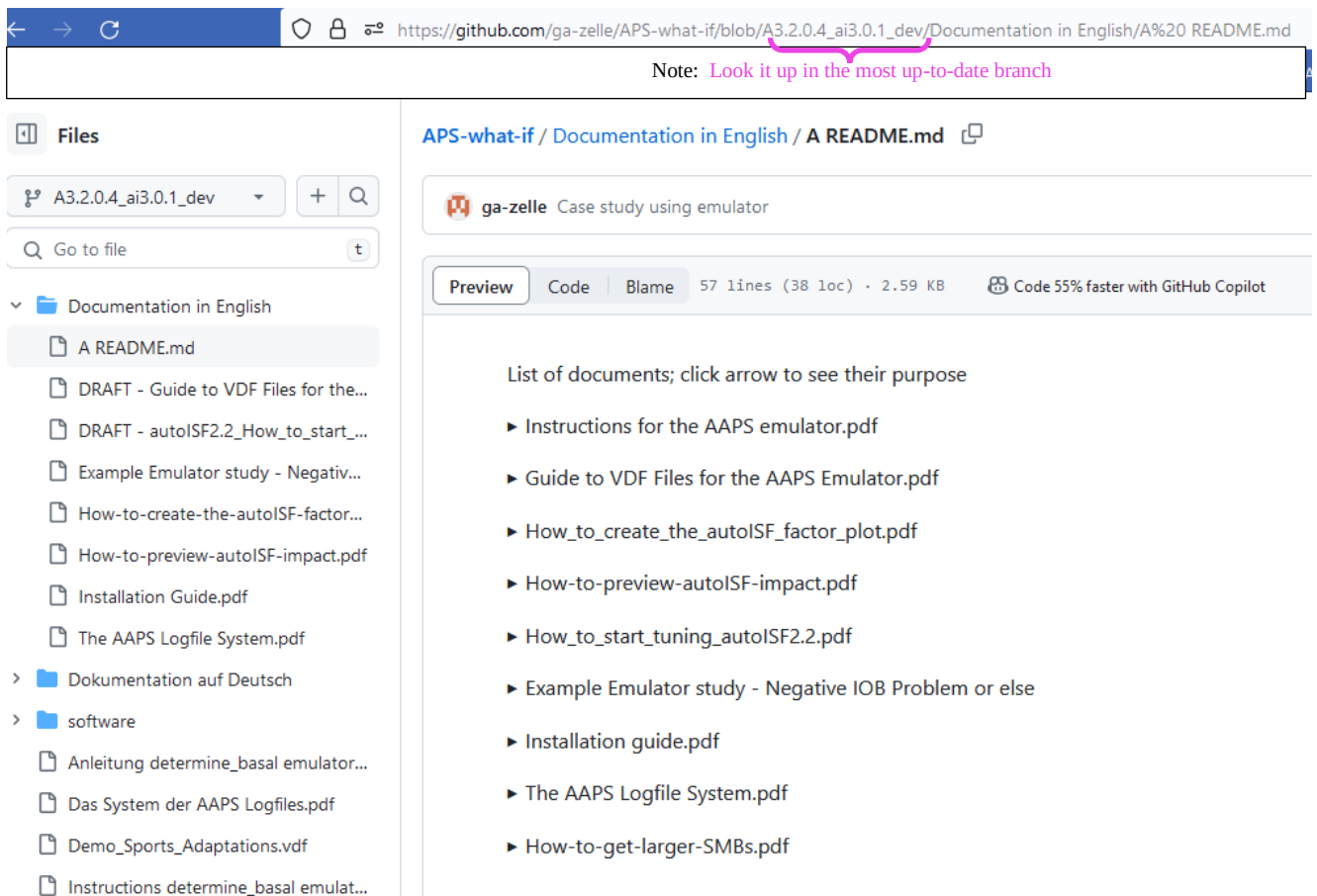
Available related case studies:

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

Files

A3.2.0.4_ai3.0.1_dev

Go to file

- 📁 Documentation in English
  - 📄 A README.md
  - 📄 DRAFT - Guide to VDF Files for the...
  - 📄 DRAFT - autoISF2.2_How_to_start_...
  - 📄 Example Emulator study - Negativ...
  - 📄 How-to-create-the-autoISF-factor...
  - 📄 How-to-preview-autoISF-impact.pdf
  - 📄 Installation Guide.pdf
  - 📄 The AAPS Logfile System.pdf
- 📁 Dokumentation auf Deutsch
- 📁 software
  - 📄 Anleitung determine_basal emulator...
  - 📄 Das System der AAPS Logfiles.pdf
  - 📄 Demo_Sports_Adaptations.vdf
  - 📄 Instructions determine_basal emulat...

APS-what-if / Documentation in English / **A README.md**

ga-zelle Case study using emulator

Preview | Code | Blame   57 lines (38 loc) · 2.59 KB   Code 55% faster with GitHub Copilot

List of documents; click arrow to see their purpose

▸ Instructions for the AAPS emulator.pdf

▸ Guide to VDF Files for the AAPS Emulator.pdf

▸ How_to_create_the_autoISF_factor_plot.pdf

▸ How-to-preview-autoISF-impact.pdf

▸ How_to_start_tuning_autoISF2.2.pdf

▸ Example Emulator study - Negative IOB Problem or else

▸ Installation guide.pdf

▸ The AAPS Logfile System.pdf

▸ How-to-get-larger-SMBs.pdf

40

41

42 In the emulator, you can see in tabular and graphical form, which autoISF component, and other

43 settings, contributed to SMB values that determined the glucose curve.

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

45 Application examples for tuning are given in associated case studies (we need more).

46

47   Note that iOS based variants of autoISF (Trio or iAPS) can not use the emulator.

48   Refer to section 11.3.

49

50

51 Join https://discord.gg/n3tD5eXExC for seeking (and giving) help with the emulator set-up or use,

52 and to exchange experience.

53

54

10.1 Installation of the emulator on your PC

56

10.1.1 Downloads

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

58 • First download **QPython 3L** onto your PC (detailed instruction see ... APS-what-if / **Instruc-**
59 **tions** determine_basal emulator

60   • Then download from: https://github.com/ga-zelle/APS-what-if/ **software** : each of these 5 py resp.

61   config. files. To do this, you must press, for each one, here
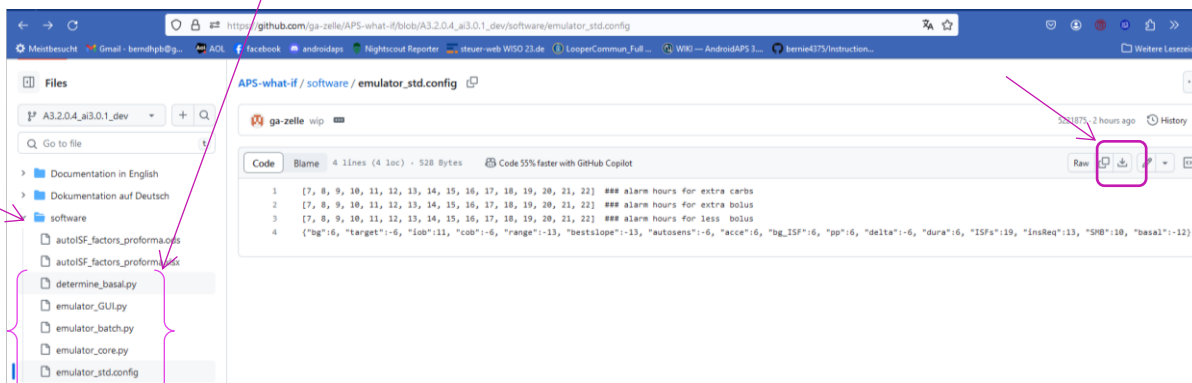
62
63   Always make sure you use the files from the branch with the same version number as your AAPS version (in
64   the example above: These files will work with AAPS dev version 3.2.0.4 with autoISF version 3.0.1).   .
65   Always keep your AAPS x autoISF and also the emulator related files up-to-date. If you can't get your
66   Emulator run, look in the Github repo whether there is a newer .py file (even with the same name; there may be
67   updates that iron out problems that may have been reported only with certain AndroidOS versions etc etc))!!
68
69
70   10.1.2 Create your PC folder structure (- you can of course deviate from the specific example given below)

71   Retrieve these 5 downloaded files on your PC (list of recent downloads), and shift them into a folder in

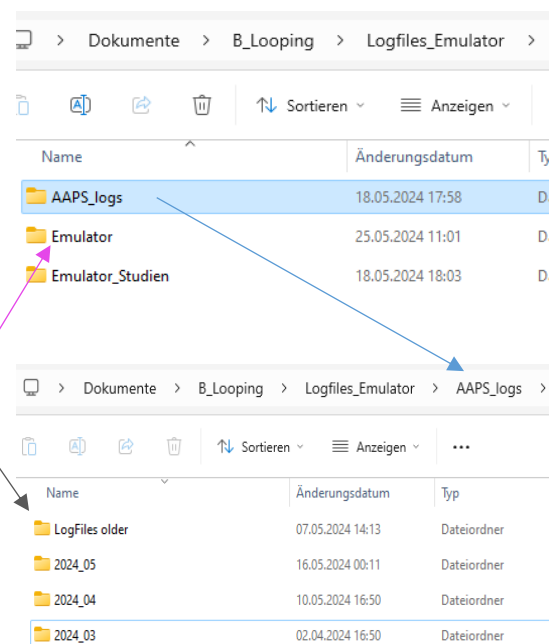72   which, or neighboring to which, you also keep your logfiles.

73   (These you must copy-in from your phone before they get auto-

74   matically erased there after x days; check your x out; ~ 2 weeks

75   normally, smaller with Libre3 (1min.)).)-

76

77   My folder structure for Logfiles and Emulation on the PC

78   has 3 monthly folders, plus one folder with data from previous

79   months and years which I am not likely to analyze any more.

80
      It is advisable to additionally store a pdf from **Nightscout**
      Reporter in the file for every month. From it, you can
      much easier find which days and times are of high interest
      to analyze with the emulator.

              Always factor in the time difference between your AAPS
              phone and the "Z" time used by AAPS.
      Neighboring the AAPS_logs is the Emulator file where the
      5 downloaded files went:

81

82   Note: Use 1minute:emulator_std.config in case you use Libre3 (1 min) as your CGM

83   **10.1.3 Create a "start emulation" button on your desktop**

84   One of these files is "**emulator_GUI.py**"

85   • Create a **link to** it

86   • Drag that link onto your **desktop**

87   • Name it something like Emulator start.

88

89   **10.1.4 Fetch two more .py files**

90   Repeat steps 10.1.1. and step 10.1.2. for 2 more py files from https://github.com/ga-zelle/Scan-APS-

91   logfiles/blob/main   and download also these:

92
93
94   Include these in your emulator file (as in example shown in 10.1.2 above).

95 10.2 Analyzing loop decisions in logfiles

96 Instead of making many screenshots every 5 (or, w/ Libre3, every 1) minutes after meals, and analyzing them

97 later, a much more elegant and powerful way to analyze your loop decisions (and how you might want to

98 influence them with different settings, see section 10.3 for this), is to use the emulator.
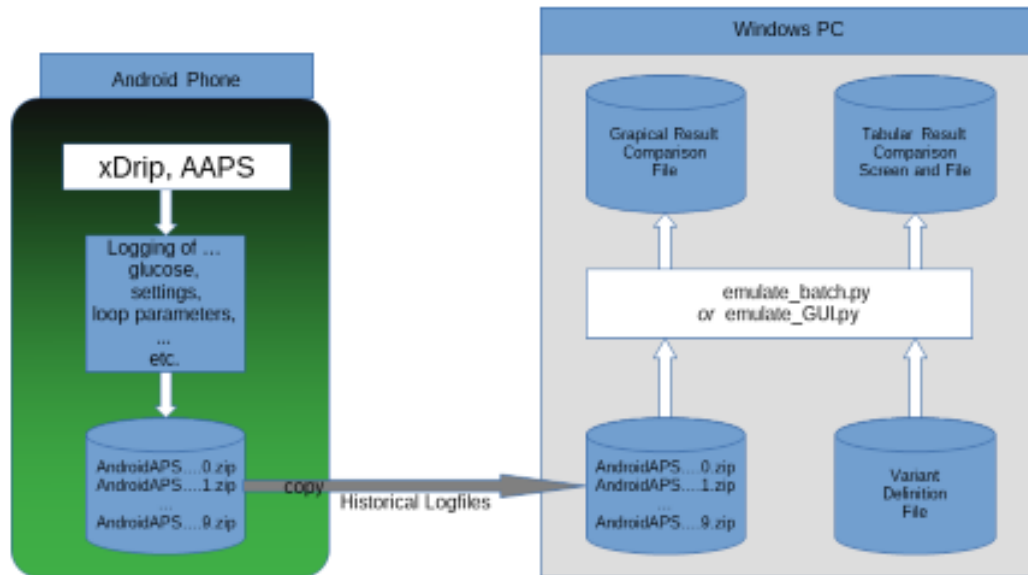
99 Sketch of Running the Emulator on a Windows PC   Github/ga-zelle / APS-what-if

100

101 10.2.1 Set up a "no change" .vdf file.

102

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

104   Note: for "what-if" analysis, entries will be made (in a second .vdf later, see section 10.3)
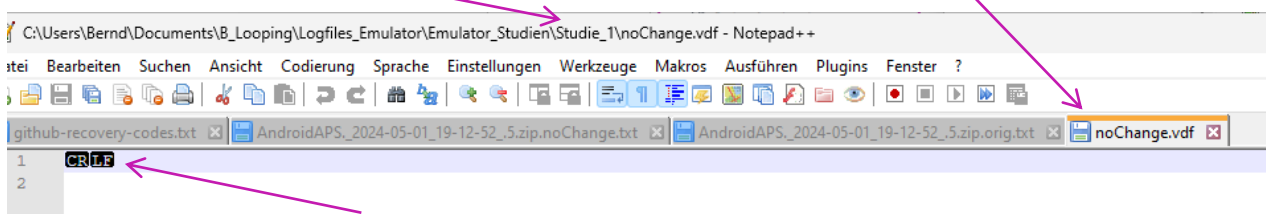
105

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

107

108 Store that in a file of your current emulator project you are about to start (see my storage path in top line

109 here)

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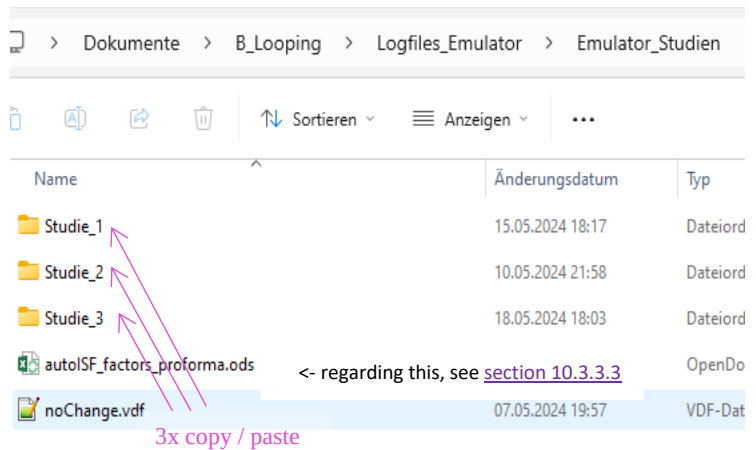

111
112 Erase any entries after CR LF   and also in lines 2 ff, if any

113

114

115   Store this noChange.vdf in your Emulator-

116   studies file (it neighbors your Emulator and

117   Logfiles files).

118

119   From that position, you always make a

120   copy and paste *into each* Study_*1 …n :*

121

122

123

124

125   10.2.2 Locate relevant logfiles

126

127 Make sure you have the AAPS logfiles that you want to analyze in an immediate-neighboring file to

128 your Emulator-- and to your Emulator_Studies- files (File structure as suggested and shown

129 above).

130 Copy (not: move!) your noChange.vdf (see above) also into your Study file (must be in all of them).

131

132

133   10.2.3 Prepare the emulator

134

135 Now go to your PC **desktop**, and start the emulator by **just pressing the button "emulator start"**

136 that you installed in step 10.1.3

137

138

139 This opens a big dialogue box with 3 fields that you must fill in with the applicable path (*without* any

140 quotation marks "..") from your Windows Explorer file system, best done via (3x) Browse button:

141
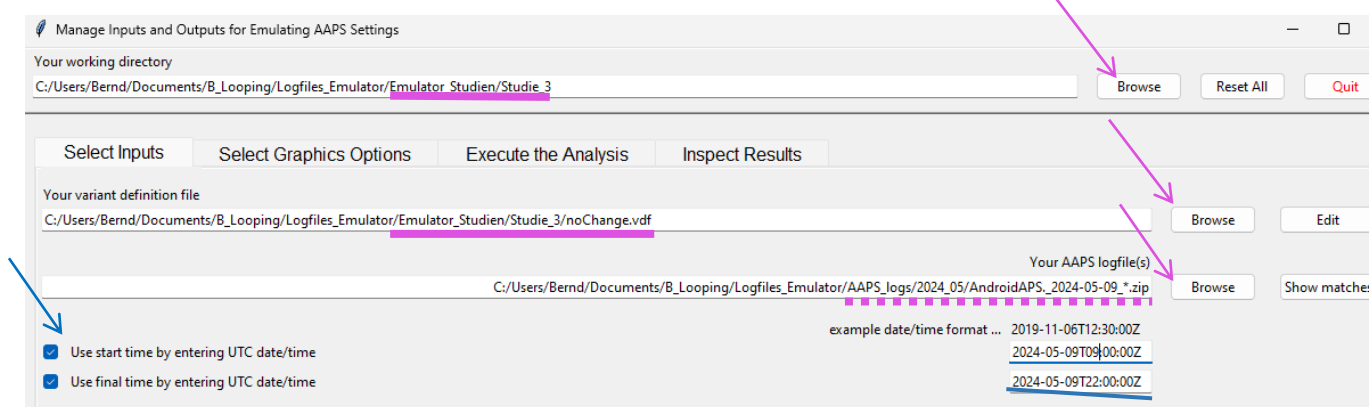
a) The top box marks the path to your current emulator project ("Studie_3" is where I want to
   store results)

b) The middle box marks the path to your current vdf (what kind of analysis; here:
   "…noChange.vdf" = *read-only*; see section 10.3 for *what-if*)

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 ex-
  ample)

- 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 :

    o clicked the bottom left two boxes

    o copied the date 2024-05-09 over the default date in the bottom right two data fields

    o 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.

162

163

164

165 10.2.4 Run emulation

166

167 Now we are ready to go: Press "Run emulation"

168   This produces sometimes an error message (e.g. if you have a syntax error, or incompatible software

169   versions: => seek help, in the Github materials provided by ga-zelle, or in Discord/Full-Closed-

170   Looping/emulate-aaps here: https://discord.gg/n3tD5eXExC

171

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

173 you could have a quick look into the **.log** file to see whether the run had errors (see section 3.)

174

175 10.2.4.1   SMB tab contents in (date..) **noChange.txt** result file

176 This basically gives you "all the SMB tabs" without needing to make screenshots every 5 minutes.

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

178   By using the **search function** you can jump, in that super long list, to all places that e.g.

179   have „autoISF" in it or „script debug", or „SMB disabled" (if you want to analyze when that

180   happened). Precise spelling, as in this .txt (or in SMB tab) is of course important.

181

182
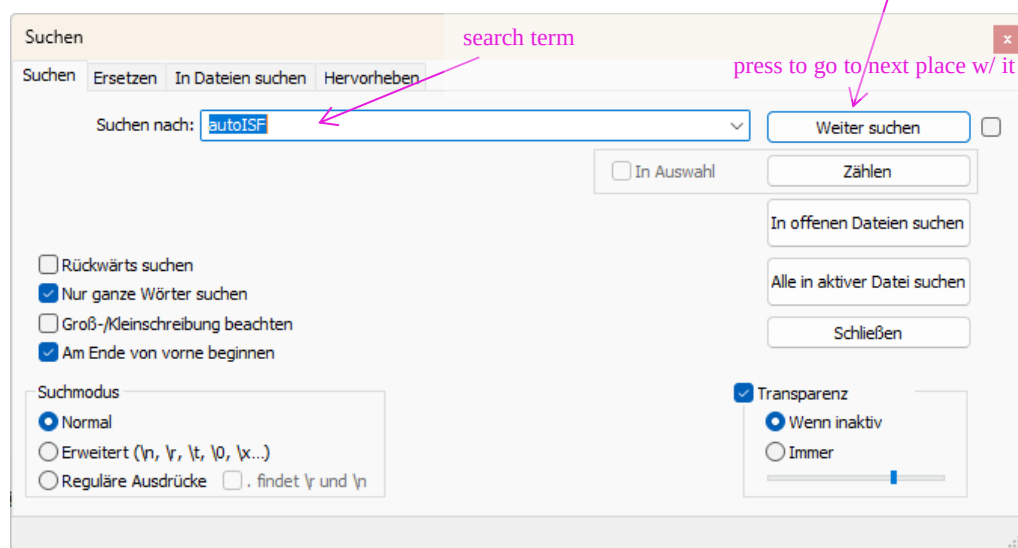
183 10.2.4.2   Table of results (…**noChange.csv** file)

184

185 The .csv file in your project folder gives a tabular presentation of how parameters like bg, iob,

186 iobTH, the various ISF contributors, bg target, insulinRequired etc. develop every 5 minutes, and

187 what SMB size and %TBR resulted.

188 It is a vast table, so you may want to reduce it to something more "digestable", either after transfer

189 to your standard calculation program (next section 10.2.4.3). You can also make settings to suppress

190 information you are usually not interested in (or do not know how to interpret, anyways) under "Select

191 Graphics Options" when you open the emulator, before executing any analysis:

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

193 Then select whether you want "All" possible outputs in the graph, or "Most" = all except those you tick "off"

194 in the boxes for each output parameter.

195   In case you would use "Some/just a few", you would have to tick those few you that do want to see, by ticking

196   the corresponding boxes.

197 Recommendation is to look at (nearly) everything offered (as your default setting that you can leave

198 untouched in most of your emulator runs):
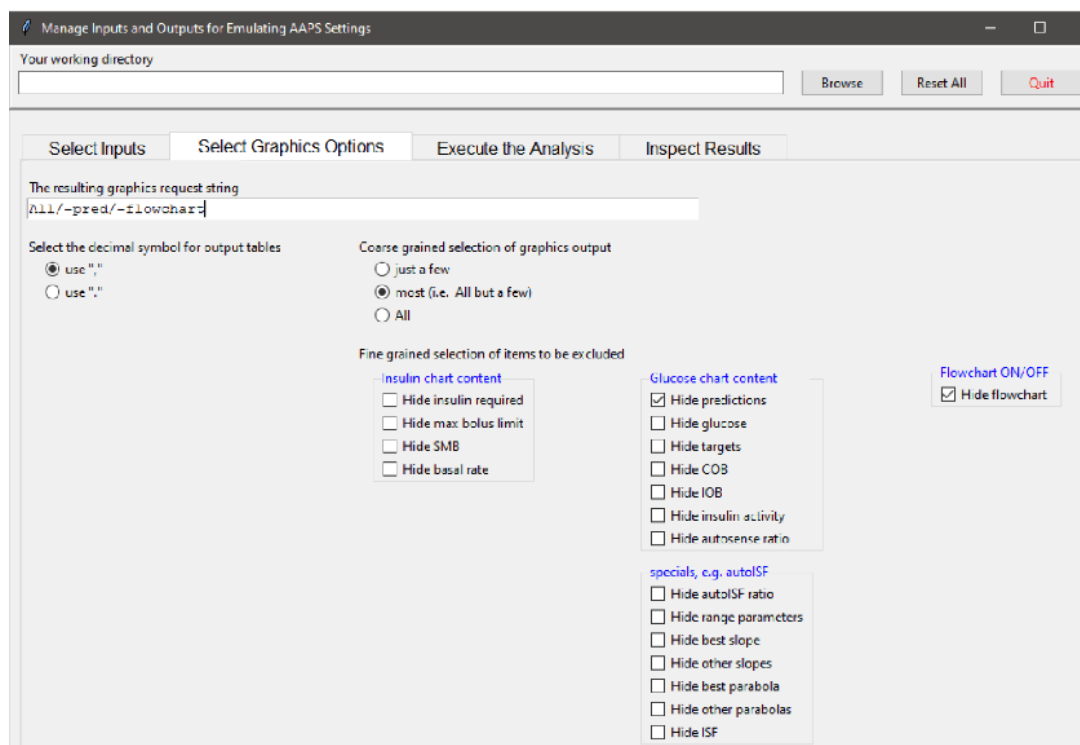
199

200 It might be easier, to not deal with customizing the csv file, and rather copy the data into your

201 favorite calculation program:

202

203 10.2.4.3 Analysis of the **noChange.csv** table in Excel or LibreOffice calc.

204

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

206 • add right next to the standard world time your corresponding "AAPS **time"**

207   For instance, adding +2/24 translates the UTZ column into central European summer time

208   column next to it (where currently a row of Z stands). Likewise, subtract like -5/24 from UTZ

209   for an US East Coast time scale.

210   Highlight all time fields (the entire columns), and switch from hh:mm:ss format to hh:mm.

211   (While the seconds are important for the loop's calculations, for our comparison with

212   Nightscout or other charts and data, it is much easier without the seconds attached)

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

214   That way, "boxes" (data fields) retain their original position in tables

215 Also, in case later you want to look into additional info, you can simply un-hide the relevant
216 columns … or time lines:

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

218

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

223 In section 10.3.4 we present an extra tool that does a standardized table reduction and color marking
224 for you!

225

226 You may be able to formulate a hypothesis or two, what settings (…ISF_weights, iobTH%,
227 SMB_range_extention, autoISFmax …) should be changed for improvement (then go to 10.3)

228
229 10.2.4.4..   Graph **noChange.pdf**
230
231 After your emulation run, under Inspect Results, you can open the pdf file that is last in the results list
232 offered.

233

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

235 • Red: the bg curve

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

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

240 • Dark blue line: iob (exceeding twice the iobTH, with temp. SMB shut-off

241 As bg did not convincingly come down enough, one could hypothesize that iobTH should be
242 elevated. ((But, again, this would have to be confirmed also with other kinds of meals)).
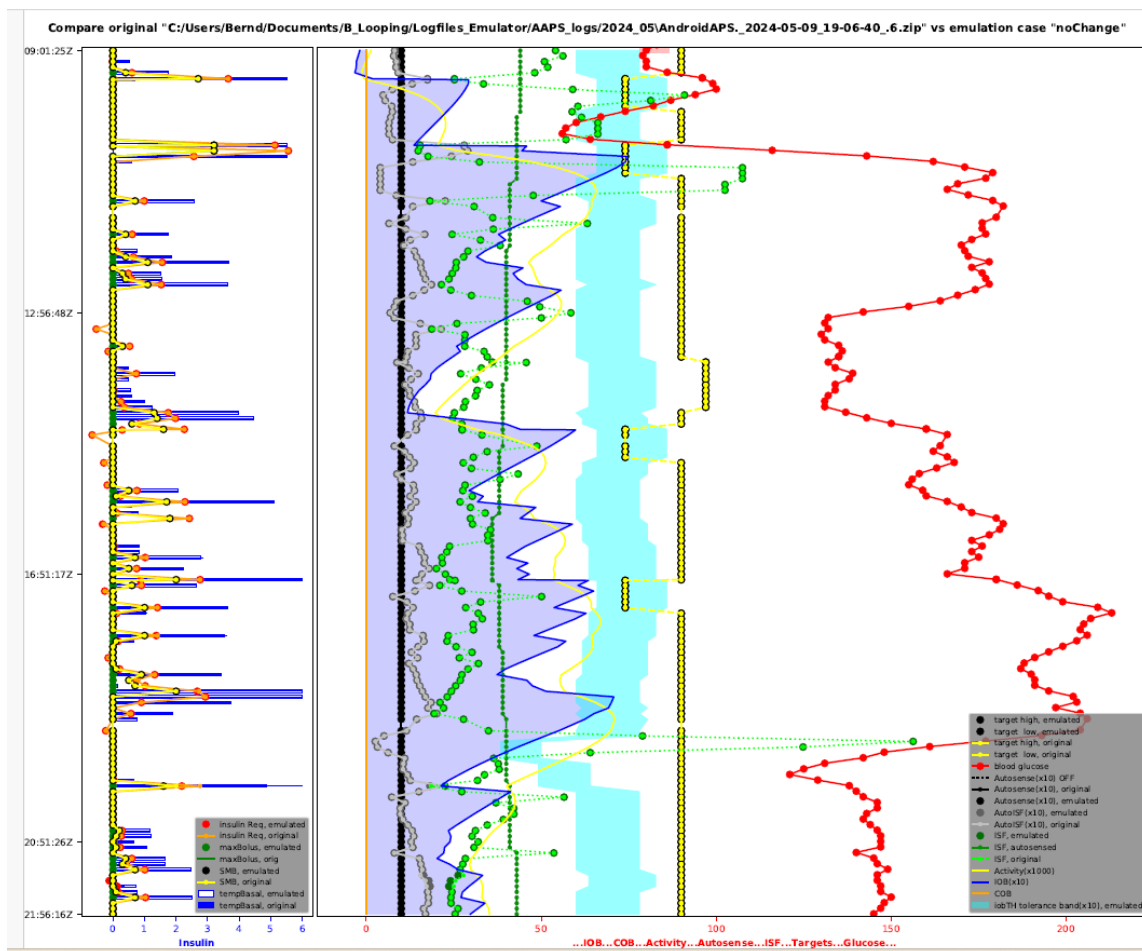
243 • Thin yellow line: Insulin activity

244 • Green dotted line: ISF as would result from AAPS w/Autosens

245 • Green scatter points: autoISF ISF no Chage (lighter points) or what-if (darker points)

246 • Black line: Profile ISF

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

248   • Orange line: cob=0 at all times (in FCL)

249
250

251   More see discussed together with (yourChanges).pdf in section 10.3.3.4

252
253

254   **10.3   "What-if" analysis using the emulator**

255

256   In the following you see an example how you can analyze a day of logfiles, and selecting the time span of

257   interest, for instance 11-24 h to look at how autoISF managed lunch and dinner.

258

259   You have to go through the emulator twice. You already did it (section 10.2.) using the no change.vdf, and

260   now start over with the same emulator with your (desired changes).vdf. Repeat, if you have two or more such

261   vdf defined. (Just clear results before executing analysis each time. No worries, all results are automatically

262   captured for all runs, all in your selected study file).

263

264

265

266

267 10.3.1 Define your investigated changes in a, or in several, (yourChanges).vdf

268

269 Define for which one to max three parameters in your current profile settings you want to look into a

270 different setting. Recommendation is to use a factor, like for example   current setting * 0.9 , or current

271 setting * 1.2, and use that in your naming for this vdf file, too. Check APS-what-if /Documentation in

272 English/Guide to VDF Files for the AAPS Emulator.pdf in section 3.

273

274 Within the same study, you can make several runs with several vdf files.

275 All results, like the csv results table, will appear then several times in your study file, only with different

276 name endings as in the underlying vdf.

277

278 Example: I like to check in my actual data (they are in my noChange.vdf emulator run**), in which time

279 points the following parameter changes would make a** (how) big **difference** in the loop's decision:

280   • 20% higher bgAccel_ISF_weight to boost the first SMBs stronger: How would that tend to ramp

281   up early iob; and might that get too strong in other parts of the data? Or does it bounce into a

282   restriction (maxSMB size; autoISFmax; iobTH…) that I might need to widen?

283   • Doubling my cautiously set bgBrake_ISF_weight shall give me insight into the workings of that

284   parameter (and whether using a much smaller weight than for bgAccel_ISF_weight is really

285   what I should keep doing)

286   • As my bg came down from a persistent high quite slowly, I elevate the dura_ISF by 20%

287

288   Actually, it would make more sense to first find my "optimal", maybe indeed elevated,

289   bgAccel_ISF_weight. *Then*, do a noChange (!) run **with that,** plus a (yourChanges) run with the stronger

290   dura weight, investigated on that basis.

291   Reason: 1) As we always say, better do only one change at a time. 2) A better job with bg control via

292   bgAccel_ISF will reduce the peak height and provide a different (easier) scenario for dura_ISF to manage.

293

294 Now, to **write** your **(yourChanges). vdf for the emulator** (this is same procedure as you did in section

295 10.2.1 for the noChange.vdf):

296

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

298   Alternatively you can also take another pre-existing vdf file, and copy it into your current project

299   giving it a new name (re-name it)

300   • name your vdf (in our example: 1.2_bgAccel_2.0_bgBrake_1.2_dura.vdf) …
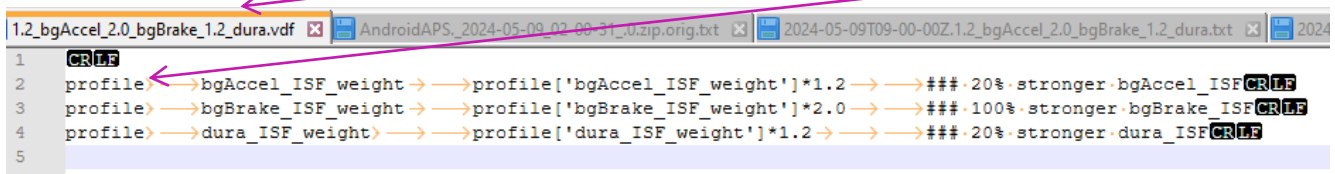
301 • … and store that in a file of your current emulator project you are about to start (see my storage
302   path in top line here)

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

305

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

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

308
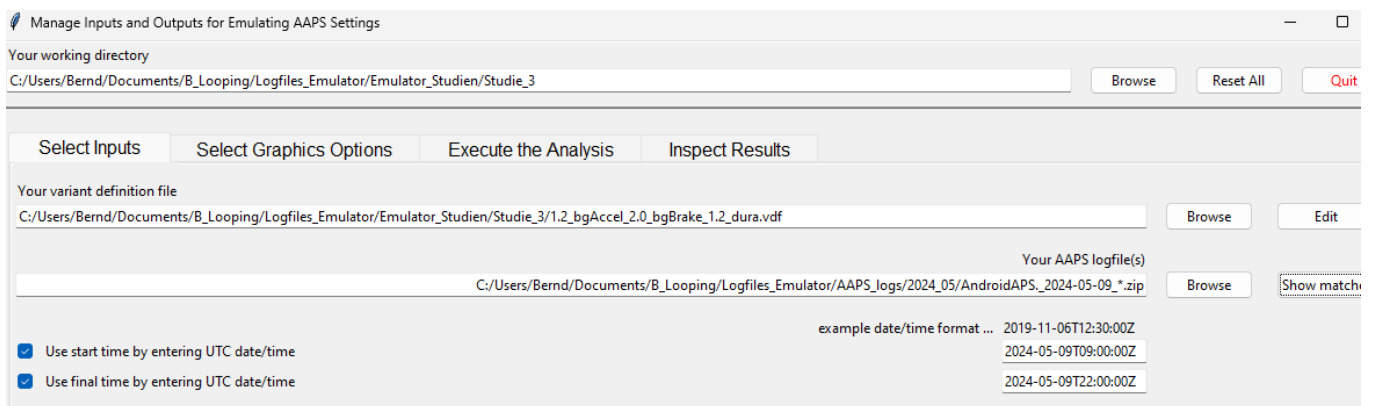309   The (yourChanges) .vdf should look like something like this:
310

311
312   CR =   LF=   Erase any entries after CR LF   and also any entries in lines below, if any

313

314 10.3.2 Run the emulator with (yourChanges).vdf

315

316 The "what-if" emulator run is done the same way as you did the noChanges run (section 10.2), however,
317 now , the **(yourChanges).vdf** must be loaded into the 2nd input field, where formerly you had the
318 noChange.vdf.:

319

320
321 In the 3rd input field, give the path to your stored logfiles. A good way to do this is:

322 • Browse in your Windows Explorer to any logfile from the desired day (2024-05-09 in above ex-
323   ample)

324 • Replace the time with an asterix * (this means you look at all-day data, in UTZ time). Check
325   whether this will work by pressing Show matches . You should see all logfiles from that day in a
326   pop-up info box.

327 • As I wanted to look at 11 am –midnight for lunch and dinner related data, I :

328   o   clicked the bottom left two boxes

329   o   copied the date 2024-05-09 over the default date in the bottom right two data fields

330   o   after T (for time), I entered the desired time of analysis AFTER conversion into my local

331   time (Central EU summer time minus 2 hours = UTZ; so to look at 11 to midnight of

332   my AAPS screen, I must enter here 09.00:00Z, and below it 22:00:00Z).

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

334 Emulation, you can look the results up under "Inspect Results". First you could have a quick look into

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

336

337 10.3.3 Emulation results

338

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

341 already stored, in our example into the "Studie 3" file, below:

342

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

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

345   bgAccel_ISF (with the intention to test this, and a noChange, on a low carb meal later.

346

347

The file browser shows:

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
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
1.2_bgAccel_2.0_bgBrake_1.2_dura.vdf	17.05.2024 20:38	VDF-Datei	1 KB
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 **(yourChanges).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 (yourChanges) 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.

368 Note that any relevant change would put your bg curve on a different trajectory, so that would influence *all*
369 *following* results. Therefore, what you get here is **not** a complete modelling how your bg would have
370 developed in the alternative scenario.
371 But you can investigate in which stages the parameter(s) you are looking at in your current "what-if" had big
372 influence, and in which direction the changes would go. (see also charts shown in <u>section 10.3.3.4</u>).
373 Analyzing how to safely come down from a high glucose plateau while limiting hypo danger towards the end
374 of digestion is also to some extent possible.
375
376 A good other way to employ the what-if analysis is real time, on your smartphone, using speech synthesis
377 (see <u>section 11</u>): Then you get real-time info, as to exactly when a significantly different proposal would
378 emerge, and can decide (and watch!) real-time whether to follow the new idea and not was probably better.
379
380 Observe that a setting change must work well for you
381   •   not just in one point of time, and
382   •   not just for one kind of meal,
383 but you must look at all time slots in the investigated meal, plus analyze with the same tool a totally different
384 meal within your usual spectrum, how things work out there
385
386 10.3.3.2 Tabular results
387
388   A)   .csv results table and spreadsheet copies of it
389
390 The **noChange.csv** table gives all relevant data. Besides development of bg and iob you see the calculated
391 insulinRequired in each loop decision, and how each of the autoISF categories contributed to the decision
392 (notably regarding SMB size).
393
394   Note that the "**acce_ISF**" results are only in case of positive acceleration (that is our main focus)
395   driven by the bgAccel_ISF_weight setting. (These are all positions > 1.0 in the "acce ISF" columns).
396
397   **In case of negative acceleration** (decelerating rise, positions **< 1.0** in the **"acce ISF"** columns),
398   **bgBrake_ISF_weight is applied**. As discussed in <u>section 4.4</u>, bgBrake_ISF might be most
399   important (and interesting to analyze) in slowly resorbing meals.
400
401   Note: maxBolus=0 means in this table that SMBs were not capped by maxBolus.
402
403
404 The **(your changes).csv** shows in detail how **every single** loop **decision** would be influenced by the different
405 settings you are investigating.

406 To inspect that huge table, click on the Z behind the start UTC time entry (see black box in the Z column of

407 the table, next page).

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

409   box.

410 Then, go to window/fix. Now you can scroll through the data and always see headline and time (or time and

411 bg level).

412   To further ease analysis, feel free to temporarily erase (hide) any columns that you (think you) do not

413   need for the intended analysis. More suggestions see in section 10.2.4.2

2024-05-09T09-00-00Z .1.2_pp_0.8_bgAccel.csv - LibreOffice Calc

Datei Bearbeiten Ansicht Einfügen Format Extras Daten Fenster Hilfe

Fenster menu:
- Neues Fenster
- Fenster schließen   Strg+W
- Teilen
- **Fixieren**
- • 2024-05-09T09-00-00Z .1.2_pp_0.8_bgAccel.csv - LibreOffice Calc
- autoISF_factors_proforma.ods - LibreOffice Calc

Liberation Sans | 10

C4   Z

id	UTC time	Z	UNIX time	bg accel	bg brake	target low orig	orig	emul	emul	cob	iob	emul	emul	act	orig	orig	dura min-utes	a
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	

2024-05-09T09-00-00Z .1.2_pp_0.8_bgAccel

Tabelle 1 / 1   Standard   Summe=0   100 %

414
415

416 Still, the csv tables are overwhelming. You could proceed **in either of two** directions now:

417

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

autoISF_factors_proforma.ods is provided as an **extra tool** that you download from here:
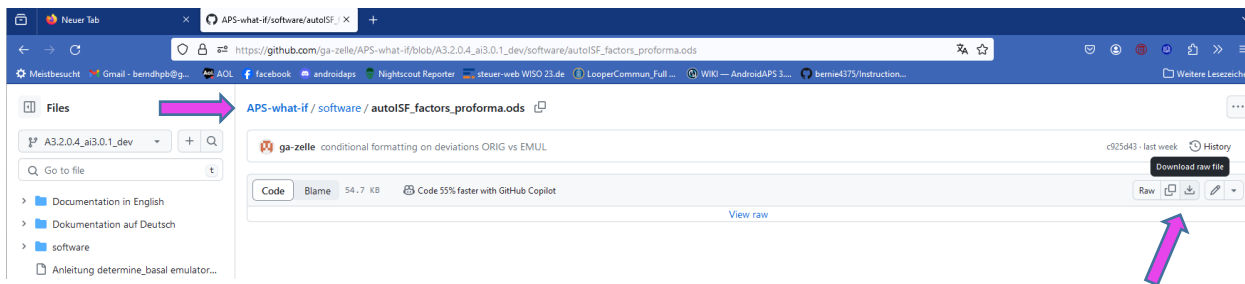
Put that file on your PC one level above the single files for all your studies:
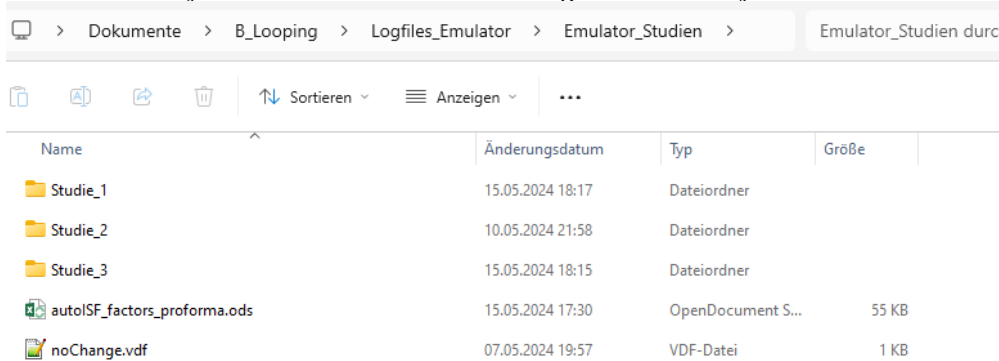
Now, if we want to use this tool on the two csv files of our Studie_3 file, we must proceed as follows (for *each* of the two .csv files, *separately*):

1. Click on the .csv file and open in Libre office calculator.

2. Make sure the time column is set to US_English:

437
438
439   3.   Now start, in Libre office calculator, the autoISF_factors_proforma.ods …

440   This turns the first 30-some lines of your csv table (left side) into a form in which important effects are

441   highlighted in color, and formatting is improved:

442
443
444   Now, you want this for the entire table.

445   In the autoISF_factors_proforma table, highlight 20 or more lines (not including the first or last), and mouse

446   right hand/insert above …

447

448

449   Do this as often as you need to create the number of lines that your emulated csv file comes with.

450   If you ended up with too many lines, erase the superfluous number (any four, in the example):

autoISF_factors_proforma.ods - LibreOffice Calc

Datei Bearbeiten Ansicht Einfügen Format Extras Daten Fenster Hilfe

Arial   10

A128:AMJ131

	A	B	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1				auto	final	dura		lin.fit		parab	parab	parab	parab	auto	acce	bg	pp	d
2		UTC		sens	ISF	min-	dura	min-	lin.fit	fit	fit	fit	fit	sens	ISF	ISF	ISF	IS
3	id	ime	act	orig	orig	utes	avg.	utes	delta	correl	durat	last-Δ	next-Δ	emul	emul	emul	emul	e

Context menu:
- Zellen formatieren...
- Zeilenhöhe...
- Optimale Zeilenhöhe...
- Zeile(n) oberhalb einfügen
- Ausgewählte Zeilen löschen
- Zeileninhalte löschen...
- Ausblenden
- Einblenden
- Ausschneiden
- Kopieren
- Einfügen
- Inhalte einfügen...

	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
	1,78	0	117	30	2,2	0,99	15	7,2	11,2	1	1,78	1,01	1,16
	0,56	5	123,5	35	2,9	0,98	15	0,4	-4,1	1	0,55	1,02	1
	0,7	15	121,5	40	2,7	0,98	20	-2,37	-6,09	1	0,65	1,01	1
	0,66	0	116	70	1,8	0,98	25	-5,89	-9,53	1	0,65	1,01	1
	0,99	5	115,5	15	-3	0,92	20	-3,57	-4,28	1	0,98	1,01	1
	1,26	10	114,7	10	-1,5	0,96	15	-0,8	1,2	1	1,26	1	1
	1,26	15	115,3	25	-1,7	0,94	20	2,77	5,49	1	1,26	1,01	1,08
	1,08	20	115,8	10	2,5	0,88	25	3,39	5,35	1	1	1,01	1,02
	1,1	10	119,3	15	3,1	0,95	15	3,6	4,1	1	1,06	1,02	1,1
	1,65	0	130	10	6	1	15	7,4	10,4	1	1,65	1,02	1,14
	1,81	0	141	10	9,1	1	20	10,7	13,84	1	1,81	1,05	1,22
	1,81	0	141	10	9,1	1	20	10,7	13,84	1	1,81	1,05	1,22
	1,69	0	141	5	11,4	1	20	10,7	13,84	1	1,69	1,05	1,22

	A	B	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
147	18	11:34:21	0	1	1,28	0	149	5	8,1	0,99	20,1	10,03	11,39	1	1,28	1,06	1,16
148	19	11:39:14	0	1	0,4	5	149	25	7,1	1	15	1,05	-4,39	1	0,38	1,06	1
149	20	11:44:15	0	1	0,57	10	148,7	40	5,4	0,99	20	-2,41	-6,96	1	0,5	1,06	1
150	21	11:49:16	0	1	1,11	20	146,6	10	-1,5	1	14,9	-2	-3,01	1	0,88	1,06	1
151	22	11:54:16	0	1	1,1	25	145,7	10	-3,5	0,99	15	-4,6	-6,6	1	0,77	1,05	1
152	23	11:59:14	0	1	0,86	5	138,5	10	-5	1	24,9	-5,53	-7	1	0,83	1,03	1
153	24	12:04:14	0	1	1,22	10	136,7	15	-4,4	1	15	-3,4	-2,4	1	1,22	1,03	1
154	25	12:08:45	0	1	1,21	10	136,7	5	-3	1	15	-3,4	-2,4	1	0,88	0,99	1
155	26	12:10:37	0	1	1,02	15	136	5	1	0,99	15	0,6	3,6	1	1,63	0,99	1
156	27	12:14:14	0	1	0,67	15	133,3	35	-3	0,96	30	-1,64	-1,07	1	0,96	0,98	1

your_title   graphs

Tabelle 1 / 2   4 Zeilen, 1024 Spalten ausgewählt   PageStyle_2023-02-20T20.empty   Summe=0   100 %

Then just copy it in, by selecting all data lines in the emulated csv, and pasting (paste special, values only)

into box A4 of your "elonged" autoISF_factors_performa.ods.

The bottom tab "your_title" should be re-named by you, best with day of log you analyze, and your what-if

parameters (so, the name of your csv file could be put in here)

Now you have a table with optimized lay-out that incorporates key data from both your no change

AND of your investigated changes.csv files.

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).

465
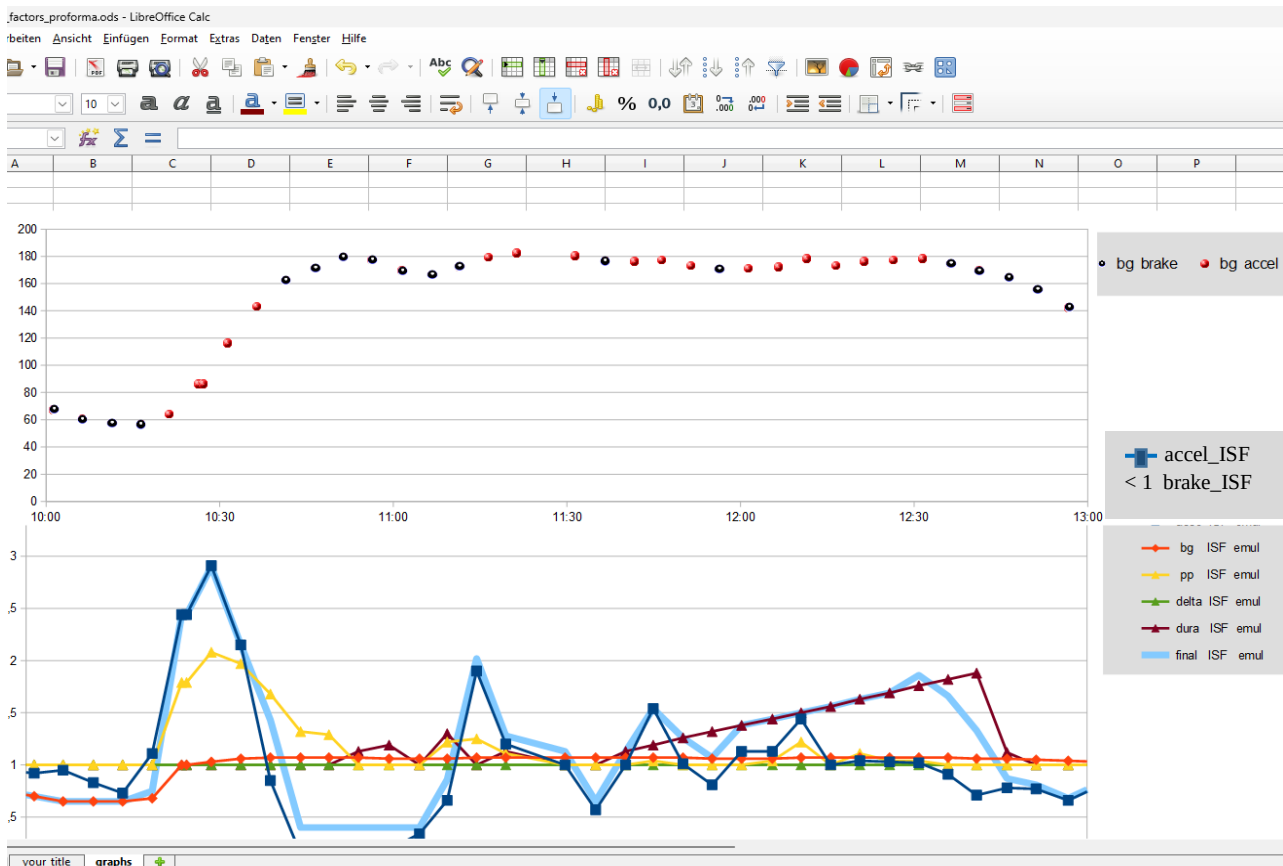
466 The bottom graph (do one for each, the noChange or the (yourChanges) case) shows the amplification factors

467 coming from each autoISF category, and the overall resulting ISF amplification.

468   You probably have to widen the time scale (double click on the time axis, and type the desired time

469   span (min and max UTC)(and spacing of data points, 00:30:00 or 00:15:00) into this box:

470

471 In the given example above, the 2.5 hours were not enough yet to analyze this 10:30 UTC (12:30 AAPS)

472 lunch; we need to look until bg is near target (hopefully before dinner starts).

473

474   A similar graph is available on the (i-)phone if you use the autoISF dev variant of iAPS (and maybe

475   of Trio, in the near future). See also section 11.3)

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 detrailed 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 too aggressive or to 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).

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

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

- 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)

513   •   Dark blue line: iob (exceeding twice the iobTH, with temp. SMB shut-off

514   As bg did not convincingly come down enough, one could hypothesize that iobTH should be

515   elevated. ((But, again, this would have to be confirmed also with other kinds of meals)).

516   •   Thin yellow line: Insulin activity

517   •   Green dotted line: ISF as would result from AAPS w/Autosens

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

519   Foreseeably, this is the strongest difference between our noChange (left) and 120% bgAcel_ISF_weight

520   (right) in the picture below. (Note the red bg curve is *both times* the really seen bg, because the what-if

521   case only looks at each single loop decision). The first (->) time the dark green dot is far to the right, this

522   *would* get the bg down, we *would start to see* a (<-) bg lowering effect, shifting the red curve to the left

523   •   Black line: Profile ISF

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

525   •   Orange "baseline" (only in FCL, as cob=0 at all times)

526
527
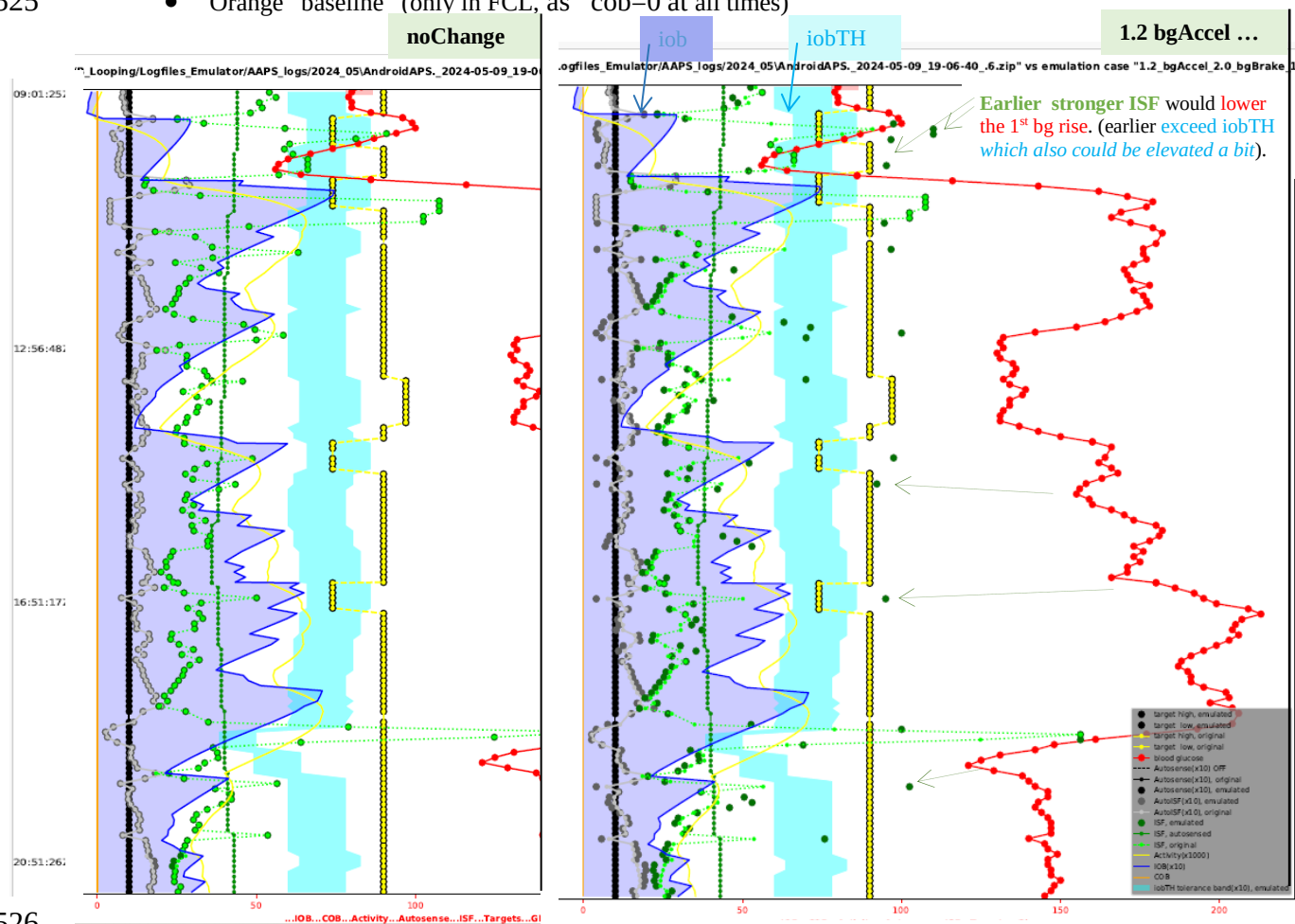
528 Regarding the other changed parameters: Stronger dura_ISF would suggest more insulin towards the end of

529 plateaus; this should have helped in the 1st plateau (red curve, top right quadrant of the picture). However,

530 same setting would have to work also on 2nd plateau; the chart cuts off there, so too early to see whether a

531 hypo danger might result.

532 Effect from doubling the bgBrake_ISF effect are hard to evaluate. Better probably to look in .csv tables, or

533 run a separate emulation for that change only.

534

535 Always check for 2 or 3 kinds of your meals whether the "new" parameter settings really are on average

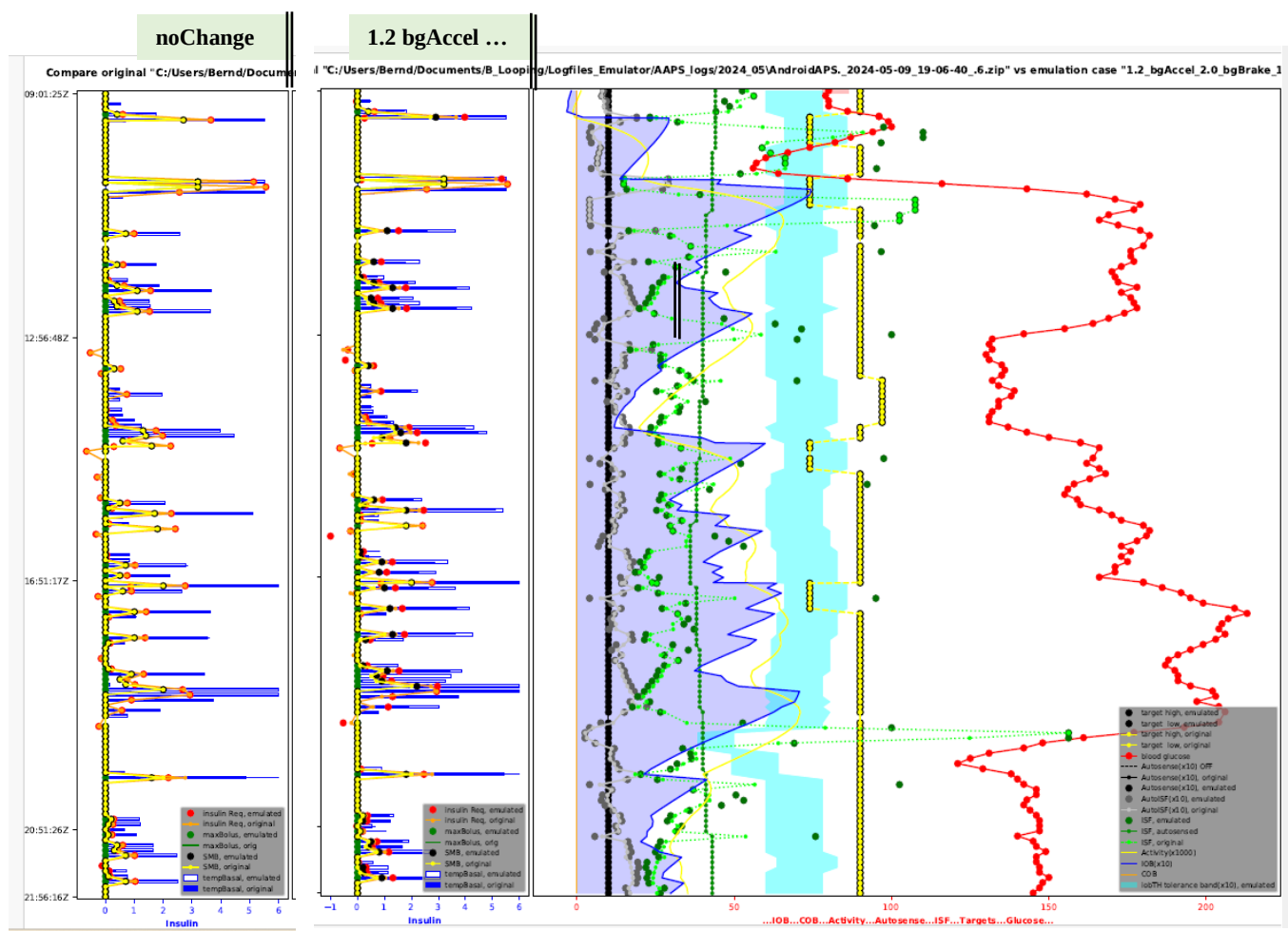536 better. (See negative example in case study 8.2!)

537
538
539 Part of both above shown charts (left side of each, with blue peaks) was cut out.….

540
541 (Unfinished / to be explained later)   (…note: yourChanges = 1.2_bgAccel_2.0_bgBrake_1.2_dura)

542
543

544
545

546

547

548 Please share your experiences with the emulator in Discord / Full-Closed-Looping / HOW

549 TO /_emulate-aaps, at: https://discord.gg/n3tD5eXExC

550

551