

Table of Contents

Anwendungsbereich.....	2
Wie es im Prinzip funktioniert.....	2
Wie man mit minimalem Risiko verbesserte Einstellungen findet.....	2
Anmerkungen und aktuelle Einschränkungen.....	3
Geprüfte Systemeinstellungen.....	3
Installation.....	3
Unter Windows10.....	3
Auf dem AAPS Telefon.....	4
Definieren Sie Ihre Variante in der VDF Datei (Variant Definition File).....	4
Ausführen der Analyse auf einem PC mit Windows.....	6
Beispiel.....	9
Die Ausgabeoption "pred".....	12
Die Ausgabeoption „flowchart“.....	14
Führen Sie die Analyse über das GUI-Frontend aus.....	15
Eingaben auswählen (Select inputs).....	15
Grafikoptionen auswählen (Select Graphics Options).....	16
Ausführen der Analyse (Execute the Analysis).....	17
Ergebnisse inspizieren (Inspect Results).....	18
Ausführen der Analyse auf dem AAPS-Telefon.....	19
Start der Emulation.....	19
Die Ergebnisse.....	22
Beenden Sie die Emulation.....	23
Tipps und tricks.....	23
Anhang A – Wie man Logfiles nutzt ohne autoISF-Inhalt oder vor Version 2.2.6.....	24
Szenario 1: Der Logfile stammt aus regulärem AAPS, d.h. ohne autoISF.....	24
Szenario 2: Der Logfile stammt aus AAPS mit einer autoISF-Version vor 2.2.6.....	24
Szenario 3: Der Logfile stammt aus regulärem AAPS, d.h. ohne autoISF und Sie wollen prüfen wie autoISF die Situation ändern würde.....	24
Anhang P – Definieren eines Pumpenprofils.....	25
Anhang V – Einige weitere Beispiele für VDF Einträge.....	26

Anwendungsbereich

Dieses Dokument richtet sich an AAPS-Benutzer, die untersuchen wollen, welche Auswirkungen bestimmte Einstellungen in AAPS auf die Insulinabgabe haben. Es beschreibt die Installation und Nutzung auf Android und PC. Die letzte Version ist stark auf die Unterstützung von autoISF ausgerichtet, aber die einfachen AAPS-Sachen funktionieren immer noch wie am Anfang. Wenn Ihr AAPS kein autoISF enthält oder Sie ältere Logfiles studieren, die die Schlagworte von autoISF nicht enthalten, können Sie den Emulator trotzdem verwenden, indem Sie einige Befehle am Anfang der VDF-Datei hinzufügen, wie in Anhang A beschrieben.

Andere Dialekte von AAPS wie AIMI, Boost oder DynamicISF werden nicht unterstützt.

Wie es im Prinzip funktioniert

Die Idee ist, die Risiken beim Ändern von AAPS-Einstellungen zu verringern. Ich teste die Auswirkungen solcher Änderungen zunächst offline auf einem PC, indem ich in einem Python-Skript emuliere, was der Loop tun würde. Dieses Python-Skript ist eine Übersetzung der ursprünglichen AAPS SMB "determine-basal.js" Javascript Version.

Der Code scannt eine vorhandene AAPS-Logdatei und extrahiert die Ergebnisse des ursprünglichen Laufs. Auch die Eingaben für "determine-basal" wie letzte Glukose, IOB-Array oder Profildefinition werden extrahiert. Bei diesem Profil handelt es sich nicht um das übliche tägliche Muster von Basalwerten usw., sondern um eine Sammlung von Einstellungen, z. B. welche SMB-Funktion ein- oder ausgeschaltet ist; für Einzelheiten gehen Sie auf die Registerkarte "SMB" in AAPS und scannen Sie den Abschnitt "Profil".

Nach dem Sammeln all dieser Eingaben liest der Emulator auch Ihre Definition von alternativen AAPS-Einstellungen wie

- anderes Temp Target
- andere SMB-Einstellungen
- unterschiedliche basale Sicherheitsfaktoren
- andere ISF (hier "sens" genannt!) oder IC-Werte

Einstellungen, die sich auf bereits eingenommene Mahlzeiten oder bereits verabreichtes Insulin beziehen, werden in anderen Modulen von AAPS behandelt und sind daher hier nicht möglich. Aus dem gleichen Grund können Änderungen an Autosense nicht behandelt werden und die einzige sinnvolle Änderung, die Sie vornehmen können, ist, es auf "1" zu setzen, was bedeutet, es auszuschalten.

Anschließend wird `determine_basal.py` aufgerufen, um die Analyse erneut auszuführen und Ergebnisse im Arbeitsordner zu erzeugen, um die ursprüngliche Situation mit dem emulierten Szenario zu vergleichen.

Wie man mit minimalem Risiko verbesserte Einstellungen findet

Da es sich bei diesem Tool um eine virtuelle Analyse handelt, können Sie viele Veränderungen zunächst "in silico" nachahmen, ohne ein Risiko für Ihre Gesundheit einzugehen, sollten sich diese Veränderungen als ungünstig erweisen. Die Schritte, die Sie mit Hilfe dieses Tools durchführen können, lauten wie folgt:

1. Identifizieren Sie eine Situation, die Sie verbessern möchten, und besorgen Sie sich die entsprechenden Logfiles
2. Lesen Sie den entsprechenden Abschnitt in den Dokumenten
3. Listen Sie die Einstellungen auf, die einen Einfluss haben könnten
4. Verwenden Sie diesen Emulator auf dem PC, um zu testen, welche dieser Einstellungen die gewünschte Wirkung haben

5. Passen Sie die Werte der bevorzugten Einstellung an, um die gewünschte Verbesserung zu erzielen
6. Starten Sie den Emulator erneut auf dem AAPS-Telefon, damit Sie das Ergebnis quasi in Echtzeit sehen und die Einstellung feinjustieren können. Verwenden Sie dabei die Variante mit der geringeren Insulinabgabe als Master in AAPS und die andere als emulierte Version. Auf diese Weise können Sie jedes Mal, wenn eine inkrementelle Insulindosis vorgeschlagen wird, entscheiden, ob Sie diese wirklich manuell anwenden wollen, wie Sie es in Open Loop beim Erlernen von AAPS getan haben.
7. Sobald alles in Ordnung und sicher aussieht, implementieren Sie die Einstellung in AAPS selbst.

Die Schritte 4-6 sind ohne dieses Tool nicht möglich und bieten zusätzliche Sicherheit beim Ändern von Einstellungen.

Anmerkungen und aktuelle Einschränkungen

Es handelt sich nicht um einen vollständigen virtuellen Loop! Bitte beachten Sie, dass die Variante bei jedem Zeitschritt eine neue und unabhängige Entscheidung darstellt. Es handelt sich nicht um eine Abfolge von aufeinander aufbauenden Entscheidungen, da nicht vorhergesagt werden kann, wozu eine andere Entscheidung im nächsten Zeitschritt geführt hätte. Daher muss man die Ergebnisse bei jedem Zeitschritt einzeln betrachten und abwägen, ob sie sicher genug sind, um in AAPS aktiviert zu werden. Sobald dieser Emulator mir vielversprechende Hinweise gibt und der Parameter abgestimmt ist, gehe ich quasi in den "offenen Regelkreis" über, d.h. ich gebe einige Tage lang manuell eine Erhöhung des SMB als Korrekturbolus, bevor ich in den Echtbetrieb gehe.

Beachten Sie, dass `maxBolus=0` in der tabellarischen Ausgabedatei bedeutet, dass `maxBolus` hoch genug war, um die SMB nicht zu begrenzen, und daher in der ursprünglichen Logdatei von AAPS nicht explizit aufgeführt wurde.

Es wird angenommen, dass die Glukosewerte in mg/dl angegeben werden.

Der Code funktioniert nur für `oref1` im SMB-Modus, auch wenn SMB nicht aktiviert ist.

Geprüfte Systemeinstellungen

- Windows10, 64 bit
- Python 3.7, 3.8, 3.10
- einige Python-Libs wie matplotlib (siehe die Import-Anweisungen in `vary_settings_core.py`)
- AAPS-Versionen für logfile: `oref1 & SMB` für 2.3 - 3.0.0.1
- AAPS-Versionen für `determine_basal`: `oref1 & SMB` für 2.5 - 3.0.0.1

Installation

Laden Sie das Github-Repository (<https://github.com/ga-zelle/APS-what-if>) als Zip-Archiv herunter.

Unter Windows10

Entpacken Sie das Archiv in einen Ordner mit dem Inhalt

- "`determine_basal.py`" als Emulation der Original-Java-Version
- "`vary_settings_GUI.py`" zur Verwaltung der Benutzereingaben in einem interaktiven Fenster
- Erstellen Sie eine Verknüpfung auf Ihrem Desktop, die auf diese GUI-Datei verweist.
- "`vary_settings_batch.py`", die alternativ die Benutzereingaben in einem DOS-Fenster verwaltet
- "`vary_settings_core.py`" das Hauptprogramm, das den gesamten Prozess steuert
- "`<variant_label>.vdf`", das die Definition der Einstellungen enthält, die geändert werden sollen

Um Python zu installieren, wenn Sie es noch nicht auf Ihrem PC haben, laden Sie am besten das Installationsprogramm von <https://realpython.com/installing-python/#how-to-install-python-on-windows> herunter. Wenn das Installationsprogramm startet, wählen Sie die benutzerspezifischen Optionen und nicht die Standardkonfiguration. In dem folgenden Dialog wählen Sie

- python zur Systemvariablen für PATH hinzufügen
- Wählen Sie, dass auch "pip" heruntergeladen und installiert werden soll.

Nachdem die Python-Installation abgeschlossen ist, müssen Sie die matplotlib-Bibliothek herunterladen. Dies geschieht in einem DOS-Fenster durch Eingabe von "pip install matplotlib" in der Befehlszeile.

Die letzte Maßnahme, um Python für dieses Projekt fit zu machen, ist die Definition einer Systemvariablen PYTHONUTF8=1, um die Verwendung von Sonderzeichen aus dem UTF-8-Zeichensatz wie dem griechischen Δ zu ermöglichen.

Auf dem AAPS Telefon

- Laden Sie qpython3 ("QPython 3L - Python für Android" von QpythonLab) aus dem Play Store herunter und installieren Sie es.
- Dadurch sollte ein Ordner "qpython" auf der obersten Ebene erstellt werden
- Gehen Sie in dessen Unterordner "scripts3".
- Extrahieren Sie aus dem oben heruntergeladenen Github-Archiv die folgenden Skriptdateien (d. h. die Python-Programme) in diesen Unterordner
 - "determine_basal.py"
 - "vary_settings_batch.py"
 - "vary_settings_core.py"
- "<variant_label>.vdf", die VDF (siehe unten), ist dieselbe wie unter Windows und wird in denselben Ordner kopiert, der auch die AAPS-Logdateien enthält.

Definieren Sie Ihre Variante in der VDF Datei (Variant Definition File)

Erstellen oder bearbeiten Sie "<variant_label>.vdf", Ihren VDF. Die frühere Namenskonvention "<variant_label>.dat" wird zwar noch unterstützt, aber nicht mehr empfohlen. Sie definiert die Änderungen für Ihr Was-wäre-wenn-Szenario. Sie haben Zugang zu den entsprechenden Listen, wie in der Registerkarte "SMB", Abschnitte "Glucose-data" bis "Autosense-data" gezeigt. Das allgemeine Format pro Zeile ist:

```
.  
.   
<array> <item in array> <neuer Wert oder Formul>  [<optional: ### beliebiger Kommentar>]  
.   
.
```

Die Einträge können durch mehrere BLANKs oder TABs getrennt und zur besseren Lesbarkeit in Spalten angeordnet werden. Ihre Bedeutung ist wie folgt:

- <Array>
Name des Python-Dictionaries; die verfügbaren Namen sind
 - autosens_data ändert sich nicht; wird innerhalb von AAPS anderweitig berechnet
 - glucose_status nicht wirklich aussagekräftig
 - currenttemp nicht wirklich aussagekräftig
 - iob_dat ändern sich nicht; wird innerhalb von AAPS anderweitig berechnet
 - meal_data ändern sich nicht; wird innerhalb von AAPS anderweitig berechnet
 - profil die meisten Variationen werden hier stattfinden
 - new_parameter zur Aufnahme emulatorspezifischer Informationen wie AAPS-Version

Wenn array kein anerkannter Name ist, wird die Zeile übersprungen.

Hier sind keine BLANKS erlaubt.

- <Element in Array>

Element in <array>; kann neu sein oder ein bestehendes Element in diesem Array umdefinieren

Hier sind keine BLANKS erlaubt.

- <Neuer Wert oder Formel>

Boolescher, numerischer oder String-Wert oder Ausdruck, der entsprechend ausgewertet wird

Darf BLANKS enthalten, aber nicht '###'

optional auch:

- <### beliebiger Kommentar> ein optionaler Kommentar nach einem Block von '###'

Das Programm überprüft die Syntax der eingegebenen Formeln, da nicht jeder mit der hier verwendeten Python-Syntax vertraut ist. Wenn Sie zum Beispiel die Anführungszeichen für max_bg vergessen, wie hier:

```
new_parameter LessSMBbelow      profile[max_bg] + 10    ### ... bg below this value
```

erhalten Sie eine Warnmeldung wie unten und das Programm wird beendet:

```
*****
Problem in VDF-file in row reading
new_parameter LessSMBbelow      profile[max_bg] + 10    ### ... bg below this value

error message is:<class 'NameError'>
*****
```

Den detaillierten möglichen Inhalt der Arrays und ihre korrekte Schreibweise finden Sie im originalden Logfile. Suchen Sie nach Zeilen, die "[DetermineBasalAdapterSMBJS.invoke():" enthalten, oder überprüfen Sie die entsprechenden Abschnitte auf der AAPS-Registerkarte "SMB".

WARNUNG:

Während AAPS auf Einschränkungen prüft und einige SMB-Einstellungen deaktiviert, weiß der Emulator nichts davon und macht einfach weiter. Es liegt daher in Ihrer Verantwortung zu prüfen, ob AAPS die Einstellungen aus Ihrem VDF akzeptiert.

Beispiel: Dieses Repo enthält die Variantendefinitionsdatei "Demo_Sports_Adaptations.vdf". Sie wird für ein Logfile verwendet, das 2-3 Stunden Badmintonspiel umfasst. Ich möchte unter 140mg/dl bleiben, weil ich sonst die Konzentration verliere und zu langsam reagiere. In diesem Fall hat es ganz gut funktioniert, aber was ist, wenn ich die Dinge ein wenig optimiere?

```
profile    enable_autoISF          False          ### for logs before autoISF
profile    smb_delivery_ratio      0.5           ### AAPS default

profile    min_bg                  101             ### just above the temp target threshold for SMB
profile    max_bg                  profile['min_bg']    ### redefine target
#rofile    target_bg               int((profile['min_bg'] + profile['max_bg'])/2)    ### example of numerical expression
profile    tempTargetset           True            ### spelling error
profile    temptargetSet            True            ### i.e. not using NS profile target
profile    allowSMB_with_high_temptarget False        ### was True at the time
```

1. Die ersten beiden Zeilen sind nur für ältere Protokolldateien erforderlich, die erstellt wurden, bevor autoISF in AAPS aufgenommen wurde.
2. In der 3. Zeile wird das untere Ziel von 100 auf 101 geändert, um es knapp außerhalb des Bereichs "enableSMB_with_temptarget" zu verschieben.
3. Die 4. Zeile setzt den oberen Zielwert mit einem einfachen Ausdruck auf denselben Wert wie den unteren. Er muss gleich sein, weil ich die Einstellung eines temporären Ziels simulieren möchte, das keine endliche Bandbreite zulässt, wenn es in der AAPS-GUI definiert ist.
4. Die 5. Zeile wird ignoriert, weil "#rofile" kein anerkannter Array-Name ist. Aber sehen Sie sich nur die Formel an und wie komplex sie sein kann. Was die Ziele betrifft, so ist "target_bg" ohnehin als Durchschnitt von "min_bg" und "max_bg" in determine_basal definiert, so dass

diese Zeile nicht benötigt wird.

5. In der 6. Zeile wird versucht, das Flag für "tempTargetset" auf True umzudefinieren. Diese Schreibweise ist jedoch nicht ganz korrekt, da es 2 Vertauschungen in der Groß- und Kleinschreibung gibt.
6. In der 7. Zeile wird das Flag für "temptargetSet" auf True gesetzt, und zwar in der richtigen Schreibweise. Es gibt keine Prüfung, ob die Zielbandbreite in diesem Fall Null ist.
7. Schließlich deaktiviert die 8. Zeile die SMB für höhere TempTargets, d.h. über 100.

Zusammenfassend lässt sich sagen, dass sich das Ziel von 100 auf 101 erhöht, was nun mit der allgemeinen Empfehlung für Sport durch Deaktivierung von SMB übereinstimmt.

Für die erstmalige Verwendung empfehle ich, alles unverändert zu lassen und zu überprüfen, ob die Ergebnisse mit dem Original übereinstimmen. Geringfügige numerische Abweichungen sind wahrscheinlich auf Rundungen in der ursprünglichen Logdatei zurückzuführen.

Es gibt sechs Spezialfälle von Quasi-Arrays, die eingeführt wurden, um zwischenzeitliche oder zeitlich variierende Zuweisungen zu behandeln:

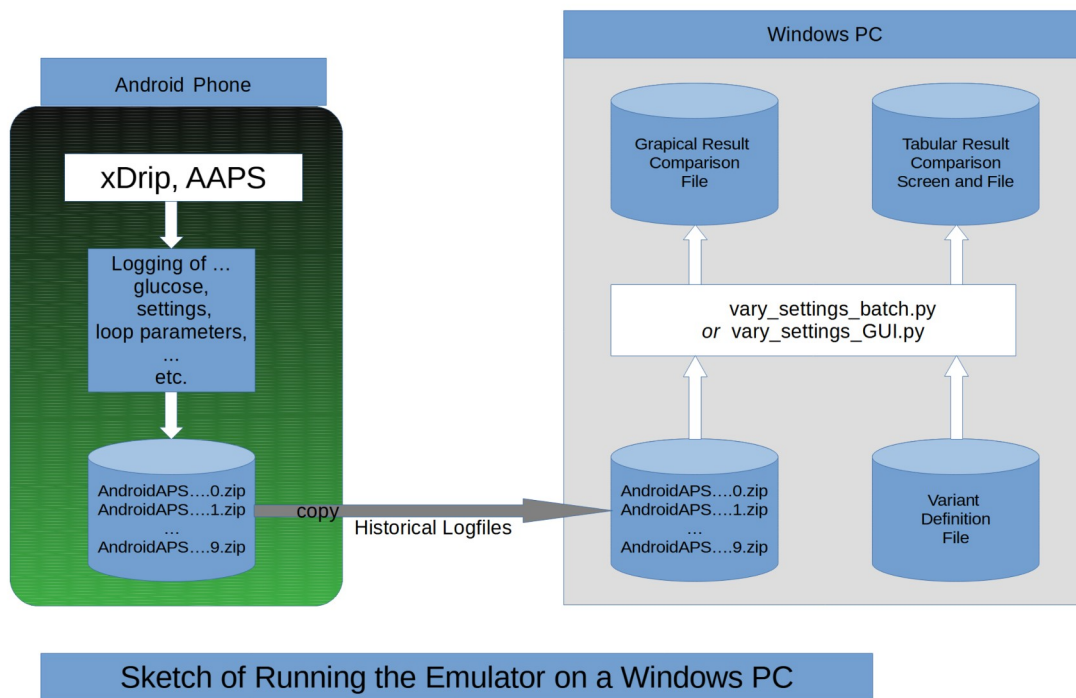
- temp <Var_name> <Wert oder Ausdruck>
Sie können Ihre eigenen Namen für Variablen verwenden, um Zwischenwerte zu speichern, auf die in den folgenden Anweisungen verwiesen werden kann
- STAIR <UTC-Datum/Zeit> <Wert ab dieser Zeit>
Dies ist nützlich für zeitabhängige Werte wie Umschalten von Zielen
- STAIR_BAS definiert Pumpenprofile; siehe Anhang STAIR ...
- STAIR_CR ... für Details ...
- STAIR_ISF ... und Beispiel
- INTERPOL <UTC-Datum/Zeit 1> <Wert zu dieser Zeit>
INTERPOL <UTC-Datum/Uhrzeit 2> <Wert zu diesem Zeitpunkt>
Dies verwendet lineare Interpolation und Extrapolation zur Definition einer zeitabhängigen Funktion

Ein Beispiel zeigt, wie die Funktionen definiert und referenziert werden können:

STAIR	2020-04-14T00:00:00Z	111	
STAIR	2020-04-14T12:50:00Z	125	
STAIR	2020-04-14T13:10:00Z	131	
INTERPOL	2020-04-14T13	100	
INTERPOL	2020-04-14T13:30:00Z	140	
profile	min_bg	round(INTERPOL) - 20	### time varying value
profile	max_bg	STAIR - 10	### time varying value

Ausführen der Analyse auf einem PC mit Windows

Für diejenigen, die weniger fit in DOS-Befehlen sind, habe ich ein GUI-Frontend erstellt, das die Eingaben in einem Formular sammelt (siehe Anhang). Hier ist eine Skizze der Daten- und Prozessflüsse bei der Verwendung des Emulators auf einem PC mit Windows:



Um zu beginnen, öffnen Sie ein DOS-Befehlsfenster, navigieren zum Arbeitsordner, der "<variant_label>.vdf" enthält, und geben ein

DOS>[python] vary_settings_batch.py <Protokolldatei(en)> <Optionen> <VDF> [<von> [<bis>]]
 Der anfängliche "python" kann weggelassen werden, wenn die DOS-Pfaddefinition das Verzeichnis für das Programm Python enthält.

Die Bedeutung der Argumente ist wie folgt:

- <Protokolldatei(en)>
 die von Android heruntergeladene AAPS-Logdatei (noch gezippt oder ungezippt). Optional können Sie die Platzhalter (? für die Suche nach einem einzelnen Zeichen; * für die Suche nach einer beliebigen Zeichenfolge) verwenden, um mehrere Logdateien im Verzeichnis zu finden, was ein längeres Zeitfenster für die Auswertung bietet. Ein Beispiel könnte "AndroidAPS._2020-04-02_*" sein, um alle Logdateien dieses Tages zu verarbeiten.
- <Optionen>
 "All" oder eine durch Schrägstrich getrennte Liste von Dingen, die angezeigt werden sollen, damit die Darstellung nicht überladen wird. Die anzuzeigenden Dinge sind Elemente aus dieser Liste:
 - insReq ist das gesamte im aktuellen Zeitschritt benötigte Insulin
 - maxBolus ist die Obergrenze des erlaubten SMB
 - SMB ist das tatsächlich zu verabreichende SMB
 - basal ist das tatsächlich abzugebende Basalinsulin
 - target sind das untere und obere Ziel
 - bg ist der ursprüngliche Glukosewert
 - cob ist der ursprüngliche COB-Wert zum besseren Verständnis der Situation
 - iob ist der ursprüngliche IOB-Wert zum besseren Verständnis der Situation
 - activity ist die ursprüngliche Insulinaktivität zum besseren Verständnis der Situation
 - as_ratio ist das Autosense-Verhältnis
 - autoISF¹ ist der endgültige ISF-Faktor
 - ISF-Faktoren¹ sind die einzelnen ISF-Faktoren (nur tabellarische Ausgaben, kein Plot)
 - ISF ist der Insulinempfindlichkeitsfaktor
 - fitsParabola¹ Seria vongefitteten Parabeln durch die letzten BG-Werte mit guter Korrelation und Darstellung in grau als gepunktete Kurve einschließlich 5-Minuten-Vorhersage

¹ Für Details und weitere Parameter siehe <https://github.com/ga-zelle/AndroidAPS/tree/autoISF>

- `bestParabola`¹ zeigt aus all diesen Fits den besten als schwarze Kurve
- `range`² Parameter der nahezu unveränderten BG, aus der `dura_ISF` bestimmt wird
- `slope`² Parameter der nahezu linearen Änderung der BG und Plotten dieser Linie
- `fitsslope`² Serie gefitteter Geraden durch die letzten BG-Werte mit guter Korrelation und Darstellung als grau gestrichelte Linie
- `bestslope`² zeigt aus all diesen Fits den besten als schwarze Linie
- `pred` stellt die Vorhersagekurven pro Zeitschritt des emulierten Szenarios dar mit Hinweisen, ob SMB deaktiviert werden musste; man erhält quasi eine Animation, indem man die Vorwärts-/Rückwärts-Tasten in der pdf-Datei benutzt, mit einer Seite pro Loop
- `flowchart` nur in der Grafikausgabe, wird gezeigt, wie `determine-basal` mit allen ausgeführten if-Verzweigungen durchlaufen wurde.

Mit dem speziellen Flag "-" vor jeder Option können Sie einzelne `<plot-options>` unterdrücken. Ich verwende normalerweise "All/-pred/-flowchart", was alles außer "pred" und "flowchart" ergibt, um das Programm bei Routineanalysen deutlich zu beschleunigen.

- `<variant_label>` oder alternativ:
- `<variant_label>.vdf` (oder aus Gründen der Abwärtskompatibilität `<variant_label>.dat`)
Eine Bezeichnung für die Namen der Ergebnisdateien und der Plots. Es ist auch der Basisname der VDF "`<variant_label>.vdf`".
Die alternative Methode, den Dateinamen direkt anzugeben, kann einfacher sein, insbesondere bei Verwendung der DOS-Type-Ahead-Methode

Mit den optionalen Parametern kann ein auszuwertendes Zeitfenster definiert werden, sei es für eine einzelne Logdatei oder eine Liste von Logdateien. Bitte beachten Sie, dass die Datums-/Zeitkombination im UTC-Format angegeben wird, d.h. in Greenwich-Zeit ohne Sommerzeit-Effekt:

- `<von>`
Definieren Sie den Beginn des Zeitfensters wie "2020-04-02T22:00:00Z" oder kurz "2020-04-02T22", was Mitternacht des 2. April ist. Wenn Sie die Startzeit weglassen wollen, weil Sie eigentlich nur die Endzeit brauchen, verwenden Sie einfach "2000" als sehr frühes Jahr.
- `<bis>`
Optional können Sie auch das Ende des Zeitfensters angeben, wiederum im lesbaren UTC-Format, das im Logfile-System verwendet wird.

Der Prozess erstellt die folgenden Dateien:

- `<Logdatei>.<Variante>.log`
Echo der verwendeten Skriptversionen und der oben aufgeführten Befehlszeilenargumente
Echo der Parameteränderungen, wie von `<variant>.vdf` angefordert
- `<Protokolldatei>.orig.txt`
Auszug aus der ursprünglichen Logdatei mit den für das Loopergebnis relevanten Einträgen; für dieselbe Logdatei wird dies immer derselbe Name sein und unabhängig von der Variante; daher ist es unschädlich, wenn eine andere Variante emuliert wird und diese überschrieben wird.
- `<Protokolldatei>.<Variante>.txt`
äquivalente, emulierte Teilmenge für das Ergebnis des Variantenloops
- `<Protokolldatei>.<Variante>.csv`
Vergleich aller Schlüsselwerte zwischen dem ursprünglichen und dem emulierten Lauf; in den unteren Zeilen werden die innerhalb des Zeitfensters gefundenen Minimal- und Maximalwerte sowie die gesamte Basal- und die gesamte SMB-Dosis angezeigt;
bereit für den Import in Ihr bevorzugtes Tabellenkalkulationsprogramm; setzen Sie das Format der Importdatei auf UTF-8 und die zweite Spalte auf das Zeitformat "hh:mm:ss"
- `<Protokolldatei>.<Variante>.pdf`
Plot der tabellarischen Ergebnisse; wenn die vorherige Datei mit diesem Namen noch im pdf-Viewer geöffnet ist, wird eine Warnung ausgegeben, bis die Fortsetzung möglich ist.
- `<logdatei>.<Variante>.delta`

2 Noch experimentell um fortgeschrittene Anpassungen zu testen wie ISF abhängig von der Steigung

Experimentelle Tabelle mit verschiedenen Quellen für Glukose-Delta

Eine Tabelle mit der in der Optionsliste ausgewählten Teilmenge von Schlüsselwerten wird ebenfalls direkt auf dem Bildschirm ausgegeben. Die Grafik wird in einer vordefinierten Auflösung (200dpi) und Größe (9 x bis zu 12 Zoll, Hochformat) gespeichert. Wenn der Emulator mit dieser DOS-Methode und ohne die "pred"-Option gestartet wurde, wird die Grafikausgabe auch interaktiv angezeigt und Dinge wie Seitenverhältnis oder Ränder können geändert werden. Nach solchen Änderungen kann die Grafik manuell durch Anklicken des Diskettensymbols gespeichert werden, vorzugsweise im jpg- oder png-Format, was die spätere Einbindung in andere Dokumente erleichtert. Mit der "pred"-Option können Sie es in einem pdf-Viewer betrachten und die Historie des Logfiles mit den Vorwärts- und Rückwärtstasten durchlaufen, eine Seite pro Loop.

Ein Dateivergleich zwischen den beiden txt-Dateien ist möglich, aber vor allem die Formatierung ist noch unterschiedlich und kennzeichnet zu viele Zeilen, obwohl sie numerisch gleich sind. Ich persönlich finde die Tabelle und den Plot nützlicher.

Beispiel

Wie sehen die Ergebnisdateien für das obige Beispiel aus? Lassen Sie uns zunächst „AndroidAPS. 2019-11-13_00-00-00_7.Demo Sports Adaptations.log“ prüfen:

```
=====
Echo of software versions used
-----
vary_settings home directory  S:\AAPS3p0_dev\
dated: 2022-05-06 01:59:23,    module name: vary_settings_GUI.py
dated: 2022-04-27 02:20:08,    module name: vary_settings_core.py
dated: 2022-04-24 17:50:29,    module name: determine_basal.py
-----
Echo of execution parameters used
-----
Logfile(s) to scan           L:/PID/AndroidAPS. 2019-11-13_00-00-00_7
Output options                All/-pred/-flowchart
Decimal symbol                .
Start of time window          2000-01-01T00:00:00Z (default)
End of time window            2099-12-31T23:59:59Z (default)
=====

===== Echo of what-if definitions actioned for variant Demo_Sports_Adaptations
===== created on Fri, 06 May 2022 02:14:05 +0200
===== for loop events found in logfile L:/PID/AndroidAPS. 2019-11-13_00-00-00_7

loop execution in row=2050 of logfile AndroidAPS. 2019-11-13_00-00-00_7 at= 2019-11-13T18:23:30Z
edited old value of 100 in profile with min_bg=101
edited old value of 100 in profile with max_bg=101
not actioned: [#rofile], [target_bg], [int((profile['min_bg'] + profile['max_bg'])/2)]
appended new entry to profile with tempTargetset=True
edited old value of True in profile with temptargetSet=True
edited old value of True in profile with allowSMB_with_high_temptarget=False

loop execution in row=5288 of logfile AndroidAPS. 2019-11-13_00-00-00_7 at= 2019-11-13T18:28:30Z
edited old value of 100 in profile with min_bg=101
edited old value of 100 in profile with max_bg=101
not actioned: [#rofile], [target_bg], [int((profile['min_bg'] + profile['max_bg'])/2)]
appended new entry to profile with tempTargetset=True
edited old value of True in profile with temptargetSet=True
edited old value of True in profile with allowSMB_with_high_temptarget=False
...
```

Die ersten Zeilen zeigen, welche Softwaremodule tatsächlich verwendet wurden. In der Vergangenheit gab es Diskrepanzen zwischen dem, was das Werkzeug tat, und dem, was es hätte tun sollen. In den nächsten Zeilen werden die Ausführungsparameter aufgeführt und ob es sich dabei um Standardeinstellungen handelt.

Nun beginnt der Bulk-Teil mit einem Echo der Änderungen, die bei jedem Zeitschritt zugewiesen wurden. Es ist sinnvoll, die Ergebnisse komplexer Ausdrücke auf Werte zu prüfen und sicherzustellen,

dass keine Rechtschreibfehler wie in den Zeilen "appended ..." aufgetreten sind.

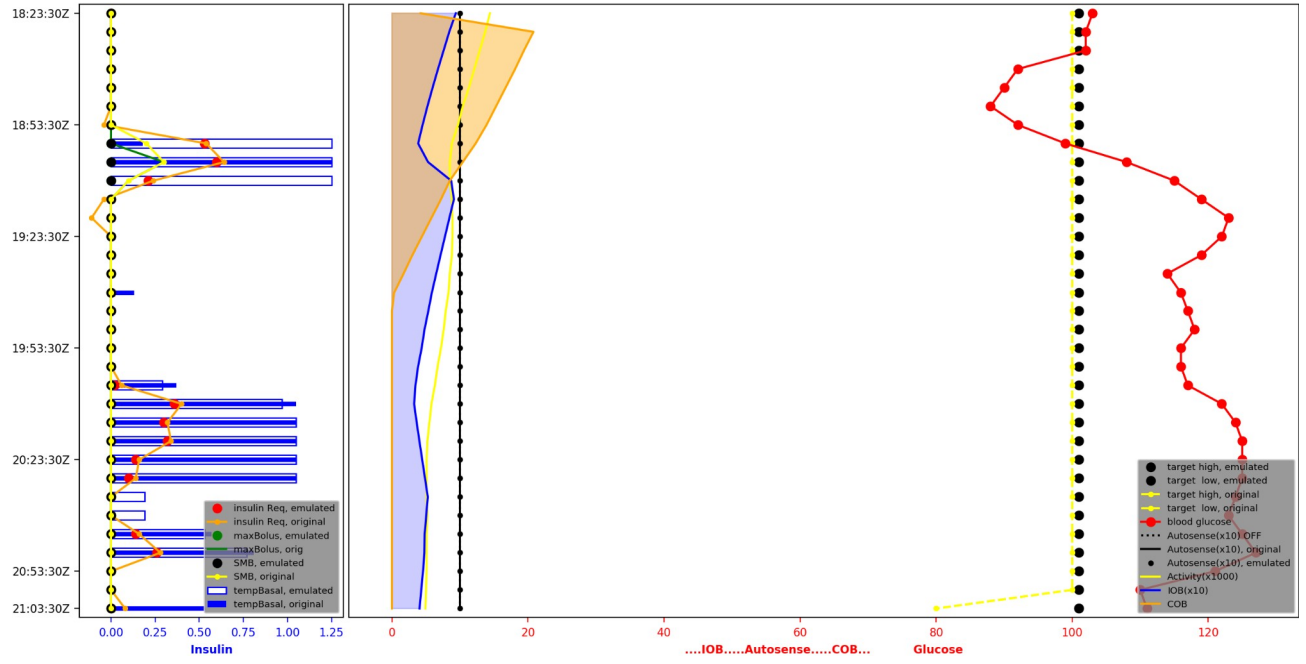
Prüfen Sie als nächstes "AndroidAPS._2019-11-13_00-00-00_.7.Demo_Sports_Adaptations.tab" (bitte beachten Sie, dass das neue Executable stattdessen eine CSV-Datei erstellt und wesentlich mehr Spalten zum Ausdrucken enthält als hier dargestellt):

Printed Event Log: Insulin Bolus and Basal Rate History																				
id	time	--UNIXtime--	bg	cob	iob	act	-Autosens-		-----target-----		insulin Req		-maxBolus-		---SMB---		---tmpBasal---			
							orig	emul	orig	emul	orig	emul	orig	emul	orig	emul	orig	emul		
0	18:23:30Z	1573669410.3	103	4.2	0.94	0.014	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
1	18:28:30Z	1573669710.2	102	20.8	0.84	0.014	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
2	18:33:30Z	1573670010.3	102	19.4	0.76	0.013	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
3	18:38:30Z	1573670310.3	92	18.1	0.68	0.012	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
4	18:43:30Z	1573670610.3	90	16.7	0.6	0.011	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
5	18:48:30Z	1573670910.3	88	15.3	0.52	0.01	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
6	18:53:30Z	1573671210.3	92	13.9	0.45	0.01	1.0	1.0	100-100	101-101	-0.04	0	0	0	0	0	0	0		
7	18:58:30Z	1573671510.3	99	12.3	0.38	0.009	1.0	1.0	100-100	101-101	0.54	0.53	0	0	0.2	0	0.18	1.25		
8	19:03:30Z	1573671810.6	108	10.4	0.53	0.009	1.0	1.0	100-100	101-101	0.64	0.6	0.3	0	0.3	0	1.25	1.25		
9	19:08:30Z	1573672110.3	115	8.5	0.87	0.009	1.0	1.0	100-100	101-101	0.24	0.21	0.1	0	0.1	0	0	1.25		
10	19:13:30Z	1573672410.3	119	7.2	0.91	0.009	1.0	1.0	100-100	101-101	-0.04	0	0	0	0	0	0	0		
11	19:18:30Z	1573672710.3	123	5.8	0.84	0.009	1.0	1.0	100-100	101-101	-0.11	0	0	0	0	0	0	0		
12	19:23:30Z	1573673010.3	122	4.4	0.78	0.009	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
13	19:28:30Z	1573673310.3	119	3.0	0.71	0.009	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
14	19:33:30Z	1573673610.2	114	1.7	0.65	0.009	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
15	19:38:40Z	1573673920.6	116	0.3	0.58	0.008	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0.13	0		
16	19:43:30Z	1573674210.4	117	0	0.53	0.008	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0.03	0.01		
17	19:48:30Z	1573674510.4	118	0	0.48	0.008	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
18	19:53:30Z	1573674810.3	116	0	0.44	0.007	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
19	19:58:30Z	1573675110.4	116	0	0.38	0.007	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
20	20:03:30Z	1573675410.6	117	0	0.35	0.006	1.0	1.0	100-100	101-101	0.06	0.02	0	0	0	0	0.37	0.29		
21	20:08:30Z	1573675710.4	122	0	0.33	0.006	1.0	1.0	100-100	101-101	0.4	0.36	0	0	0	0	1.05	0.97		
22	20:13:30Z	1573676010.3	124	0	0.36	0.006	1.0	1.0	100-100	101-101	0.32	0.3	0	0	0	0	1.05	1.05		
23	20:18:30Z	1573676310.4	125	0	0.4	0.005	1.0	1.0	100-100	101-101	0.34	0.32	0	0	0	0	1.05	1.05		
24	20:23:30Z	1573676610.3	125	0	0.44	0.005	1.0	1.0	100-100	101-101	0.16	0.14	0	0	0	0	1.05	1.05		
25	20:28:30Z	1573676910.8	125	0	0.48	0.005	1.0	1.0	100-100	101-101	0.14	0.1	0	0	0	0	1.05	1.05		
26	20:33:30Z	1573677210.3	124	0	0.53	0.005	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0.19		
27	20:38:30Z	1573677510.4	123	0	0.5	0.005	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0.19		
28	20:43:30Z	1573677810.3	125	0	0.48	0.005	1.0	1.0	100-100	101-101	0.16	0.14	0	0	0	0	0.57	0.53		
29	20:48:30Z	1573678110.7	127	0	0.48	0.005	1.0	1.0	100-100	101-101	0.28	0.26	0	0	0	0	0.81	0.77		
30	20:53:30Z	1573678410.3	121	0	0.46	0.005	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
31	20:58:30Z	1573678710.3	110	0	0.43	0.005	1.0	1.0	100-100	101-101	0	0	0	0	0	0	0	0		
32	21:03:30Z	1573679010.7	111	0	0.41	0.005	1.0	1.0	80- 80	101-101	0.08	0	0	0	0	0	0.53	0		
Totals:																	0.6	0.0	0.76	0.91

In dieser Tabelle sind die wichtigsten Vergleiche zwischen dem ursprünglichen Fall und dem emulierten Fall aufgeführt. In den Zielspalten zeigt die letzte Zeile, dass das ursprüngliche Ziel von 100 auf 80 reduziert wurde, weil der Sport vorbei war. Da die Variante mit 101 fortgesetzt wurde, wird dieser letzte Zeitschritt bei der endgültigen Beurteilung einfach ignoriert. Der Insulinbedarf ist erwartungsgemäß etwas niedriger, da das Ziel etwas höher liegt. Die SMB-Spalten bestätigen, dass alle ursprünglichen SMBs deaktiviert wurden. Das tempBasal muss nun den Mangel an Insulin in diesen Zeitschritten ausgleichen und ist ansonsten aufgrund des geringeren Bedarfs leicht reduziert. Die Summenzeile am unteren Rand gibt einen Hinweis auf die Veränderungen, aber der Zeitpunkt der größten Unterschiede ist wichtig. Während die SMBs von 0,6 Einheiten abgegeben wurden, summierten sich die Basalwerte in diesen 15 Minuten auf 0,12 Einheiten, was einen Gesamtbetrag von 0,72 Einheiten ergibt. Im Fall der Variante summiert sich das gesamte Basalinsulin in diesem kurzen Zeitraum auf 0,31 IE. Zusammenfassend lässt sich sagen, dass dieser Unterschied zu gering ist, um eine Änderung der Einstellungen zu rechtfertigen, die an diesem Tag gut funktioniert haben, und dass die nächste Woche ohnehin anders aussehen wird, weil sich der Körper in einem anderen Zustand befinden wird. Der Vorgang könnte jedoch für die Logdateien der entsprechenden Aktivitäten an anderen Tagen wiederholt werden.

Die grafische Darstellung des Vergleichs sagt dasselbe aus. Die beste Auswertungsmethode besteht darin, bei der Beurteilung des Vergleichs sowohl die Grafik als auch die Tabelle zu betrachten. Das Diagramm für das Beispiel sieht wie folgt aus:

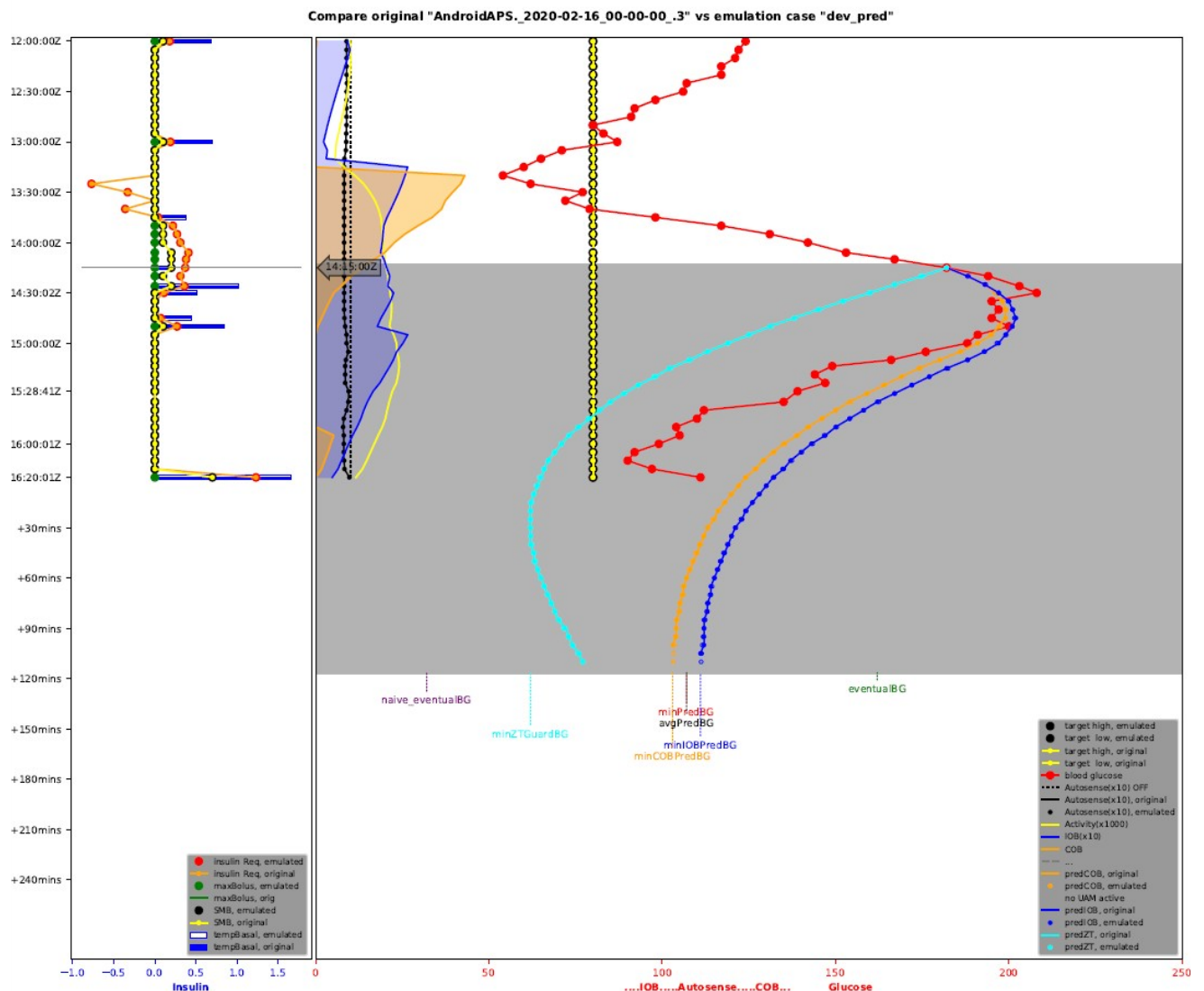
Compare original "AndroidAPS_2019-11-13_00-00-00_7" vs emulation case "Demo_Sports_Adaptations"



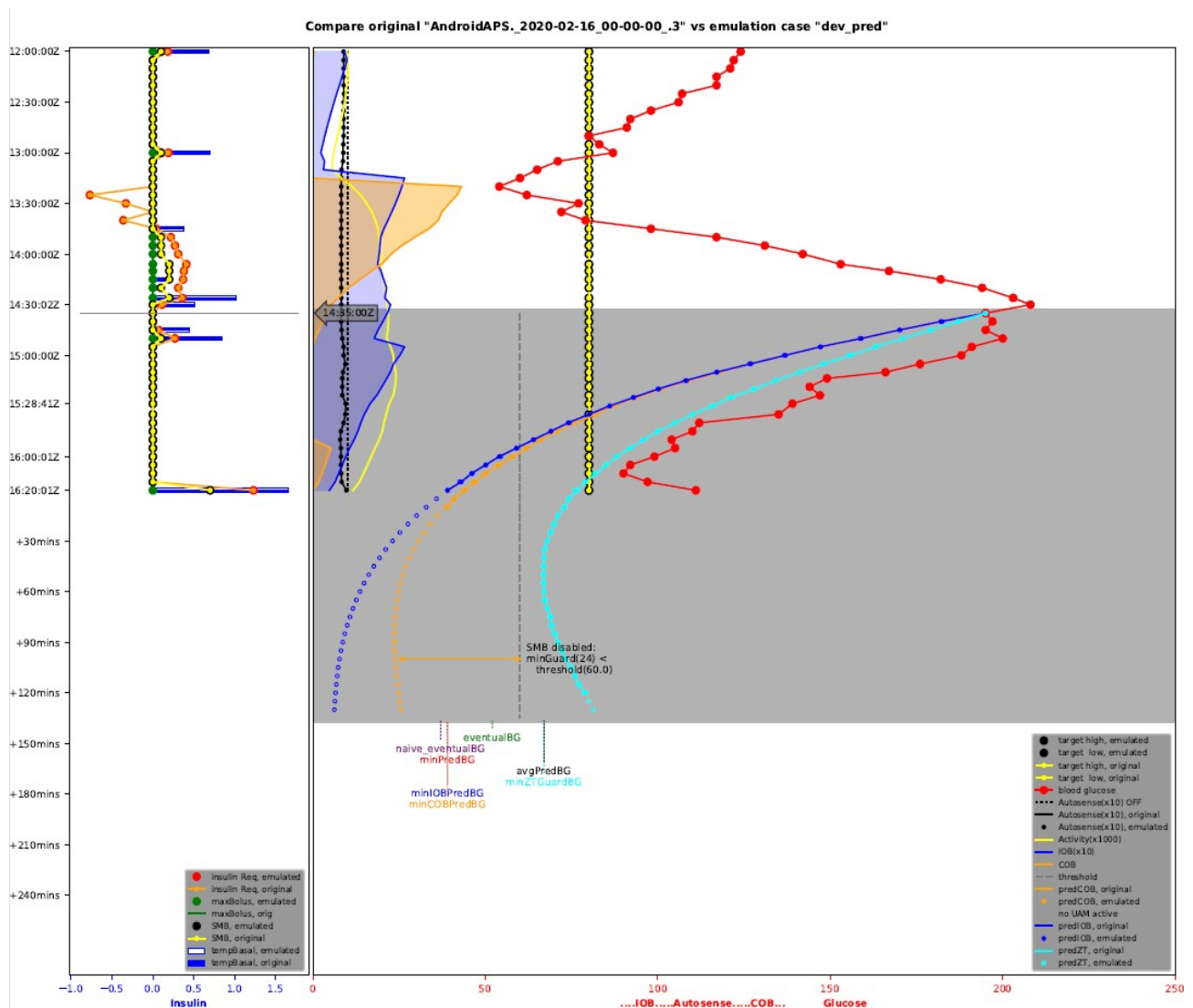
Die Ausgabeoption "pred"

Sie überlagert das Diagramm mit den Vorhersagelinien, die aus der AAPS-Registerkarte "Home" bekannt sind. Mit den Rückwärts- und Vorwärtspfeilen im pdf-Viewer können Sie eine Zeitlupenanimation des Verlaufs erhalten. Man hofft, dass dies das Verständnis für die Funktionsweise dexts Loops verbessern wird.

Das folgende Beispiel stammt aus einer anderen Protokolldatei ohne aktive Variation und zeigt einen Moment, in dem die Vorhersagen mit dem zukünftigen Blutzucker recht gut übereinstimmen. Das bedeutet, dass die Einstellungen etwa eine Stunde lang recht gut mit dem realen Leben übereinstimmen und dass es keine äußeren Störungen wie neue Kohlenhydrate oder Zieländerungen gab.



Einige Zeitschritte später meldet der Loop auf der Registerkarte "SMB" einen Grund, warum SMB deaktiviert wurde. Eine solche Situation wird in das Diagramm aufgenommen, wenn sie zutrifft, um das Verständnis der Argumentation zu erleichtern. Interessant finde ich, dass die Schleife die Vorhersagen mit dem Schwellenwert und nicht mit dem unteren Zielwert vergleicht. Dieser Schwellenwert ist definiert als 50 % des unteren Zielwerts plus 20 mg/dl, d. h. er liegt immer unter dem unteren Zielwert!

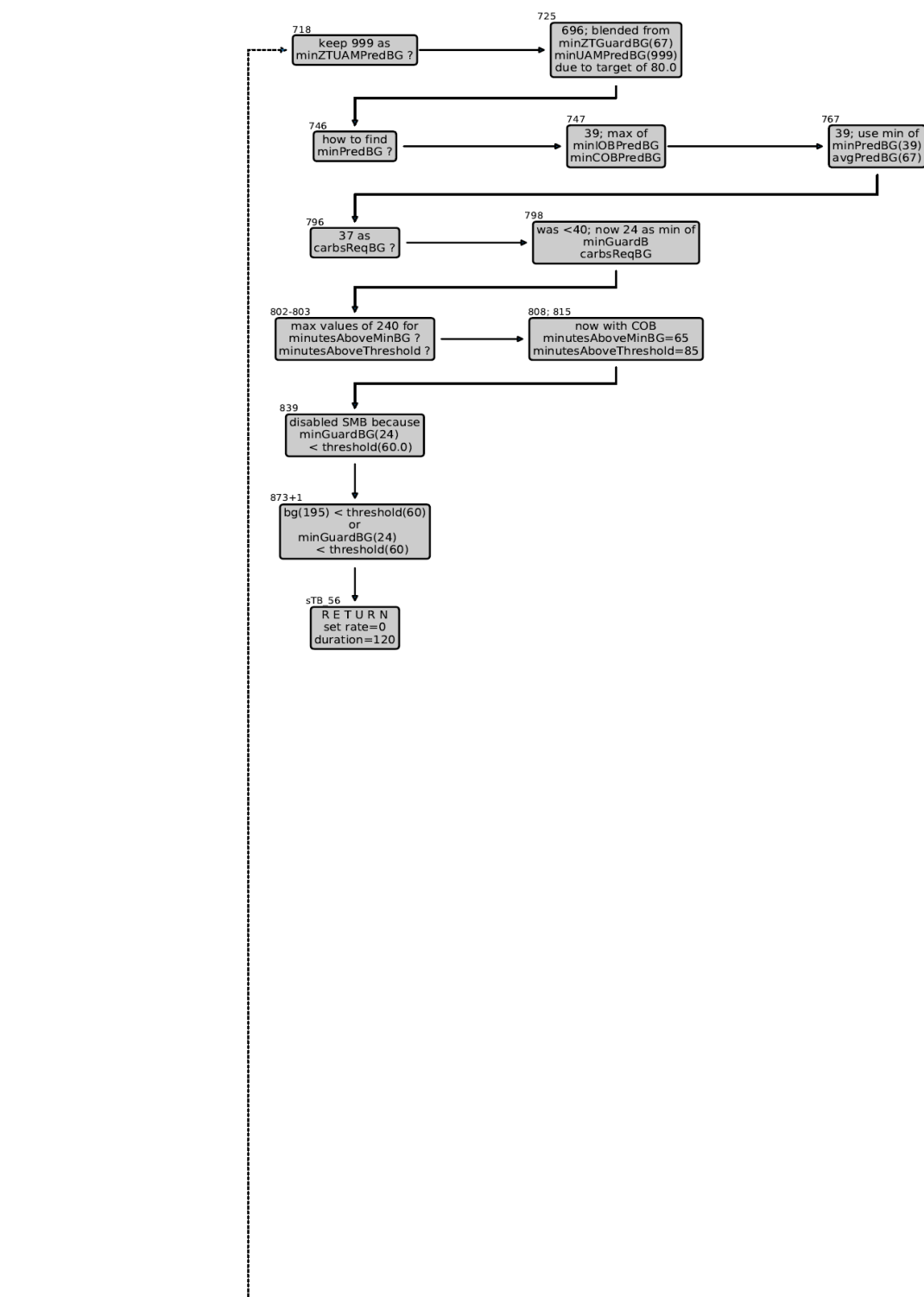


Die offenen Kreise in den Vorhersagen zeigen übrigens die ursprünglichen Berechnungen, die später in AAPS gekappt, abgeschnitten und gerundet werden, und diese endgültigen Werte sind als gefüllte Kreise dargestellt.

Die Ausgabeoption „flowchart“

Um die Logik und die Entscheidungen in determine-basal besser zu verstehen, habe ich diese zusätzliche Grafik hinzugefügt. Wie bei der Option "pred" wird für jeden Zeitschritt ein neues Flussdiagramm erstellt. Als Beispiel ist hier ein vergrößerter Ausschnitt für den gleichen Zeitschritt wie zuvor, d.h. die Deaktivierung von SMB um 14:35:00Z:

00-00-00.3" vs emulation case "dev_pred"



Flowchart and decision logic at time 14:35:00Z

Sie zeigt die Quelle der Entscheidung in Zeile 839³ der ursprünglichen Datei determine-basal.js.

³ Diese Zeilennummern waren in Version 2.6 gültig und haben sich inzwischen verschoben

Führen Sie die Analyse über das GUI-Frontend aus

Klicken Sie zunächst auf die Verknüpfung, die während der Installation auf dem Desktop erstellt wurde. Daraufhin öffnet sich ein DOS-Fenster. Das DOS-Fenster sollte leer bleiben, aber es kann sein, dass Fehlermeldungen auftauchen, und ich empfehle, diese zu überprüfen, bevor Sie die grafische Benutzeroberfläche verlassen. Ich habe versucht, solche Meldungen im Code für die Anzeige im Formular selbst abzufangen, aber man weiß ja nie.

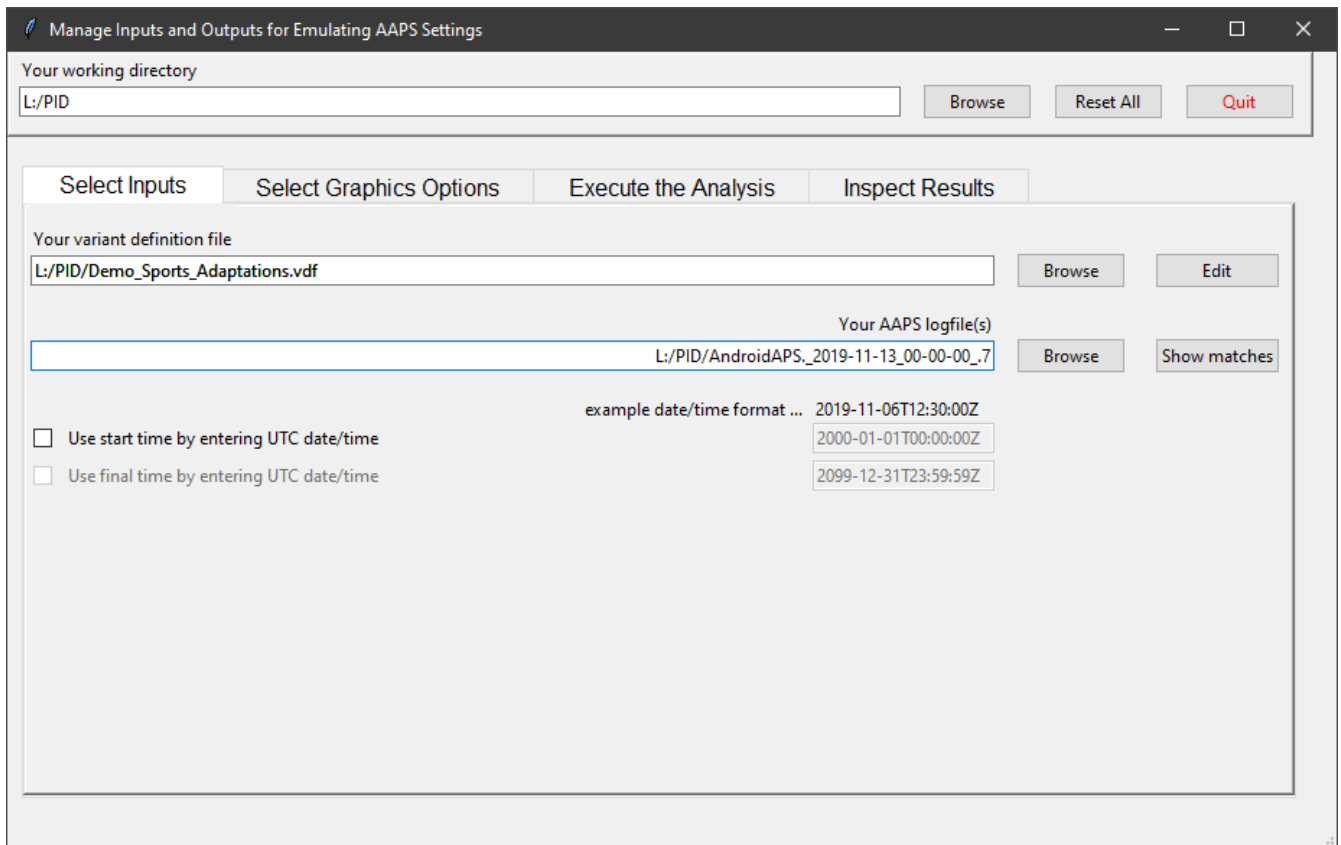
Nach dem Öffnen des DOS-Fensters öffnet sich ein weiteres Fenster, das ein Formular enthält, das den gesamten Prozess steuert. Die Größe des Fensters kann so eingestellt werden, dass längere Dateinamen ohne Scrollen angezeigt werden. Im Allgemeinen werden Kästchen mit weißem Hintergrund für die Eingabe von Text, meist Dateinamen, verwendet. Kästchen mit abgerundeten Ecken und grauem Hintergrund sind Befehlsschaltflächen zum Starten bestimmter Aktionen:

- Browse - startet den standardmäßigen Dateiauswahldialog, um die entsprechende Datei zu finden
- Edit - lädt die Datei in Ihren Standard-Editor (sofern Ihr System eine Standardanwendung für diesen Dateityp zugewiesen hat)
- Show - ähnlich wie "Edit", mit Ausnahme des PDF-Ergebnisses, bei dem Ihr Standard-PDF-Viewer verwendet wird

Der Prozess läuft von oben nach unten und von links nach rechts ab, aber Sie können natürlich überall hin springen. Zunächst wählen Sie den Arbeitsordner aus, d.h. den Ordner, in dem sich Ihre Variantendefinitionsdatei und später die Ergebnisdateien befinden.

Eingaben auswählen (Select inputs)

Klicken Sie auf die Registerkarte "Select Inputs", falls diese nicht bereits aktiv ist. Die beiden erforderlichen Eingaben sind die Namen der Variantendefinitionsdatei und der AAPS-Logdatei(en).



Wenn Sie mehrere Logdateien für längere Zeitfenster scannen müssen, bearbeiten Sie den Dateinamen, indem Sie "*" oder "?" an der entsprechenden Stelle des Dateinamens einsetzen. In solchen Fällen können Sie über die Schaltfläche "Show matches" prüfen, welche Dateien auf die Wildcard-

Spezifikation zutreffen.

Für die Angabe eines optionalen Startdatums/einer Startzeit aktivieren Sie das Kontrollkästchen. Das Zeitfeld ist dann nicht mehr ausgegraut und Sie können das Startdatum und die Startzeit eingeben/bearbeiten. Oberhalb des Feldes sehen Sie das Format für die Angabe von Datum und Uhrzeit. Denken Sie daran, dass AAPS das interne UTC-Format verwendet, das der Greenwich-Zeit ohne Sommerzeitverschiebung entspricht, d. h. in MESZ (Mittleuropäische-Sommerzeit) liegt es numerisch 2 Stunden zurück. Normalerweise benutze ich Kopieren und Einfügen aus diesem Teil des Logfile-Feldes und nehme dann die detaillierten Anpassungen vor. Wenn auch die Endzeit gewünscht ist, kreuzen Sie wieder das Kästchen an, kopieren Sie die Startzeit und passen Sie sie an.

Wenn Sie auf "Reset All" klicken, werden alle Eingabefelder gelöscht und die Standardeinstellungen wiederhergestellt.

Grafikoptionen auswählen (Select Graphics Options)

Klicken Sie auf die Registerkarte „Select Graphics Options“, falls sie nicht bereits aktiv ist. In diesem Formular wird ausgewählt, welche Informationen in der grafischen Ausgabe und in einer Tabelle auf dem Bildschirm angezeigt werden sollen. Es verwendet Optionsgruppen für die grobe Auswahl und zeigt gegebenenfalls Ankreuzfelder für die einzuschließenden oder auszuschließenden Details an. Die resultierende Zeichenkette, wie sie in der DOS-Methode verwendet wird, wird dynamisch aktualisiert, um die Ankreuzaktionen widerzuspiegeln, kann aber auch manuell bearbeitet werden.

Die Optionsfelder für das Dezimalsymbol ermöglichen es Ihnen, Ihre gewohnte Ansicht zu verwenden und das in Ihrer bevorzugten Tabellenkalkulation verwendete Trennzeichen bei der Erstellung der CSV-Datei auszuwählen.

Die anfängliche Auswahl für die gewünschten Ausgaben ist "most", die am häufigsten verwendete

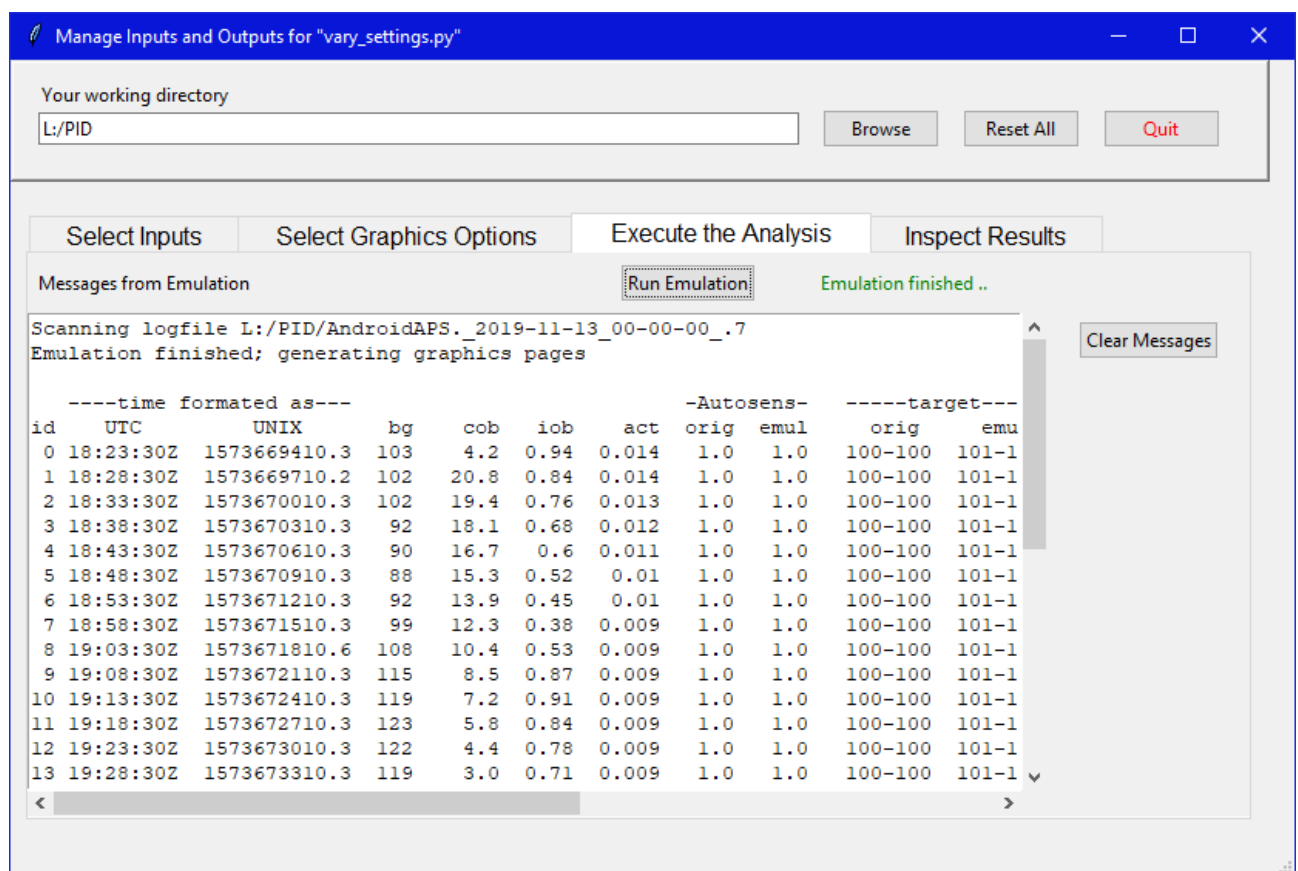
Option. Im Modus "most" kreuzen Sie die Optionen an, an denen Sie nicht interessiert sind.

Im Modus "some" ist es umgekehrt, d. h. Sie wählen die Optionen aus, an denen Sie besonders interessiert sind.

Im Modus "Alle" sind die feingliedrigen Auswahlgruppen im unteren Teil deaktiviert und ausgeblendet, da sie in diesem Kontext keinen Sinn machen.

Ausführen der Analyse (Execute the Analysis)

Klicken Sie auf die Registerkarte „Execute the Analysis“, wenn sie nicht bereits aktiv ist. Mit der Schaltfläche "Execute the Analysis" wird die Analyse gestartet. Der große weiße Rahmen zeigt Meldungen an, die sonst im DOS-Fenster angezeigt werden. Hier kann es sinnvoll sein, die Fenstergröße anzupassen, um mehr Inhalt ohne Scrollen anzuzeigen. Die vertikale Größenanpassung ist auf 30 Zeilen begrenzt, damit das Fenster auf einen normalen Monitor passt.

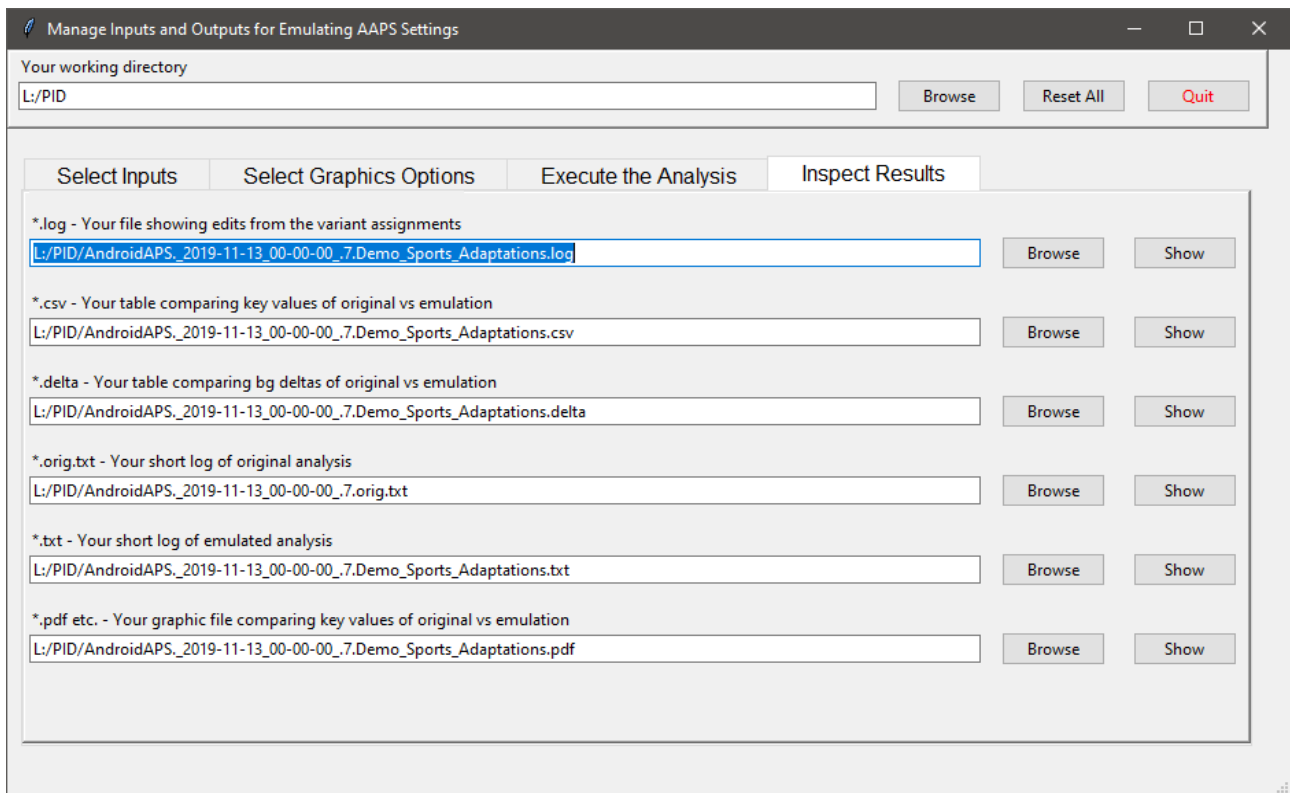


Diese tabellarische Auflistung sieht bei den späteren Softwareversionen etwas anders aus.

Fehlermeldungen oder **Hinweise auf fehlende Eingaben** werden in rot angezeigt. Die Schaltfläche "Clear Messages" löscht natürlich alle Inhalte in der Meldungsbox. Dies ist nützlich, bevor Sie die nächste Analyse starten.

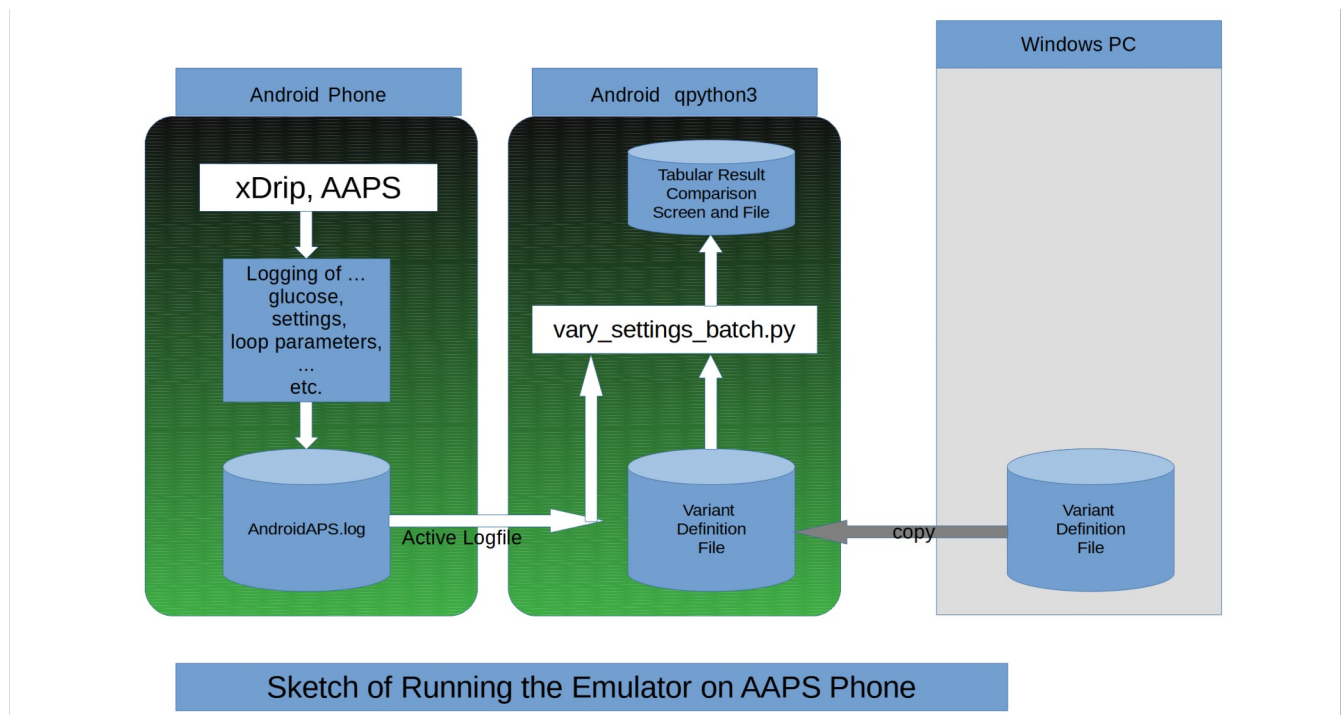
Ergebnisse inspizieren (Inspect Results)

Klicken Sie auf die Registerkarte „Inspect Results“, wenn sie nicht bereits aktiv ist. Diese Registerkarte wird ebenfalls automatisch aktiviert, sobald die Analyse abgeschlossen ist. Auch die Dateinamen sind nach der Analyse bereits vorausgefüllt. Sie können jedoch die Schaltflächen "Browse" verwenden, um den Inhalt von zuvor durchgeführten Analysen auszuwählen und anzuzeigen.



Ausführen der Analyse auf dem AAPS-Telefon

Hier ist eine Skizze der Daten- und Prozessflüsse bei Verwendung des Emulators auf dem Android-Telefon mit AAPS:



Die Verwendung des Emulators auf dem Telefon ist viel einfacher als unter Windows, da nur sehr wenige Eingabeparameter erforderlich sind. Diese wenigen Parameter werden in den ersten Dialogen eingegeben. Auch die Eingabe von Logdateien ist nicht erforderlich, da der Emulator immer mit der gerade aktiven Logdatei arbeitet. Auf diese Weise beschattet der Emulator effektiv den aktiven n Loop und Sie können sofort sehen, ob die alternativen Einstellungen andere SMBs oder TBRs geliefert hätten.

Wenn die SMB-Werte höher oder niedriger als im Master sind, wird der Benutzer durch eine Meldung per Sprachsynthese informiert. So kann man das Telefon versteckt tragen, bis man alarmiert wird und dann entscheiden, ob man das inkrementelle SMB als zusätzlichen Bolus manuell anwenden möchte.

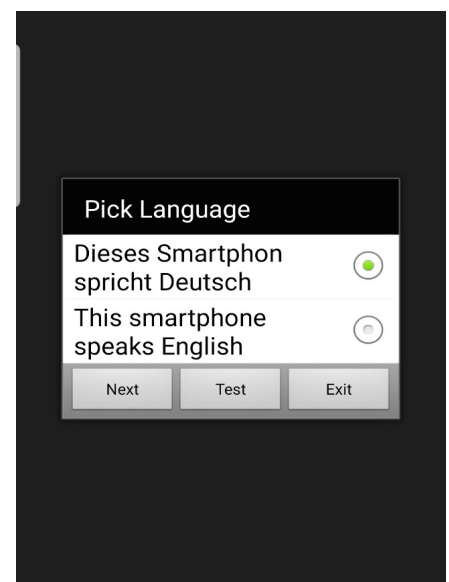
Als weitere Sicherheitsmaßnahme wird das Logfile regelmäßig auf die "add'l carbs req" gescannt. Wenn diese beim letzten Loop gefunden werden, wird wieder eine Meldung über die Sprachsynthese ausgegeben.

Start der Emulation

Drücken Sie auf dem Telefon die bei der Installation erstellte Schaltfläche "QPython3L". Drücken Sie dort auf "Programme", wählen Sie `vary_settings_batch.py` und wählen Sie schließlich "Run".

Der erste Dialog dient dazu, die Sprache für die Sprachsynthese auszuwählen. Warum das? Ursprünglich hatte ich Deutsch eingestellt, aber manchmal bestand die Dame darauf, alles auf Englisch auszusprechen. Gelegentlich wechselte sie sogar mitten in einer Sitzung die Sprache.

Das bedeutete, dass ich das Skript neu starten musste. Ich vermute, dass diese Umschaltungen von xDrip+ ausgelöst wurden, als "Speak Readings" aktiv war.



Klicken Sie auf "Test" um sich ein Beispiel für die Sprachsynthese anzuhören.

Klicken Sie auf "Next" um fortzufahren.

Die nächsten drei Dialoge dienen dazu, alle Stunden des Tages auszuwählen, in denen Sie gesprochene Alarmer zulassen möchten. So können Sie beispielsweise Alarmer während der Schlafenszeit deaktivieren.

Es gibt drei Situationen, die gesprochene Alarmer auslösen:

- Ankündigung von "add'l carbs required", wenn dies beim Scannen der AAPS-Meldungen entdeckt wird
- Ankündigung eines zusätzlichen Bolus, falls der SMB höher gewesen wäre
- Ankündigung eines reduzierten Bolus (sorry - jetzt zu spät), falls das SMB niedriger gewesen wäre

Die Bildschirmkopie auf der rechten Seite zeigt das Beispiel für den ersten Fall, die zusätzlich benötigten Kohlenhydrate.

Offensichtlich müssen Sie nach oben und unten scrollen, um alle Punkte auf der Liste zu erreichen.

Pick alarm hours for extra carbs required	
00	<input type="checkbox"/>
01	<input type="checkbox"/>
02	<input type="checkbox"/>
03	<input type="checkbox"/>
04	<input type="checkbox"/>
05	<input type="checkbox"/>
06	<input type="checkbox"/>
07	<input checked="" type="checkbox"/>
08	<input type="checkbox"/>
09	<input checked="" type="checkbox"/>
10	<input checked="" type="checkbox"/>
11	<input checked="" type="checkbox"/>
12	<input checked="" type="checkbox"/>
13	<input checked="" type="checkbox"/>
14	<input checked="" type="checkbox"/>
15	<input checked="" type="checkbox"/>
16	<input checked="" type="checkbox"/>
17	<input checked="" type="checkbox"/>
18	<input checked="" type="checkbox"/>
19	<input checked="" type="checkbox"/>
20	<input checked="" type="checkbox"/>
21	<input checked="" type="checkbox"/>
22	<input checked="" type="checkbox"/>
23	<input type="checkbox"/>

NextExit

Im nächsten Dialogfenster können Sie alle Elemente auswählen, die auf dem Bildschirm des Telefons angezeigt werden sollen. Sie sehen die anfängliche Standardauswahl. Je nach Größe Ihres Bildschirms im Querformat kann es sein, dass Sie nicht alles in eine einzige Zeile pro Loop unterbringen können und daher weniger interessante Spalten abwählen müssen.

Möglicherweise müssen Sie den Bildschirm nach oben und unten scrollen, um alle Optionen zu erreichen.

Die anfängliche Standardauswahl ist für die autoISF⁴-Einstellung geeignet, und Sie werden diese Auswahl wahrscheinlich ändern.

Diese Standardeinstellung erfordert 93 Spalten in der Anzeige.

Klicken Sie auf "Test", um eine Sprachausgabe zu erzeugen, die die erforderliche Anzahl von Spalten für die gewählte Auswahl ansagt.

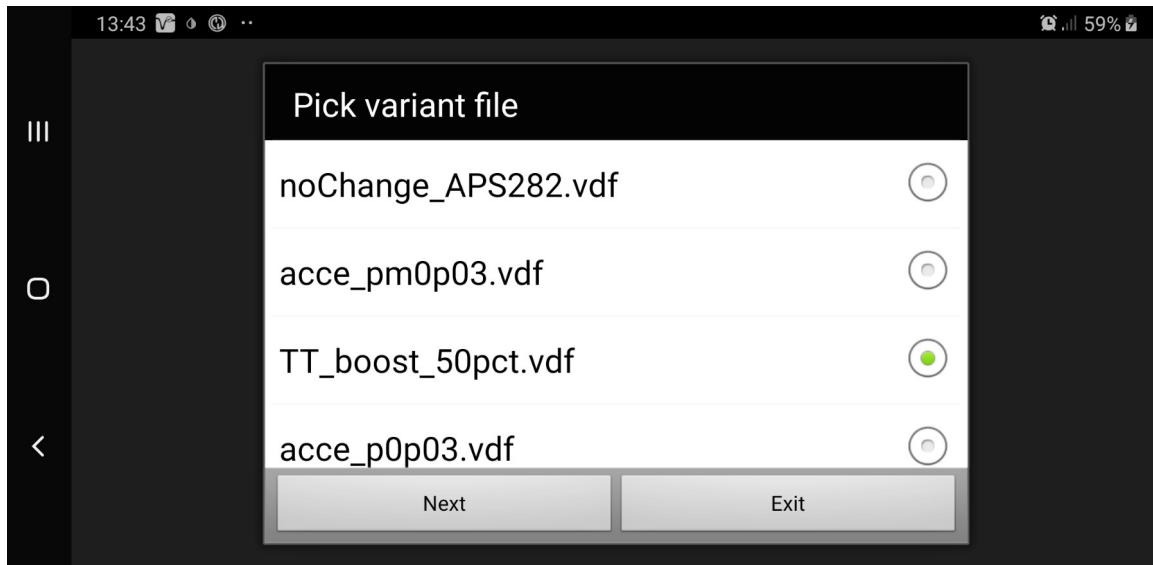
Klicken Sie auf „Next“ um fortzufahren.

Pick outputs	
bg	<input checked="" type="checkbox"/>
target	<input type="checkbox"/>
iob	<input type="checkbox"/>
cob	<input type="checkbox"/>
range	<input type="checkbox"/>
bestslope	<input type="checkbox"/>
ISF-factors	<input checked="" type="checkbox"/>
ISFs	<input checked="" type="checkbox"/>
insReq	<input checked="" type="checkbox"/>
SMB	<input checked="" type="checkbox"/>
basal	<input type="checkbox"/>

Next Test Exit

4 Siehe meinen Branch autoISF

Das letzte Dialogfenster, das sich öffnet, dient der Auswahl Ihrer Varianten-Definitionsdatei. Dies ist ein guter Zeitpunkt, um das Telefon ins Querformat zu drehen, damit längere Dateinamen schön angezeigt werden, und um die Anzeige der Ergebnistabelle vorzubereiten.



Klicken Sie auf „Next“ um fortzufahren.

Die Ergebnisse

Die regulären Ergebnisdateien sind im Ordner der Logfile-Datei gespeichert, falls Sie sie benötigen. Bitte beachten Sie, dass keine Grafikdatei erstellt wird, da ich die matplotlib-Bibliothek für Android nicht finden konnte.

Das Hauptergebnis der Android-Version ist die dynamische Tabelle mit den wichtigsten Ergebnissen. Sie umfassen die letzte Stunde oder so. Ältere Einträge werden gelöscht und neuere werden angehängt. Diese Tabelle wird nun auch unter Windows gedruckt, allerdings ist sie dort nicht auf maximal 14 Zeilen beschränkt.

13:29

54%

← No. 2

NEWCTRLTAKE WAKELOCK

UTC

-----ISF factors-----

-----ISFs-----

insulin Req

---SMB---

time	bg	auto	acce	bg	pp	delta	dura	orig	prof	emul	orig	emul	orig	emul
10:29Z	125	1.0	0.84	1.15	1	1	1.11	97.3	95	98.1	0	0	0	0
10:34Z	127	1.0	0.88	1.16	1.02	1	1.16	88.9	95	92.9	0	0	0	0
10:39Z	128	1.0	0.91	1.16	1.02	1	1.22	81.8	95	85.4	0.02	0.0	0	0
10:41Z	128	1.0	0.91	1.16	1.02	1	1.22	81.8	95	85.2	0.05	0.02	0	0
10:44Z	129	1.0	0.94	1.17	1.02	1	1.28	75.7	95	78.6	0.12	0.1	0.1	0
10:49Z	129	1.0	0.94	1.17	1.01	1	1.34	73.1	95	75.9	0.04	0.01	0	0
10:54Z	128	1.0	0.88	1.17	1	1	1.4	77.3	95	77.3	0	0	0	0
10:59Z	126	1.0	0.88	1.17	1	1	1.45	74.4	95	74.3	0	0	0	0
11:04Z	121	1.0	0.77	1.17	1	1	1.5	82	95	81.9	0	0	0	0
11:09Z	121	1.0	1.03	1.16	1	1	1.55	61.4	95	61.3	0	0.0	0	0
11:14Z	121	1.0	1.17	1.16	1	1	1.6	59.6	95	59.4	0.17	0.17	0.1	0.1
11:18Z	123	1.0	1.07	1.15	1.02	1	1.65	57.7	95	57.6	0.28	0.28	0.2	0.2
11:24Z	123	1.0	1	1.15	1.01	1	1.7	56	95	55.8	0.07	0.07	0	0
11:29Z	120	1.0	0.72	1.15	1	1	1.75	75.5	95	75.3	0	0	0	0

Variant "TT_boost_50pct"

waiting 300 sec for next loop at 13:34

Manchmal ging beim Drehen des Telefons die Spaltenausrichtung verloren und ich musste auf die

nächste Aktualisierung des Loops warten. Der Grund dafür könnte sein, dass auf meinem Telefon die maximale Breite im Hochformat 46 beträgt, also die Hälfte der obigen Tabellenbreite. Daher füge ich ein BLANK an und das Drehen des Telefons ist weniger störend. Wenn Sie andere Bildschirmabmessungen haben und sich darüber ärgern, lassen Sie es mich wissen, und ich werde vielleicht eine benutzerspezifische Konfigurationsdatei hinzufügen.

Beenden Sie die Emulation

Klicken Sie auf die "Back"-Taste des Telefons und wählen Sie dann "NEIN", d.h. lassen Sie es nicht im Hintergrund laufen.

Tipps und tricks

- Falls die Tastatur die Ergebnistabelle überlagert oder vom Bildschirm verdrängt, klicken Sie auf die "Zurück"-Taste des Telefons.
- Falls Sie das Telefon zwischen Hoch- und Querformat drehen, kann die Ausrichtung der Spalten durcheinander geraten. Dies wird bei der nächsten Ausführung der Schleife behoben und sieht dann wieder ordentlich aus.
- Falls die Tabelle anfangs nur ein paar Zeilen enthält, wurde das Skript kurz nach dem Start eines neuen Logfiles gestartet. Dieses frische Logfile umfasst weniger als 1 Stunde.
- Die erste Spalte listet die Ausführungszeiten im UTC-Format auf. Für die Sommerzeit in Mitteleuropa liegt dieses Label also 2 Stunden hinter Ihrer Ortszeit und der des Telefons.
- Manchmal verschwindet die qpython3-App aus der Liste der letzten Apps (auf meinem Telefon scheint es eine Grenze von 4 zu geben). In solchen Fällen beende ich die App (Android-Einstellungen → Apps → Qpython3 → Force Stop) und starte sie neu.
- Die Meldung "add'l carbs req" erscheint, obwohl genügend Kohlenhydrate in den Bolusrechner eingegeben wurden. Ich habe einen Filter eingebaut, der diese Meldung bei kleineren Mengen und längeren Zeitspannen unterdrückt. Wenn die Meldung immer noch über die Sprachsynthese ausgegeben wird, überprüfe ich, ob ich wirklich schnelle Kohlenhydrate zusätzlich benötige oder ob die Kohlenhydrate an Bord schnell genug sind, um mich auf der sicheren Seite zu halten.

Status: 30-Jun-2022 @ 14:43

Anhang A – Wie man Logfiles nutzt ohne autoISF-Inhalt oder vor Version 2.2.6

Unter dieser Überschrift gibt es 3 mögliche Szenarien:

Szenario 1: Der Logfile stammt aus regulärem AAPS, d.h. ohne autoISF

```
profile      enable_autoISF      False      ### for logs before autoISF
profile      smb_delivery_ratio   0.5        ### AAPS default
```

Dieser Fall wurde bereits im obigen Beispiel der Sportanpassung angesprochen. Die VDF-Datei benötigt diese beiden zusätzlichen Zeilen, die am Anfang eingefügt werden, um ihr mitzuteilen, dass autoISF deaktiviert ist und dass das SMB-Abgabeverhältnis dem des normalen AAPS entspricht.

Szenario 2: Der Logfile stammt aus AAPS mit einer autoISF-Version vor 2.2.6

```
profile      enable_autoISF      profile['use_autoisf']      ### get from previous name
profile      autoISF_max          profile['autoisf_max']      ### get from previous name
profile      dura_ISF_weight      profile['autoisf_hourlychange'] ### get from previous name
profile      enable_dura_ISF_with_COB profile['enableautoisf_with_COB'] ### get from previous name
profile      autoISF_min          -999                      ### no limit at the time
profile      enable_pp_ISF_always False                          ### no limit at the time
profile      pp_ISF_hours          0                            ### no limit at the time
new_parameter pp_ISF              1                            ### effectively unused
#ew_parameter delta_ISF          1                            ### effectively unused
```

In diesem Fall ist der autoISF2.1-Algorithmus in AAPS enthalten, aber die Logik hat sich geändert und die Parameternamen ebenfalls. Sie müssen der VDF-Datei mitteilen, welchen alten Namen sie dem entsprechenden neuen zuordnen soll. Die Details hängen davon ab, welche "Zwischen"-Version von autoISF zu diesem Zeitpunkt aktiv war. Wenn es nicht funktioniert, lesen Sie die Fehlermeldung, die Ihnen sagen wird, welcher Parameter fehlt oder falsch geschrieben ist.

Szenario 3: Der Logfile stammt aus regulärem AAPS, d.h. ohne autoISF und Sie wollen prüfen wie autoISF die Situation ändern würde

```
profile      enable_autoISF      True      ### activate all of autoISF
profile      autoISF_min          0.7
profile      autoISF_max          1.2
profile      smb_delivery_ratio   0.5
profile      enable_dura_ISF_with_COB True
profile      dura_ISF_weight      0.5

new_parameter acce_ISF          1.0      ### inactive
new_parameter bg_ISF            1.0      ### inactive
new_parameter pp_ISF            1.0      ### inactive
new_parameter delta_ISF         1.0      ### inactive
```

Hier ist ein Beispiel, wie Sie den dura_ISF-Effekt aktivieren und die anderen inaktiv lassen würden, um zu sehen, ob und wann er Ihre Insulinabgabe verändert hätte.

Wenn Sie alternativ untersuchen möchten, wie pp_ISF bei Ihrer Gastroparese geholfen haben könnte, dann würden Sie dura_ISF = 1.0 definieren und anstelle der beiden Profilparameter für dura_ISF die Parameter auflisten, die zu pp_ISF beitragen.

Anhang P – Definieren eines Pumpenprofils

Wenn Sie Automatisierungsregeln zum Ändern von Profilen verwendet haben, möchten Sie vielleicht zu einem normalen Profil zurückkehren. So können Sie ein Pumpenprofil innerhalb einer VDF-Datei definieren:

```
STAIR_ISF      00:00:00Z  45      ### 01h_M(ittel)E(uropäischde)Z(eit) oder 02h_MESZ
STAIR_ISF      01:00:00Z  44      ###
STAIR_ISF      02:00:00Z  42      ###
...
STAIR_ISF      17:00:00Z  36      ### 18h_MEZ
STAIR_ISF      18:00:00Z  38      ###
...
STAIR_ISF      22:00:00Z  43      ### 23h_MEZ
STAIR_ISF      23:00:00Z  44      ### 00h_MEZ
profile      sens      STAIR_ISF  ###

STAIR_CR       00:00:00Z  8.0      ### 01h_M(ittel)E(uropäischde)Z(eit) oder 02h_MESZ
STAIR_CR       01:00:00Z  7.5      ###
...
STAIR_CR       20:00:00Z  7.5      ###
STAIR_CR       21:00:00Z  8.0      ###
STAIR_CR       22:00:00Z  9.0      ### 23h_MEZ
STAIR_CR       23:00:00Z  9.0      ### 00h_MEZ
profile      carb_ratio  STAIR_CR  ###

STAIR_BAS      00:00:00Z  0.41     ### 01h_M(ittel)E(uropäischde)Z(eit) oder 02h_MESZ
STAIR_BAS      01:00:00Z  0.43     ###
STAIR_BAS      02:00:00Z  0.44     ###
STAIR_BAS      03:00:00Z  0.50     ###
...
STAIR_BAS      19:00:00Z  0.75     ### 20h_MEZ
STAIR_BAS      20:00:00Z  0.75     ###
STAIR_BAS      21:00:00Z  0.60     ###
STAIR_BAS      22:00:00Z  0.45     ### 23h_MEZ
STAIR_BAS      23:00:00Z  0.43     ### 00h_MEZ
profile      current_basal  STAIR_BAS  ###
```

Zur besseren Lesbarkeit wurden in diesem Beispiel einige Zeilen weggelassen. Wichtig zu beachten ist die Zeitverschiebung zwischen UTC auf der einen Seite und MEZ (Mittleuropäische Zeit) bzw. MESZ (Mittleuropäische Sommerzeit) auf der anderen Seite. Die Zeilen müssen nach der UTC-Zeit sortiert sein. Daher muss die erste Zeile (Winterzeit) bzw. die ersten beiden Zeilen (Sommerzeit) aus Ihrer Pumpendefinition abgeschnitten und am Ende angehängt werden. Wenn Ihr Profil nicht für 24 Stunden vollständig ausgefüllt ist, kann dies auch bedeuten, dass Sie zunächst einen Pumpeneintrag um 01 bzw. 02 Uhr erstellen müssen.

Anhang V – Einige weitere Beispiele für VDF Einträge

Diese können als Ideensammlung und als Proforma für Ihre eigenen Studien dienen.

Array	Element	Wert/Ausdruck	Bemerkung
Beispiele zum Ändern der Zielwerte			
profile	max_bg	85	### fester Wert
profile	min_bg	profile['min_bg']+10	### jetzigen Wert um 10mg/dl erhöhen
profile	min_bg	profile['min_bg']*1.2	### jetzigen Wert um 20% erhöhen
profile	target_bg	(profile['min_bg'] + profile['max_bg']) / 2	### korrekte Anweisung; ist aber nicht nötig, weil AAPS das sowieso macht
Beispiele für Änderungen von Elementen im Array "profile"			
profile	enableSMB_always	True	### funktioniert für alle CGMs; beachten Sie den Großbuchstaben in "True"
profile	sens	95	### fester Wert
profile	carb_ratio	12.5	### fester Wert
profile	enableUAM	True	
profile	enableSMB_with_high_temp_target	False	
profile	maxUAMSMBBasalMinutes	180	### rein theoretisch, in AAPS nicht möglich
Beispiel zur Änderung des ISF abhängig von der Abweichung des BZ vom Ziel			
temp	BG	glucose_status['glucose']	### speichere BZ in der Hilfsvariablen BG
temp	Aim	(profile['min_bg'] + profile['max_bg'])/2	### speichere mittleres Ziel in der Hilfsvariablen Aim
temp	Dev	(temp['BG'] - temp['Aim'])/temp['Aim']	### speichere relative Abweichung in der Hilfsvariablen Dev
profile	sens	profile['sens'] / temp['Dev']	### stärke oder schwäche ISF abhängig von der Zielabweichung
Beispiel zur Stärkung des ISF falls delta>10mg/dl/5min und BZ>100mg/dl			
temp	steep	glucose_status['delta'] > 10	### 1 falls delta >10mg/dl/5min; sonst 0
temp	high	glucose_status['glucose'] > 100	### 1 ifalls BZ > 100mg/dl; sonst 0
temp	strong	1 + temp['high'] * temp['steep'] * 0.2	### 1.2 falls beide Bedingungen zutreffen; sonst 1
profile	sens	profile['sens'] / temp['strong']	### stärke ISF um 20% falls beide Bedingungen zutreffen; sonst belassen
Beispiele für Änderungen abhängig von Datum und Uhrzeit			
STAIR	2020-04-14T03:00:00Z	120	### Wert ab 3:00 UTC, also 05:00 MESZ um sich auf Sport vorzubereiten
STAIR	2020-04-14T05:00:00Z	150	### Wert ab 07:00 MESZ, dem Beginn der morgendlichen Übungen
STAIR	2020-04-14T06:00:00Z	90	### Wert ab 08:00, MESZ, dem Ende der morgendlichen Übungen
profile	max_bg	STAIR	### sich zeitlich ändernder Wert zwischen 5 und 8 Uhr MESZ
INTERPOL	2020-04-14T20	0	### Wert um 20:00 UTC
INTERPOL	2020-04-14T22	4	### Wert um 22:00 UTC
profile	carb_ratio	INTERPOL+10	### linear ansteigender Wert von CR einschließlich Inter- und Extrapolation
Weitere Möglichkeiten			
new_parameter	AAPS_Version	"<2.7"	### Nutze AAPS 2.6.1 Algorithmus statt 2.7 oder 2.8; beachte die Anführungszeichen um Text zu kennzeichnen
autosens_data	ratio	1	### schaltet Autosens aus
autosens_data	ratio	1.3	### rein theoretisch, macht kaum Sinn