

Entwicklung eines Konverters für die Harmonisierung von Hypnogrammen



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences



**Centrum für biomedizinische
Bild- und Informationsverarbeitung**

Forschung. Innovation. Inkubation

Sarah-Lee Mendenhall

September 25, 2020

Abstract

Zur Erkennung unterschiedlicher Erkrankungen in der Schlafmedizin nimmt die Auswertung von Hypnogrammen eine zentrale Bedeutung ein. In der forschungsbasierten Arbeit stehen Datenpoole verschiedener Hypnogramme zur Verfügung, welche sich in der Herkunft und damit auch in der Formatierung unterscheiden können. Um diese einheitlich nutzbar zu machen, bietet sich eine Konvertierung auf ein einheitliches Format an. Durch das Fehlen eines Standards, nähert sich diese Arbeit einem möglichen Standard an und konvertiert werden beispielhaft vier verschiedene Ausgangsformate.

Contents

1	Zielsetzung	2
2	Einführung	2
2.1	Schlafstadien	2
2.1.1	Rechtschaffen und Kales	3
2.1.2	AASM	3
2.2	Hypnogramme	3
3	Formate	4
3.1	Quellformate	4
3.2	Zielformat	5
4	Umsetzung und Ergebnisse	5
4.1	Programmablauf	5
4.2	Pakete	6
4.3	Selektion	6
4.4	Parameterübergabe	7
4.5	Mapping	7
4.6	Template	9
4.7	Ausgabeformat	10
5	Zusammenfassung und Ausblick	10

1 Zielsetzung

Als Ausgangssituation liegen Hypnogramme in unterschiedlichen Formaten und Aufbau vor. Um sie in anderen Anwendungen einheitlich nutzen zu können, sollen sie in ein gemeinsames Format überführt werden. Beispielfür, werden vier häufige Dateiformate verschiedener Herkunft in ein gemeinsames JSON Format übertragen. Da es bislang keinen einheitlichen Standard für die Formatierung von Hypnogrammen gibt, soll sich das Endformat zumindest einem bekannten Standard des Datenaustausches im Gesundheitswesen annähern. Als Ergebnis wird ein terminalbasiertes Programm basierend auf plattformunabhängigen ausführbaren Python 3 Code erwartet.

2 Einführung

Als wichtigstes diagnostisches Verfahren in der Schlafmedizin dient die Polysomnographie. Hierbei werden diverse physiologische Messungen während des Schlafs aufgezeichnet. Als Teil dieser Messungen wird ein Hypnogramm erstellt, welches die Verteilung der einzelnen Schlafstadien während der Aufzeichnung grafisch darstellt.

2.1 Schlafstadien

Der Schlaf wird in verschiedene Arten unterteilt, die s.g. Schlafstadien. Diese werden durch spezifische Gehirnaktivitäten gekennzeichnet und anhand von EEG, EMG und EOG erfasst und ausgewertet. Den Zusammenschluss aller aufgetretenen Schlafstadien nennt man Schlafprofil. Für die Beurteilung eines gesunden Schlafprofils wird sowohl die Verteilung der einzelnen Schlafstadien, als auch der Zeitpunkt des Eintretens einer Schlafphase benötigt. Als Goldstandard zur Ableitung der polysomnografischen Daten in ein Schlafprofil galt bis 2007 der von Rechtschaffen und Kales 1968 publizierte Vorschlag zur Schlafstadieneinteilung. Die von Rechtschaffen und Kales publizierte Einteilung galt bis 2007 als Leitlinie. Die American Academy of Sleep Medicine (AASM) publizierte ein neues Manual 2007 und löste damit das s.g. "R&K Manual" ab[9]. Vor der eigentlichen Einteilung in einzelne Schlafstadien wird zunächst in REM-Schlaf und NON-REM-Schlaf unterschieden [9]. Eine kurze Betrachtung beider Nomenklaturen ist an dieser Stelle notwendig, da einige der zu konvertierenden Hypnogramme noch aus den 90er Jahren stammen.

2.1.1 Rechtschaffen und Kales

Nach den s.g. 'R&K' Regeln wird der Schlaf in fünf unterschiedliche Bereiche eingeteilt. Diese sind die Stadien Wach, S1 bis S4 und REM.

Table 1: Beschreibung der Schlafstadien nach R&K[8]

Schlafstadium	Beschreibung
Wach	Wachzustand
S1	Übergangsphase zwischen Wach und Schlaf
S2	stabiler Schlaf
S3	Tiefschlaf
S4	Tiefschlaf
REM	REM-Schlaf
Movement Time	Bewegung

2.1.2 AASM

Das von der American Academy of Sleep Medicine neu vorgelegte Manual fasste die Stadien aus dem R&K Manual auf neue Weise zusammen. Hierbei wird nun eine Einteilung in Non-REM(N)1, N2 und N3 sowie REM (R) vorgenommen[9]. Die Schlafstadien 3 und 4 wurden zu N3 zusammengefasst.

Table 2: Beschreibung der Schlafstadien nach AASM [7]

Schlafstadium	Beschreibung
W	Wachzustand
N1	Übergangsphase zwischen Wach und Schlaf
N2	stabiler Schlaf
N3	Tiefschlaf
R	REM-Schlaf

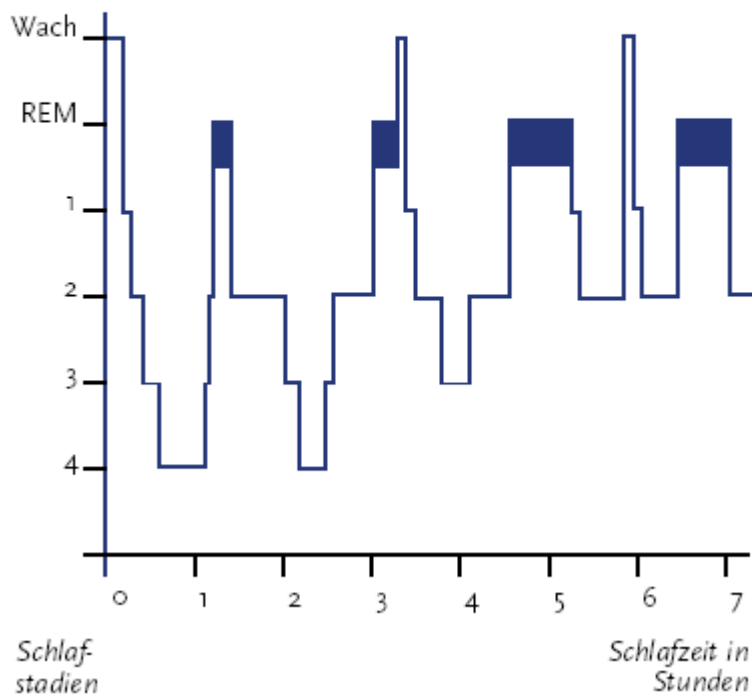
2.2 Hypnogramme

Ein Hypnogramm dient als grafische Darstellung des Schlafprofils. Hierbei wird eine Zuordnung der einzelnen Schlafstadien in 30 Sekunden Fragmente vorgenommen. Das entspricht pro Fragment einem Schlafstadium. Über die Schlafstadien hinaus werden hierbei auch Aufzeichnungen erfasst die nicht der allgemeinen Nomenklatur entsprechen, wie z.B. Artefakte. In der grafischen Auswertung kann gewählt werden, ob diese auch im Hypnogramm dargestellt

werden sollen. Die X-Achse der Grafik gibt hierbei das erreichte Schlafstadium an und stellt dies über die Y-Achse in eine zeitliche Abhängigkeit. Nach aktuellem Stand, gibt es derzeit keinen einheitlichen Standard um Hypnogramme zu erzeugen oder zu speichern.

Idealtypisches Schlafprofil (Hypnogramm) eines jungen Erwachsenen

Das Stadium des REM-Schlafes ist gekennzeichnet durch schnelle Augenbewegungen (Rapid Eye Movement). Die Leichtschlafstadien sind mit 1 und 2, die Tiefschlafstadien mit 3 und 4 bezeichnet.



Quelle: [6]

Figure 1: Hypnogramm

3 Formate

3.1 Quellformate

Für die Umsetzung des Konverters wurden vier repräsentative Hypnogramme ausgewählt. Als Ausleseformate liegen eine XML, eine RML, eine CSV und eine EDF-Datei vor. Diese unterscheiden sich sowohl in der Formatierung,

als auch in der Herkunft. Die Herkunft zu kennen ist wichtig um die einzelnen Werte korrekt auslesen zu können, da die Bezeichnungen nicht einheitlich vorgenommen werden. Dies zeigt sich vor allem in der Kodierung der einzelnen Schlafstadien. Die vorliegenden Hypnogramme unterscheiden sich auch inhaltlich. Hierfür musste zunächst analysiert werden, welche gemeinsamen Werte vorliegen. Nicht jedes Hypnogramm beinhaltet ein Datum oder nähere Angaben zum Patienten. Da es für die Verarbeitung und Darstellung nur auf die Schlafstadien mit den entsprechenden Epochen ankommt, werden zur Verarbeitung auch nur diese Daten ausgelesen. Dies gewährleistet eine Vereinheitlichung der Daten.

3.2 Zielformat

Um sich zumindest einem allgemeinen Standard zu nähern, wurde zunächst als Zielformat ein geeigneter Standard gesucht. Hierfür bot sich grundsätzlich etwas aus dem Datensatz des FHIR an, da dieses für den Datenaustausch im Gesundheitswesen immer mehr an Bedeutung gewinnt[1]. Der FHIR Standard ist modular aufgebaut, hierbei werden einzelne, untereinander verlinkbare Ressourcen angelegt. Da das Hypnogramm nur einen Teil der Polysomnographie darstellt und zumeist zusammen mit anderen Aufzeichnungen betrachtet wird, bietet sich die Anlage als Ressource an. Hierfür wurde die Ressource mit der höchsten Ähnlichkeit ausgewählt, in diesem Fall "Observation" [2]. Um die Kodierung der einzelnen Schlafstadien zu vereinheitlichen, wurde sich hier auf die numerische Kodierung im LOINC Format geeinigt[4].

4 Umsetzung und Ergebnisse

4.1 Programmablauf

Zur Ausführung des Programms muss zunächst das Modul EdfReader installiert werden über: "pip3 install -user edfrd" Eventuell muss vorher noch der Paketmanager installiert werden, was abhängig vom Betriebssystem ist. "sudo apt-get install python3-pip" Zum Starten des aktuellen Programms, ist es notwendig die Datei converter.py zu starten. Hierbei werden ein benötigter und ein optionaler Parameter übergeben.

"python3 converter.py [Dateiname][opt.: neuer Dateiname]" Die Ergebnisse werden auf der Konsole ausgegeben und in eine json gespeichert im Zielformat.

4.2 Pakete

Das European Data Format ist eigentlich ein Dateiformat, welches für die Speicherung von Biosignalen genutzt wird. Da Hypnogramme aus polysomnografischen Aufzeichnungen resultieren, werden diese teilweise auch als EDF bereitgestellt. Python bietet keine Standardpakete für den Umgang mit EDF Dateien, hierfür wurde der EDF Reader von Christoph Jansen verwendet[5]. Für das Einlesen von XML und RML Dateien wurde die ElementTree XML API verwendet. Diese bietet einen einfachen Zugriff auf XML Strukturen[3]. Zum Einlesen und verarbeiten der CSV wurde auf die Standard Python-pakete zurückgegriffen.

```
1 # %%
2 from edfrd import read_header, read_data_records
3 import json
4 import xml.etree.ElementTree as ET
5 import csv
6 import argparse
```

4.3 Selektion

Da es sich bei der Eingabe um unterschiedliche Formate handelt, musste ein Selektionsparameter festgelegt werden. Dieser beruht auf dem Ursprung der Datei. Zunächst erfolgt eine Erkennung des Dateityps anhand des Inhalts einer Datei. Für die Funktionsweise des Konverters ist es wichtig, dass die Eingabedateien nicht nur im Dateityp übereinstimmen, sondern auch in der inhaltlichen Struktur. Nach der Erkennung des Typs, erfolgt der jeweilige Funktionsaufruf zum Auslesen der Datei.

```
1 def recognize():
2     parameter = None
3     ergebnis = None
4     originfile = open(inputfile, 'r')
5     # print(datei.read(30))
6     ergebnis = originfile.read(150)
7
8     if "X,,4,09.05.1999,21:55:42,10.05" in ergebnis:
9         parameter = "RemLogic"
10        functionremlogic(parameter)
11    elif "<CMPStudyConfig>" in ergebnis:
12        parameter = "CMPStudy"
13        cmpstudy(parameter)
14    elif "www.respironics.com" in ergebnis:
15        parameter = "Respironics"
16        functionRespironics(parameter)
17    elif "Converted by EDFtoEDF CDatentechnik" in ergebnis:
```



```

18     parameter = "CDatentechnik"
19     functionCDatentechnik(parameter)
20 else:
21     print("nicht zulässiger Dateityp")

```

Derzeit wird davon ausgegangen, dass die Herkunft auch das zu konvertierende Dateiformat bestimmt. Wenn keines der zulässigen Dateistrukturen erkannt wird, erzeugt das Programm eine Konsolenausgabe mit Hinweis auf einen nicht erkannten Dateityp und beendet das Programm an dieser Stelle.

4.4 Parameterübergabe

Es handelt sich um ein terminalbasiertes Programm. Für die Übergabe der Parameter wurde das Paket `argparse` verwendet. Neben dem Aufruf der Datei muss auch die zu konvertierende Datei übergeben werden. Optional kann ein Parameter zur Dateibenennung der Ausgabe übergeben werden, per default wird dieser Name aus dem ursprünglichen Dateinamen gebildet.

```

1 parser = argparse.ArgumentParser()
2
3     parser.add_argument("Dateiname", help="Name der zu
4     konvertierenden Datei")
5     argsdefault = parser.parse_args()
6     parser.add_argument("Output", type=str, help="Name der
7     neuen Datei", nargs="?", default=argsdefault.Dateiname+".
8     json")
9
10    args = parser.parse_args()
11    inputfile = args.Dateiname
12    # parameter = args.Herkunft
13    outputfile = args.Output

```

4.5 Mapping

Das Scoring der Schlafstadien unterscheidet sich in allen vorliegenden Hypnogrammen. Für die angestrebte Vereinheitlichung der Hypnogramme ist ein Mapping auf die entsprechenden LOINC Kodierungen erforderlich. Auch hierbei wurde als Selektionsparameter die Herkunft der Datei gewählt.

```

1 # mapping sleepstages on LOINC Format
2 def functionmapping(maplist, givenperiod, parameter):
3     convlist = []
4
5     def switch_stages(mapped):
6         global switcher

```

```

7         if parameter == "RemLogic":
8             switcher = {
9                 # corresponds to Sleep stage unspecified in
Loinc
10                 "10": 23664,
11                 "11": 23672,
12                 "12": 23680,
13                 # "RemSpindle": 23688,
14                 "13": 23696,
15                 "14": 23704,
16                 "15": 23712,
17             }
18         if parameter == "CDatentechnik":
19             switcher = {
20                 # corresponds to Sleep stage unspecified in
Loinc
21                 # hier muss die Zuordnung nochmals
abgeglichen werden
22                 9: 23656,
23                 6: 23664,
24                 0: 23672,
25                 5: 23680,
26                 7: 23688,
27                 1: 23696,
28                 2: 23704,
29                 3: 23712,
30                 4: 23720,
31             }
32         if parameter == "CMPStudy":
33             switcher = {
34                 "9": 23656,
35                 "6": 23664,
36                 "0": 23672,
37                 "5": 23680,
38                 "7": 23688,
39                 "1": 23696,
40                 "2": 23704,
41                 "3": 23712,
42                 "4": 23720
43             }
44         if parameter == "Respironics":
45             switcher = {
46                 # entspricht Sleep stage unspecified in Loinc
47                 "NotScored": 23656,
48                 "": 23664,
49                 "Wake": 23672,
50                 "REM": 23680,
51                 "RemSpindle": 23688,
52                 "NonREM1": 23696,

```

```

53         "NonREM2": 23704,
54         "NonREM3": 23712,
55         "NonREM4": 23720,
56     }
57     convlist.append(switcher.get(mapped, stage))
58
59     for stage in maplist:
60         switch_stages(stage)
61     functionjson(convlist, givenperiod)

```

4.6 Template

Als Template für die Konvertierung wurde eine JSON Datei angelegt, welche die Ressource "Observation" repräsentiert(s.Kapitel 3.2). Diese kann jederzeit angepasst werden und wird bei Erstellung der konvertierten Datei importiert.

```

1  {
2      "resourceType": "Observation",
3      "id": "Hypnogram",
4      "text": "Hypnogram",
5      "status": "final",
6      "category": [
7          {
8              "coding": [
9                  {
10                     "system": "http://terminology.hl7.org/
CodeSystem/observation-category",
11                     "code": "procedure",
12                     "display": "Procedure"
13                 }
14             ]
15         }
16     ],
17     "code": {
18         "coding": [
19             {
20                 "system": "??",
21                 "code": "22664",
22                 "display": "MDC_EEG_SCORE_SLEEPSTG"
23             }
24         ]
25     },
26     "subject": {
27         "reference": "Patient/xyz",
28         "display": "xyz"
29     },
30     "effectiveDateTime": "xyz",

```

```

31     "performer": [
32         {
33             "reference": "Practitioner/abz",
34             "display": "abc"
35         }
36     ],
37     "valueSampledData": {
38         "origin": {
39             "value": 0
40         },
41         "period": 30,
42         "factor": 1,
43         "lowerLimit": 0,
44         "upperLimit": 9,
45         "dimensions": 1,
46         "data": "None"
47     }
48 }

```

4.7 Ausgabeformat

Bei der Erstellung der neuen konvertierten Datei findet eine Extraktion der relevanten Parameter aus der Ursprungsdatei statt. Hierbei wird das aus der zuvor ausgeführten Funktion zur Extraktion der Datensätze den neuen zugeordnet. Der Datensatz der Schlafstadien wird dem Unterpunkt valueSampledData - Data zugeordnet.

```

1 # Create the new converted json
2 def functionjson(stadien, period):
3     # Read in template for hlhir format
4
5     with open('template.json', 'r') as json_file:
6         hlhir = json.load(json_file)
7         hlhir['valueSampledData']['data'] = stadien
8         hlhir['effectiveDateTime'] = "test"
9         hlhir['valueSampledData']['period'] = period
10
11     with open("Ausgabedateien/" + outputfile, "w") as
12         write_file:
13         json.dump(hlhir, write_file, indent=2)

```

5 Zusammenfassung und Ausblick

Die Annäherung an den FHIR Standard ist lediglich ein prototypischer Versuch. Die Definition dieses Standards war nicht Bestandteil dieser Arbeit.

Der Konverter kann derzeit exemplarisch vier unterschiedliche Dateiformate konvertieren. Es erscheint sinnvoll diesen noch weiter auszubauen. Hierbei sollte die Möglichkeit mit einbezogen werden, dass die verschiedenen Herkunftsgeräte unterschiedliche Exportfunktionen anbieten. So kann das Hypnogramm selben Ursprungs in verschiedenen Formaten vorliegen. Dieser Fall wird derzeit noch nicht abgedeckt. Des Weiteren sollten Binärformate ergänzt werden, um den gesamten Datenpool an Hypnogrammen konvertieren zu können. Sollte sich im Laufe der Zeit ein Standard für Hypnogramme ergeben, sollte der Konverter in seiner Zielformatierung angepasst werden. Möglich wäre auch eine Möglichkeit, statt einzelner Hypnogramme, ganze Ordner als Datenmenge zu übergeben. Dieses würde die Effizienz des Konverters deutlich steigern.

References

- [1] <http://hl7.org/fhir>. Index - fhir v4.0.1, abgerufen 8/27/2020.
- [2] <http://hl7.org/fhir/observation>. Observation - fhir v4.0.1, abgerufen 8/27/2020.
- [3] <https://docs.python.org/3/library/xml.etree.elementtree.html>. xml.etree.elementtree — the elementtree xml api — python 3.8.5 documentation, abgerufen 8/27/2020.
- [4] <https://loinc.org/>. Home – loinc, abgerufen 8/27/2020.
- [5] <https://pypi.org/project/edfrd>. edfrd, abgerufen 8/27/2020.
- [6] <http://www.gbe-bund.de/>. Schlafstörungen ++ Gesundh.-Berichterstattung des Bundes, Themen-Heft 27., abgerufen 8/27/2020.
- [7] C. Iber, S. Ancoli-Israel, A. L. Jr. Chesson, and S. F. Quan. The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications. 1st edition. Technical report, American Academy of Sleep Medicine., 2007.
- [8] Allan Rechtschaffen and Anthony Kales. *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects*. Publication (National Institutes of Health (U.S.)) ; no. 204,. U. S. National Institute of Neurological Diseases and Blindness, Neurological Information Network, Bethesda, Md, 1968. Brain Information Service/Brain Research Institute, Los Angeles.

- [9] Martin Winterholler. Beurteilung der Schlafstadien und Auswertung einer Polysomnographie nach den Kriterien der American Academy of Sleep Medicine. *Das Neurophysiologie-Labor*, 38(3):110–116, 2016.