

## 1. ÜBER DIESES DOKUMENT

Dieses Dokument ist *work in progress* und dient vorerst nur der Sammlung diverser (aus dem Quellcode von bwview extrahierter) Formeln.

## 2. ANALYSIS.C

Die Datei *analysis.c* enthält die Funktionen zur Analyse der EEG-Daten.

**2.1. Vorbemerkung: Analysis types.** Das Tool bietet drei verschiedene Analyse-Typen:

- 0 : Blackman-Fenster (default)
- 1 : IIR Biquad Filter mit  $Q = 0,5$
- 2 : IIR Biquad Filter mit  $Q = 0,72$

Im folgenden wird vorerst nur die Analyse mit Analyse-Typ 0 (Blackman-Fenster) betrachtet, da wir in der Anwendung bisher auch nur diese genutzt haben.

**2.2. struct BWSetup.** Ein Struct mit den grundlegenden Parametern:

```
1 typedef struct BWSetup BWSetup;
2
3 struct BWSetup {
4     int typ;           // Analyse-Typ (0-2, s.o.)
5     int off;           // Offset im input file (in samples, gezaehlt ab 0)
6     int chan;          // anzuzeigender Kanal (gezaehlt ab 0)
7     int tbase;         // Time-base (Samples pro Datenpunkt (horizontal))
8     int sx;            // Zahl der zu berechnenden Spalten (size-X)
9     int sy;            // Zahl der zu berechnenden Spalten (size-Y)
10    double freq0, freq1; // oberste und unterste Frequenz (Achtung: s.u.)
11    double wwrat;       // Verhaeltnis von Fensterbreite zur Wellenlaenge der
                        // Mittenfrequenz
12 };
```

**2.3. Oberste und unterste Frequenz.** Erläuterung zu den beiden Frequenzen *freq0* und *freq1* aus dem struct *BWSetup* (s.o.): Auf die Zeilen (Anzahl: *BWSetup.sy*) werden äquidistante Frequenzbänder (*noch zu übersetzen: in log-freq-space*) zwischen *freq0* und *freq1* verteilt. Das bedeutet, dass das oberste Band knapp unter *freq0* und das unterste Band knapp über *freq1* liegt. Um also z.B. sechs Bänder zwischen 128 Hz und 256 Hz zu erhalten, setzt man *BWSetup.sy* = 6, *BWSetup.freq0* = 128, *BWSetup.freq1* = 256.

**Achtung(!):** obiges ist eine Übersetzung der Kommentare, die etwas widersprüchlich sind, was *freq0* und *freq1* angeht (und welches von beiden die obere und welches die untere Frequenz ist). **Nach allem, was bisher vom Code gesichtet wurde, ist *freq0* die untere und *freq1* die obere Frequenz** (das Beispiel stimmt also, aber der Satz davor nicht).

**2.4. Makro *PLAN\_SIZE(n)*.** Ein Makro zur Berechnung, wie groß das Array für einen *fftw-plan* sein muss.

*n* mod 3 bestimmt den Typen des Plans:

- 0 : real  $\mapsto$  komplex
- 1 : komplex  $\mapsto$  real
- 2 : komplex  $\mapsto$  komplex

Das Makro berechnet:

$$PLAN\_SIZE(n) = \begin{cases} 3 \cdot 2^{\lfloor \frac{n}{6} \rfloor}, & \text{wenn } n \bmod 3 \text{ ungerade.} \\ 2 \cdot 2^{\lfloor \frac{n}{6} \rfloor}, & \text{wenn } n \bmod 3 \text{ gerade.} \end{cases}$$

**2.5. static void copy\_samples(BWAnal \*aa, fftw\_real \*arr, int off, int chan, int len, int errors).** Eine Funktion, die den mit *off* und *len* ausgewählten Bereich der EEG-Daten eines Kanals (*chan*) einem *BWBlock* (struct definiert in *file.c*, Details sind noch to do) in ein Array kopiert. Geht der gewählte Bereich über die Grenzen der Datei hinaus, werden entsprechend Nullen in das Array eingetragen.

*Anmerkung: Code ist nicht ins Detail überprüft, die Ausführungen beruhen nur auf den Kommentaren und einem groben Blick auf den Code.*

2.6. **static void recreate\_arrays(BWAnal \*aa).** Führt erst einmal *free* auf alle Arrays des BWAnal-Structs aus, um sie dann anhand der im Struct eingetragenen Zeilen- und Spaltengrößen (*aa->c.sx* und *aa->c.sy*) neu zu allokkieren.

2.7. **BWAnal \* bwanal\_new(char \*fmt, char \*fnam).** Erzeugt einen neuen BWAnal-Struct für die Datei *fnam*, die mit den in *fmt* übergebenen Formatspezifikationen geladen wird. Dabei wird (neben einigen Default-Parametern, die später wieder überschrieben werden) folgender Parameter gesetzt:

Block size *bsiz* = 1024

2.8. **void bwanal\_start(BWAnal \*aa).** Die Funktion *bwanal\_start(BWAnal \*aa)* Startet die Berechnungen.

*Vorbemerkung:* Von den drei

*sx* := Anzahl Spalten

*sy* := Anzahl Zeilen

*tbase* := Samples pro Datenpunkt/Pixel

*wwrat* := Ratio of window width to the centre-frequency wavelength

$\log 0 = \ln(f_{MAX})$

$\log 1 = \ln(f_{MIN})$

**Für jede Zeile  $a < sy$ :**

Mittenfrequenz:  $f_a = e^{\log 0 + \frac{a+0.5}{sy}(\log 1 - \log 0)}$

(Logische) Fenstergröße in Samples:  $wwid_a = \frac{f_{SAMPLE}}{f_a} wwrat$

$siz = sx \cdot tbase + \lfloor wwid_a \rfloor + 2 + 10$

$b = \lfloor \log_2(siz) \rfloor \cdot 6$

**Solange  $PLAN\_SIZE(b) > siz$ :** {  $b = b - 1$  }

Index auf zu benutzenden fft Plan:  $ftp_a = b$

Dazugehörige Fenstergröße:  $awwid_a = PLAN\_SIZE(b)$

Maximale Plangröße:  $maxsiz = PLAN\_SIZE(b)$ ;

}

*to be continued...*