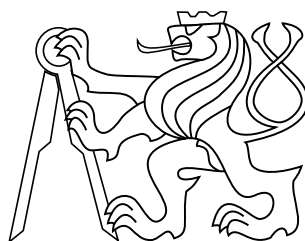


Bakalářská práce

Modulární systém pro zpracování PSG záznamů

Petr Štěpánek



Květen 2013

Vedoucí práce : Ing. Václav Gerla, Ph.D.

České vysoké učení technické v Praze
Fakulta elektrotechnická, Katedra kybernetiky

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Petr Štěpánek**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Modulární systém pro zpracování PSG záznamů**

Pokyny pro vypracování:

1. Upravte PSGLab MATLAB toolbox [1], který umožňuje dávkové zpracování polysomnografických (PSG) záznamů, tak, aby umožňoval spouštění vybraných metod (načítání EEG/PSG záznamu, segmentace signálu, extrakce popisných parametrů a klasifikace) s využitím přehledného GUI.
2. Navržený systém by měl být modulární a měl by umožňovat snadné přidávání dalších modulů bez nutnosti měnit hlavní kód.
3. Dále navrhnete vhodnou datovou strukturu pro uchovávání nastavení jednotlivých metod a vybraných mezivýsledků, s možností snadného exportu.
4. Ověřte realizovaný systém nad reálnými EEG/PSG daty.

Nepovinná část zadání: implementace vybraných metod předzpracování signálu (např. 50/60 Hz notch filtr, nebo převzorkování PSG signálů na jinou vzorkovací frekvenci).

Seznam odborné literatury:


- [1] V. Gerla, V. Djordjevic, L. Lhotska, V. Krajca. PSGLab Matlab Toolbox for Polysomnographic Data Processing: Development and Practical Application, Proceedings of 10th International Conference on Information Technology and Applications in Biomedicine, IEEE, 2010. Online: <http://bio.felk.cvut.cz/psglab/>
- [2] V. Gerla. Automated Analysis of Long-Term EEG Signals. PhD thesis, The Czech Technical University in Prague, 2012.
- [3] E. Niedermeyer, F. Lopes da Silva. Electroencephalography – Basic principles, clinical applications and related field. Philadelphia: Lippincott William & Wilkins, S. 1165-97, ISBN 0-7817-5126-8, 2005.

Vedoucí: Ing. Václav Gerla, Ph.D.

Platnost zadání: do konce zimního semestru 2013/2014


prof. Ing. Michael Sebek, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 2. 1. 2013

Poděkování

Děkuji svému vedoucímu bakalářské práce Ing. Václavovi Gerlovi, Ph.D. za poskytnutí cenných rad k dané problematice. Také bych chtěl poděkovat své rodině za toleranci v posledních několika měsících.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne 23. 5. 2013

.....
Podpis

Abstrakt

Práce se zabývá návrhem modulárního systému pro zpracování biomedicínských signálů. Cílem je vytvořit grafickou nadstavbu nad PSGLab Matlab toolboxem, která zjednoduší zpracování multikanálových záznamů (EEG, PSG, apod.) a umožní uživateli efektivně nastavit parametry jednotlivých metod. Realizovaná aplikace obsahuje tyto hlavní části: modul pro načtení multikanálového signálu, segmentace signálu, výpočet popisných parametrů a klasifikace. Součástí řešení je také návrh vhodné datové struktury pro uchovávání nastavení jednotlivých metod a vybraných mezivýsledků. Na konci práce jsou možnosti realizovaného GUI demonstrovány na reálných datech.

Klíčová slova

EEG, GUI, MATLAB toolbox, zpracování signálů

Abstrakt

The focus of this work is to design modular system for biomedical signal processing. Aim is to create graphical user interface for PSGLab Matlab toolbox to ease processing of multichannel records (EEG, PSG, etc.) and to allow user to effectively set parametres of individual methods. Realized solution consists of these main parts: modul for multichannel signal loading, signal segmentation, evaluation of features and classification. Simultaneously data structure for storing method setting and results was proposed. At the end of thesis functionality of realized GUI is demonstrated on set of real data.

Keywords

EEG, GUI, MATLAB toolbox, signal processing

Obsah

1. Úvod	1
2. Polygrafické signály	2
2.1. EEG	2
2.2. Standardní zapojení elektrod	3
3. Zpracování signálů	4
3.1. Předzpracování signálů	4
3.2. Segmentace	4
3.3. Výpočet příznaků	4
3.4. Klasifikace	5
3.5. Přehled existujících řešení	5
4. PSGLab	6
4.1. Stěžejní vlastnosti toolboxu	6
5. Blokové schéma navržené aplikace	7
5.1. Adresářová struktura navržené aplikace	7
5.2. Datové toky uvnitř GUI	9
6. Datové struktury	11
6.1. Předávání parametrů	11
6.2. Načítání nastavení funkcí	12
6.2.1. Segmentace	13
6.2.2. Výpočet příznaků	14
6.2.3. Klasifikace	15
6.2.4. Speciální případ funkcí v nulovém počtem parametrů	16
6.3. Obnovení nastavení GUI z předchozího chodu	17
6.4. Uložení výsledků jednotlivých částí procesu	19
6.4.1. Segmentace	19
6.4.2. Výpočet příznaků	19
6.4.3. Klasifikace	20
7. Jednotlivé části GUI	21
7.1. Hlavní část aplikace	21
7.1.1. Načtení souboru s uloženými daty	21
7.1.2. Popis komponent GUI	21

7.1.3.	Důležité funkce	23
7.1.4.	Nastavení skupin kanálů	24
7.2.	Segmentace	25
7.2.1.	Popis komponent GUI	25
7.2.2.	Důležité funkce	26
7.2.3.	Nastavení segmentačních funkcí	27
7.3.	Výpočet příznaků	27
7.3.1.	Popis komponent GUI	28
7.3.2.	Důležité funkce	29
7.4.	Klasifikace	30
7.4.1.	Popis komponent GUI	30
7.4.2.	Důležité funkce	31
8.	Ověření nad reálnými daty	33
8.1.	Krátkodobý záznam	33
8.2.	Střednědobý záznam	36
8.3.	Dlouhodobý záznam	38
9.	Možná rozšíření	41
9.1.	Podpora vstupu různých datových formátů	41
9.2.	Podpora různých vzorkovacích frekvencí kanálů	41
9.3.	Grafická uživatelská rozhraní pro export dat	41
9.4.	Modul pro zpracování a definici tagů	42
9.5.	Přidávání funkcí za běhu aplikace	42
9.6.	Vizualizace výsledků klasifikace	42
9.7.	Historie operací v GUI	42
9.8.	Modul pro předzpracování signálů	42
10.	Závěr	43
	Přílohy	
A.	Zobrazení načteného záznamu	44
A.1.	Původní verze modulu	44
A.2.	Stávající verze modulu	44
	Literatura	46

Použité zkratky

CNS	Centrální nervová soustava
EKG	Elektrokardiogram
EEG	Elektroencefalogram
EMG	Elektromyogram
EOG	Elektrookulogram
GUI	Grafické uživatelské rozhraní
KNN	K-nejbližší soused
PNG	Pneumogram
PSG	Polysomnografie

1. Úvod

Ruční zpracování PSG záznamů je časově náročná úloha. K usnadnění tohoto procesu vznikl na Katedře kybernetiky PSGLab toolbox, který obsahuje množinu funkcí k předzpracování, segmentaci, výpočtu příznaků a klasifikaci dat. Bez detailní znalosti celého toolboxu a způsobu nastavení jednotlivých metod, je práce s ním pro běžného uživatele velice obtížná. Cílem této práce je vytvořit grafickou nadstavbu nad tímto toolboxem a usnadnit tak zpracování PSG záznamů.

Důraz je kladen na vytvoření modulárního systému, který uživateli poskytne nástroj ke zpracování biomedicínských signálů, ať už za pomoci funkcí PSGLab Matlab toolboxu nebo funkcí vlastních. Uživatelské funkce je nejprve nutné upravit do speciálního formátu. Pro tyto účely byla navržena základní struktura funkcí, kterou je nutné dodržet. K uložení nastavení funkcí, ke komunikace mezi hlavními částmi aplikace a k uložení mezivýsledků byla navržena celá řada datových struktur.

Navržená aplikace byla rozdělena do několika separátních částí. Hlavní část aplikace spravuje veškerá data, umožňuje načtení vybraného záznamu, změnu skupin kanálů načteného záznamu, export vybraných mezivýsledků a volání zbylých částí. Části pro segmentaci, výpočet popisných příznaků, klasifikaci a do jisté míry i část pro změnu skupin kanálů, byly navrženy tak, aby jejich ovládání bylo intuitivní a co možná nejvíce navzájem podobné.

Následující dvě kapitoly se obecně zabývají popisem polygrafických kanálů a způsoby jejich zpracování. Čtvrtá kapitola uzavírá teoretickou část a poskytuje náhled do struktury a funkčnosti PSGLab toolboxu. V kapitole číslo pět je uvedeno blokové schéma výsledné aplikace. Kapitoly šest a sedm detailně popisují jednotlivé prvky GUI a použité datové struktury. V osmé kapitole je ověřena funkce aplikace nad reálnými daty. V závěrečných dvou kapitolách jsou uvedeny návrhy na možné rozšíření aplikace a shrnutí celé práce.

2. Polygrafické signály

Jako biosignály můžeme označit veškeré signály, jejichž existenci můžeme zaznamenat v živých organismech. Může se jednat o průběhy elektrických napětí, proměnlivá magnetická pole, změny chemických koncentrací, mechanické pohyby, zvuky, změny teplot aj. Můžeme je registrovat v důsledku spontánní aktivity biologického systému (nativní signály) anebo jako důsledek nějakých úmyslných podnětů (evokované signály, provokace apod.), podle (Heřman, 1999) [2]

Termínem polygrafie je označován simultánní záznam biosignálů různé povahy. Mezi PSG signály se řadí EEG (záznam mozkových bioelektrických potenciálů z povrchu hlavy), EKG (záznam srdeční činnosti), EMG (záznam svalové činnosti), EOG (snímání pohybu očí), PNG (záznam plicní činnosti).

2.1. EEG

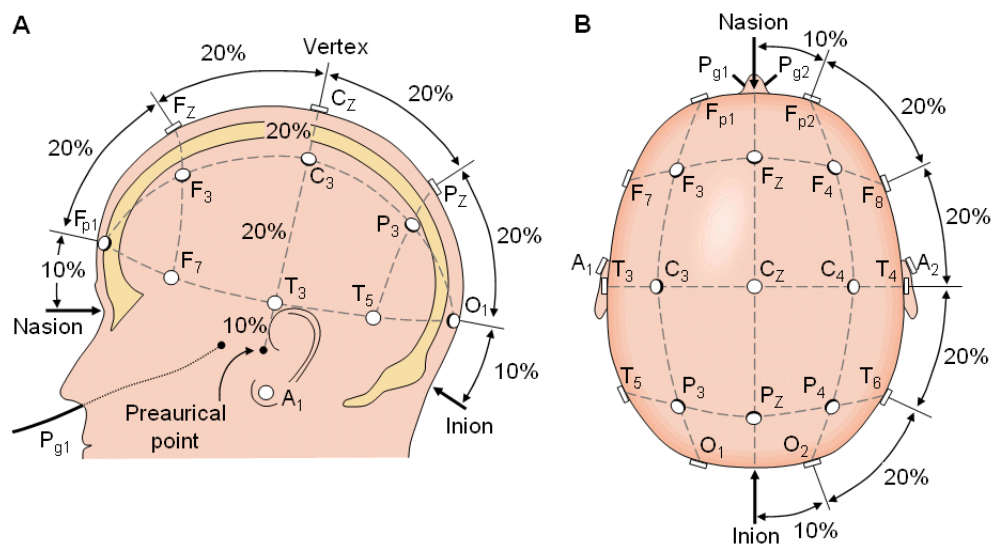
EEG je standardní neinvazivní metoda funkčního vyšetření elektrické aktivity centrální nervové soustavy. Sumační signály z neuronů jsou snímány elektrodami z povrchu hlavy. Problémem je, že průchodem tkáněmi a lebečními kostmi je amplituda signálu zeslabena na úroveň řádově desítek mikrovolt. Vzhledem k tomu, že EEG signál vzniká jako důsledek vážené sumace aktivity extrémně vysokého množství neuronů, nejsme již v EEG signálu schopni odlišit jednotlivé akční potenciály buněk tak, jako např. v EMG. Nejznámější je aktivita alfa s frekvencí přibližně 12 Hz , kterou sledujeme u dospělých v okcipitální oblasti hlavy při zavřených očích. Pomalejší frekvence (theta a delta) mohou být v bdělém stavu u dospělých patologickým příznakem. Během spánku jsou naproti tomu identifikátorem různých spánkových stadií, čehož se využívá ve spánkových laboratořích; u dětí mohou být tyto frekvence měřítkem vyzrálosti CNS, podle (Heřman, 1999) [2]

Tato práce se věnuje primárně EEG záznamům, neboť tato oblast není doposud efektivně zpracována. To je dáno jednak tím, že se měří velké množství kanálů (32, 64, 128, apod.) a dále proto, že se v praxi často měří i mnohahodinové záznamy. Hodnocení takových signálů je velmi komplexní úloha. Hodnocení EEG

záznamů v lékařské praxi se věnují například publikace [12] [14].

2.2. Standardní zapojení elektrod

Standardně je používáno takzvané "10/20" zapojení [13], které je zobrazeno na Obrázku 1. Toto označení vychází z běžně označovaných oblastí: F – frontální (čelní), Fp – frontopolární, C – centrální (střední), P – parietální (temenní), O – okcipitopolární (týlní, zátylní), T – temporální (spánkový), podle (Rieger, 2004) [3]



Obrázek 1. Standard "10/20" rozmístění elektrod pro měření EEG.

3. Zpracování signálů

Zpracováním signálů můžeme označit proces, při němž na zkoumaný záznam aplikujeme množinu operací, za účelem získat pro nás důležitá data charakterizující jeho vlastnosti. Ke zpracování obecných PSG záznamů byl navržen proces skládající se z předzpracování signálu, výpočtu popisných parametrů, klasifikace a vizualizace. Pro účely této bakalářské práce, byla segmentace vyčleněna z předzpracování signálů. Více informací k dané tématice je uvedeno v [1].

3.1. Předzpracování signálů

Účelem předzpracování je odstranit ze signálu šum a připravit ho ke zpracování. Šum je nutné z naměřeného záznamu odstranit, jinak by mohl ovlivnit samotnou analýzu signálu. Nejpoužívanějšími metodami jsou: filtrace, převzorkování, odstranění posouvající se základny, detekce artefaktů a průměrující metody.

3.2. Segmentace

Obecný PSG signál, který je nestacionární, je rozdělen do přibližně stacionárních úseků, nebo-li segmentů. Signál lze rozdělit do stejných nebo do vzájemně se lišících úseků. K tomu jsou využívány, mimo jiné, metody konstantní a adaptivní segmentace. Metody konstantní segmentace [19] rozdělí signál do úseků stejné délky, kdežto metody adaptivní [20] [21] signál rozdělí do segmentů na základě jejich lokálního chování.

Metody segmentace použité v práci byly převzaty z PSGLab toolboxu. Jejich detailní popis je možné nalézt v [1].

3.3. Výpočet příznaků

Po přípravě signálu dvěma předchozími kroky, je možné segmentům signálu vypočítat vybrané popisné příznaky, podle kterých budou rozřazeny do tříd. Počítané

příznaky mohou být statistické vlastnosti segmentů, jejich frekvenční spektra nebo jejich vzájemná podobnost.

Před přistoupením ke klasifikaci může být z celkové množiny příznaků, vybrána podmnožina příznaků, nějakým způsobem nejlepších vzhledem ke klasifikaci.

Metody výpočtu příznaků použité v práci byly převzaty z PSGLab toolboxu. Jejich detailní popis je možné nalézt v [1]. Další metody extrakce příznaků jsou popsány v [15] [16] [17] [18].

3.4. Klasifikace

Klasifikace je proces, při kterém jsou segmenty rozděleny do tříd podle jejich vlastností, charakterizovaných vektorem vypočítaných příznaků. Segmenty je možné klasifikovat metodami s učitelem nebo bez učitele. Při klasifikaci s učitelem je klasifikátor nejprve naučen na trénovací množině a posléze aplikován na zkoumaná data. Mezi tyto klasifikátory patří například KNN, přirozený Bayesův klasifikátor nebo neuronové sítě. Klasifikace bez učitele využívá pouze informací obsažených v datech. Do této skupiny patří například hierarchické shlukování nebo k-means shlukování.

3.5. Přehled existujících řešení

V závěru této kapitoly je uveden orientační seznam již existujících řešení zabývajících se zkoumanou problematikou:

- *PSGLab*, jeho grafické nadstavbě je věnována tato práce. [4]
- *EEGLAB*. [6]
- *MEET*. [7]
- *EEGVIS*. [8]
- *LORETA*. [9]
- *ASA Lab*. [10]
- *BioSig*. [11]

4. PSGLab

PSGLab je Matlab toolbox zaměřený na zpracování biomedicínských signálů. Poskytuje celou řadu funkcí pro předzpracování signálů, výpočet a výběr příznaků, klasifikaci s učitelem a bez učitele, a vizualizaci dat. S využitím těchto funkcí jsou z dat extrahovány a následně interpretovány důležité informace. PSGLab umožňuje hromadné zpracování záznamů v případě, že je nutné zpracovat velké objemy dat.

4.1. Stěžejní vlastnosti toolboxu

Stručný obsah nejdůležitějších prvků toolboxu:

- Import/Export dat v různých formátech (EaSys, ASCII, formáty používané v EEGLab a další binární formáty).
- Předzpracování signálů: filtrace (50Hz notch filter), převzorkování, odstranění artefaktů, změna amplitudy kanálu.
- Segmentace signálů: konstantní a adaptivní.
- Výpočet příznaků: zaměření na výpočet EEG příznaků.
- Výběr příznaků: výběr nejvhodnějších příznaků pro klasifikaci.
- Normalizace příznaků.
- Klasifikace.
- Vizualizační metody.

Podrobnosti o PSGlab toolboxu jsou uvedeny na stránce:

<http://bio.felk.cvut.cz/psglab/>.

5. Blokové schéma navržené aplikace

Navržená aplikace je rozdělena do několika částí, každá z nich má na starosti jinou činnost. Hlavní část aplikace zajišťuje načítání veškerých potřebných dat, konkrétně: soubor určený ke zpracování, nastavení používaných funkcí a uložené nastavení. Zároveň slouží jako prostředník pro komunikaci ostatních částí.

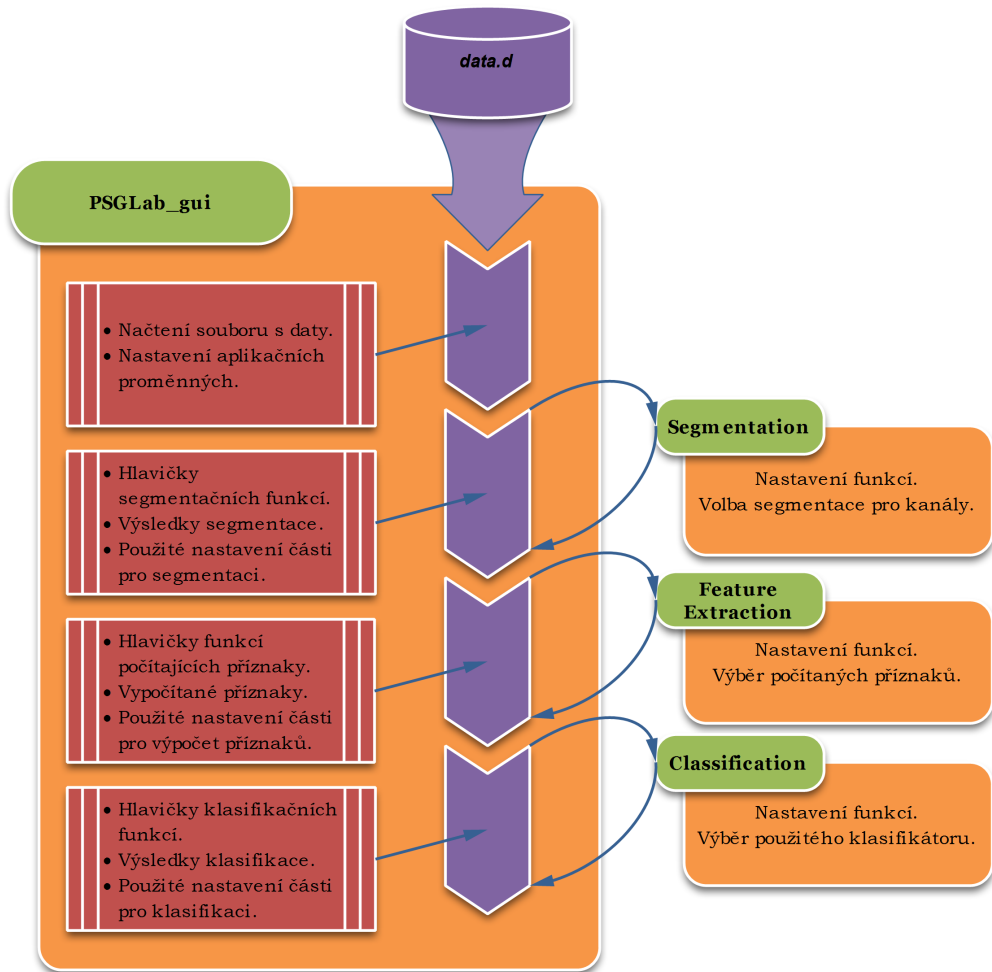
Načtené kanály ze souboru jsou dány k dispozici části pro segmentaci, která data zpracuje a výsledek zašle zpět. Hlavní část aplikace po obdržení výsledku umožní pokračovat dalším krokem. Část pro výpočet příznaků si vyzvedne z hlavní části aplikace výsledek segmentace a další potřebná data. Vypočítané příznaky jsou opět zaslány zpět. Stejným způsobem pak klasifikační část vyzvedne potřebná data, provede vlastní výpočet a zašle výsledky zpět.

Blokové schéma aplikace je znázorněno na Obrázku 2.

5.1. Adresářová struktura navržené aplikace

V této podkapitole je stručně popsána adresářová struktura navržené aplikace a obsah těchto adresářů.

```
/PSGLab_GUI
  /Classification
  /Data_preview
  /Feature_Extraction
  /Functions
    /Classification
    /Data_input
    /easy2matlab
    /Feature_Extraction
    /Segmentation
  /Results
```

Obrázek 2. Blokové schéma reprezentující strukturu GUI.

/Segmentation

Domovský adresář *PSGLab_GUI* obsahuje m-file a figure hlavní části aplikace, *PSGLab_gui*. M-file grafického uživatelského rozhraní pro nastavení skupin kanálů, *PSGLab_gui_channelGroup.m*, a soubor *Setting.mat* s uloženým nastavením, jeho strukturou se zabývá kapitola 6.3.

Adresář *Classification* obsahuje m-file GUI pro klasifikaci :

PSGLab_gui_classification GUI pro klasifikační část, kapitola 7.4.

PSGLab_gui_classification_fncSetting Vedlejší GUI pro nastavení klasifikačních funkcí.

Adresář *Feature_Extraction* obsahuje m-file GUI pro výpočet příznaků:

PSGLab_gui_feature_extraction GUI pro část extrakce příznaků, kapitola 7.3.

PSGLab_gui_feature_extraction_fncSetting Vedlejší GUI pro nastavení funkcí, které počítají příznaky.

Adresář *Results* slouží k ukládání mezivýsledků, pokud se uživatel rozhodne data exportovat. Jméno uloženého souboru se skládá ze jména právě načteného souboru s daty a z identifikátoru části které náleží. Například *soubor1Segmentation.mat* jsou exportovaná data segmentace souboru *soubor1.d*.

Adresář *Segmentation* obsahuje m-file GUI pro segmentaci:

PSGLab_gui_segmentation GUI pro segmentační část, kapitola 7.2.

PSGLab_gui_segmentation_fncSetting Vedlejší GUI pro nastavení segmentačních funkcí.

Adresář *Data_preview* obsahuje m-file GUI pro zobrazení kanálů, podrobněji v příloze A.

Adresář *Functions* obsahuje m-file používaných funkcí :

Classification Úložiště pro segmentační funkce, kapitola 6.2.3.

Data_input/easy2matlab Funkce používané k načtení *.d souborů.

Feature_Extraction Úložiště pro funkce počítající příznaky, kapitola 6.2.2.

Segmentation Úložiště pro segmentační funkce, kapitola 6.2.1.

5.2. Datové toky uvnitř GUI

PSGLab_gui: Při spouštění načte soubor *Setting.mat* a nastavení používaných funkcí. Na podnět uživatele načte soubor určený ke zpracování a nastaví všechny potřebné aplikační proměnné. Přijímá výsledky částí pro segmentaci, výpočet příznaků a klasifikaci. Sleduje zda některá z částí připravila své nastavení k uložení.

PSGLab_gui_segmentation : Od *PSGLab_gui* si vyzvedne potřebná data k inicializaci GUI. Na základě podnětů od uživatele určí hranice segmentů kanálů. Po výpočtení hranic segmentů zašle data do *PSGLab_gui*.

PSGLab_gui_feature_extraction : Od *PSGLab_gui* si vyzvedne potřebná data

5. Blokové schéma navržené aplikace

k inicializaci GUI. Na základě podnětů od uživatele vypočítá požadované příznaky. Po vypočtení příznaků zašle data do *PSGLab_gui*.

PSGLab_gui_classification : Od *PSGLab_gui* si vyzvedne potřebná data k inicializaci GUI. Na základě podnětů od uživatele provede klasifikaci dat. Po dokončení klasifikace zašle data do *PSGLab_gui*.

6. Datové struktury

Aplikace využívá ke komunikaci, ukládání mezivýsledků a k načítání hlaviček funkcí pro segmentaci, výpočet příznaků a klasifikaci, celou řadu datových struktur. Jejich funkce a význam je podrobně vysvětlen v této kapitole.

6.1. Předávání parametrů

Vzhledem k povaze problému, je nutné, aby mohla hlavní část aplikace, *PSGLab_gui*, komunikovat se zbývajícími částmi a naopak. K tomuto účelu jsou využity funkce Matlabu:

getappdata(H, NAME) Z objektu definovaného pomocí handle H vezme aplikací definovaná data se jménem NAME.

setappdata(H, NAME, VALUE) V objektu definovaného pomocí handle H vytvoří aplikací definovaná data se jménem NAME a hodnotou VALUE.

Při spouštění aplikace je zjištěn handle na *PSGLab_gui* a je uložen do Root příkazem:

```
setappdata(0, 'hPSGLab_gui', gcf);
```

Nula specifikuje, že se proměnná uloží do Root, prostřední člen představuje její jméno, podle kterého k ní bude přistupováno a *gcf* je samotný handle na naposledy otevřený figure.

Hlavní část pak následně vyzvedne tuto proměnnou a pomocí ní nastaví veškerá data aplikace, která je potřeba sdílet. Jako příklad je uvedeno sdílení proměnné počtu kanálů:

```
hPSGLab_gui = getappdata(0, 'hPSGLab_gui');  
setappdata(hPSGLab_gui, 'nch', handles.nch);
```

Data uložená v Root jsou viditelná i pro ostatní části aplikace. To se využívá například v případě, pokud část pro segmentaci, požaduje od *PSGLab_gui* data

a nebo naopak, chce-li odeslat výsledek segmentace.

Nejprve je vyzvednut handle na *PSGLab_gui* z Root a následně je pomocí něho možné přistupovat k datům hlavní části aplikace.

```
hPSGLab_gui = getappdata(0, 'hPSGLab_gui');
promenna_segmentace = getappdata(hPSGLab_gui, 'nch');
```

Pokud chce vedlejší část aplikace aktualizovat data hlavní části, použije příkaz:

```
setappdata(hPSGLab_gui, 'nch', 10);
```

Proměnné uložené v Root se při ukončení aplikace automaticky neodstraní. V funkci volané při zavírání *PSGLab_gui* je proto nutné použít příkaz:

```
rmappdata(0, 'hPSGLab_gui');
```

6.2. Načítání nastavení funkcí

Jedním z důležitých cílů návrhu aplikace, bylo sjednocení způsobu nastavení veškerých funkcí, využitých při zpracování signálu. Dalším cílem bylo umožnit uživateli jednoduše přidat nové funkce nebo odebrat stávající, aniž by musel modifikovat zdrojový kód aplikace. Pro tyto účely byl navržen následující způsob komunikace *PSGLab_gui* a používaných funkcí:

1. *PSGLab_gui* nejprve při konstrukci GUI prohledá adresáře, v nichž jsou funkce uloženy.
2. Každá funkce je zavolána bez vstupních parametrů a je tak donucena k vrácení datové struktury obsahující informace o ní.
3. *PSGLab_gui* si tyto informace uloží do odpovídajících struktur a následně je nastaví jako své aplikační proměnné způsobem popsaným v kapitole 6.1.

Popsaná funkcionálnita je zaručena, pokud mají funkce tuto strukturu :

```
function [ output_args ] = jmeno_funkce( input_args )
    switch nargin
        case 0,
            vlastnosti_metody_definovane_strukturou
            ...
            output_args = definice_funkce
        otherwise,
```

```

        telo_funkce
        ...
        output_args = vysledek
    end;
end

```

Vlastní obsah struktur definující funkce pro segmentaci, výpočet příznaků a klasifikaci se vzájemně liší a jejich tvar je popsán v následujících odstavcích.

6.2.1. Segmentace

Aby bylo možné načíst při startu aplikace nastavení segmentačních funkcí, musí být uloženy v adresáři :

```
/hPSGLab_gui/Functions/Segmentation
```

Detailní popis datové struktury pro segmentační funkce:

```

fnc_definition.const = 3;
    const_label.c1 = 'Popis první konstanty';
    const_label.c2 = 'Popis druhé konstanty';
    const_label.c3 = 'Popis třetí konstanty';
fnc_definition.const_name = const_label;
    fnc_params.p1 = 10;
    fnc_params.p2 = 0.2;
    fnc_params.p3 = -5;
fnc_definition.default_setting = fnc_params;
    fnc_param_definition.p1 = [1 1];
    fnc_param_definition.p2 = [1 0];
    fnc_param_definition.p3 = [-1 1];
fnc_definition.const_description = fnc_param_definition;
fnc_definition.data = 0;
fnc_definition.tags = 0;

```

fnc_definition.const Určuje počet konstantních parametrů funkce. V uvedeném příkladu je počet roven třem, tudíž počet proměnných následujících položek musí být také tři.

fnc_definition.const_name V jednotlivých proměnných je popis použitých konstant pro uživatele. Popisky jsou ukládány do proměnných c_i , kde i je index proměnné.

fnc_definition.default_setting Defaultní nastavení proměnných. Hodnoty jsou ukládány do proměnných p_i , kde i je index proměnné.

fnc_definition.const_description Popis konstant pro GUI. Definice konstant jsou ukládány do proměnných p_i , kde i je index proměnné.

1. složka vektoru:

- -1 představuje konstantu menší než nula.
- 0 představuje konstantu menší nebo rovno nule.
- 1 představuje konstantu větší než nula.

2. složka vektoru:

- 0 představuje reálné číslo.
- 1 představuje celé číslo.

fnc_definition.data Hodnota 1 říká, že funkce ke svému chodu potřebuje matici s daty kanálu. Hodnota 0 značí, že data nejsou potřeba.

fnc_definition.tags Hodnota 1 říká, že funkce ke svému chodu potřebuje tagy pro daný kanál. Hodnota 0 značí, že tagy nejsou potřeba.

Výsledkem segmentačních metod je matice o rozměrech $1 \times (2 * Počet_segmentu)$. Jednotlivé prvky matice jsou indexy hranic segmentů v daném kanálu.

6.2.2. Výpočet příznaků

Aby bylo možné načíst při startu aplikace nastavení funkcí pro výpočet příznaků, musí být uloženy v adresáři:

```
/hPSGLab_gui/Functions/Feature_Extraction
```

Detailní popis datové struktury pro funkce hledající příznaky:

```
fnc_definition.const = 2;
    const_label.c1 = 'Popis první konstanty';
    const_label.c2 = 'Popis druhé konstanty';
fnc_definition.const_name = const_label;
    fnc_params.p1 = 15;
    fnc_params.p2 = -0.4;
fnc_definition.default_setting = fnc_params;
    fnc_param_definition.p1 = [1 1];
    fnc_param_definition.p2 = [-1 0];
fnc_definition.const_description = fnc_param_definition;
fnc_definition.fsamp = 0;
```

fnc_definition.const Určuje počet konstantních parametrů funkce. V uvedeném příkladu je počet roven třem, tudíž počet proměnných následujících položek musí být také tři.

fnc_definition.const_name V jednotlivých proměnných je popis použitých konstant pro uživatele. Popisky jsou ukládány do proměnných c_i , kde i je index proměnné.

fnc_definition.default_setting Defaultní nastavení proměnných. Hodnoty jsou ukládány do proměnných p_i , kde i je index proměnné.

fnc_definition.const_description Popis konstant pro GUI. Definice konstant jsou ukládány do proměnných p_i , kde i je index proměnné.

1. složka vektoru:

- -1 představuje konstantu menší než nula.
- 0 představuje konstantu menší nebo rovno nule.
- 1 představuje konstantu větší než nula.

2. složka vektoru:

- 0 představuje reálné číslo.
- 1 představuje celé číslo.

fnc_definition.fsamp Hodnota 1 říká, že funkce ke svému chodu potřebuje frekvenci, s níž je daný kanál vzorkován. Hodnota 0 značí, že vzorkovací frekvence není potřeba.

Výsledky funkcí počítající příznaky jsou ve formátu:

```
result.featName = 'Název příznaku';  
result.varName = {'Jméno proměnné 1', 'Jméno proměnné 2'};  
result.data = Vlastní vypočtená data funkce;
```

result.FeatName Položka se jménem počítaného příznaku.

result.varName Seznam jmen jednotlivých výsledků funkce.

result.data Vypočtené příznaky segmentu ve formě matice o rozměrech $1 \times \text{Počet_výsledků}$.

6.2.3. Klasifikace

Aby bylo možné načíst při startu aplikace nastavení klasifikačních funkcí, musí být uloženy v adresáři:

```
/hPSGLab_gui/Functions/Classification
```


Detailní popis datové struktury pro klasifikační funkce :

```
fnc_definition.const = 1;
    const_label.c1 = 'Popis první konstanty';
fnc_definition.const_name = const_label;
    fnc_params.p1 = 2;
fnc_definition.default_setting = fnc_params;
    fnc_param_definition.p1 = [1 1];
fnc_definition.const_description = fnc_param_definition;
fnc_definition.supervised = 0;
```

fnc_definition.const Určuje počet konstantních parametrů funkce. V uvedeném příkladu je počet roven třem, tudíž počet proměnných následujících položek musí být také tři.

fnc_definition.const_name V jednotlivých proměnných je popis použitých konstant pro uživatele. Popisky jsou ukládány do proměnných c_i , kde i je index proměnné.

fnc_definition.default_setting Defaultní nastavení proměnných. Hodnoty jsou ukládány do proměnných p_i , kde i je index proměnné.

fnc_definition.const_description Popis konstant pro GUI. Definice konstant jsou ukládány do proměnných p_i , kde i je index proměnné.

1. složka vektoru:

- -1 představuje konstantu menší než nula.
- 0 představuje konstantu menší nebo rovno nule.
- 1 představuje konstantu větší než nula.

2. složka vektoru:

- 0 představuje reálné číslo.
- 1 představuje celé číslo.

fnc_definition.supervised Hodnota 1 říká, že je jedná o klasifikaci s učitelem, hodnota 0 značí funkce bez učitele.

Výsledkem klasifikace je matice o rozměrech $1 \times \text{Počet_klasifikovaných_segmentů}$. Jednotlivé prvky matice jsou pořadová čísla tříd, do kterých byl segment zařazen.

6.2.4. Speciální případ funkcí v nulovém počtem parametrů

V případě, že funkce nepožaduje od uživatele nastavení nějakého svého konstantního parametru, definice funkce je v následujícím tvaru :

```
fnc_definition.const = 0;
fnc_definition.const_name = 0;
fnc_definition.default_setting = 0;
fnc_definition.const_description = 0;
```

Význam položek je popsán například v kapitole 6.2.1.

6.3. Obnovení nastavení GUI z předchozího chodu

K urychlení zpracování množiny souborů s podobnými daty, GUI automaticky načítá stav jednotlivých komponent z předchozího chodu. K načtení nastavení dojde pouze pokud jsou splněny podmínky:

- Počet kanálů právě načteného souboru s PSG daty je shodný s počtem kanálů minulého souboru.
- Ve složce *PSGLab_gui/Functions/Segmentation* je stejný soubor funkcí jako při zpracování minulého souboru.
- Ve složce *PSGLab_gui/Functions/Feature_Extraction* je stejný soubor funkcí jako při zpracování minulého souboru.
- Ve složce *PSGLab_gui/Functions/Classification* je stejný soubor funkcí jako při zpracování minulého souboru.

Soubor *Setting.mat* s nastavením je uložen v domovském adresáři *PSGLab_gui*. V souboru je uloženo nastavení pouze z předchozího běhu. GUI neumožňuje uložit současně několik nastavení a později si po spuštění vybrat, kterou z těchto šablon použít.

Pro uchování všech potřebných dat byla navržena struktura:

```
dataPreviewFlag: 0
preprocessingFlag: 0
segmentationFlag: 1
featureExtractionFlag: 1
classificationFlag: 1
dataIn: 1
nch: 21
favouriteDir: 'C: ...\MATLAB\PSGLab_GUI\Test_data\'
chGroup: [1 2 3 ... 19 20 21]
mainGuiPosition: [103.8000 15.6923 138.2000 45.9231]
```

```

segmentationSetting: [1x1 struct]
featureExtractionSetting: [1x1 struct]
classificationSetting: [1x1 struct]

```

dataPreviewFlag Logická proměnná, určuje existenci uloženého nastavení části pro vizualizaci signálů.

preprocessingFlag Logická proměnná, určuje existenci uloženého nastavení části pro předzpracování signálů.

segmentationFlag Logická proměnná, určuje existenci uloženého nastavení části pro segmentaci signálů.

featureExtractionFlag Logická proměnná, určuje existenci uloženého nastavení části pro výpočet příznaků signálů.

classificationFlag Logická proměnná, určuje existenci uloženého nastavení části pro klasifikaci signálů.

dataIn Logická proměnná, určuje zda byl v minulém chodu otevřen nějaký soubor s daty a tedy zda existuje následující položka.

nch Počet kanálů naposledy otevřeného souboru s daty.

favouriteDir Absolutní cesta k naposledy otevřenému souboru. Při otvírání nových dat, je započato prohledávání právě v tomto adresáři.

chGroup Matice definující použité skupiny kanálů.

mainGuiPosition Poloha a rozměry hlavního okna GUI při zavření.

Dále následují, pokud existují, struktury s nastavením jednotlivých částí. Struktury pro segmentaci, výpočet příznaků a klasifikaci jsou si velice podobné. V následujícím příkladu je popsána struktura pro segmentaci:

```

guiPosition: [100 100 1355 200]
userSetting: [1x4 struct]
segMap: [4x21 double]

```

guiPosition Pozice a velikost části pro segmentaci.

userSetting Uživatelské nastavení parametrů metod, pokud uživatel ponechal původní nastavení, je shodné s nastavením načteným z segmentačních funkcí.

segMap Matice o rozměrech *Počet_segmentačních_funkcí* × *Počet_kanálů*. Uchovává informaci o tom, které segmentační funkce byly v minulém chodu použity pro příslušné kanály.

Struktura pro výpočet příznaků, respektive klasifikaci, se od uvedené struktury pro segmentaci liší pouze v poslední položce. Tato položka je nahrazena *featMap*, respektive *classMap*.

6.4. Uložení výsledků jednotlivých částí procesu

Budeme-li brát načtení dat ze souboru jako jednu část celého procesu, tak zmiňme, že načtená data, jsou uložena do matice $Počet_kanálů \times Počet_vzorků_na_kanál$, a následně jako ostatní důležitá data jsou pak uložena do struktury *handles*, kterou grafická uživatelská rozhraní používají k uchovávání všech proměnných.

6.4.1. Segmentace

Výsledky segmentace jsou uloženy v následujícím formátu:

```
segmentationResult.ch1 = Hranice_segmentů_prvního_kanálu;
segmentationResult.ch2 = Hranice_segmentů_druhého_kanálu;
...
segmentationResult.chN = Hranice_segmentů_N-tého_kanálu;
```

Konstanta N je rovna počtu kanálů. Grafické uživatelské rozhraní zaručuje, že veškeré členské proměnné ch_i struktury *segmentationResult*, obsahují nějakou segmentaci příslušných kanálů.

6.4.2. Výpočet příznaků

K uchování vypočtených příznaků byla navržena struktura:

```
featExtResult(N).featureM.segmentK = Data_příznaku_pro_segment;
featExtResult(N).featureM.featName = Jméno_počítaného_příznaku;
featExtResult(N).featureM.varName = Jména_jednotlivých_proměnných;
featExtResult(N+1).featMap = Počítané_příznaky_pro_kanály.;
```

Konstanta N odpovídá počtu kanálů, konstanta M počtu funkcí počítajících příznaky a konstanta K počtu segmentů příslušného kanálu. Pro lepší ilustraci je uveden následující příklad:

```
featExtResult(1)
```

jsou vypočítané příznaky pro první kanál.

```
feature1: [1x1 struct]
feature2: []
feature3: []
feature4: []
feature6: [1x1 struct]
```

Tedy pro první kanál byly vypočítány příznaky jedna a šest. Poznamenejme, že pro každý kanál musí být vypočítán alespoň jeden příznak a pro kanály ve stejné skupině, musí být vypočítány stejné příznaky. O kontrolu uvedených podmínek se automaticky stará GUI.

```
segment1: 3.4088
segment2: 11.0562
segment3: 9.4470
segment4: 12.5233
featName: 'FeatExt_std'
  varName: {'std'}
```

Ve struktuře prvního příznaku jsou data funkce pro jednotlivé segmenty kanálu, jméno počítaného příznaku a název proměnné. Víme tedy, že funkce *FeatExt_std* vypočítala jednu hodnotu pro každý segment a že se jedná o směrodatnou odchylku.

6.4.3. Klasifikace

Výsledky klasifikace jsou uloženy ve struktuře:

```
classificationResult.ch1 = Třídý_segmentů_prvního_kanálu;
classificationResult.ch2 = Třídý_segmentů_druhého_kanálu;
...
classificationResult.chN = Třídý_segmentů_N-tého_kanálu;
```

Konstanta N je opět rovna počtu kanálů. Každý kanál musí mít před začátkem klasifikace přiřazenou klasifikační funkci. Kanály ve stejné skupině jsou klasifikovány společně za použití stejné klasifikační funkce.

7. Jednotlivé části GUI

Navržená aplikace je rozdělena do několika částí. Kapitola 7 je zaměřena na popis nejdůležitějších funkcí těchto částí a na vysvětlení významu jednotlivých komponent.

7.1. Hlavní část aplikace

O načtení dat a obecně o správu veškerých datových struktur se stará hlavní část aplikace, *PSGLab_gui*. Tato část grafického uživatelského rozhraní byla vytvořena za pomoci nástroje *GUIDE*.

7.1.1. Načtení souboru s uloženými daty

GUI využívá funkci *EASYS2*, který umožňuje načítání souborů s příponou **.d*. Soubory s příponou **.d* v systému *EASYS2* mají následující strukturu:

- Hlavička - obecné informace o datech a struktuře souboru
- Data - datová reprezentace kanálů
- Příloha - obsahuje dodatečné informace týkající se dat

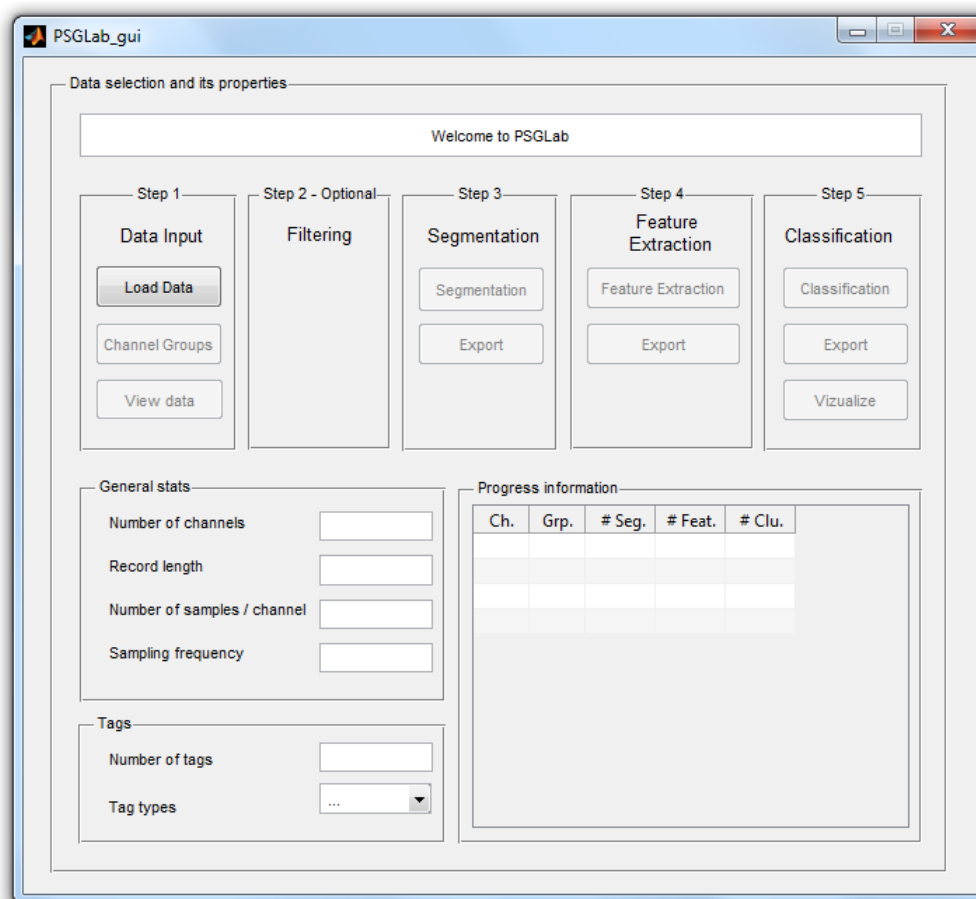
Naměřené vzorky kanálů v daném časovém okamžiku jsou v sekvenci uloženy za sebou v datové části souboru.

7.1.2. Popis komponent GUI

Stav GUI po spuštění aplikace je uveden na Obrázku 3. Ve spodní sekci se nacházejí obecné informace o načteném souboru a informace o průběhu zpracování. Nad touto sekci je pětice panelů, které reprezentují řetěz zpracování dat. Na úplném vrcholu je okno, které zobrazuje důležité informace o stavu aplikace.

Panel *Step 1*:

7. Jednotlivé části GUI



Obrázek 3. Hlavní okno aplikace.

Load Data Po stisku tlačítka je otevřeno dialogové okno, které uživateli umožní otevřít jím vybraný soubor.

Channel Group Po stisku tlačítka, je otevřeno GUI, které uživateli umožní přenastavit skupiny kanálů. Více v kapitole 7.1.4

View Data Po stisku tlačítka je otevřeno GUI, které dokáže zobrazit a částečně modifikovat načtená data.

Panel *Step 3*:

Segmentation Po stisku tlačítka je otevřeno GUI, které uživateli umožní vypočítat hranice segmentů kanálů dle jeho specifikací.

Export Po stisku tlačítka, jsou data segmentace uložena do souboru.

Panel *Step 4*:

Feature Extraction Po stisku tlačítka je otevřeno GUI, které uživateli umožní vypočítat příznaky kanálů dle jeho specifikací.

Export Po stisku tlačítka, jsou vypočítané příznaky uloženy do souboru.

Panel *Step 5*:

Classification Po stisku tlačítka je otevřeno GUI, které uživateli umožní klasifikovat kanály dle jeho specifikací.

Export Po stisku tlačítka, jsou výsledky klasifikace uloženy do souboru.

Vizualize Po stisku tlačítka je zobrazena grafická reprezentace klasifikace kanálů.

Panel *General stats*:

Sampling frequency Frekvence s níž jsou vzorkovány načtená data, uvedeno v jednotkách [Hz].

Export Po stisku tlačítka jsou výsledky klasifikace uloženy do souboru.

Number of samples per channel Počet vzorků na kanál.

Record length Délka záznamu ve formátu [hh : mm : ss].

Number of channels Počet kanálů v načteném souboru.

Panel *Tags*:

Number of tags Počet tagů, které byly uloženy v souboru společně s daty.

Tag types Výčet typů tagů, které byly uloženy v souboru společně s daty.

Panel *Progress information*:

Ch. Pořadové číslo kanálu.

Grp. Číslo skupiny, ke které daný kanál přísluší.

Seg. Počet vypočtených segmentů pro příslušný kanál.

Feat. Počet vypočtených příznaků pro příslušný kanál.

Clu. Počet vypočtených shluků pro příslušný kanál.

7.1.3. Důležité funkce

Z hlediska funkčnosti jsou nejdůležitější tyto tři funkce:

1. *PSGLab_gui_OpeningFcn*
2. *PSGLab_gui_CloseRequestFcn*
3. *loadDataBTT_Callback*

PSGLab_gui_OpeningFcn: Jedná se o funkci, kterou GUI vykoná před tím než se zobrazí. Patří zde veškeré operace, které je možné provést bez zásahů uživatele před spuštěním aplikace. Seznam úkonů prováděných v této funkci:

- Určení adresáře ze kterého je GUI spouštěno a přidání do cesty Matlabu.
- Načtení nastavení segmentačních funkcí, funkcí pro výpočet příznaků a klasifikačních funkcí do příslušných datových struktur.
- Uložení handle na *PSGLab_gui* do Root a nastavení potřebných aplikačních dat.
- Otevření a načtení souboru s uloženým nastavením.

PSGLab_gui_CloseRequestFcn: Funkce, která je vykonána před zavření *PSGLab_gui*. Před uzavřením aplikace se provede :

- Kontrola *Flag* proměnných a případné uložení nastavení z tohoto běhu do struktury tomu určené.
- Uložení struktury s nastavením do souboru *Setting.mat*.
- Odstranění handle na *PSGLab_gui* z Root.

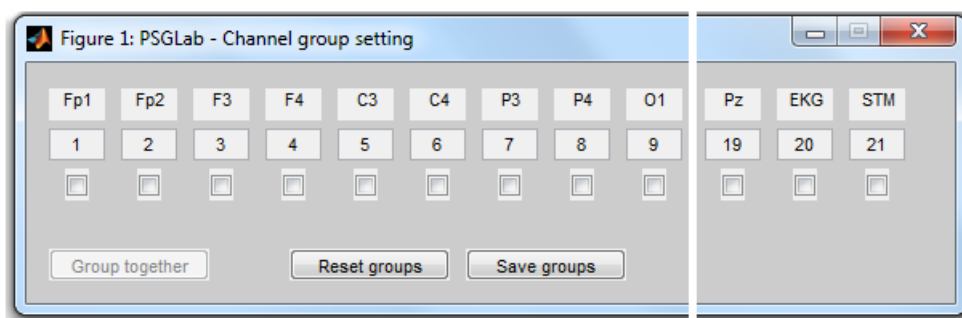
loadDataBTT_Callback: Funkce je vykonána při stisku tlačítka *Load Data*. Seznam úkonů, které funkce provede:

- Načtení uživatelem zvoleného souboru.
- Kontrola, zda byly splněny všechny potřebné podmínky pro načtení nastavení z předchozího chodu.
- Odstranění starých dat a přenastavení tlačítek při otevření nového souboru.
- Nastavení zbylých aplikačních dat a inicializace komponent GUI v závislosti na otevřeném souboru.

7.1.4. Nastavení skupin kanálů

Pokud chceme zpracovávat některé kanály společně, toto jednoduché GUI uživateli umožní přeuspořádat kanály do skupin. Pro skupiny kanálů se počítají stejné příznaky a následně se kanály ze skupiny klasifikují dohromady. Kód GUI je uložen v souboru *PSGLab_gui_channelGroup.m* a při jeho tvorbě nebyl použit nástroj *GUIDE*.

V horním řádku jsou uvedeny jména kanálů. Pod ním jsou čísla skupin, ke kterým kanál přísluší. Na posledním řádku jsou přepínací boxy, které slouží k vymezení intervalu kanálů jež budou zařazeny do stejné skupiny.



Obrázek 4. Okno pro nastavení skupin kanálů.

Group together Jakmile uživatel zvolí interval, tlačítko se stane aktivním. Kanály z vybraného intervalu zařadí do stejné skupiny a přepočítá zbylá čísla skupin.

Reset groups Obnoví čísla skupin do původního stavu. V každé skupině je jeden kanál a číslo skupiny je rovno pořadovému číslu kanálu.

Save groups Vybrané nastavení skupin zašle do *PSGLab_gui*.

7.2. Segmentace

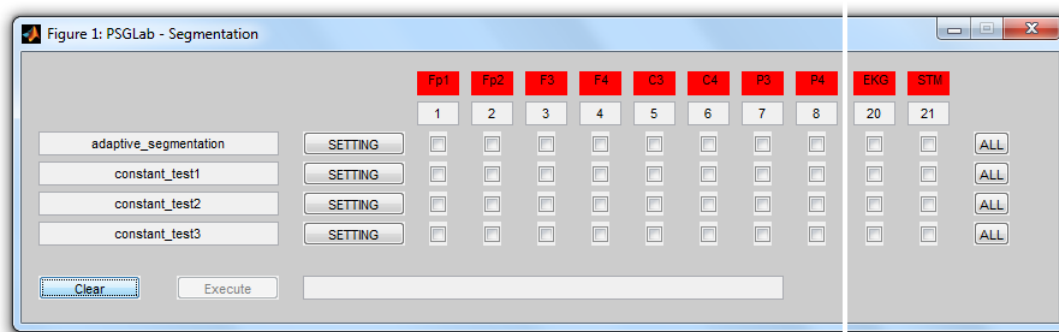
Prvním povinným krokem při zpracování načtených dat je segmentace. Tato část GUI umožňuje uživateli nastavit parametry segmentačních funkcí a zvolit jakou funkcí se budou počítat segmenty příslušného kanálu. Zdrojový kód části pro segmentaci je uveden v souboru *PSGLab_gui_segmentation.m*. Při tvorbě nebyl použit nástroj *GUIDE*. Stav GUI po otevření v případě, že nebylo načteno nastavení z předchozího běhu, je možné shlédnout na Obrázku 5.

7.2.1. Popis komponent GUI

V horní části GUI jsou nejprve uvedena jména kanálů, pod nimi jsou skupiny, do kterých kanály náleží. Změnou barvy pozadí jmen kanálů, je uživateli indikováno, zda pro daný kanál je zvolena segmentační funkce. Červená indikuje, že zatím nebyla funkce zvolena. Zelená naopak značí připravený kanál. Jakmile jsou pozadí všech jmen kanálů v zelené barvě, je uživateli umožněno pokračovat dále.

V levé části GUI se nacházejí dvojice jméno funkce a její nastavení. Jména funkcí jsou shodná s názvem souboru, v níž je funkce uložena.

Ve spodní části GUI je textové pole, v němž je zobrazován průběh výpočtů. Tedy pro jaký kanál se právě počítají hranice segmentů.



Obrázek 5. Okno pro nastavení segmentace.

Setting Při stisku je vyvoláno GUI umožňující nastavit parametry příslušné segmentační funkci, více v kapitole 7.2.3.

All Všem kanálům přiřadí stejnou segmentační funkci.

Clear Po stisku jsou odstraněny veškeré volby z matice přepínačů.

Execute V případě, že pro každý kanál je vybrána právě jedna segmentační funkce, stiskem je spuštěn výpočet.

7.2.2. Důležité funkce

Pro správnou funkci GUI jsou důležité funkce :

1. *settingButton_Callback*
2. *executeEnable*
3. *executeButton_Callback*

settingButton_Callback: Funkce je vykonána při stisku tlačítka *Setting*. Přehled úkonů, které funkce vykoná:

- Určení indexu objektu, který událost vyvolal.
- Zavolání GUI pro nastavení funkcí s výše určenými vstupními parametry.

executeEnable: Funkce kontrolující zda byly splněny veškeré podmínky pro spuštění výpočtu hranic segmentů. Funkce je volána při :

- Jakékoliv změně stavu přepínačů.
- Stisku tlačítka *Clear* nebo *ALL*.
- Po obnově stavu přepínačů z minulého běhu.

executeButton_Callback: Funkce je vykonána po stisku tlačítka *Execute*. Tlačítko je povoleno pouze pokud jsou splněny podmínky pro zahájení procesu segmentace. V těle funkce se provede:

- Při opakovaném spuštění segmentace jsou nejprve odstraněna stará data segmentace, výpočtu příznaků a klasifikace.
- V závislosti na stavu přepínačů se volají segmentační funkce s příslušnými vstupními parametry.
- Výsledky segmentace jsou uloženy do datové struktury a jsou zaslány do *PSGLab_gui*.
- Příslušná *Flag* proměnná je nastavena a je uloženo nastavení.
- Tabulka *Progress information* je aktualizována a je změněn stav některých tlačítek v hlavní části GUI.

7.2.3. Nastavení segmentačních funkcí

Aby bylo možné pohodlně měnit konstantní parametry funkcí, bylo navrženo grafické uživatelské rozhraní, které parametry přehledně zobrazí. Kód GUI je uložen v souboru *PSGLab_gui_segmentation_fncSetting.m* a při jeho tvorbě nebyl použit nástroj *GUIDE*.

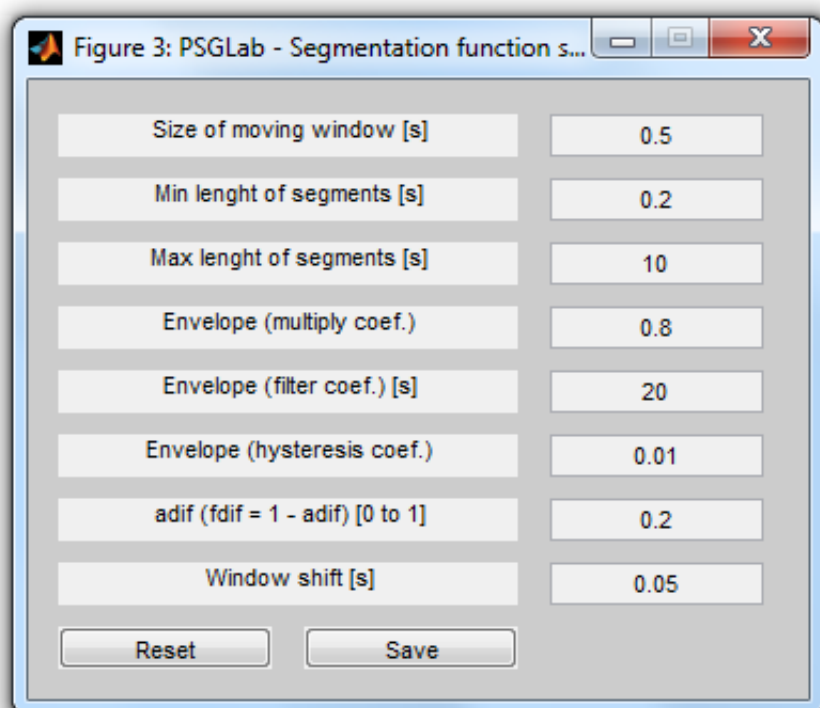
V horní části GUI jsou umístěny dvojice popis konstantního parametru a jeho hodnota. Zobrazená hodnota je buď defaultní nastavení parametrů, a nebo uživatelské nastavení.

Reset Obnoví defaultní nastavení funkce.

Save Uloží právě zobrazené nastavení parametrů jako uživatelské nastavené funkce.

7.3. Výpočet příznaků

Po vypočtení hranic segmentů kanálů je povoleno tlačítko *Feature Extraction*. Kliknutím se zobrazí GUI, které umožní nastavit parametry metod pro výpočet příznaků a zvolit příznaky, jenž se mají pro daný kanál počítat. Zdrojový kód GUI pro výpočet příznaků je uložen v souboru *PSGLab_gui_feature_extraction.m*. Při tvorbě nebyl použit nástroj *GUIDE*. Vzhled neinicializovaného GUI je uveden na Obrázku 7.



Obrázek 6. Okno pro nastavení parametrů segmentační funkce.

7.3.1. Popis komponent GUI

V horní části GUI jsou nejprve uvedena jména kanálů, pod nimi jsou skupiny, do kterých kanály náležejí. Změnou barvy pozadí jmen kanálů, je uživateli indikováno, zda pro daný kanál je zvolen některý z příznaků. Červená indikuje, že zatím nebyla funkce zvolena. Zelená naopak značí připravený kanál. Jakmile jsou pozadí všech jmen kanálů v zelené barvě, je uživateli umožněno pokračovat dále.

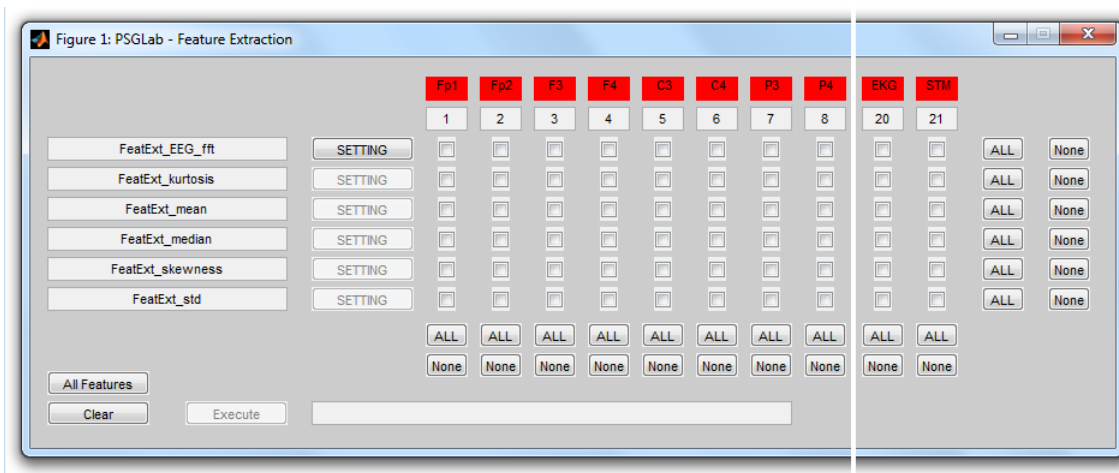
V levé části GUI se nacházejí dvojice jméno funkce a její nastavení. Jména funkcí jsou shodná s názvem souboru, v níž je funkce uložena.

Ve spodní části GUI je textové pole, v němž je zobrazován průběh výpočtů. Tedy pro jaký kanál se právě počítají příznaky.

All Features Po stisku jsou všem kanálům přiřazeny všechny příznaky.

Setting Po stisku je otevřeno GUI umožňující nastavit parametry funkcí počítající příznaky. Princip činnosti je shodný s nastavením segmentačních funkcí popsaným v kapitole 7.2.3.

All Tlačítko *All* v levé části GUI přiřadí všem kanálům příslušný příznak. Tlačítko *All* ve spodní části GUI přiřadí příslušnému kanálu všechny příznaky.



Obrázek 7. Okno pro nastavení části pro výpočet příznaků.

None Tlačítko *None* v levé části GUI odstraní daný příznak všem kanálům. Tlačítko *None* ve spodní části GUI odstraní všechny příznaky příslušného kanálu.

Clear Po stisku jsou odstraněny veškeré volby z matice přepínačů.

Execute V případě, že pro každý kanál je vybrán alespoň jeden příznak, stiskem je spuštěn výpočet.

7.3.2. Důležité funkce

Pro správnou funkci GUI jsou důležité funkce:

1. *settingButton_Callback*
2. *executeEnable*
3. *executeButton_Callback*

settingButton_Callback: Funkce je vykonána při stisku tlačítka *Setting*. Přehled úkonů, které funkce vykoná:

- Určení indexu objektu, který událost vyvolal.
- Zavolání GUI pro nastavení funkcí s výše určenými vstupními parametry.

executeEnable: Funkce kontrolující zda byly splněny veškeré podmínky pro spuštění výpočtu příznaků. Funkce je volána při :

- Jakékoliv změně stavu přepínačů.
- Stisku tlačítka *Clear*, *ALL* nebo *None*.
- Po obnově stavu přepínačů z minulého běhu.

executeButton_Callback: Funkce je vykonána po stisku tlačítka *Execute*. Tlačítko je povoleno pouze pokud jsou splněny podmínky pro zahájení výpočtu příznaků. V těle funkce se provede :

- Při opakovaném spuštění výpočtu příznaků jsou nejprve odstraněna stará data výpočtu příznaků a klasifikace.
- V závislosti na stavu přepínačů se volají funkce počítající příznaky s příslušnými vstupními parametry.
- Vypočítané příznaky jsou uloženy do datové struktury a jsou zaslány do *PSGLab_gui*.
- Příslušná *Flag* proměnná je nastavena a je uloženo nastavení.
- Tabulka *Progress information* je aktualizována a je změněn stav některých tlačítek v hlavní části GUI.

7.4. Klasifikace

Jakmile je dokončen výpočet příznaků, nebo-li pro každý kanál byl vypočítán alespoň jeden příznak a pro skupiny kanálů byly vypočítány stejné příznaky, je povoleno tlačítko *Classification*, které po stisku spustí část GUI pro klasifikaci. Zdrojový kód této části je uložen v souboru *PSGLab_gui_classification.m* a při jeho tvorbě nebylo využito nástroje *GUIDE*. Stav GUI po otevření bez načtení nastavení z předchozího chodu je uveden na Obrázku 8.

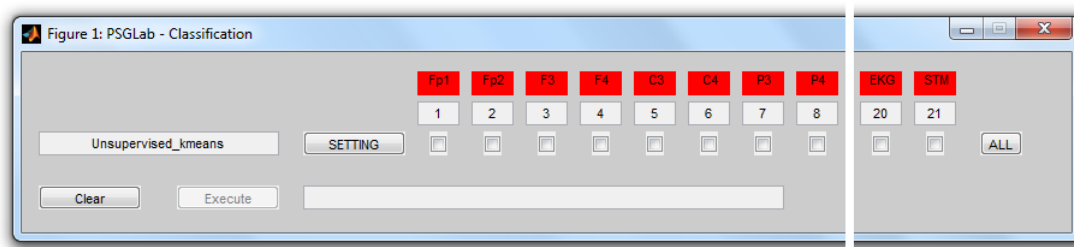
7.4.1. Popis komponent GUI

V horní části GUI jsou nejprve uvedena jména kanálů, pod nimi jsou skupiny, do kterých kanály náleží. Změnou barvy pozadí jmen kanálů, je uživateli indikováno, zda pro daný kanál je zvolena nějaká klasifikační funkce. Červená indikuje, že zatím nebyla funkce zvolena. Zelená naopak značí připravený kanál. Jakmile jsou pozadí všech jmen kanálů v zelené barvě, je uživateli umožněno pokračovat dále.

V levé části GUI se nacházejí dvojice jméno funkce a její nastavení. Jména funkcí jsou shodná s názvem souboru, v níž je funkce uložena.

Ve spodní části GUI je textové pole, v němž je zobrazován průběh výpočtů. Tedy pro jaký kanál se právě normalizují příznaky a následně počítá klasifikace.

Setting Při stisku je vyvoláno GUI umožňující nastavit parametry příslušné klasifikační funkci. Princim činnosti je opět shodný s nastavením segmentačních funkcí popsaným v kapitole 7.2.3.



Obrázek 8. Okno pro nastavení klasifikace signálů.

All Všem kanálům přiřadí stejnou klasifikační funkci.

Clear Po stisku jsou odstraněny veškeré volby z matice přepínačů.

Execute V případě, že pro každý kanál je vybrána právě jedna klasifikační funkce, stiskem je spuštěn výpočet.

7.4.2. Důležité funkce

Pro správnou funkci GUI jsou důležité funkce:

1. *settingButton_Callback*
2. *executeEnable*
3. *executeButton_Callback*

settingButton_Callback: Funkce je vykonána při stisku tlačítka *Setting*. Přehled úkonů, které funkce vykoná:

- Určení indexu objektu, který událost vyvolal.
- Zavolání GUI pro nastavení funkcí s výše určenými vstupními parametry.

executeEnable: Funkce kontrolující zda byly splněny veškeré podmínky pro spuštění klasifikace. Funkce je volána při:

- Jakékoliv změně stavu přepínačů.
- Stisku tlačítka *Clear* a *ALL*.
- Po obnově stavu přepínačů z minulého běhu.

executeButton_Callback: Funkce je vykonána po stisku tlačítka *Execute*. Tlačítko je povoleno pouze pokud jsou splněny podmínky pro zahájení klasifikace. V těle funkce se provede:

- Při opakovaném spuštění klasifikace jsou nejprve odstraněna stará data.
- Vypočítané příznaky jsou přeuspořádány do maticového tvaru vhodného pro klasifikaci.

7. Jednotlivé části GUI

- V závislosti na nastavení skupin kanálů jsou sloučeny kanály dohromady.
- Výsledné matice příznaků jsou normalizovány.
- V závislosti na stavu přepínačů se volají klasifikační funkce s příslušnými vstupními parametry.
- Pokud došlo k nějakému sloučení kanálů jsou skupiny roztrhány na jednotlivé kanály.
- Výsledky klasifikace jsou uloženy do příslušné datové struktury a jsou zaslány do *PSGLab_gui*.
- Příslušná *Flag* proměnná je nastavena a je uloženo nastavení.
- Tabulka *Progress information* je aktualizována a je změněn stav některých tlačítek v hlavní části GUI.

8. Ověření nad reálnými daty

Funkčnost aplikace je ověřena nad několika typy reálných dat lišících se v délce nebo počtu kanálů. Z ilustračních důvodů je pozornost věnována pouze několika ukázkovým typům.

8.1. Krátkodobý záznam

Vzhled hlavního okna aplikace po načtení krátkodobého záznamu je možné shlédnout na Obrázku 9.

V sekci *General stats* jsou zobrazeny základní informace o vlastnostech načteného souboru. Jedná se o záznam s 21 kanály, jeho délka je 36s, každý kanál má 9150 vzorků a všechny kanály byly vzorkovány s frekvencí 250Hz. V sekci *Tags* je zobrazen počet tagů obsažených v datech a následuje jejich výčet.

V části *Progress information* jsou zobrazena jména kanálů, jejich pořadové číslo a skupina do které přísluší. V tomto případě byly ponechány původní skupiny kanálů.

Po stisku tlačítka *Segmentation* je otevřeno okno umožňující nastavit parametry segmentace. Tato část se zaměřením na prvních 8 kanálů je uvedena na Obrázku 10.

Pro všechny kanály byla zvolena funkce počítající konstantní segmentaci.

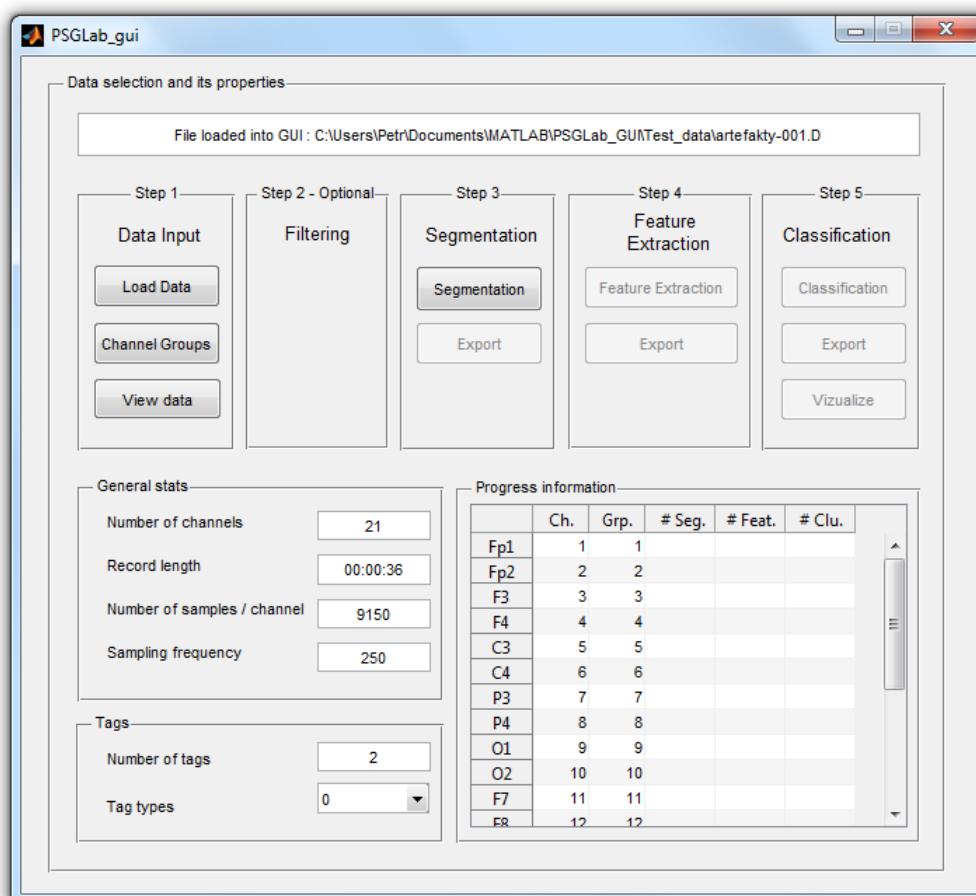
Nastavení vybrané funkce bylo provedeno po stisku tlačítka *Setting* v příslušném řádku. Rozhraní pro nastavení funkce je uvedeno na Obrázku 11. Tato rozhraní jsou totožná s rozhraními pro výpočet příznaků a klasifikaci, a proto v dalším textu nebudou uvedena.

Délka segmentu byla nastavena na 4s. Defaultní hodnota parametru funkce byla 10s.

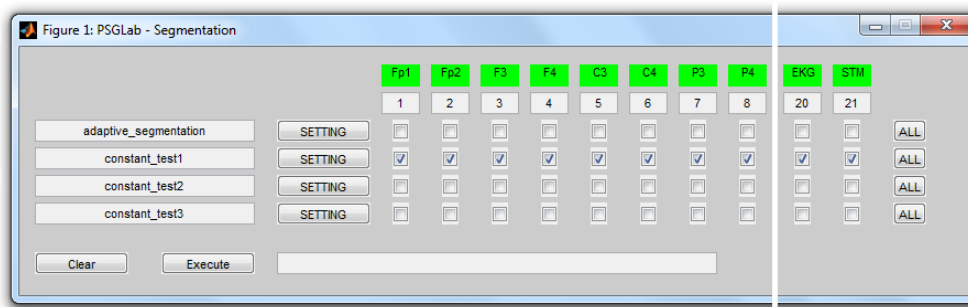
Po dokončení segmentace je zpřístupněno, mimo jiné, tlačítka *Feature Extraction* a jeho stiskem je možné zobrazit rozhraní pro výběr počítaných příznaků a nastavení těchto funkcí. Rozhraní pro výpočet příznaků je uvedeno na Obrázku 12.

V tomto případě mají kanály navzájem různá čísla skupin, pokud je vybrán

8. Ověření nad reálnými daty



Obrázek 9. Hlavní okno po načtení krátkodobého záznamu.



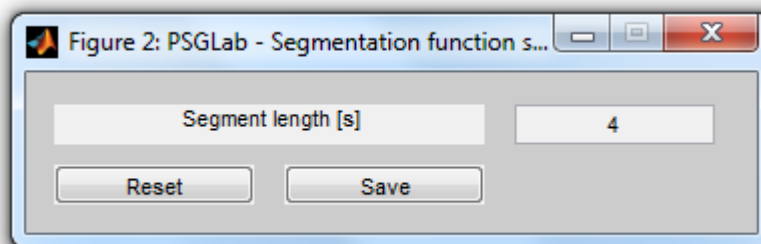
Obrázek 10. Okno pro segmentaci po načtení krátkodobého záznamu.

alespoň jeden příznak pro každý kanál, je možno přejít k výpočtu. Příznaky je jinak možné volit libovolně.

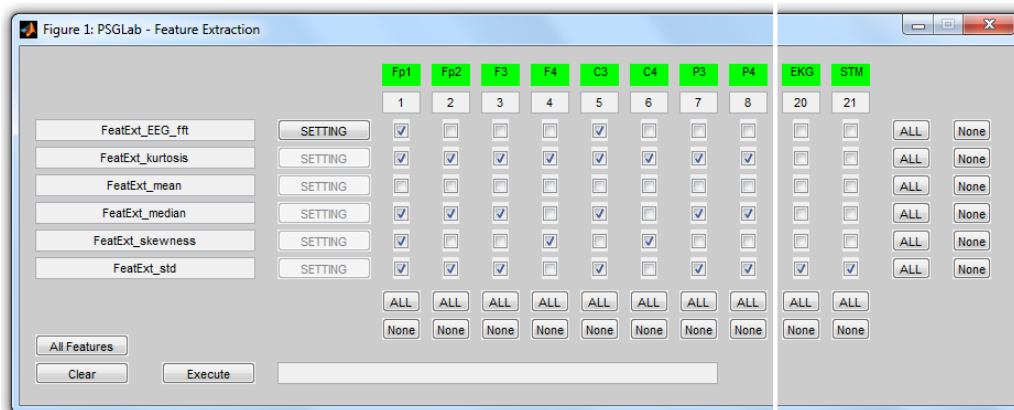
Jakmile je dokončen výpočet příznaků, je možné stiskem tlačítka *Classification* přistoupit k nastavení klasifikace dat. Okno pro výběr klasifikačních funkcí a jejich nastavení je uvedeno na Obrázku 13.

Stav hlavního okna aplikace po dokončení klasifikace je uvedeno na Obrázku 14.

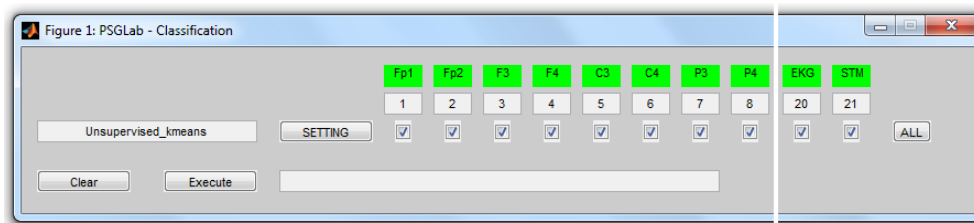
8. Ověření nad reálnými daty



Obrázek 11. Okno pro nastavení segmentační funkce.

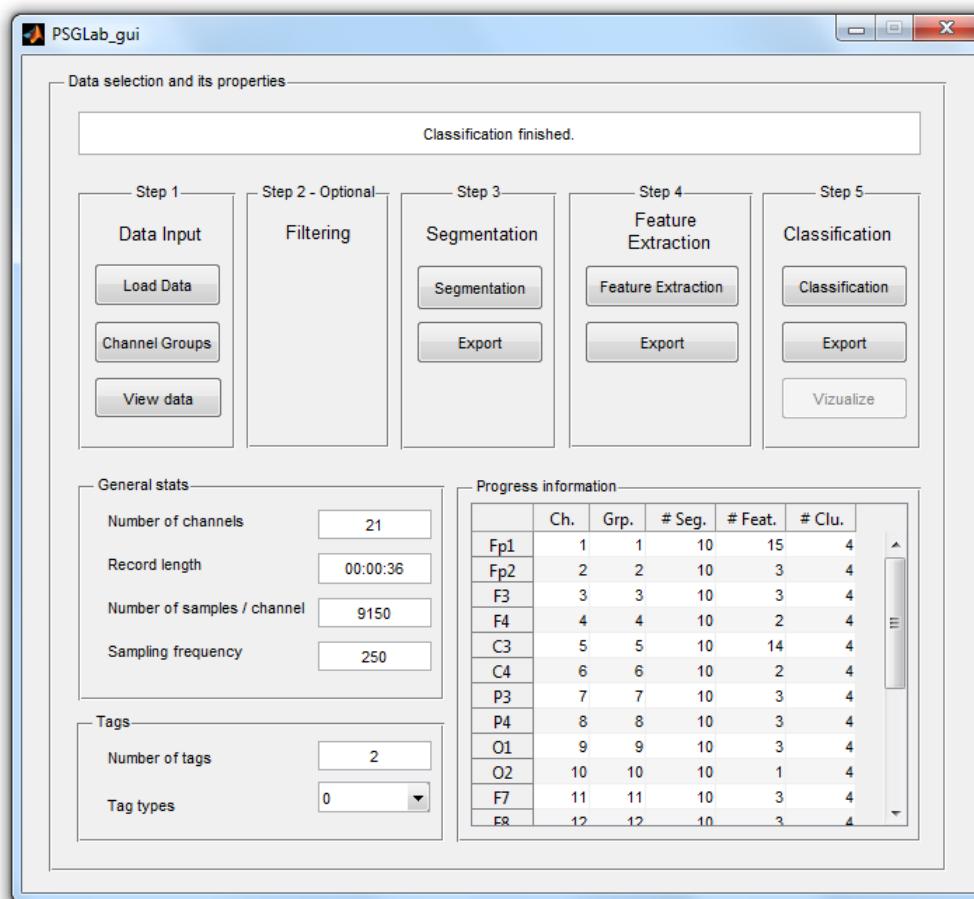


Obrázek 12. Okno pro výpočet příznaků po načtení krátkodobého záznamu.



Obrázek 13. Okno pro klasifikaci po načtení krátkodobého záznamu.

V sekci *Progress information* jsou zobrazeny výsledky procesu. Pro všechny kanály byla zvolena konstantní segmentace, a proto jsou ve třetím sloupci počty segmentů kanálů shodné. V čtvrtém sloupci je zobrazen počet příznaků vypočítaných pro daný kanál. V posledním sloupci je počet počet shluků do kterých byly segmenty kanálů klasifikovány.



Obrázek 14. Hlavní okno po zpracování krátkodobého záznamu.

8.2. Střednědobý záznam

V tomto případě byl načten střednědobý záznam a jeho kanály 2 až 6 byly sloučeny do jedné skupiny. Stav hlavního okna po načtení je uveden na Obrázku 15.

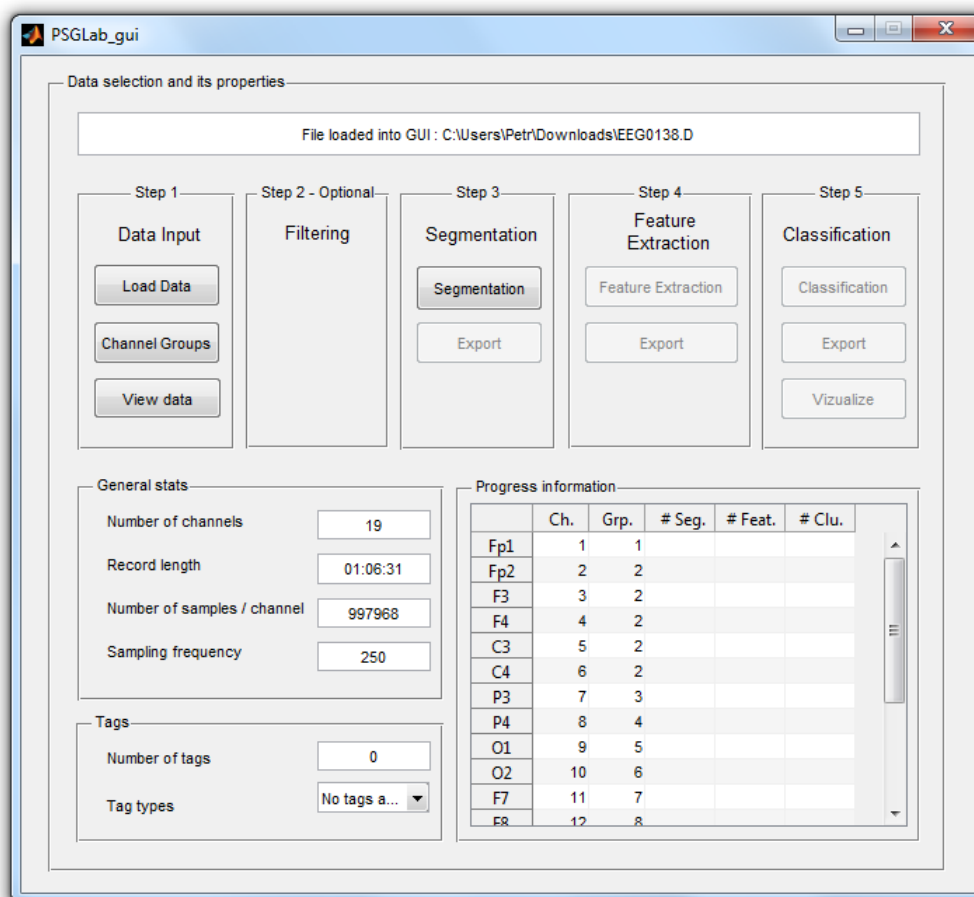
Načtený záznam má 19 kanálů, jeho přibližná délka je $1h6m$ a každý kanál má přibližně 10^6 vzorků. Záznam byl opět vzorkován frekvencí $250Hz$. Na rozdíl od předešlého krátkodobého záznamu, tento záznam neobsahuje žádné tagy.

Segmentaci je i v tomto případě možné volit libovolně, tedy kanály ve stejné skupině mohou být segmentovány různými metodami. Stav okna pro segmentaci je možné shlédnout na Obrázku 16.

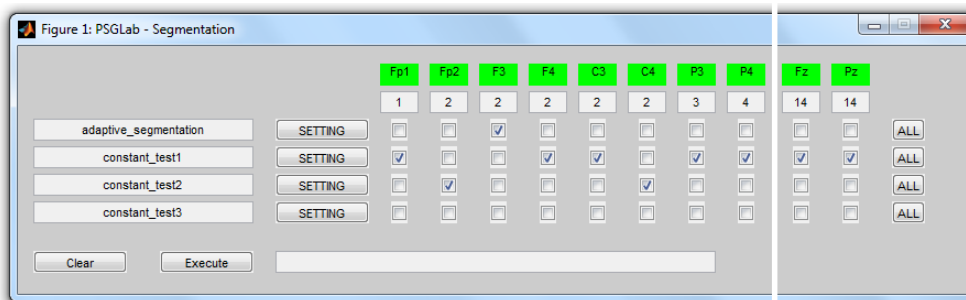
Po dokončení segmentace je možné spustit okno pro výběr počítaných příznaků. Je nutné, aby pro kanály ve stejné skupině, byly počítány totožné příznaky. Tento fakt, je ilustrován na Obrázku 17.

Klasifikace je provedena standardně jako v předešlém případě. Opět však platí,

8. Ověření nad reálnými daty



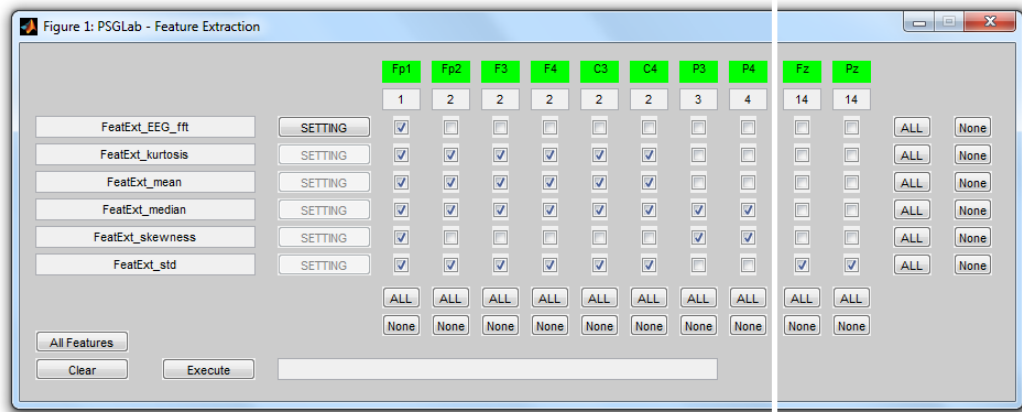
Obrázek 15. Hlavní okno po načtení střednědobého záznamu.



Obrázek 16. Segmentační okno pro zpracování střednědobého záznamu.

že kanály ve stejné skupině, musí být zpracovány stejnou klasifikační funkcí. Kanály ve stejné skupině jsou během klasifikace sloučeny dohromady. Jsou normalizovány a následně klasifikovány jako by se jednalo o jeden kanál. Na závěru klasifikace jsou skupiny rozděleny zpět do jednotlivých kanálů.

Po ukončení klasifikace dat je stav grafického uživatelského rozhraní uveden na Obrázku 18.



Obrázek 17. Okno výpočtu příznaků pro zpracování střednědobého záznamu.

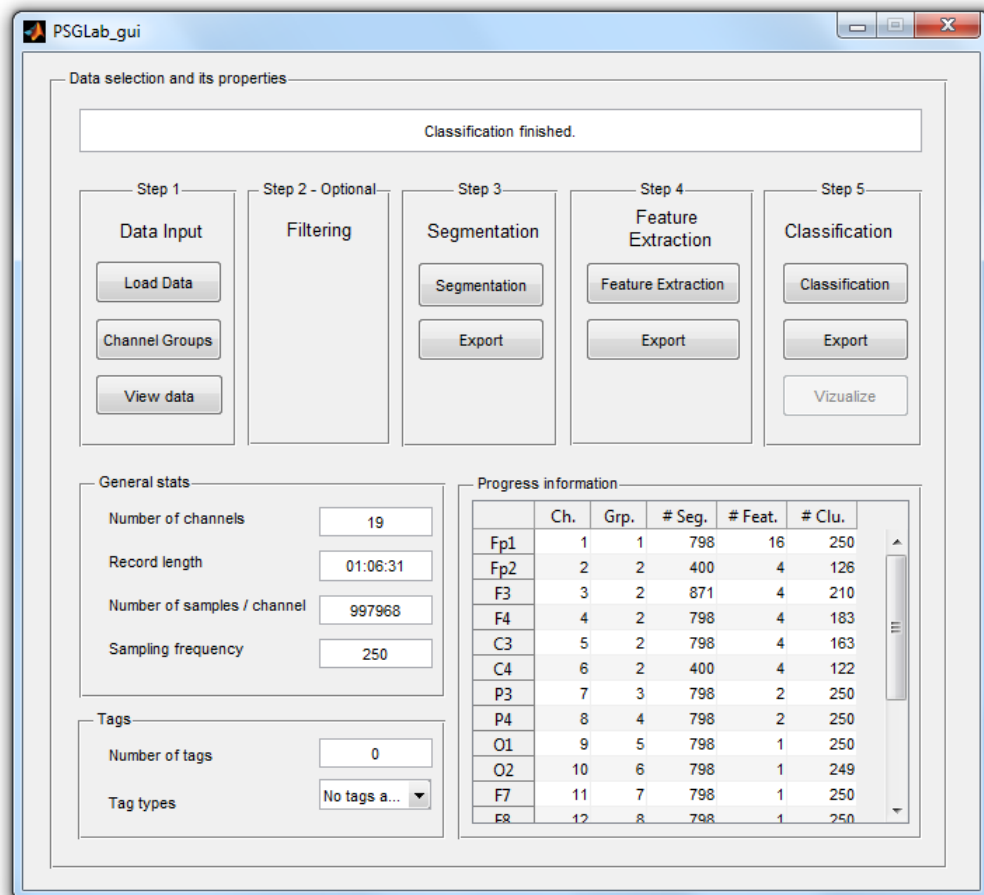
8.3. Dlouhodobý záznam

Na závěr této kapitoly byly možnosti aplikace otestovány na dlouhodobém záznamu. Způsob práce s částmi pro segmentaci, výpočet příznaků a klasifikaci je stejný bez ohledu na délku záznamu. Z tohoto důvodu zde bude uveden pouze stav hlavního GUI po dokončení klasifikace.

Načtený záznam obsahuje 24 kanálů, každý s přibližně $7,4 \cdot 10^6$ vzorky o celkové délce $8h13m$. Kanály byly vzorkovány s frekvencí $250Hz$. Soubor s daty obsahoval 994 tagů, jejich typy jsou uvedeny pod tímto údajem.

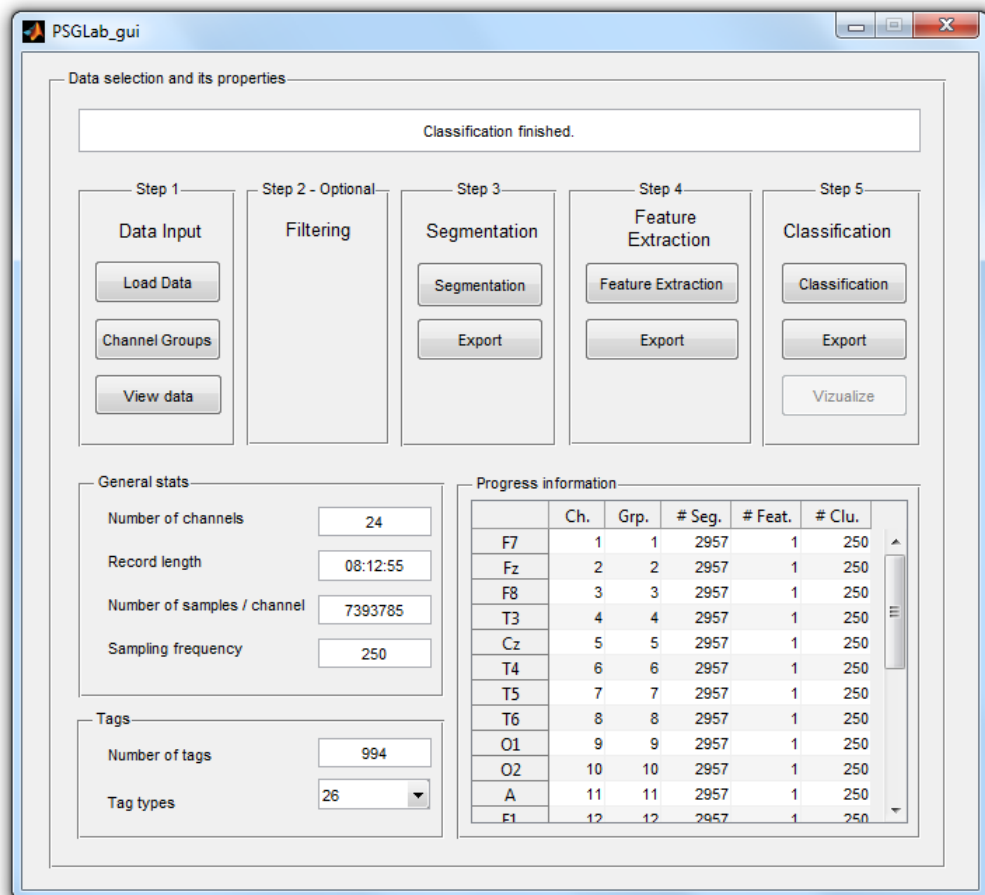
Stav hlavního okna poté co byly všechny kanály segmentovány do úseků délky $10s$, byl pro ne vypočítán jeden příznak a po rozdělení segmentů kanálů do 250 shluků je uveden na Obrázku 19.

8. Ověření nad reálnými daty



Obrázek 18. Hlavní okno po ukončení zpracování střednědobého záznamu.

8. Ověření nad reálnými daty



Obrázek 19. Hlavní okno po ukončení zpracování dlouhodobého záznamu.

9. Možná rozšíření

Účelem této práce bylo navrhnout základní kostru aplikace pro zpracování dlouhodobých biomedicínských záznamů. V této kapitole je navrženo několik možných rozšíření aplikace.

9.1. Podpora vstupu různých datových formátů

V současné době aplikace podporuje pouze soubory ve formátu **.d*, které jsou načítány za pomoci funkcí systému *EASYS2*. Jedním z možných rozšíření, by tedy byl modul umožňující načítání souborů i v jiných formátech. Modul, který by v závislosti na příponě souboru, použil vhodnou funkci k načtení dat do GUI a korektně pak nastavil odpovídající aplikační proměnné.

9.2. Podpora různých vzorkovacích frekvencí kanálů

Navržená aplikace předpokládá, že jsou všechny kanály vzorkovány stejnou vzorkovací frekvencí. Předpoklad vychází ze struktury právě podporovaného datového typu načítaných souborů. Existují však systémy podporující rozdílné vzorkovací frekvence pro jednotlivé kanály. Dalším možným rozšířením by byla úprava aplikace pro podporu různých vzorkovacích frekvencí.

9.3. Grafická uživatelská rozhraní pro export dat

Stávající aplikace umožňuje export vybraných mezivýsledků do souborů. Umístění, jméno a formát těchto souborů je předem dáno. Aplikaci by bylo možné rozšířit o GUI umožňující uživateli si zvolit umístění, jméno a případně i formát dat, které si přeje uložit.

9.4. Modul pro zpracování a definici tagů

Navržená aplikace umožňuje pouze zobrazit již existující značky, nebo-li tagy. Mohla by být rozšířena o modul poskytující nástroje k editaci či tvorbě uživatelských značek.

9.5. Přidávání funkcí za běhu aplikace

Pokud chce uživatel použít stejnou funkci počítající například hranice segmentů pro dva kanály s různým nastavením, nemá jinou možnost, než fyzicky vytvořit v odpovídajícím adresáři kopii funkce, tu posléze přejmenovat a spustit aplikaci znovu. Části pro segmentaci, výpočet příznaků a klasifikaci by mohly být rozšířeny o nástroj umožňující vytvořit lokální kopii funkce a dovolit tak použít stejnou funkci s různým nastavením během jednoho výpočetního cyklu.

9.6. Vizualizace výsledků klasifikace

Výstupem klasifikace jsou pořadová čísla shluků do kterých byl příslušný segment zařazen. Hlavní okno aplikace pak zobrazuje počet těchto shluků v kanálech. Dalším možným rozšířením by mohl být modul pro grafickou vizualizaci příslušnosti segmentu do shluku.

9.7. Historie operací v GUI

Důležité informace pro uživatele jsou zobrazována v horní části hlavního okna aplikace. V okně je vždy zobrazena poslední důležitá zpráva. Toto okno by mohlo být rozšířeno na víceřádkovou verzi uchovávající historii operací s GUI i včetně časového údaje.

9.8. Modul pro předzpracování signálů

Předzpracováním signálu je možné odstranit celou řadu nežádoucích jevů zanesených do dat při měření. Je možné upravit vzorkovací frekvenci kanálů nebo odstranit rušivý vliv elektrické rozvodné sítě. Aplikace by mohla být rozšířena o modul fungující na stejné bázi jako části pro segmentaci, zpracování příznaků nebo klasifikaci, ale provádějící vybrané předzpracování kanálů.

10. Závěr

Výsledkem práce je navržený modulární systém pro zpracování biomedicínských záznamů. Uživateli je umožněno načítat a zpracovávat soubory ve formátu **.d*. Vybrané vlastnosti načteného souboru jsou zobrazeny v hlavní části aplikace. Po načtení souboru je možné pozměnit příslušnost kanálů do skupin, například pokud chce uživatel zpracovávat všechny EEG kanály dohromady.

Dalším krokem při zpracování je segmentace signálu. Část pro segmentaci využívá funkcí uložených v příslušné složce. Hlavní část aplikace se při startu nejprve dotáže těchto funkcí na jejich vlastnosti. Získané údaje pak předá segmentační části. Uživateli je umožněno přidávat vlastní funkce pro segmentaci, dodrží-li navrženou strukturu hlavičky funkce. Funkce ze složky je kdykoliv možné odebrat nebo naopak přidat nové. Nastavení parametrů používaných funkcí je možné nastavit dle libosti uvnitř segmentačního okna. Segmentační okno obecně uživateli poskytuje přehledný nástroj, ve kterém jsou voleny funkce a jejich nastavení použité pro daný kanál.

Pro určené segmenty jsou v následující části vypočítány příznaky. Tak jako v případě segmentace, i zde existuje složka určená funkcím počítající příznaky, která má stejné vlastnosti a význam. Okno pro výpočet příznaků je vizuálně velice podobné segmentačnímu oknu a do jisté míry je ovládáno stejně. Jediným významným rozdílem je zacházení se skupinami kanálů. Grafické uživatelské rozhraní zajišťuje, že se pro kanály ve stejné skupině počítají tytéž příznaky.

Posledním krokem zpracování je klasifikace. Segmenty jednotlivých kanálů se rozřadí do tříd na základě vypočítaných příznaků. Funkce provádějící klasifikaci mají také svoji složku, ve které jsou uloženy. Ovládání této části je v principu stejné jako u předchozích částí. Navržená aplikace využívá shlukových metod klasifikace jako zástupce učení bez učitele.

Vybrané mezivýsledky je možné exportovat do souborů. Před ukončením aplikace je uloženo nastavení používaných funkcí a stavu jednotlivých částí GUI. Při novém startu aplikace je toto nastavení načteno a v závislosti na právě zpracovávaných datech je použito i v tomto běhu. Výrazně je tak usnadněno zpracování několika podobných záznamů.

Příloha A.

Zobrazení načteného záznamu

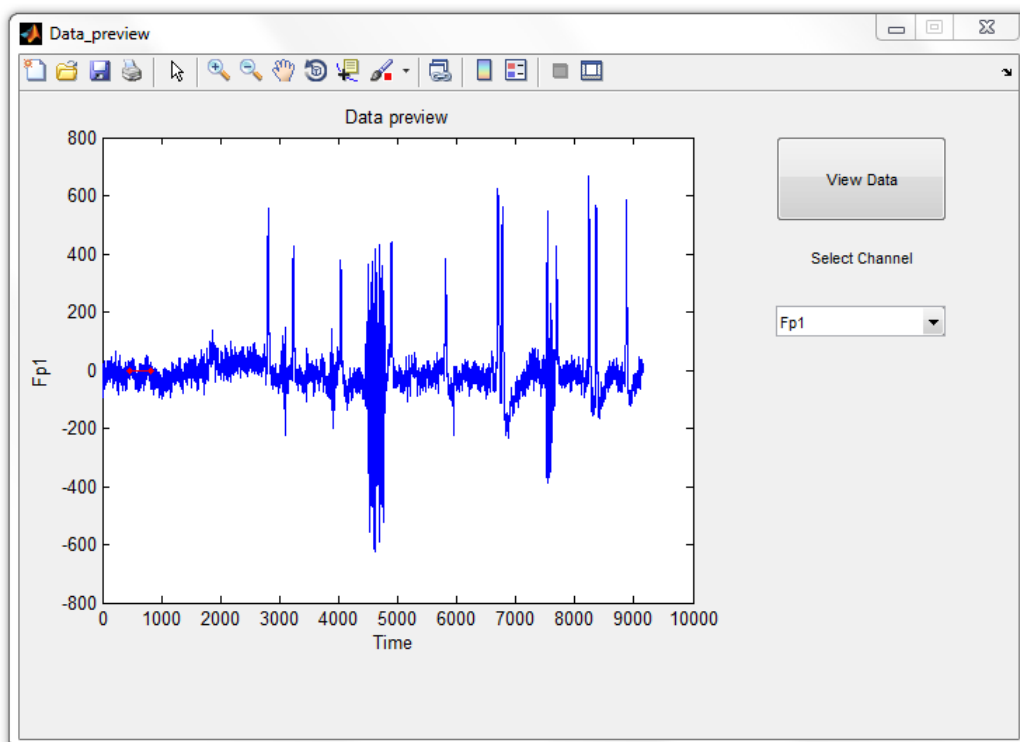
Součástí aplikace je také modul pro zobrazení načteného záznamu. Autorem tohoto modulu je Václav Příhoda, který převzal základní verzi verzi pro jednoduché zobrazení záznamu a upravil ji do stávající podoby.

A.1. Původní verze modulu

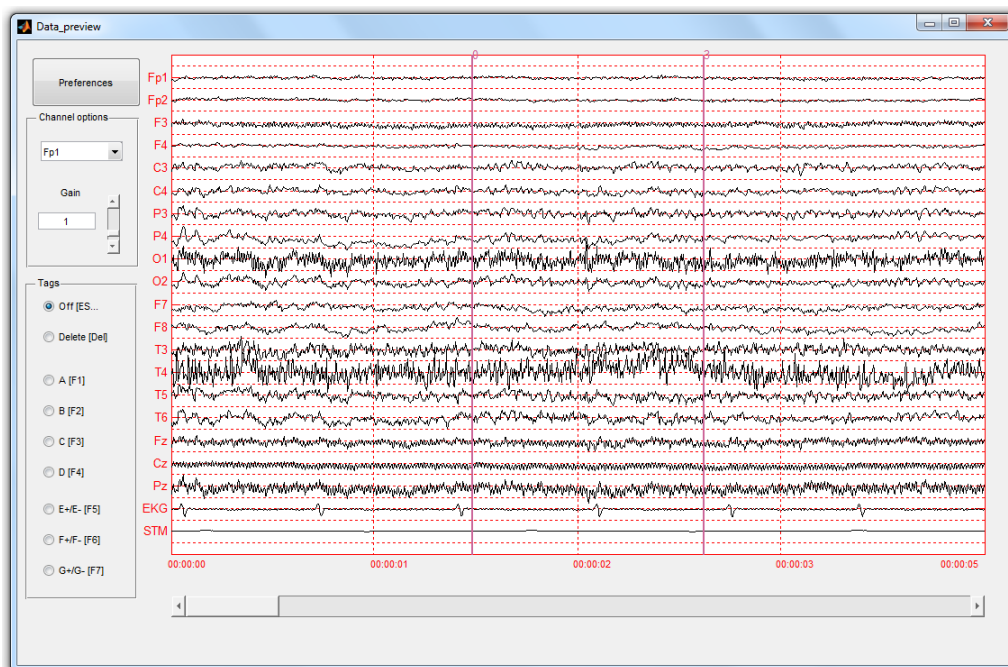
V rámci mého individuálního projektu byl z důvodu ověření funkčnosti načítání záznamů vytvořen modul *Data Preview*, který velice jednoduchým způsobem umožňoval zobrazit jeden kanál, bez možnosti změny jakéhokoliv nastavení. Původní verze modulu je uvedena na Obrázku 20.

A.2. Stávající verze modulu

Jak již bylo zmíněno výše, Václav Příhoda v rámci bakalářské práce s názvem *Anotátor dlouhodobých EEG záznamů*, upravil původní verzi do současné podoby. Vzhled modulu po načtení stejných dat jako v případě původní verze je uveden na Obrázku 21.



Obrázek 20. Původní vzhled modulu pro zobrazení záznamů.



Obrázek 21. Současný vzhled modulu pro zobrazení záznamů.

Literatura

- [1] V. Gerla. *Automated Analysis of Long-Term EEG Signals*. PhD thesis, The Czech Technical University in Prague, 2012.
- [2] Petr Heřman. *Biosignály z pohledu biofyziky*. 1999. ISBN: 80-902899-7-5
- [3] J. Rieger. *Zpracování dlouhodobých EEG záznamů*. Diplomová práce, České vysoké učení technické v Praze, 2004.
- [4] V. Gerla, V. Djordjevic, L. Lhotska, V. Krajca. *PSGLab Matlab Toolbox for Polysomnographic Data Processing: Development and Practical Application*, Proceedings of 10th International Conference on Information Technology and Applications in Biomedicine, IEEE, 2010. Online: <http://bio.felk.cvut.cz/psglab> (visited on 20/5/2012).
- [5] E. Niedermeyer, F. Lopes da Silva. *Electroencephalography - Basic principles, clinical applications and related fields*. Philadelphia: Lippincott William & Wilkins, S. 1165-97, ISBN: 0-7817-5126-8, 2005.
- [6] *EEGLab*. Online: <http://scn.ucsd.edu/eeglab/> (visited on 21/5/2012).
- [7] J. Štastný. *MEET: Modular EEG Processing Toolbox*. Online: http://amber.feld.cvut.cz/fpga/publications/meet_abstract.pdf (visited on 21/5/2012).
- [8] Robins KA. *EEGVIS: A Matlab toolbox for Browsing, Exploring, and Viewing Large Datasets*. 2012. Online: <http://visual.cs.utsa.edu/research/projects/eegvis> (visited on 21/5/2012).
- [9] R. D. Pascual-Marqui. *LORETA: low resolution brain electromagnetic tomography*. Online: <http://www.uzh.ch/keyinst/loreta.htm> (visited on 21/5/2012).
- [10] *ASA Lab*. Online: <http://www.seren-medical.com/asalaba.html> (visited on 21/5/2012).

- [11] *BioSig*. Online: <http://biosig.sourceforge.net/index.html> (visited on 21/5/2012).
- [12] J. M. Stern and J. Engel. *Atlas of EEG patterns. LWW medical book collection*. Lippincott Williams & Wilkins, 2005.
- [13] E. S. Goldensohn, A. D. Legatt, S. Koszer, and S. M. Wolf. *Goldensohn's EEG interpretation: problems of overreading and underreading*. 2nd Completely Revised Edition. Armong, 1999.
- [14] W. H. Spriggs. *Essentials of Polysomnography*. Jones & Bartlett Learning, 2009.
- [15] MathWorks Inc. *MATLAB: Statistics toolbox: user's guide*. The MathWorks, 2007.
- [16] B. V. Mohamed, K. Y. Farooq, and K. Y. Uzzaman. *Automatic seizure detection using higher order moments*. In Proceedings of the 2010 International Conference on Recent Trends in Information, Telecommunication and Computing, ITC '10, pages 159-163, Washington, DC, USA, 2010. IEEE Computer Society.
- [17] B. R. Greene, S. Faul, W. P. Marnane, G. Lightbody, I. Korotchikova, and G. B. Boylan. *A comparison of quantitative EEG features for neonatal seizure detection*. Clin Neurophysiol, 119(6):1248-1261, 2008.
- [18] K. Paul, V. Krajča, Z. Roth, J. Melichar, and S. Petránek. *Comparison of quantitative EEG characteristics of quiet and active sleep in newborns*. Sleep Med, 4(6):543-552, 2003.
- [19] B. H. Jansen. *EEG segmentation and classification: an explorative study*. Cooperative Drukkerij P.E.T., 1979.
- [20] J. S. Barlow, O. D. Creutzfeldt, D. Michael, J. Houchin, and H. Epelbaum. *Automatic adaptive segmentation of clinical EEGs*. Electroencephalography and Clinical Neurophysiology, 51(5):512-525, May. 1981.
- [21] N. Sulaiman, M. N. Taib, S. Lias, Z. H. Murat, S. A. M. Aris, and N. H. A. Hamid. *EEG-based stress features using spectral centroids technique and K-nearest neighbor classifier*. In Proceedings of the 2011 UKSim 13th International Conference on Modelling and Simulation, UKSIM '11, pages 69-74, Washington, DC, USA, 2011. IEEE Computer Society.