

**UNIVERZITA PALACKÉHO V OLOMOUCI**

**PRÍRODOVEDECKÁ FAKULTA**

KATEDRA BIOFYZIKY

**BAKALÁRSKA PRÁCA**

**Fraktálne charakteristiky biologických signálov a štruktúr**



Vypracovala: Andrea Mikulášová

Študijný odbor: Biofyzika

Vedúci bakalárskej práce: RNDr. Tomáš Füst, Ph.D.

Olomouc 2021

## Bibliografická identifikácia

Meno a priezvisko autora:	Andrea Mikulášová
Názov práce:	Fraktálne charakteristiky biologických signálov a štruktúr
Typ práce:	Teoretická
Pracovisko:	Katedra Biofyziky, Univerzita Palackého v Olomouci
Vedúci práce:	RNDr. Tomáš Fürst, Ph.D.
Rok obhajoby práce:	2021
Abstrakt:	<p>Mnoho biologických štruktúr a signálov pochádzajúcich zo živých organizmov vykazuje fraktálne charakteristiky, vyznačujúce sa tým, že sa rovnaké motívy opakujú na rôznych priestorových či časových škálach. Srdce má fraktálny charakter prejavujúci sa v jeho štruktúrnom, ako aj časovom aspekte. Normálna elektrická aktivita srdca je tradične označovaná termínom „regulárny sínusový rytmus“. Avšak intervaly medzi dvoma údermi srdca zdravých jedincov fluktuujú v zdanlivo nepravidelnom a komplexnom štýle, aj počas ich zotrvania v kľude. Existuje mnoho rôznych matematických prístupov ku kvantifikácii fraktálnych charakteristík signálov a štruktúr. V tejto práci sa zaoberáme skúmaním dynamiky tlkotu ľudského srdca cez fraktálnu analýzu nestacionárnych časových radov intervalov medzi jednotlivými údermi srdca (R-R intervalov) pomocou Detrendovanej fluktuačnej analýzy (DFA) a Multiškálovej entropie (MSE) v programovacom jazyku Python.</p>
Kľúčové slová:	fraktál, fraktálne charakteristiky, časové rady, biologické signály, R-R intervaly, normálny sínusový rytmus, fibrilácia predsiení
Počet strán:	52
Počet príloh:	2
Jazyk:	Slovenský

## **Bibliographical identification**

Author's first name and surname:	Andrea Mikulášová
Title of thesis:	Fractal characteristics of biological signals and structures
Type of thesis:	Theoretical
Department:	Department of Biophysics, Palacký University Olomouc
Supervisor:	RNDr. Tomáš Füst, Ph.D.
The year of defence:	2021
Abstract:	<p>Many biological structures and signals derived from living organisms exhibit fractal properties characterized by the same motifs being repeated on different spatial or temporal scales. The heart has a fractal character manifested in its structural as well as temporal aspects. Normal electrical activity of the heart is traditionally referred to as “regular sinus rhythm”. However, the intervals between two heartbeats of healthy individuals fluctuate in seemingly irregular and complex manner, even while remaining at rest. There are many different mathematical approaches to quantifying the fractal characteristics of signals and structures. In this thesis we investigate the human heart rate dynamics through fractal analysis of non-stationary interbeat interval (R-R interval) time series using Detrended Fluctuation Analysis (DFA) and Multiscale Entropy (MSE) in the Python programming language.</p>
Key words:	fractal, fractal characteristics, time series, biological signals, R-R intervals, normal sinus rhythm, atrial fibrillation
Number of pages:	52
Number of attachments:	2
Language:	Slovak

Chcela by som srdečne poďakovať svojmu vedúcemu bakalárskej práce RNDr. Tomášovi Fürstovi, Ph.D. nielen za jeho trpezlivosť, odbornú pomoc a usmernenie pri písaní tejto práce, ale aj za inšpiráciu a motiváciu, ktorou ma zahŕňal počas môjho štúdia. Ďalej ďakujem prof. RNDr. Janovi Naušovi, CSc. za jeho rady, ochotu a poskytnutú literatúru. V neposlednom rade ďakujem svojej rodine a priateľom za ich podporu.

Prehlasujem, že som bakalársku prácu vypracovala samostatne, pod vedením RNDr. Tomáša Fürsta, Ph.D. a s použitím literatúry uvedenej v závere práce.

V Olomouci dňa . . . . .

# OBSAH

1	ÚVOD.....	1
2	MATEMATICKÉ FRAKTÁLY .....	2
2.1	HISTORICKÝ KONTEXT .....	2
2.2	BENOIT B. MANDELBROT.....	4
2.3	DEFINÍCIA .....	6
2.3.1	<i>Geometrická podobnosť</i> .....	7
2.3.2	<i>Dimenzia</i> .....	8
2.3.3	<i>Výpočet HB fraktálnej dimenzie</i> .....	9
2.3.4	<i>Metóda počítania boxov</i> .....	12
3	PRÍRODNÉ FRAKTÁLY A APLIKÁCIE TÝCHTO ŠTRUKTÚR .....	13
3.1	PEÚCA .....	14
3.2	CIEVNY SYSTÉM.....	14
3.3	STROMY .....	14
3.4	BUNKY A TKANIVÁ.....	15
3.5	DLA SYSTÉMY .....	15
3.6	DNA .....	16
3.7	APLIKÁCIE .....	17
3.8	SRDCE .....	18
4	FRAKTÁLY V ČASOVÝCH RADOCH.....	20
4.1	DETREDOVANÁ FLUKTUAČNÁ ANALÝZA DFA ( <i>DETRENDED FLUCTUATION ANALYSIS</i> ) .....	20
4.1.1	<i>Interpretácia metódy DFA aplikovanej na intervaly medzi údermi srdca</i> .....	23
4.2	MULTIŠKÁLOVÁ ENTROPIA MSE ( <i>MULTISCALE ENTROPY</i> ).....	24
4.2.1	<i>Interpretácia metódy MSE aplikovanej na intervaly medzi údermi srdca</i> .....	26
5	CIEĽ PRÁCE .....	27
6	METÓDY .....	28
6.1	ZÍSKANIE EKG ZÁZNAMOV .....	28
6.2	URČENIE R-R INTERVALOV .....	29
6.3	METÓDA DFA.....	29
6.4	METÓDA MSE .....	30
6.5	CHARAKTERISTICKÝ DFA EXPONENT A MSE KRIVKA.....	31
7	VÝSLEDKY A DISKUSIA .....	32
7.1	ZÍSKANIE R-R INTERVALOV Z EKG ZÁZNAMOV .....	32
7.2	METÓDA DFA.....	34
7.3	METÓDA MSE .....	40
7.4	CHARAKTERISTICKÝ DFA EXPONENT A MSE KRIVKA.....	44

8	ZÁVER .....	47
9	ZOZNAM LITERATÚRY .....	49
10	INTERNETOVÉ ZDROJE .....	51
11	ZDROJE OBRÁZKOV .....	52
	PRÍLOHA 1.....	
	PRÍLOHA 2.....	

## ZOZNAM SKRATIEK A ZNAČIEK

AF	Skupina subjektov s fibriláciou predsiení
ANS	Autonómny nervový systém
cDNA	Komplementárna deoxyribonukleová kyselina
CO	Oxid uhľnatý
Cu	Meď
DFA	Detrendovaná fluktučná analýza ( <i>Detrended Fluctuation Analysis</i> )
DLA	Difúzne limitovaná agregácia ( <i>Diffusion Limited Aggregation</i> )
DNA	Deoxyribonukleová kyselina
EKG	Elektrokardiogram
EEG	Elektroencefalogram
HB	Hausdorffovo-Besicovitchova (dimenzia)
MIT-BIH	Massachusetts Institute of Technology – Boston's Beth Israel's Hospital
MSE	Multiškálová Entropia ( <i>Multiscale Entropy</i> )
MWI	Integrácia pohybujúcou sa vlnou ( <i>Moving Wave Integration</i> )
N	Skupina subjektov s normálnym sínusovým rytmom
SampEn/S <sub>E</sub>	Vzorková entropia ( <i>Sample Entropy</i> )
SD	Smerodajná odchýlka ( <i>Standard Deviation</i> )
STM	Skenovací tunelovací mikroskop
WFDB	Waveform-database





# 1 ÚVOD

Ako opísať prírodné objekty, ktoré vykazujú nejakú mieru komplexity? Oblaky, stromy, lastúry či paprade alebo vetvenia riek. V živej prírode sa stretávame so zložitými funkčnými systémami s veľkým počtom komponentov, ku ktorým sa pridávajú emergentné javy a miera komplexity narastá. Jedným z konceptov matematiky 20. storočia opisujúcich komplexné prírodné štruktúry sú fraktály. Fraktály sú definované ako matematické objekty, ktorých geometrické štruktúry sa presne alebo približne opakujú v nich samých v rôznych merítkach a ktorých fraktálna dimenzia striktno presahuje ich topologickú dimenziu. Prírodné objekty alebo javy, ktoré dokážeme charakterizovať fraktálnou množinou, voľne nazývame prírodnými fraktálmi. Fraktály v prírode nemusia mať len viditeľnú fraktálnu podobu, často sa totižto vyskytujú v časových radoch fluktuácií veličín alebo hodnôt, či fyziologických signáloch. Na ich analýzu a klasifikáciu boli vyvinuté početné metódy. Táto práca sa zaoberá aplikáciou dvoch metód – Detrendovanej fluktuačnej analýzy a Multiškálovej entropie – na intervaly medzi údermi srdca, teda tzv. R-R intervaly. Tieto metódy slúžia na kvantifikáciu komplexity časových radov a na rozlíšenie normálneho a patologických stavov.

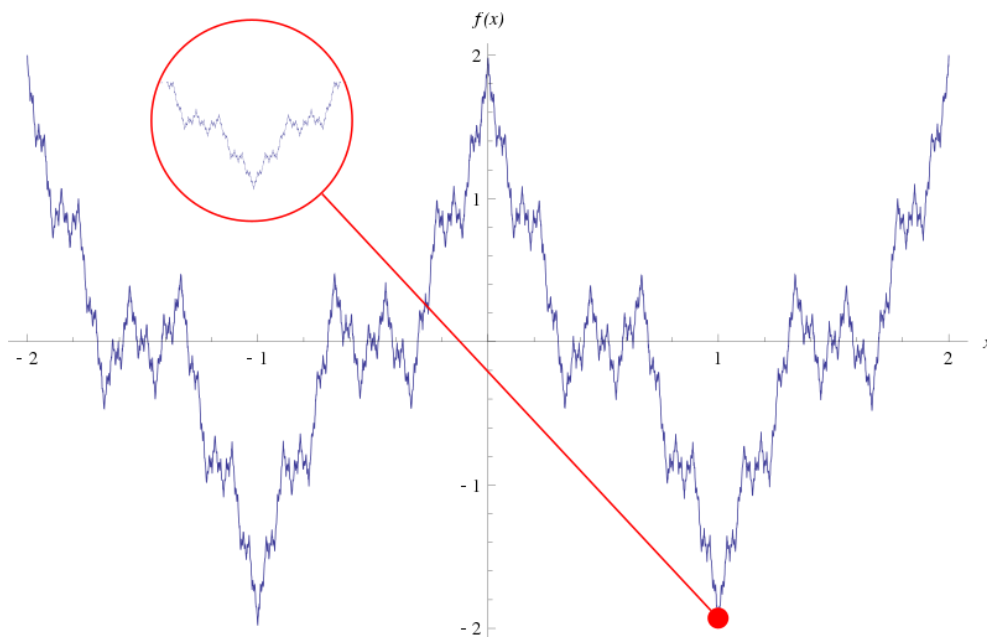
Práca je rozdelená na teoretickú časť ponúkajúcu prehľad problematiky a predstavenie konceptu fraktálnych štruktúr a časť obsahujúcu výsledky analýzy EKG záznamov s popisom metód a ich diskusiou.

## 2 MATEMATICKE FRAKTÁLY

### 2.1 Historický kontext

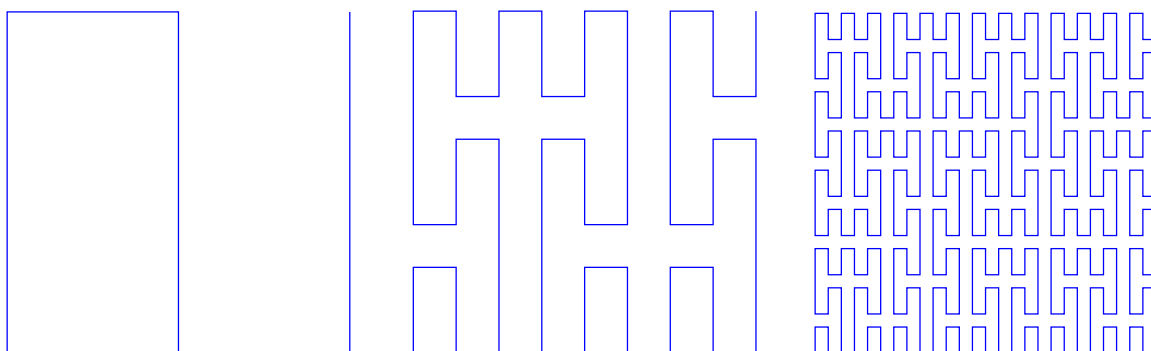
Fraktály hrajú významnú rolu vo vývoji modernej matematiky a v revolúcii oddeľujúcej myšlienky klasickej matematiky 19. storočia a tej modernej z 20. storočia. Cantorova teória množín, Peanova krivka vyplňujúca plochu a im podobné štruktúry podnecujúce túto revolúciu boli označované ako „patologické“, či „galériou monštier“ (Dyson 1978).

Odvtedy ako Newton vyvinul kalkulus, sa matematici, inšpirovaní prírodou a fyzickým svetom, sústredili len na spojité funkcie. Preto keď v roku 1872 Karl Weierstrass predstavil spojitú funkciu, ktorá nebola hladká a nemala deriváciu ani v jednom bode, zdvihla sa proti nemu vlna kritiky a odporu od slávnych matematikov ako Charles Hermite, Émile Picard alebo Henri Poincaré. Weierstrassovo „monštrum“ (na Obrázku 1) bolo akceptované až v roku 1904, keď Albert Einstein prišiel s modelom Brownovho pohybu. Častice sa v tekutine pohybujú náhodným smerom, kvôli zrážkam s molekulami tekutiny, vďaka čomu trajektórie týchto častíc nie sú nikdy hladké (Kucharski 2014).



Obrázok 1: Weierstrassova funkcia. Dostupné z: <https://www.pngwing.com/en/free-png-nkaow>

V roku 1890 Giuseppe Peano skonštruoval spojité zobrazenie z jednotkového intervalu do jednotkového štvorca, ktoré je známe ako Peanova krivka v limite vyplňujúca plochu (Obrázok 2). O rok na to David Hilbert zostrojil svoju vlastnú krivku vyplňujúcu plochu a ako prvý do svojej publikácie zahrnul aj ilustráciu takéhoto objektu (Sagan 1991).



*Obrázok 2:* Peanova krivka, tri iterácie. Dostupné z:  
<https://commons.wikimedia.org/wiki/File:Peanocurve.svg#/media/File:Peanocurve.svg>

V roku 1904 švédsky matematik Helge von Koch predstavil Kochovu krivku (na Obrázku 7), ktorá mala v limite nekonečnú dĺžku, avšak zaberala konečnú plochu. Vytvoril tak nediferencovateľnú krivku (spojitú krivku bez konečnej derivácie vo všetkých bodoch) pomocou elementárnej geometrie. Neskôr sa ukázalo, že tieto už spomenuté „škaredé“ funkcie ako Weierstraussova funkcia alebo Peanova krivka a nekonečne dlhé krivky ohraničujúce konečnú plochu ako tá Kochova, sú len špičkou ľadovca v teórii, ktorá dobre opisuje správanie dynamických systémov. Klasická euklidovská geometria bola totižto dovtedy jedinou formou akceptovanej geometrie. Opisovala geometricky ideálne útvary (ako štvorce, kocky, trojuholníky alebo gule) pomerne jednoducho, avšak komplikovanejšie štruktúry takto opísať nedokázala (Zelinka a kol. 2006).

Silnou súčasťou matematiky bola topológia, ktorá riešila niektoré z problémov klasickej geometrie. Nezaoberala sa vlastnosťami útvarov, ako krivosť alebo vzdialenosť, ale vlastnosťami nemennými pri spojitých transformáciách. Topológia totižto predpokladá, že všetky šálky rôznych tvarov s uškami sú rovnocenné, pretože všetky môžu byť transformované jedna na druhú bez potreby vytvorenia nového otvoru alebo zacelenia už

existujúceho otvoru. Tak podobne sú ekvivalentné aj útvary ako štvorec a kruh, avšak úsečka a kruh už nie. Tak isto aj šálka bez uška a šálka s uškom už nie sú rovnocenné, pretože pri transformácií tam by musel vzniknúť nový otvor a naspäť by musel jeden zaniknúť. V prípade topológie však môžeme takto upadnúť do tvrdení, že línie pobreží všetkých ostrovov nadobúdajú rovnakú formu, pretože sú topologicky identické s kružnicou. A teda, že ich topologická dimenzia (bližšie v sekcii 1.3.2) je rovnaká pre tieto línie, ako aj pre kružnice, a teda rovná jednej (Mandelbrot 1982).

Teóriou riešiacou tieto problémy opisov matematických a neskôr reálnych objektov a štruktúr bola fraktálna geometria, založená B.B. Mandelbrotom v 70. rokoch 20. storočia. Akonáhle počítače umožnili komplexné výpočty iteračných systémov, mnoho citlivých nelineárnych dynamických systémov bolo lepšie preskúmaných práve vďaka teórii chaosu a fraktálnej geometrii (bližšie časť 2.2) (Korolj a kol. 2019).

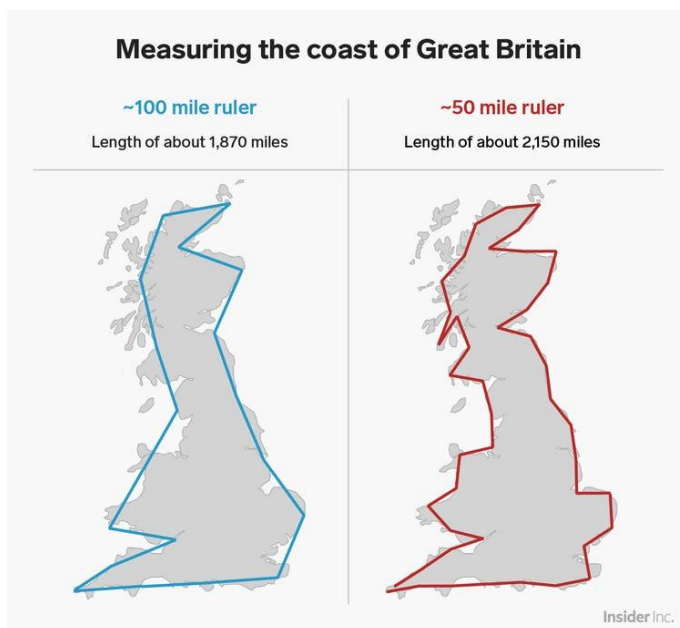
## 2.2 Benoit B. Mandelbrot

Fraktál, z latinského *fractus*, nepravidelný, rozbitý, bol definovaný B. B. Mandelbrotom, výnimočným osobnosťou a uvažovaním, v roku 1975. Mandelbrot, priekopník vo svojom obore, fraktálnej geometrii, sa narodil v Poľsku. V roku 1936 s rodinou emigrovali do Paríža, neskôr do stredného Francúzska. Po oslobodení Paríža v roku 1944 sa tam opäť vrátil a zložil prijímacie skúšky na vtedy popredných vedeckých školách. Ich úspešné zloženie bolo považované za veľmi nezvyčajné, keďže svoje predošlé roky strávil samoštúdiom. Vďaka jeho vrozenému daru, dokázal nájsť geometrické doplnenia takmer ku každému analytickému problému. V Paríži, kolíske klasickej analýzy, však v tej dobe panoval názor, že geometria je mŕtvy obor z detských kníh matematiky, z ktorých musí každý správny matematik vyrásť. Toto všeobecné presvedčenie a zatvrdnutosť matematickej analýzy ho donútili odchyliť sa zo smeru, ktorý ho neskôr preslávil. Dva roky na Caltechu, obhájenie doktorandskej práce na Parížskej Univerzite, rok na Princetone, svadba v roku 1955, profesúra. Ozajstný prevrat však nastal v lete 1958, keď sa dostal ako návštevník do firmy IBM. Vtedy bol výskum IBM omnoho lepší ako ten univerzitný kdekoľvek inde v krajine. Aj keď sa Mandelbrotovi dostávalo už v tej dobe slávy a bol navštevujúcim profesorom na mnohých katedrách rôznych oborov, žiadna univerzita ho nechcela ako stáleho profesora s

tak nepredvídateľnými záujmami. Kdežto IBM podporovala jeho projekty a po dvoch rokoch na Harvarde mu dala aj jeho vlastných zamestnancov. Počas týchto rokov sa Mandelbrot zaoberal širokým spektrom tém z rôznych odvetví. Do lingvistického Zipfovho zákona frekvencií výskytu slov vniesol štatistické vysvetlenia a tiež sa podieľal na vysvetlení Paretovho princípu a distribúcie príjmov z ekonomického hľadiska. V oblasti ekonómie sa zdržal skúmaním akciového trhu a cenami komodít. Dovtedy boli pohyby na trhu predikované pomocou fyzikálneho modelu plynu v rovnovážnom stave. Mandelbrot ako prvý pristúpil k tomuto problému skrz sebedobnosť, čo omnoho lepšie popisovalo pozorované variácie cien. Ďalší pokrok následne prišiel so skúmaním šumu, turbulencií a distribúcií porúch telekomunikačných kanálov a distribúcií klastrov galaxií. Všetky jeho riešenia problémov boli postavené na sebedobnosti (Barcellos 2008). V roku 1967 však verejne položil veľmi známu otázku: „Aké dlhé je pobrežie Británie?“. Odpoveď znie: „záleží“. Záleží na veľkosti inštrumentu, ktorým meriame. Mandelbrot totižto tvrdil, že geografické krivky sú štatisticky sebedobné a že ich miera komplikovanosti môže byť kvantifikovaná pomocou dimenzie  $D$ , ktorá je zároveň aj exponentom podobnosti. V prípade, že by geograf chcel zmerať dĺžku pobrežia, najskôr si musí zvoliť mierku  $G$ , pod ktorou sú už ostatné črty považované za detaily. Je schopný odhadnúť dĺžku pobrežia  $L(G)$  vykreslením úsečiek dĺžky  $G$  aproximujúcich pobrežie čo najkratšou cestou (znázornené na Obrázku 3). Odhad dĺžky  $L(G)$  značne závisí na zvolenej mierke  $G$ . Po určení  $L(G)$  pre viaceré hodnoty  $G$  geograf zistí, že so zmenšujúcim sa  $G$  dĺžka  $L(G)$  rýchlo rastie, v limite až donekonečna. Analytický vzorec prepájajúci  $L(G)$  s  $G$ , čisto empiricky zistený, predložil Lewis F. Richardson v roku 1961

$$L(G) = M \cdot G^{1-D}, \quad (1)$$

kde  $M$  je kladná konštanta a  $D$  je konštanta väčšia alebo rovná jednej. Ak  $D = 1$  ide o rovnú hranicu, kdežto v prípade extrémne nepravidelnej línie Británie  $D = 1,25$ . Najviac hladkým na mape je pobrežie Južnej Afriky s  $D = 1,02$  (Mandelbrot 1967).



Obrázok 3: Meranie dĺžky pobrežia Veľkej Británie. Vľavo použité merítko dĺžky 100 míľ, vpravo použité merítko dĺžky 50 míľ. Dostupné z: <https://www.businessinsider.com/fractals-and-the-coast-of-great-britain-2019-2>

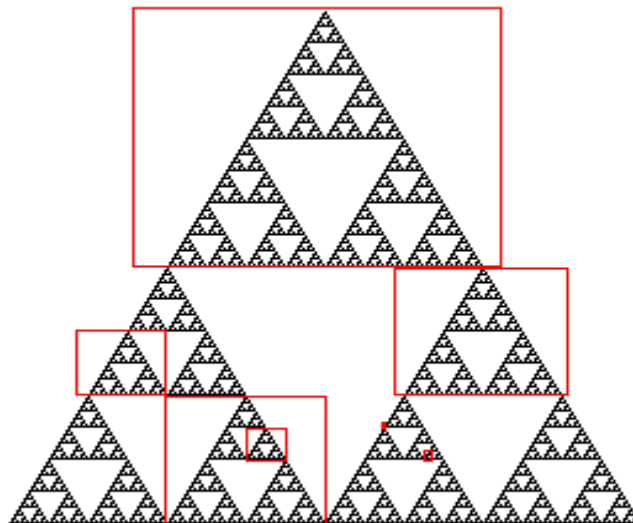
V roku 1973 bol Mandelbrot svojim strýkom pozvaný urobiť na College de France väčšiu prednášku. Pri príprave tejto prednášky si uvedomil, že jeho práca skrz obory bola prepojená, kompletnejšia a homogénnejšia než očakával. V roku 1975 bola publikovaná Fraktálna geometria prírody v prvej francúzskej verzii. Svoje nádeje z mladosti, nájst' usporiadanosť tam, kde všetci ostatní videli len chaos, teda naplnil (Barcellos 2008).

## 2.3 Definícia

Fraktály nie sú určené striktne krátkou definíciou. Definujú ich mnohé zobrazenia a kontexty, s ktorými sa na ne odvolávame (Barnsley 1988a). Dva najčastejšie pohľady však hovoria o fraktáloch skrz ich geometrickú podobnosť a dimenziu.

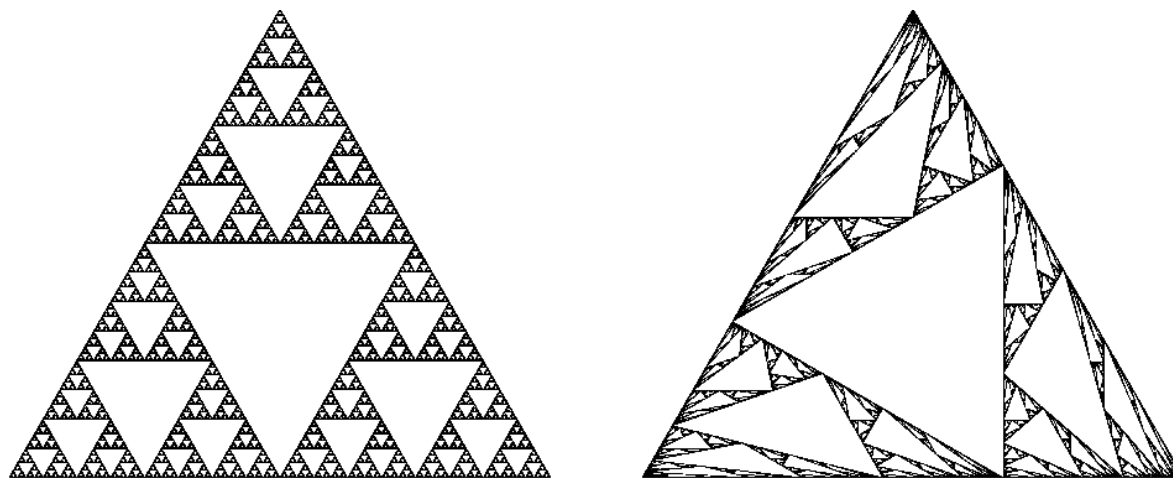
### 2.3.1 Geometrická podobnosť

Skrz geometrickú podobnosť sú fraktály definované ako objekty, ktorých geometrické štruktúry sa presne alebo približne opakujú v nich samých v rôznych merítkach až do nekonečna (v matematickom slova zmysle; vo fyzikálnom svete vždy narazíme na hranice systému). Z tohto hľadiska delíme fraktály na sebedobné a sebebízne. Sebedobné fraktály sú čisto matematické objekty, pretože akákoľvek ich časť je presnou kópiou pôvodného objektu (Zelinka a kol. 2006). Na Obrázku 4 sa nachádza Sierpiňského trojuholník z roku 1916, ktorý vzniká z rovnostranného trojuholníka. Body v strede každej zo strán sú spojené a vzniknutý útvar – opäť trojuholník – je z útvaru odstránený. Postup sa znova a znova opakuje a je zjavné, že ten istý vzor sa vyskytuje v rôznych merítkach.



Obrázok 4: Sierpiňského trojuholník a demonštrácia geometrickej sebedobnosti. Dostupné z: [http://pages.cs.wisc.edu/~ergreen/honors\\_thesis/similar.html](http://pages.cs.wisc.edu/~ergreen/honors_thesis/similar.html)

Spoločným znakom tohto typu fraktálov je ich nemennosť do určitej miery. Nemennosť po posuve a nemennosť po zmene škály/rozsahu. Fraktály invariantné práve po zmene škály označujeme preto teda aj ako „*scaling fractals*“ (Mandelbrot 1982). Sebebízne fraktály sú naopak podobnou kópiou celku vytvorenou rozdielnym škálovaním v rôznych smeroch (napríklad v smere  $x$  a  $y$ ). Príklad sebebíznej štruktúry sa nachádza na Obrázku 5.



(A) The classical self-similar Sierpiński gasket  $X_1$ .

(B) A self-affine gasket  $X_2$  which is not self-similar.

*Obrázok 5:* Porovnanie sebedobného a sebezpríbuzného fraktálu. Vľavo (A) klasický sebedobný Sierpiňského trojuholník, vpravo (B) sebezpríbuzný trojuholník. Dostupné z: <https://www.semanticscholar.org/paper/A-converse-statement-to-Hutchinson's-theorem-and-a-Morris-Sert/ebd6a99025cd62fc664c29ea9f0658a946850b13>

### 2.3.2 Dimenzia

Počas svojej krízy v období 1875–1925 matematici tušili, že v prípadoch iregularít, fragmentácií, ako aj regularít a súvislosti množín im nepostačí klasická definícia dimenzie ako počet súradníc. Existujú dva pohľady na dimenziu v súvislosti s fraktálmi. V každom Euklidovskom priestore je ku akokoľvek „patologickej“ množine priradené reálne číslo, ktoré nazývame dimenzia. Tou intuitívnejšou z dvoch je topologická dimenzia  $D_T$ , pretože je laicky vnímaná ako rozmer priestoru (Mandelbrot 1982). Topologická dimenzia bola na základe poznatkov Henri Lebesguea prvýkrát formálne definovaná Eduardom Čechom a nazýva sa preto aj Čech-Lebesgueova pokrývajúca dimenzia. Topologický priestor  $X$  má dimenziu  $n$ , ak existuje konečné otvorené zjemnenie každého konečného otvoreného pokrytia  $X$  tak, aby bol každý bod priestoru  $X$  obsiahnutý v prieniku najviac  $n+1$  množín tohto zjemnenia (Coornaert 2015). Druhou je dimenzia popísaná Hausdorffom v roku 1919 a upravenou Besicovitchom (viac v sekcii 1.3.3). Ide o Hausdorffovu-Besicovitchovu (HB) dimenziu označovanú ako  $D$ . Táto dimenzia  $D$  nemusí byť striktné celočíselná, avšak topologická dimenzia  $D_T$  áno.



Nemusia sa zhodovať, ale vždy platí

$$D \geq D_T. \quad (1)$$

Fraktál je teda z definície množina, pre ktorú platí, že HB dimenzia striktne presahuje jej topologickú dimenziu. Takisto každá množina s neceločíselnou dimenziou  $D$  je fraktálom, ako napríklad Cantorova množina alebo Kochova krivka (viď Obrázok 6 a 7). Preto je HB dimenzia označovaná ako fraktálna dimenzia (Mandelbrot 1982). Pomocou jej výpočtu dokážeme od seba odlišiť rôzne druhy fraktálov a tiež popísať ich mieru štruktúrovanosti (Zelinka a kol. 2006)

Fraktálna dimenzia teda poskytuje objektívny prostriedok na porovnávanie jednotlivých fraktálov. Je dôležitá, pretože môže byť definovaná v spojení s nameranými dátami a zameraná s určitou aproximáciou v rámci daného experimentu (Barnsley 1988b). Fraktálne dimenzie môžu byť priradené objektom reálneho sveta, akými sú napríklad línie pobreží (Mandelbrot 1982), štruktúry rastu lišajníkov a rastlín v porovnaní s abiotickým okolím (Azua-Bustos a Vega-Martínez 2013), alebo aj ľudský mozoček (Liu a kol. 2003). Tieto čísla nám potom umožňujú porovnať zložité množiny z reálneho sveta s matematickými fraktálmi (Barnsley 1988b).

### 2.3.3 Výpočet HB fraktálnej dimenzie

Geometricky hladká krivka má topologickú dimenziu rovnú jednej a pri pohľade v rôznych merítkach sa toto číslo nemení. Dĺžka pobrežia, tiež s topologickou dimenziou rovnou jednej, keďže ide o krivku, však so zmenšovaním merítka narastá. Táto krivka zaberá v rovine omnoho viac priestoru ako hladká krivka, a zároveň nie je rovinou. Preto je jej fraktálna dimenzia väčšia ako jeden, a zároveň menšia ako dva. Fraktálna Hausdorffova-Besicovitchova dimenzia je určená vzťahom

$$D = \lim_{\varepsilon \rightarrow 0} \frac{\log N(\varepsilon)}{\log\left(\frac{1}{\varepsilon}\right)}, \quad (2)$$

kde  $N(\varepsilon)$  je minimálny počet elementárnych útvarov (napríklad v  $R^3$  kocôčok) a  $\varepsilon$  označuje merítko (dĺžku strany daných kocôčok). Najjednoduchším príkladom je výpočet HB

dimenzie úsečky. Úsečku rozdelíme na  $N$  dielikov a merítka novovzniknutej úsečky potom je

$$\varepsilon = \frac{1}{N}. \quad (3)$$

Platí, že

$$N\varepsilon^D = 1, \quad (4)$$

a teda dimenzia  $D$  potom je

$$D = \frac{\log N}{\log(\frac{1}{\varepsilon})} = \frac{\log N}{\log N} = 1. \quad (5)$$

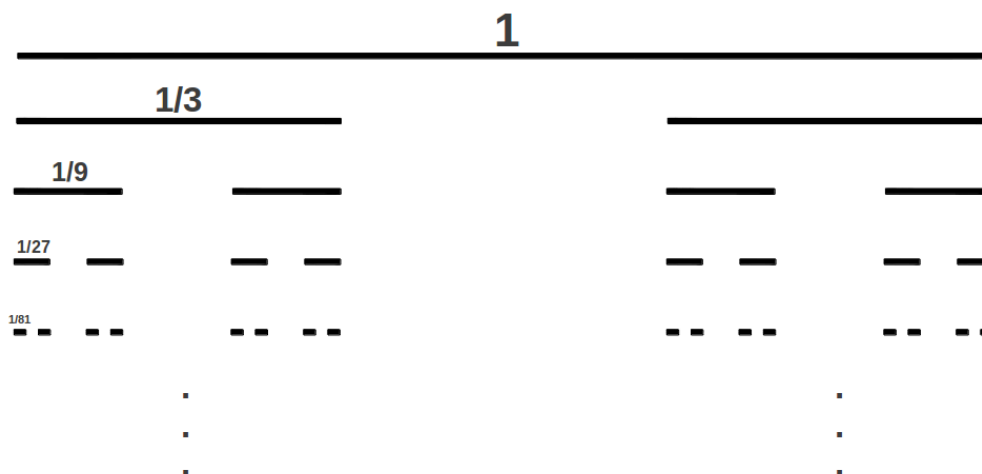
Ako je známe z Euklidovskej geometrie, topologická dimenzia úsečky je rovná jednej. V prípade štvorca, ktorý po dvojnásobnom zväčšení nadobúda štvornásobnú plochu, platí

$$\varepsilon = \frac{1}{N^{1/2}}, \quad (6)$$

a jeho dimenzia je rovná dvom. Ďalej pre kocku platí, že

$$\varepsilon = \frac{1}{N^{1/3}}, \quad (7)$$

a jej dimenzia je rovná trom. Najjednoduchším a najpreskúmanejším fraktálnym útvarom je Cantorova množina, reprezentovaná na Obrázku 6.

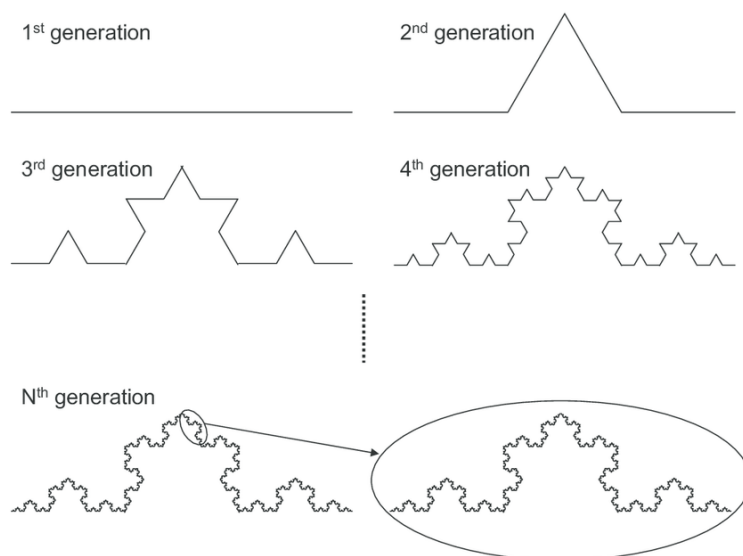


Obrázok 6: Cantorova množina. Dostupné z: <http://georgcantorbyelithompson.blogspot.com/2015/02/the-cantor-set.html>

Leží na intervale  $[0,1]$  a vzniká opakovaným vynechaním prostrednej tretiny všetkých intervalov, ktoré zvýšili po predchádzajúcej iterácii. Po vynechaní v prvom kroku ostávajú intervaly  $[0, 1/3]$  a  $[2/3, 1]$ , teda  $N = 2$  intervaly s merítkom  $\varepsilon = 1/3$ . Každý zo zostávajúcich intervalov sú podrobené rovnakej operácii, vznikajú tak 4 ďalšie intervaly a tak ďalej. Preto jej fraktálna dimenzia, podľa vzťahu (2), nadobúda hodnotu

$$D = \frac{\log 2}{\log 3} = 0,6309. \quad (8)$$

Táto množina je množina bodov s topologickou dimenziou 0, avšak skladá sa z nekonečného počtu bodov a jej HB dimenzia je neceločíselná. Na ploche je najjednoduchším fraktálom Kochova krivka, ktorej vytvorenie je zobrazené na Obrázku 7.



Obrázok 7: Kochova krivka. Dostupné z: [https://www.researchgate.net/figure/Generation-of-the-Koch-curve-produced-by-a-simple-iterative-transformation-beginning\\_fig4\\_47934714](https://www.researchgate.net/figure/Generation-of-the-Koch-curve-produced-by-a-simple-iterative-transformation-beginning_fig4_47934714)

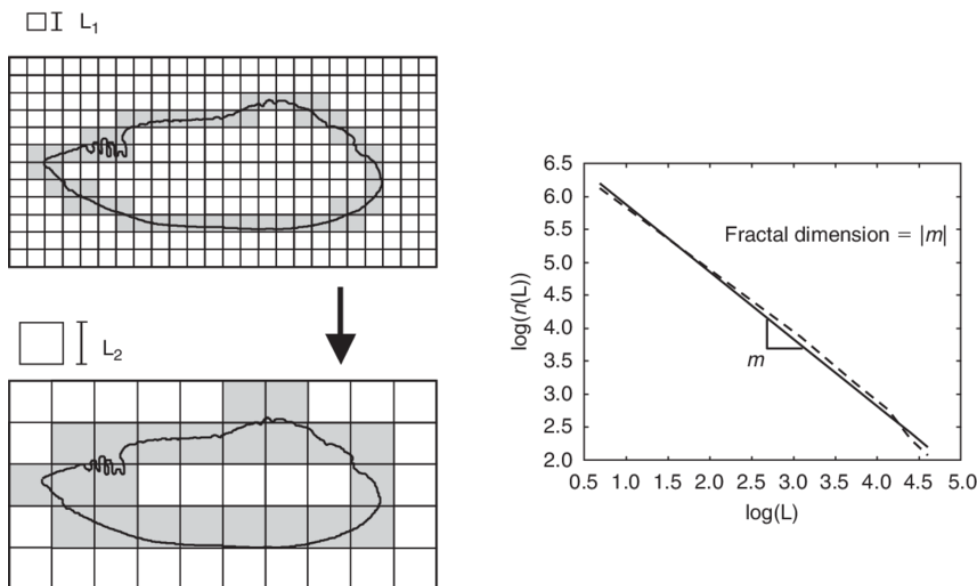
Dĺžka každej hrany sa rozdelí na tretiny, nad prostrednou tretinou sa zostrojí rovnostranný trojuholník a jeho základňa sa odstráni. Z pôvodnej hrany vznikajú takto 4 podobné úseky a postup sa opakuje na každom jednom z nich. Jej fraktálna dimenzia je

$$D = \frac{\log 4}{\log 3} = 1,2619. \quad (9)$$

Keďže fraktálna dimenzia tejto krivky je väčšia ako jej topologická dimenzia, ide o fraktál (Zelinka a kol. 2006).

### 2.3.4 Metóda počítania boxov

V prírode však tak pravidelné fraktály, aké boli spomenuté vyššie, neexistujú. Fraktálna dimenzia množín reálneho sveta potom musí byť určená experimentálne (Barnsley 1988b). Metóda počítania boxov (*the box counting method*) je najčastejšou metódou výpočtu fraktálnej dimenzie. Ide o postupné delenie obrázku mriežkou, kde je plocha alebo obvod objektu zmeraná naprieč rôznymi veľkosťami boxov mriežky (Korolj a kol. 2019). Ako je znázornené na Obrázku 8, objekt záujmu, v tomto prípade krivka, je preložený mriežkou. Počítané sú všetky boxy, v ktorých sa táto krivka nachádza. Boxov s veľkosťou  $L_1$  je 57, kdežto boxov s veľkosťou  $L_2$  je len 22.



Obrázok 8: Metóda počítania boxov a určenie fraktálnej dimenzie. Dostupné z: [https://www.researchgate.net/figure/Illustration-of-the-box-counting-method-for-estimating-the-fractal-dimension-of-the\\_fig2\\_228950126](https://www.researchgate.net/figure/Illustration-of-the-box-counting-method-for-estimating-the-fractal-dimension-of-the_fig2_228950126)

Všeobecne platí, že pri znižovaní sa boxov mriežky, stúpa počet boxov, v ktorých sa nachádza skúmaný objekt. Dáta sú potom vynesené do grafu s log-log škálou a smernica priamky aproximujúcej tieto dátové body je približnou hodnotou fraktálnej dimenzie objektu (Barnsley 1988b).

### 3 PRÍRODNÉ FRAKTÁLY A APLIKÁCIE TÝCHTO ŠTRUKTÚR

Prírodné štruktúry a vzory vykazujú často takú mieru iregularity a zložitosti, že klasická Euklidovská geometria zaostáva v schopnosti opísať ich. Príroda totižto vykazuje úplne iný druh komplexity (Mandelbrot 1982). Termín komplexita nie je jednoduché exaktne a jasne definovať. Vo všeobecnosti však ide o spôsob zmýšľania o kolektívnom správaní sa veľkého počtu jednotiek. Tieto jednotky medzi sebou interagujú a ich správanie sa vyvíja v čase. V tomto prípade je však celok viac než súhrn svojich častí, pretože interakcie týchto základných jednotiek dávajú vzniknúť tzv. emergentným vlastnostiam alebo priamo štruktúram, prinášajúcim často prekvapivý a neintuitívny vývoj komplexných systémov v čase. Komplexita sa rodí z nelinearity. Práve nelinearita je zodpovedná za zdanlivo náhodné chaotické správanie sa systémov a spôsobuje, že malé zmeny na určitých úrovniach organizácie vyvolávajú veľké zmeny na iných. S ich dôsledkami sa v živej prírode stretávame často a práve pohľad na problémy skrz komplexitu môže priniesť prirodzené a zaujímavé riešenia (Coveney a Highfield 2003)

Objekty spomenuté v Úvode, ako aj mnohé iné javy, s ktorými sa stretávame v bežnom živote, by mohli byť súhrnne pomenované ako prírodné fraktály. Termín prírodný fraktál totižto voľne opisuje prírodný vzor, štruktúru, ktorú môžeme reprezentovať fraktálnou množinou. Napríklad trajektórie Brownovho pohybu častíc sú fraktálnou množinou a fyzikálny Brownov pohyb je prírodným fraktálom (Mandelbrot 1982).

Väčšina prírodných fraktálov sa zaraďuje medzi neuniformné fraktály – ich dimenzia  $D$  a/alebo  $D_T$  nadobúda rozdielnych hodnôt pre rozdielne časti množiny. Naopak uniformné fraktály boli opísané v sekcii 2 (Cantorova množina, Kochova krivka, Sierpiňského trojuholník a iné). V prírode sa nachádza omnoho viac ďalších fraktálnych štruktúr a teórií ako sú opísané v tejto sekcii. Fraktálny pohľad na klastre galaxií, turbulenciu, dynamiku pohybu tekutín, či objekty ako špongie, mriežky a kvety a mnohé iné zaujímavé fenomény musíme však kvôli rozsahu práce opomenúť.

### 3.1 Pľúca

Ako sa ukazuje, v prírode je fraktálna štruktúra častým riešením mnoho problémov. Jednou z výhod fraktálnych objektov je ich veľká plocha v pomere k objemu objektu. Túto vlastnosť majú napríklad ľudské pľúca. Pľúca cicavcov sa skladajú zo zmenšujúcich a deliacich sa sebepríbuzných vetiev ukončených alveolami. Ľudské pľúca so svojou plochou tenisového kurtu, a objemom len okolo 6 litrov, zabezpečujú efektívnu výmenu dýchacích plynov (Havlin a kol. 1995). Bolo zistené, že fraktálna dimenzia stien dýchacích ciest astmatikov trpiacich sťaženým dýchaním bola nižšia, než v prípade zdravých pľúc. Naopak fraktálna dimenzia pacientov s idiopatickou pľúcnou fibrózou bola vyššia (Korolj a kol. 2019). Nielen geometria dýchacích ciest môže byť opísaná pomocou fraktálov. Časovo závislé vlastnosti dýchania vykazujú tiež fraktálne správanie (Havlin a kol. 1995).

### 3.2 Cievny systém

Cievny systém, ktorý rozvádza kyslík a živiny po celom tele, tak isto musí mať fraktálnu štruktúru, aby bol tento transport a ich výmena so splodinami metabolizmu efektívne zabezpečená pre každú bunku (Havlin a kol. 1995). Morfológia cievneho systému a jeho priestorová organizácia sa ale mení v chorobných stavoch. Sieť cievneho systému sietnice sa používa ako reprezentácia celého cievneho systému a poskytuje informácie nielen o zdraví očí, ale aj o vetvení ciev v celom tele (Korolj a kol. 2019).

### 3.3 Stromy

Podobne, ako u cievneho systému v sekcii 3.2, majú vetvenie stromov a rastlín, koreňový systém a žilnatina listov fraktálny charakter (Havlin a kol. 1995). Už Leonardo da Vinci si všimol, že súčet priemerov vetiev (hrúbka vetvy) na každej úrovni sa rovná priemeru vetvy pod nimi. Preto sa pri opise stromov, ale aj ciev alebo priedušiek, pridáva k fraktálnej dimenzii  $D$  aj exponent priemeru  $\Delta$  určujúci vzťah medzi priermi vetiev jednotlivých úrovní fraktálnej štruktúry.

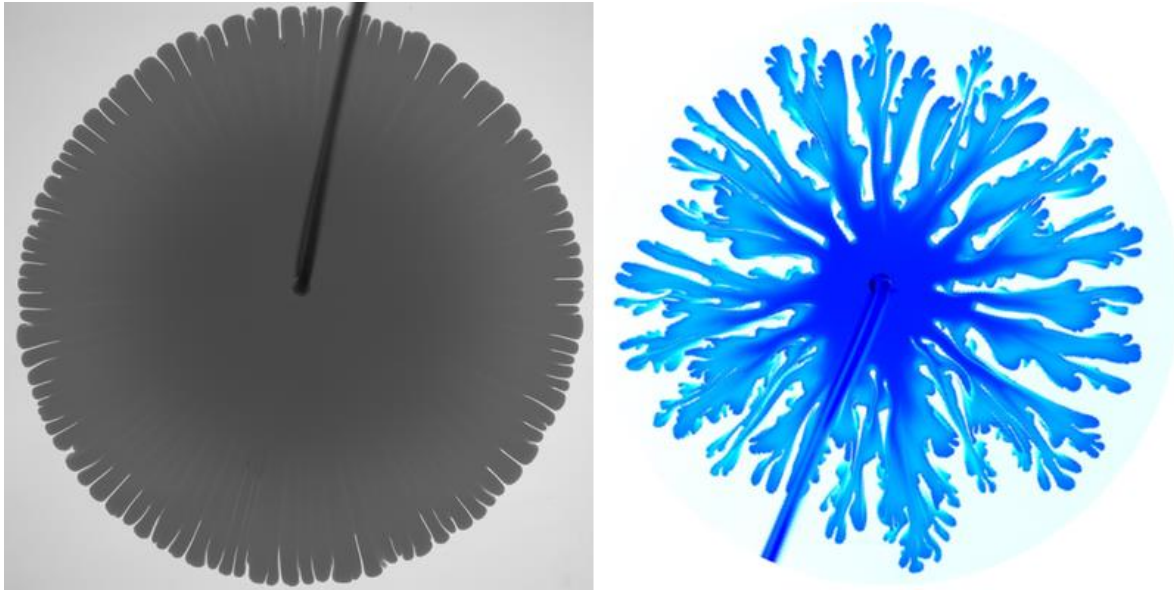
Tak isto ak les nie je nijak systematicky udržiavaný a je pozorovaný z výšky, stromy v ňom vytvárajú zhluky. Línie zhlukov stromov sa javia kostrbaté a ku každému zhlukovi je zaznamenaná aj jeho plocha. Tvorenie zhlukov sa však líši u jednotlivých druhov stromov, napríklad cyprusy sa zhlukujú omnoho viac ako niektoré listnaté stromy (Mandelbrot 1982).

### 3.4 Bunky a tkanivá

Viacero typov tumorov bolo klasifikovaných použitím výpočtu ich fraktálnej dimenzie na základe rozdielov morfológie jednotlivých buniek a tkanív. T-lymfocyty pri detekcii a rozpoznávaní antigénov počas „skenovania“ povrchu používajú fraktálne vzorce. Populácie kmeňových buniek sú kvôli svojím spôsobom delenia sa a migrácie považované za fraktálne siete (*fractal networks*). Kolektívne správanie sa buniek a ich migrácia sú v súčasnosti považované za kľúčové vo vysvetlení mechanizmov vzniku vetviacich sa tkanív v orgánoch, ako sú pľúca a obličky, alebo štruktúr ako mliečne žľazy a cievny systém. Od organizácie kryštálov minerálov a kolagénu v kostiach až po ich lamelárnu a trabekulárnu štruktúru hrajú fraktálne vzory dôležitú úlohu v evolúcií optimálnej tvrdosti a tuhosti kostí a odzrkadľujú niekoľko typov rôznych abnormalít a chorôb (Havlin a kol. 1995).

### 3.5 DLA systémy

Tvar neurónov a ich vývin môžu byť kvantifikované pomocou fraktálnych konceptov a rastových DLA (difúzne limitovaná agregácia – *diffusion limited aggregation*) modelov. Tieto DLA modely sa zdajú byť aj odpoveďou na otázku: „Prečo sa žalúdok sám nestrávi?“. Steny žalúdka sú pokryté vrstvou mukusu, ktorý slúži ako ochranná bariéra voči spätnej difúzií tráviacich štiav. Zabezpečuje totižto gradient pH od 1-2 v dutine po 6-7 na povrchu buniek steny. Ako však prechádza kyselina chlorovodíková zo žalúdočných žliaz do dutiny bez toho, aby zmenila pH vrstvy mukusu? Prechod kyseliny totižto zahŕňa tzv. viskózne prsty (*viscous fingering*), fenomén, pri ktorom tekutiny s nižšou viskozitou vstreknuté do tekutiny s vyššou viskozitou tvoria vzor v podobe rozbiehajúcich sa prstov (Obrázok 9).



Obrázok 9: Príklad viskózných prstov. Kvapalina s nižšou viskozitou vstreknutá do kvapaliny s vyššou viskozitou. Dostupné z: <https://physics.aps.org/articles/v10/60>

V prípade kyseliny chlorovodíkovej a mucínu ich tvorba závisí na pH. In vitro pokusy ukazujú, že kyselina tvorená žľazami prechádza mucínovou vrstvou s pH 5-7 vďaka tvorbe prstov. Naopak vyššia viskozita mukusu zabraňuje kyseline z dutiny žalúdka s pH 2 difundovať späť k bunkám steny žalúdka. Ďalším príkladom DLA systému je rast bakteriálnych kolónií na agarovom gély. Kolónie sa rozrastajú tvoriac sebepríbuzné štruktúry (Havlin a kol. 1995).

### 3.6 DNA

V ľudskom genóme len približne 5 % celkovej dĺžky kóduje proteíny. Gény sú od seba oddelené intergénovými sekvenciami (*intergenic sequences*), ktoré nekódujú proteíny a často sú tieto oblasti niekoľkokrát dlhšie ako samotné gény. Dokonca v samotných génoch sa nachádzajú inklúzie, ktoré opäť nekódujú proteíny. Tieto nekódujúce oblasti génu sa nazývajú intróny a kódujúce exóny (Havlin a kol. 1995).

Ďalekosahové mocninové korelácie (*long-range power law correlations*), vyskytujúce sa pri skúmaní až pozoruhodne veľkého množstva biologických systémov, boli objavené aj v DNA sekvenciách, použitím modelu náhodnej chôdza. DNA chôdza je definovaná pravidlom,



v ktorom „chodec“ postúpi nahor, ak sa v sekvencií objaví pyrimidín a zostúpi nadol, ak sa v sekvencií objaví purín. Takto vykreslená línia je veľmi zúbkovaná v prípade sekvencie obsahujúcej intróny, čo naznačuje prítomnosť ďalekosiahlych korelácií. V prípade cDNA (*complementary DNA*) toho istého génu a génu bez intrónov je línia menej zúbkovaná a vykazuje prítomnosť krátkodobých korelácií (Peng a kol. 1992).

### 3.7 Aplikácie

Schopnosť navrhnúť a vygenerovať fraktálne štruktúry slúži výborne pri výrobe biomateriálov, kedy mnohé z nich sú inšpirované prírodou. Niektoré ich časti ako polyméry, proteíny či kompozity sa správajú podľa škálovacieho zákona a vzorcov kritikalít. Pružné vodiče vytvorené vďaka samoorganizácii nanočastíc, fraktálne vzorované metalické vodiče s vylepšenou flexibilitou alebo vytváranie flexibilných a konformných 3D štruktúr z 2D listov silikónovej gúmy umožňujú vytvoriť nové užitočné vlastnosti pri zachovaní ich pôvodných výhod. Fraktálne vinutie metalických senzorov aj po ich umiestnení na mäkkú a pohyblivú kožu poskytuje zvýšenú odolnosť voči naťahovaniu a lepší prechod medzi elastickým a plastickým správaním materiálu.

Komponovanie fraktálnych vzorcov do klinickej praxe sa stáva čoraz častejším. Rehabilitačné zariadenia používajúce variabilnejšie vzorce v svojich stimuláciách vykazujú prirodzenejšie výsledky. Elektrické simulátory, pacemakery a mechanické ventilátory so zvýšenou variabilitou svojho pôsobenia disponujú terapeutickými benefitmi (Korolj a kol. 2019).

Ako už bolo vyššie spomenuté, vetviace sa fraktálne objekty, ako stromy alebo pľúca, s vysokým pomerom povrchu k objemu, ponúkajú kľúčové funkčné výhody. Napodobovanie týchto topológií prináša pokroky vo výrobe nových biomateriálov s proteínovým základom. Tieto sú po fosforylácii schopné vratného samozostavenia do fraktálnych štruktúr schopných zachytiť a aj vypustiť prenášané proteíny (*cargo proteins*) (Hernández a kol. 2019).

Využitie fraktálnej geometrie sa objavuje aj v kvantovej fyzike. Vlastnosti elektrónových kvantových systémov závisia na ich dimenziách. Poznatky o vplyve neceločíselnej dimenzie na správanie elektrónov boli dlhú dobu len teoretické. Táto situácia môže byť umelo vytvorená

kontrolovaným umiestnením molekúl CO na povrch Cu (111) pomocou skenovacieho tunelovacieho mikroskopu (STM). Molekuly CO vytvárajú odpudivý potenciál definujúci priestor, v ktorom sa môžu elektróny nachádzať. Elektróny, pobývajúce vo vymedzenom priestore tvaru Sierpiňského trojuholníka, majú dimenziu presne 1,58 ako je predpokladané pre tento fraktál (Kempkes a kol. 2019).

Fraktály zohrávajú tiež významnú úlohu v oblasti vizualizácie. Fraktálne algoritmy sa používajú pri kódovaní, či komprimácii digitálneho obrazu, ako aj pri simulácií reliéfov v počítačových hrách (Zelinka 2006). Richard Feynman tiež v roku 1979 vyhlásil, že fraktálne stromy mu pomohli predstaviť si a namodelovať tzv. trysky (*jets*), kužele častíc vytvorených pri hadronizácii kvarkov (Mandelbrot 1982).

### 3.8 Srdce

Srdce má fraktálny charakter prejavujúci sa v jeho štruktúrnom, ako aj časovom aspekte. Zo štruktúrneho hľadiska má srdcový sval svoju hierarchiu a jeho fraktálne vlastnosti sú založené na usporiadaní vlákien. Sebepodobnosť sa totižto objavuje na škále už od molekulárnych filamentov aktínu a myozínu spájajúcich sa do myofibrilov, myofibrilov tvoriacich myocyty a zväzky myocytov zhromažďujúcich sa do vlákien tkaniva srdcového svalu. Takáto vysoko organizovaná hierarchická štruktúra je nevyhnutná k správne fungovaniu svalu pri synchronizovanej kontrakcii. Podobne Hisov zväzok a Purkiňove vlákna, patriace k prevodovému systému srdca, tvoria sieť, ktorá je fraktálom (Korolj a kol. 2019).

Normálna elektrická aktivita srdca je tradične označovaná termínom „regulárny sínusový rytmus“. Avšak intervaly medzi dvoma údermi srdca zdravých jedincov fluktuujú v zdanlivo nepravidelnom a komplexnom štýle, aj počas ich zotrvania v kľude. Naopak u chorých jedincov (trpiacich kardiomyopatiou) sú tieto fluktuácie omnoho miernejšie. V prípade zdravých jedincov sú nepravidelné fluktuácie výhodou, pretože predchádzajú neprímerane nemennému režimu obmedzujúcemu funkčné odozvy organizmu. Podobné prechody do periodických režimov boli pozorované aj pri širšom spektre ďalších chorôb, ako sú epilepsia, náhla srdečná smrť (*sudden cardiac death*), niektoré zhubné nádory a syndróm spomalenia

rastu plodu (*fetal distress syndrome*). Teda ďalekosiahle korelácie prítomné v zdravom srdečnom tepe indikujú, že neuroautonómne kontrolné mechanizmy sa snažia odvieť systém čo najďalej od jedného rovnovážneho stavu (Havlin a kol. 1995, Peng a kol. 1995).

V tejto práci ďalej bude skúmaná práve dynamika tlkotu ľudskeho srdca cez fraktálnu analýzu nestacionárnych časových radov intervalov medzi jednotlivými údermi srdca (R-R intervalov).

## 4 FRAKTÁLY V ČASOVÝCH RADOCH

Ako bolo uvedené v predchádzajúcich sekciách, existuje niekoľko metód kvantifikovania fraktálnych charakteristík pomocou fraktálnej dimenzie. Príkladom by mohla byť metóda počítania boxov aplikovaná na pobrežia či koreňové sústavy rastlín, zameriavajúca sa na tvar objektov a množín. Časové rady sú však tvorené postupnosťou hodnôt zaznamenaných v čase, ako napríklad ceny akcií na burze, fluktuácie atmosférické tlaku alebo niektoré fyziologické signály. „Tvar“ takýchto radov nie je jasne definovateľný a zvyčajne ide o časové rady vykazujúce štatistickú sebedobnosť alebo sebebízanosť. Preto bolo nutné prísť s inými technikami skúmania týchto fraktálnych charakteristík. V sekciách 4.1 a 4.2 sú predstavené dve rôzne metódy analýzy fraktálnych časových radov, ktoré budú aplikované na fyziologický signál – intervaly medzi údermi srdca (R-R intervaly).

### 4.1 Detrendovaná fluktuačná analýza DFA (*Detrended Fluctuation Analysis*)

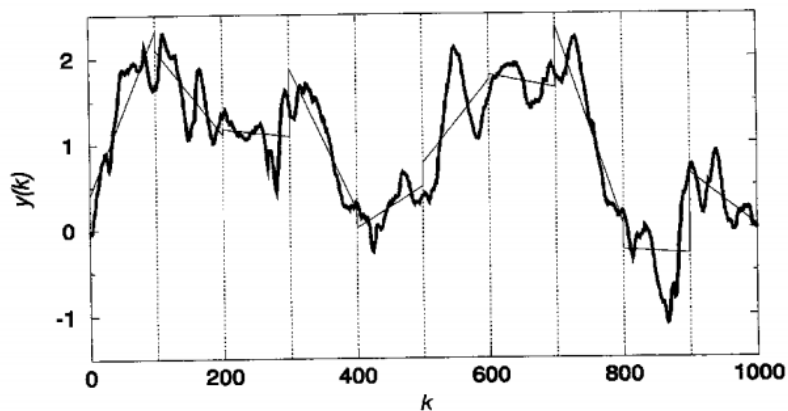
Detrendovaná fluktuačná analýza je metóda navrhnutá pre analýzu časových radov, v ktorých sa nachádzajú fenomény propagujúce sa časovým radom na veľké vzdialenosti. Metóda bola predstavená v roku 1994 (Peng a kol. 1994) a bola navrhnutá pre nestacionárne časové rady. Výsledkom je exponent  $\alpha$  podobný Hurstovmu exponentu. Ten je pomenovaný podľa H. E. Hursta, ktorého meno je spojené so štatistickou metódou jeho vlastného vynálezu. Hurst bol fascinovaný riekou Níl a pri skúmaní jej každoročného prietoku si ako prvý uvedomil dôležitosť výstavby ďalších nádrží kľúčových pre Egypt počas suchých rokov (Mandelbrot 1982). Zatiaľ čo Hurstov exponent figuruje pri analýze stacionárnych časových radov, DFA je aplikovateľná na nestacionárne signály, akými sú napríklad intervaly medzi jednotlivými údermi srdca. Tento príklad použijeme k vysvetleniu základných princípov metódy DFA.

Na analýzu fluktuácií srdečného tepu sa používajú digitalizované elektrokardiogramy. Z nich sú vyčítané intervaly medzi jednotlivými údermi srdca  $i$  a  $i+1$ , označené ako  $B(i)$ .

Tie sú najskôr integrované podľa vzťahu

$$y(k) = \sum_{i=1}^k [B(i) - B_{ave}], \quad (10)$$

kde  $B_{ave}$  je priemerná hodnota intervalov. Integrovaný časový rad  $y(k)$  je rozdelený do úsekov rovnakej dĺžky  $n$ . Na dátach v každom z týchto úsekov osobitne je metódou najmenších štvorcov skonštruovaný fit. Dáta sú ďalej zbavené trendu jednoduchým odčítaním  $y$  súradnice trendu  $y_n(k)$  v jednotlivých boxoch od integrovaného časového radu  $y(k)$ . Tento proces je ilustrovaný v Obrázku 10.



Obrázok 10: Integrovaný časový rad  $y(k)$ . Vertikálne čiary znázorňujú šírku okien  $n = 100$  a rovné čiary na týchto úsekoch znázorňujú trend odhadnutý pomocou metódy najmenších štvorcov. Prevzaté z Peng a kol. (1995)

Pre každú zvolenú dĺžku úsekov  $n$  je určená miera fluktuácie podľa vzťahu

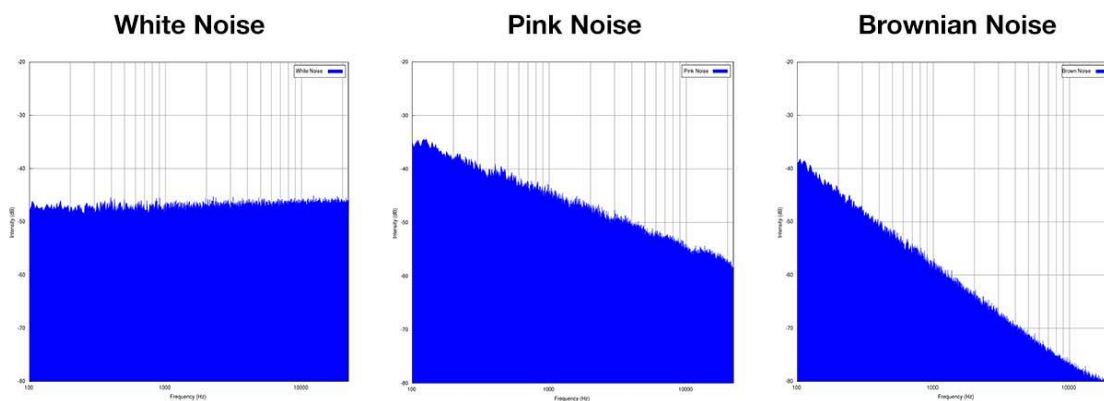
$$F(n) = \sqrt{\frac{1}{N} \sum_{k=1}^N [y(k) - y_n(k)]^2}, \quad (11)$$

kde  $N$  je celkový počet intervalov súboru. Tento výpočet je zopakovaný naprieč viacerými škálami – dĺžkami úsekov  $n$  – a slúži k odhaleniu vzťahu medzi fluktuáciami  $F(n)$  a zvolenou dĺžkou úsekov  $n$ . Vo všeobecnosti  $F(n)$  so zväčšujúcim sa  $n$  tiež zväčšuje a po vynesení hodnôt do log-log grafu je zjavný lineárny vzťah naznačujúci sebedobnosť medzi integrovanými a detrendovanými časovými radmi. Fluktuácie na malých úsekoch sú podobné fluktuáciám na väčších úsekoch mocninovým spôsobom. Smernica lineárnej rovnice popisujúcej vzťah medzi logaritmom  $F(n)$  a logaritmom  $n$  sa označuje ako škálovací

exponent (parameter sebedobnosti)  $\alpha$  (Bianchi 2020, Peng a kol. 1995). Exponent  $\alpha$  nadobúda charakteristických hodnôt uvedených v Tabuľke 1.

Tabuľka 1: Hodnoty škálovacieho exponentu a ich interpretácie

škálovací exponent $\alpha$	charakteristika procesu
$0 < \alpha \leq 0,5$	antiperzistentné ďalekodosahové mocninové korelácie
$\alpha = 0,5$	biely šum, hodnoty sú nekorelované
$0,5 < \alpha \leq 1$	perzistentné ďalekodosahové mocninové korelácie
$\alpha = 1$	1/f šum (ružový)
$\alpha > 1$	korelácie existujú, nie však mocninové
$\alpha = 1,5$	Brownov šum (hnedý), integrovaný biely šum



Obrázok 11: Spektrá bieleho, ružového a Brownovho šumu. Na ose  $x$  sa nachádza frekvencia v jednotkách Hz s logaritmickou škálou a na ose  $y$  intenzita v jednotkách dB. Dostupné z: [https://www.xpressocommunications.com/client-news/beyond-music-our-response-to-sound/attachment/blog-february\\_white-pink-brown-noise/](https://www.xpressocommunications.com/client-news/beyond-music-our-response-to-sound/attachment/blog-february_white-pink-brown-noise/)

Tento exponent slúži aj ako indikátor drsnosti pôvodných časových radov – čím vyššia je jeho hodnota, tým hladšie sú časové rady. Taktiež je možné si všimnúť, že 1/f šum je kompromisom medzi úplnou nepredpovedateľnosťou a drsnosťou bieleho šumu a omnoho hladším priebehom hnedého šumu (Peng a kol. 1995).

#### 4.1.1 Interpretácia metódy DFA aplikovanej na intervaly medzi údermi srdca

K popisu fluktuácií srdcového rytmu sa používajú ich distribučné charakteristiky, ako napríklad stredná hodnota a rozptyl. Tieto však nenesú žiadnu informáciu o dynamike, teda o postupnom vývoji v čase. Intervaly medzi jednotlivými údermi srdca fluktuujú komplexným spôsobom. S použitím techník štatistickej fyziky je možné odhaliť tzv. ďalekodosahové mocninové korelácie pretrvávajúce počas tisícok srdečných úderov. Toto správanie je označované ako so škálou nemenné (*scale-invariant*) – fraktálne. Škálovo nemenné vlastnosti biologických systémov sa ukazujú ako dôležité a úzko prepojené so schopnosťou adekvátne reagovať a prispôbiť sa zmene. Ich dôsledkom sú práve ďalekosiahle korelácie ako výsledok kolektívneho správania sa komplexného systému skladajúceho sa z viacerých komponentov, ktoré sa navzájom ovplyvňujú cez lokálne (*short-range*) interakcie (Peng a kol. 1995).

Mechanizmus zodpovedný za fluktuácie sa javí byť prepojený s autonómnym nervovým systémom (ANS). Stimulácia parasympatiku znižuje aktivitu buniek udávajúcich srdcový rytmus v sinoatriálnom uzle prevodového systému srdca. Stimulácia sympatiku má zas opačný dopad. Predpokladá sa, že práve interakcie medzi týmito dvomi vetvami ANS sú dôvodom zdanlivo náhodných fluktuácií a variability srdcového rytmu u zdravých jedincov. U zdravých jedincov prítomnosť ďalekodosahových korelácií v srdcovom rytme poukazuje na fakt, že nervové autonómne a centrálné kontrolné mechanizmy sa snažia systém odvieť od jedného rovnovážneho stavu. Literatúra ďalej uvádza, že prítomnosť Brownovho šumu alebo veľmi periodické správanie sa systému je spojené s patologickým stavom zlyhania srdca a naopak nekorelované správanie (biely šum) je spojené s niektorými srdcovými arytmiami, akou je napríklad fibrilácia (Peng a kol. 1995).

Rozdielne správanie sa ďalekodosahového škálovania je dostatočne význačné na rozlíšenie zdravých a patologických stavov. Tieto ďalekodosahové korelácie naznačujú prítomnosť komplexných nelineárnych procesov generujúcich fluktuácie na viacerých škálach. Práve absencia jednej charakteristickej škály pomáha zamedziť organizmu prehnane ľpiet' na jednom režime obmedzujúcom jeho vlastnú funkčnú odozvu na podnety okolia (Peng a kol. 1995).

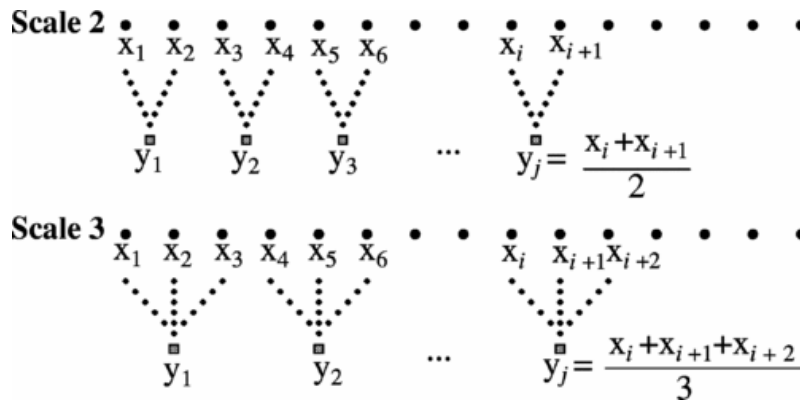
## 4.2 Multiškálová Entropia MSE (*Multiscale Entropy*)

MSE analýza bola navrhnutá ako metóda kvantifikácie komplexity signálu, či už fyzikálneho alebo fyziologického. Jej výhodou je fakt, že zohľadňuje kvantifikovanie komplexity signálu nielen na jednej časovej škále ale na viacerých zároveň. Táto analýza používa algoritmus založený na myšlienke entropie asociovanej s kvantifikovaním miery komplexity fyziologických signálov, užitočnej práve pri rozlišovaní medzi zdravými a chorobnými stavmi v medicínskom obore (Costa a kol. 2002a).

Majme časovú radu  $\{x_1, \dots, x_N\}$ . K nej sú skonštruované časová rady kľzavého priemeru (*coarse-grained time series*)  $y_j^{(\tau)}$ , znázornené na Obrázku 12, jednoduchým spriemerovaním jednotlivých dátových bodov v neprekrývajúcich sa segmentoch podľa vzťahu

$$y_j^{(\tau)} = \frac{1}{\tau} \sum_{i=(j-1)\tau+1}^{j\tau} x_i. \quad (12)$$

V rovnici  $\tau$  predstavuje škálovací faktor a  $1 \leq j \leq N/\tau$ . Nasledovne je určená tzv. vzorková entropia (SampEn) pre každú časovú radu  $y_j^{(\tau)}$ , počínajúc pôvodnou časovou radou. Hodnoty SampEn príslušné jednotlivým škálovacím faktorom  $\tau$  sú potom vynesené do grafu a ich trend je ďalej skúmaný (Costa a kol. 2002a).



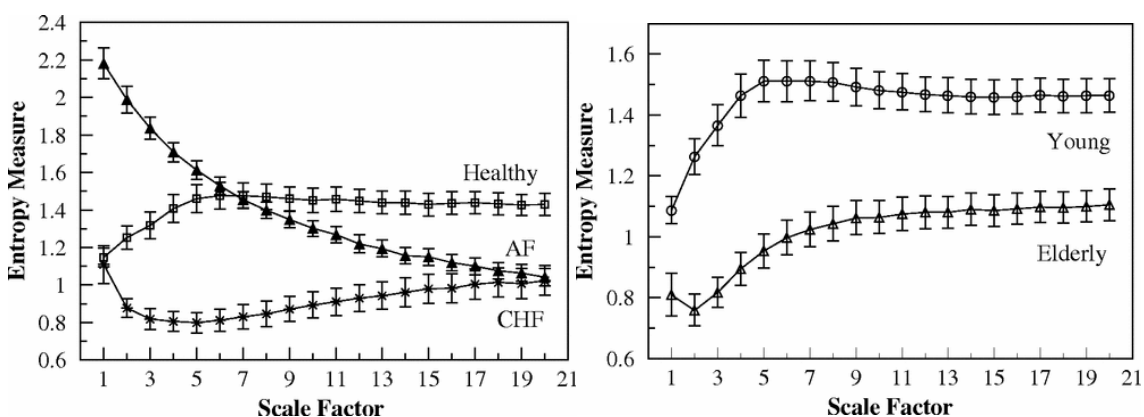
Obrázok 12: Názorné vytvorenie časových radov kľzavého priemeru pre škály 2 a 3. Prevzaté z Costa a kol. (2005)



Algoritmus SampEn bol navrhnutý na stanovenie pravidelnosti dátových rád na základe existencie vzorcov v týchto radoch. Jeho výhodou je, že bez akejkoľvek vedomosti o tom, z akého zdroja dáta pochádzajú, dokáže spoľahlivo odhadnúť náhodnosť dátovej rady. Analýza náhodnosti dátových rád vychádza z matematickej Teórie Informácie, ktorej základy spísal Claude E. Shannon. Fundamentálnym konceptom tejto teórie je entropia, kvantifikujúca množstvo informácie a stupeň náhodnosti systému. John von Neuman navrhol Shannonovi tento nový koncept nazvať entropia, aj keď nesúvisí s fyzikálnou veličinou entropia definovanou Clausiusom a Boltzmanom v termodynamike a štatistickej mechanike. Entropia informačnej teórie nie je funkciou hodnôt dátovej sady, ale funkciou ich pravdepodobnosti výskytu. Tak isto neudáva či je dátový rad náhodný alebo nie, ale ako moc náhodný je. Vo všeobecnosti, ak sa v dátových radoch nachádzajú úseky, ktoré sa opakujú, sme schopní tieto vzory vyhľadať a dátovú sadu komprimovať s ich využitím – radu považujeme za nekompexnú. Naopak, ak je dátová sada úplne náhodná, vzory v nej nenájdeme, a teda je považovaná za komplexnú. Záujem netkvie v zisťovaní celkovej entropie systému generujúceho túto radu, ale v pozorovaní ako sa entropia mení v čase. Jej výpočet a využitie pre obmedzené, zašumené a stochastické časové rady však neboli veľmi úspešné (Delgado-Bonal a Marshak 2019). Preto bola SampEn navrhnutá ako štatistika na analýzu náhodnosti sérií cez priame využitie korelačných integrálov (Richman a Moorman 2000). Jej algoritmus je skoro nezávislý na dĺžke časovej rady a pracuje priamo s hodnotami dátovej rady namiesto s pravdepodobnosťami ich výskytu. Jej nízke hodnoty popisujú systém ako jednostajný a predpokladateľný s opakujúcimi sa úsekmi, zatiaľ čo vysoké hodnoty značia náhodnosť a nezávislosť dát. V jej výpočte figurujú parametre  $m$  (*embedding dimension*) a  $r$  (*noise filter*), kde  $m$  je dĺžka porovnávaných vektorov a so zväčšujúcim sa  $r$  algoritmus nachádza viac vzorov (Delgado-Bonal a Marshak 2019). Úloha týchto parametrov bude priblížená v sekciách 6.4 a 7.3.

#### 4.2.1 Interpretácia metódy MSE aplikovanej na intervaly medzi údermi srdca

Z literatúry je známe, že korelované časové rady s  $1/f$  šumom vykazujú vyššiu komplexitu ako nekorelované časové rady s bielym šumom. Dynamika intervalov medzi údermi srdca zdravých subjektov je považovaná za komplexnejšiu než u jedincov s nejakou patológiou. Ako je znázornené na Obrázku 13 vľavo, hodnoty entropie zdravých subjektov na malých škálach stúpajú a postupne sa stabilizujú na relatívne konštantnej hodnote. U subjektov s kongestívnym zlyhaním srdca hodnoty najskôr na malých škálach klesajú a potom pozvoľne stúpajú. Krivka pre subjekty s fibriláciou predsiení najviac pripomína tvar krivky pre biely šum – hodnoty entropie monotónne klesajú (Costa a kol. 2002b, Costa a kol. 2005). Tak isto komplexita časových rád tlkotu srdca s vekom subjektov klesá, znázornené na Obrázku 13 vpravo (Costa a kol. 2002b).



Obrázok 13: vľavo MSE krivky zdravých subjektov (Healthy), subjektov s fibriláciou predsiení (AF) a kongestívnym zlyhaním srdca (CHF); vpravo MSE krivky mladých zdravých subjektov (Young) a starších zdravých subjektov (Elderly). Prevzaté z Costa a kol. 2002b

## 5 CIEĽ PRÁCE

Existuje mnoho rôznych matematických prístupov ku kvantifikácii fraktálnych charakteristík signálov a štruktúr. Cieľom práce je preskúmať možnosti uplatnenia týchto nástrojov v analýze biologických signálov a štruktúr. Nosnou témou je dynamika tlkotu ľudského srdca skúmaná cez fraktálnu analýzu nestacionárnych časových radov intervalov medzi jednotlivými údermi srdca (R-R intervalov) v programovacom jazyku Python.

## 6 METÓDY

### 6.1 Získanie EKG záznamov

PhysioNet.org (PhysioNet 2021) je platforma slúžiaca ako zdroj komplexných fyziologických signálov (Goldberger a kol. 2000). Ponúka prístup k databázam, softvérom, príručkám a výzvam zahŕňajúcim fyziologické signály ako EKG, EEG, sluchové odozvy, priebeh krvného tlaku a iné.

Každá EKG databáza sa skladá zo sady nahrávok identifikovaných názvom nahrávky. Každý názov má k sebe niekoľko súborov s rôznou príponou. Súbor s príponou `.dat` obsahuje binárny zápis jedného alebo viacerých zdigitalizovaných signálov. Súbor s príponou `.hea` je krátky textový súbor, hlavička obsahujúca názov, formát, počet signálov, vzorkovaciu frekvenciu, trvanie nahrávky či čas spustenia nahrávania. Súbor s príponou `.atr` je opäť binárny súbor obsahujúci značky, anotácie, charakteristik signálu, akými sú napríklad QRS komplexy, maximá T, či P vln a iné. Niekedy sa anotácie QRS komplexov vyskytujú osobitne v súboroch s príponou `.qrs`. K analýze signálov bola vytvorená WFDB knižnica, ktorá má implementácie v rôznych programovacích jazykoch. V tejto práci je používaný programovací jazyk Python.

Z platformy Physionet.org boli zadovážené 2 databázy – databáza MIT-BIH normálneho sínusového rytmu (MIT-BIH Normal Sinus Rhythm Database 2021) a databáza MIT-BIH fibrilácie predsiení (Moody a Mark 1983, MIT-BIH Atrial Fibrillation Database 2021).

Databáza MIT-BIH normálneho sínusového rytmu (*MIT-BIH Normal Sinus Rhythm Database*) obsahuje 18 dlhodobých nahrávok EKG zaobstaraných v nemocnici Beth Israel v Bostone. Subjekty v tejto databáze nevykazujú žiadne signifikantné arytmie. Nachádza sa v nej 5 mužov vo veku 26 až 45 rokov a 13 žien vo veku 20 až 50 rokov. Jednotlivé nahrávky sú dlhé približne 21 hodín a zaznamenané so vzorkovacou frekvenciou 128 Hz.

Databáza MIT-BIH fibrilácie predsiení (*MIT-BIH Atrial Fibrillation Database*) obsahuje 25 dlhodobých EKG záznamov subjektov trpiacich fibriláciou predsiení (zväčša paroxyzmálnou). Jednotlivé nahrávky sú dlhé 10 hodín a zaznamenané so vzorkovacou frekvenciou 250 Hz.

## 6.2 Určenie R-R intervalov

Z každej z databáz bolo vybraných 15 subjektov, nahrávky boli načítané a z nich boli získané dĺžky intervalov medzi jednotlivými údermi srdca ako vzdialenosti medzi dvomi R vrcholmi QRS komplexov. Pre subjekty s normálnym sínusovým rytmom (skupina N) sú R vrcholy nájdené pomocou algoritmu WFDB knižnice, kdežto pre subjekty s fibriláciou predsiení (skupina AF) sú pozície píkov odčítané z anotácií. Detekcia QRS komplexov pomocou algoritmov nie je triviálnou záležitosťou. Detektor WFDB knižnice používa integráciu pohybujúcou sa vlnou MWI (*moving wave integration*) s Rickerovou vlnkou (*Ricker wavelet*). Na prvých QRS komplexoch zistí parametre, spustí hlavnú detekciu komplexov signálu a iteruje cez lokálne maximá MWI signálu. Skontroluje, či ozaj ide o QRS komplex (hodnoty musia byť nad určitým prahom, nesmie ísť o T vlnu a pod.) a ak áno, aktualizuje počiatkové „naučené“ parametre. Ak nie, klasifikuje komplex ako šum a taktiež aktualizuje parametre. Algoritmus dokonca dokáže spätne vyhľadať komplexy znížením prahu, ak by sa stalo, že ďalší QRS komplex nebol detegovaný do 1,66-násobku posledného R-R intervalu (WFDB documentation 2021).

Takto je potom vytvorený časový rad intervalov medzi dvomi susediacimi údermi srdca. Na tieto intervaly s počtom dátových bodov  $N = 2 \cdot 10^4$  sú ďalej aplikované algoritmy MSE a DFA.

## 6.3 Metóda DFA

V sekcii 4.1 je opísaná metóda DFA, ktorá je aplikovaná na časové rady s využitím balíka *fathon* (Bianchi 2020). Balík ponúka rýchlu, efektívnu a flexibilnú analýzu a výpočet škálovacieho exponentu  $\alpha$ . Tento exponent je určený pre každý subjekt a následne je vypočítaný priemerný koeficient pre skupinu N a AF. Ďalej sú určené koeficienty  $\alpha_1$  a  $\alpha_2$  na úsekoch s malými a veľkými dĺžkami okien  $n$  aj ich priemerná hodnota pre obe skupiny subjektov. DFA je taktiež aplikovaná na biely a 1/f šum. Celý kód použitia tejto metódy sa nachádza v Prílohe 1.

## 6.4 Metóda MSE

Ako je uvedené v sekcii 4.2, z pôvodného časového radu sú procesom kľzavého priemeru vytvorené časové rady škály 1 až 20. Ku každému radu je vypočítaná hodnota SampEn (Sample entropy 2021), kedy sa tento výpočet realizuje v programovacom jazyku Python cez:

```
1 def sampen(L, m, r):
2     N = len(L)
3     B = 0.0
4     A = 0.0
5
6     # Tu xmj je JEDNA array vektorov ktoré odčítam od VŠETKÝCH xmi
7     # Rozdelí časový rad a uloží templates dĺžky m
8     xmi = np.array([L[i : i + m] for i in range(N - m)]) #tak aby
9     # bolo možné vytvoriť m+1 vektor
10    xmj = np.array([L[i : i + m] for i in range(N - m + 1)])
11
12    # Uloží matches a vylúči self-match, vypočíta A
13    # np.abs+max(axis = 1) --> najväčšia vzdialenosť cez akúkoľvek
14    # súradnicu
15    # -1 vyhodí self-match
16    A = np.sum([np.sum(np.abs(xmii - xmj).max(axis=1) <= r) - 1 for
17    xmii in xmi])
18
19    # Podobne pre B
20    m += 1
21    xm = np.array([L[i : i + m] for i in range(N - m + 1)])
22
23    B = np.sum([np.sum(np.abs(xmi - xm).max(axis=1) <= r) - 1 for xmi
24    in xm])
25
26    # Vráti SampEn
27    return -np.log(B/A)
```

(celý kód analýzy dát sa nachádza v Prílohe 1). Princíp metódy spočíva v rozdelení časového radu na vektory o veľkosti  $m$ . Prvý vektor, slúžiaci ako vzor (*template*), o veľkosti  $m$  je porovnávaný so všetkými nasledujúcimi vektormi. Ich vzájomná vzdialenosť je meraná Čebyševovou/max normou. Ak sa vektory nachádzajú dostatočne „blízko“ (ich vzdialenosť musí byť  $\leq r$ ) algoritmus zaznamená zhodu. Následne je druhý vektor v poradí vybraný ako vzor a je porovnaný so zvyškom vektorov. Suma všetkých zhôd pre veľkosť vektoru je označená ako  $A$ . Celý postup je zopakovaný pre vektory o veľkosti  $m+1$  a suma všetkých

zhôd je označená ako  $B$ . Hodnota SampEn pre časový rad je braná ako prirodzený logaritmus podielu sumy zhôd  $A$  a sumy zhôd  $B$ .

Hodnoty SampEn vypočítané pre každý subjekt sú vynesené do grafu v závislosti na škále za vzniku MSE krivky. Krivky sú porovnané medzi oboma skupinami. Vlastnosti MSE sú skúmané pri zmene parametrov  $m$  a  $r$  a pri aplikácií na biely a  $1/f$  šum.

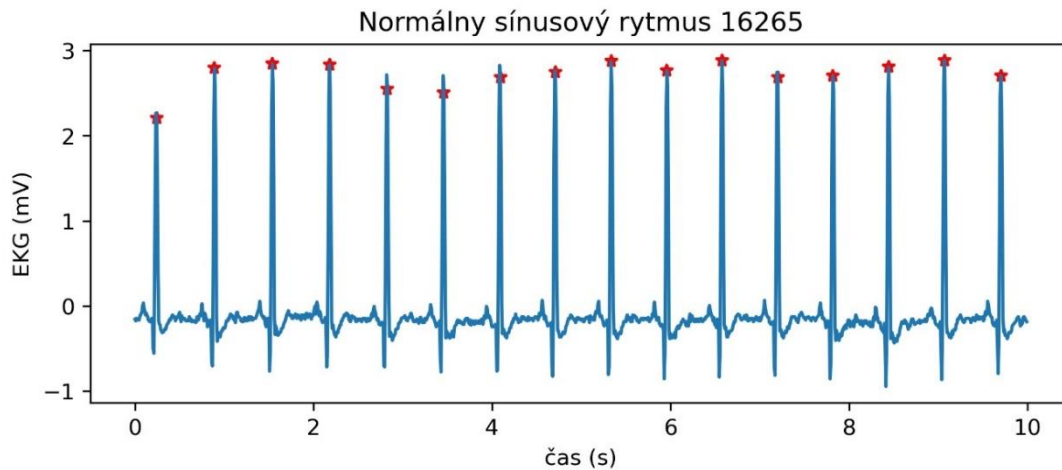
## 6.5 Charakteristický DFA exponent a MSE krivka

Zo skupiny N bolo vybraných následne 6 subjektov. Časový rad dĺžky intervalov medzi údermi srdca bol o veľkosti  $N = 8 \cdot 10^4$  bol rozkúskovaný na 8 častí po  $N = 1 \cdot 10^4$  dátových bodoch. Na každom úseku bola určená priemerná hodnota SampEn pre škály 8-20 a bol určený exponent  $\alpha_2$ . Tieto dvojice boli potom vynesené do grafov a označené odlišnou farbou pre každý subjekt osobitne. Cieľom je zistiť, či je na základe metód DFA a MSE možné rozoznať jednotlivé subjekty ak sa budeme pozerat' na neprekrývajúce sa úseky časových radov dĺžky ich R-R intervalov. Celý kód analyzujúci dáta sa nachádza v Prílohe 2.

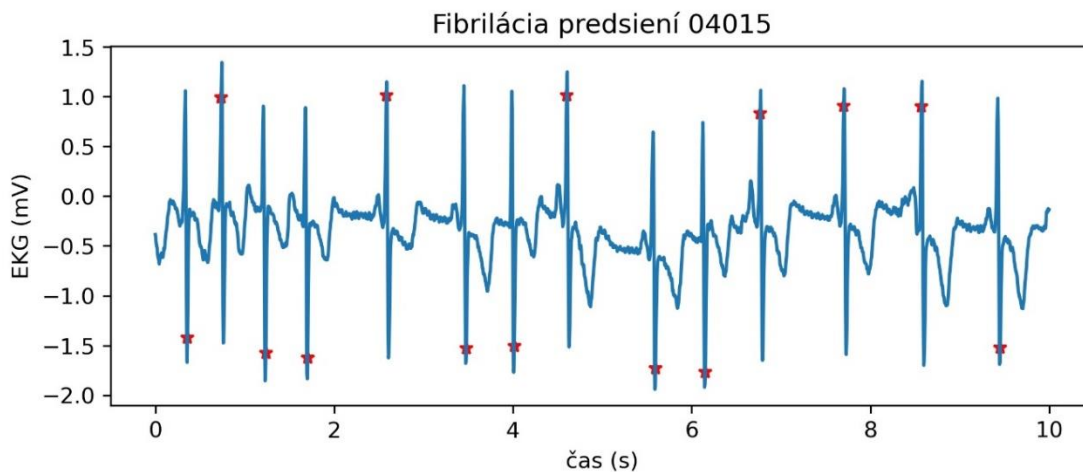
## 7 VÝSLEDKY A DISKUSIA

### 7.1 Získanie R-R intervalov z EKG záznamov

V skupinách N a AF boli EKG záznamy vynesené do grafov a boli nájdené R píky QRS komplexov. V Grafe 1 sa nachádza ukážka 10 sekúnd záznamu subjektu 16265 zo skupiny N. V Grafe 2 sa nachádza tiež 10 sekúnd dlhý EKG záznam subjektu 04015 z AF skupiny. Poloha R píkov v oboch grafoch je vyznačená červenou hviezdíčkou.



Graf 1: EKG záznam dlhý 10 sekúnd, subjekt s normálnym sínusovým rytmom. Červené hviezdíčky značia polohu R píkov.

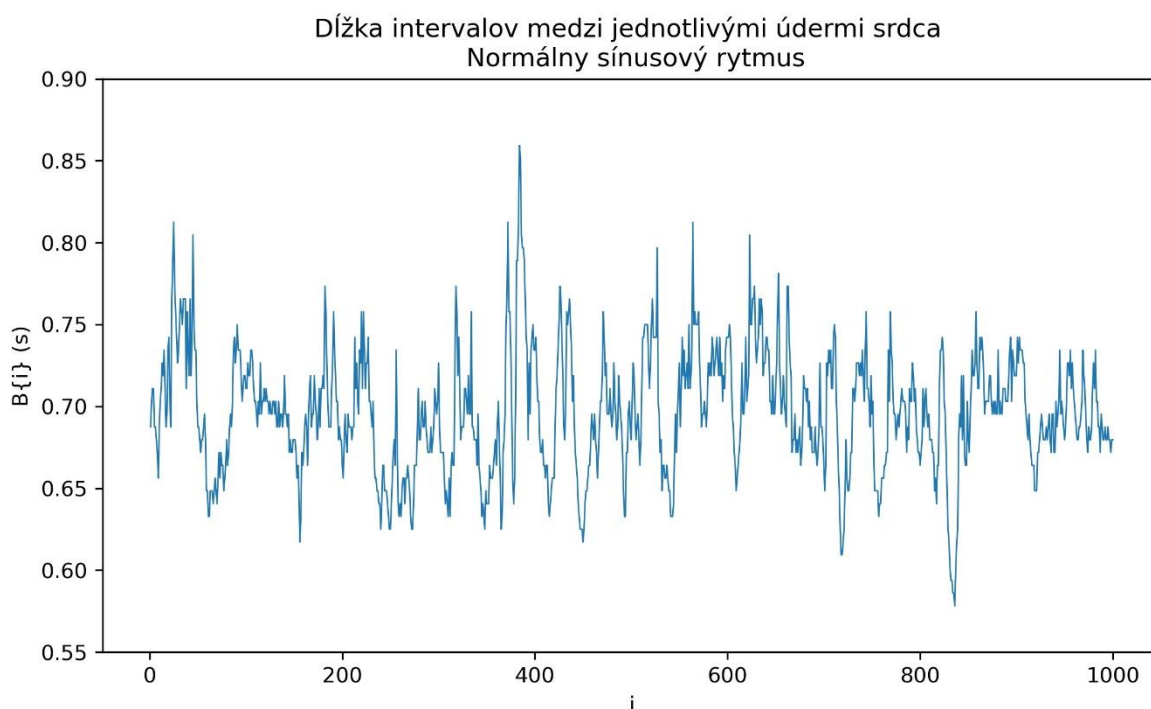


Graf 2: EKG záznam dlhý 10 sekúnd, subjekt s fibriláciou predsiení. Červené hviezdíčky značia polohu R píkov.



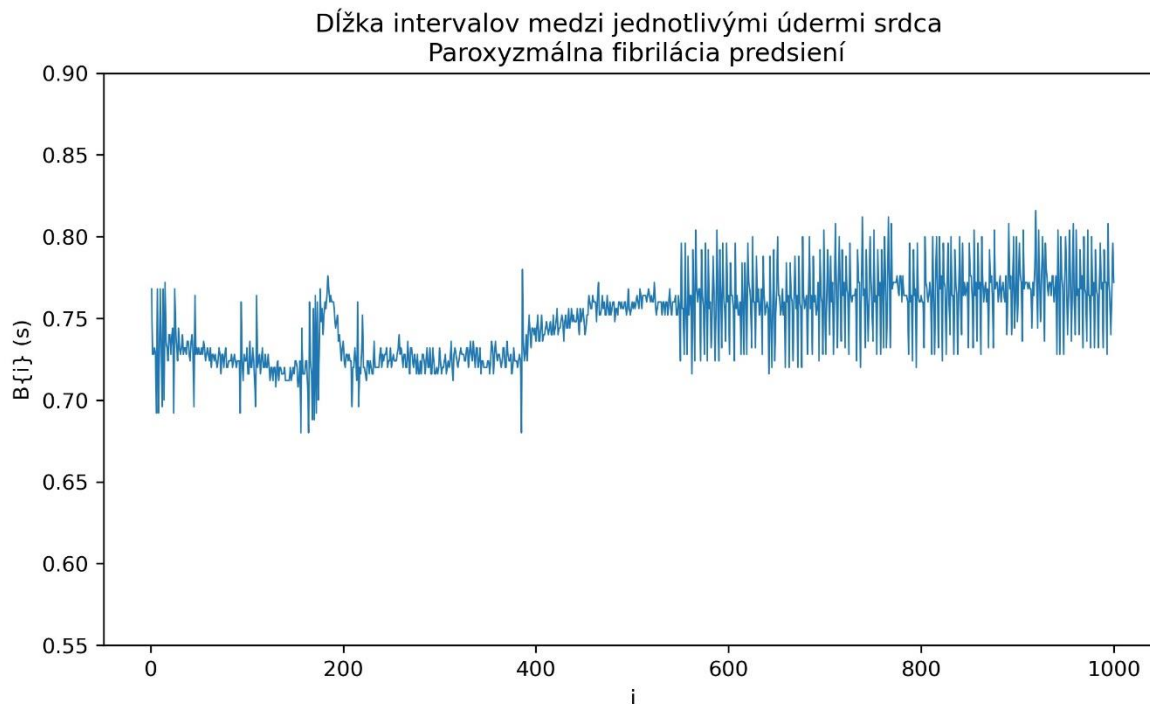
Už v Grafe 1 a 2 je zjavný signifikantný rozdiel medzi EKG signálmi skupiny s normálnym rytmom a skupiny fibrilácií. Signál v Grafe 2 je viac zašumený a jednotlivé údery prichádzajú nepravidelne. Kým v prípade N skupiny sú píky hľadané pomocou funkcie WFDB knižnice, poloha píkov skupiny AF je vyčítaná z anotácie priloženej k jednotlivým dátovým súborom. Netreba sa nechať zmiatať označením polôh v záporných hodnotách – k ich nájdeniu bol použitý algoritmus autorov databázy a pre účely tejto práce nás zaujímajú len polohy na časovej osi.

Nasledovne boli určené intervaly medzi jednotlivými údermi pomocou vzájomného odčítania polôh R píkov. V Grafe 3 sa sú vynesené dĺžky intervalov  $B(i)$  v závislosti na poradí úderu  $i$  subjektu 16265 zo skupiny N.



*Graf 3:* Dĺžka intervalov medzi údermi srdca pre 1000 úderov, subjekt s normálnym sínusovým rytmom.

V Grafe 4 sú tak isto vynesené dĺžky intervalov  $B(i)$  v závislosti na poradí úderu  $i$  subjektu 04015 zo skupiny AF.



*Graf 4:* Dĺžka intervalov medzi údermi srdca pre 1000 úderov, subjekt s fibriláciou predsiení.

Po určení dĺžky intervalov  $B(i)$  a ich vykreslení do Grafu 3 a 4 tiež možno pozorovať rozdiely medzi signálmi. Kým v prípade subjektu 16265 dĺžka intervalov medzi sťahmi srdca fluktuuje v náhodnom štýle u subjektu 04015 si môžeme všimnúť určitú formu periodicity a rigidity. Osi v oboch grafoch sú zrovnané na rovnaké hodnoty pre jednoduché porovnanie.

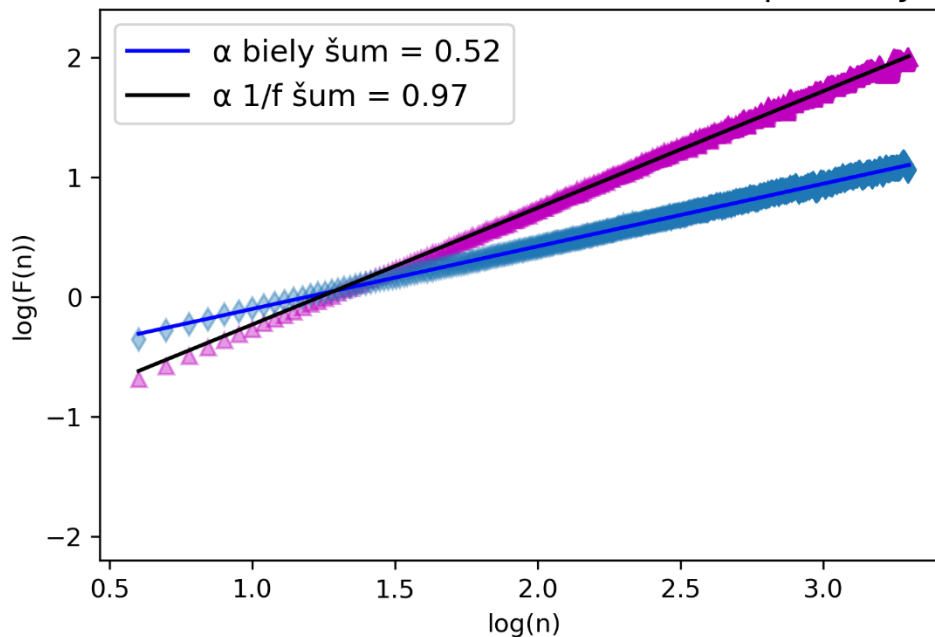
Po nadobudnutí hodnôt pre všetkých 15 subjektov v každej zo skupín boli dáta analyzované pomocou DFA a MSE metódy. Vybraných bolo  $N = 2 \cdot 10^4$  dátových bodov zodpovedajúcich dĺžke R-R intervalov.

## 7.2 Metóda DFA

Obe metódy boli najskôr aplikované na vygenerované dátové sady šumov. Biely šum totižto predstavuje postupnosť nekorelovaných náhodných hodnôt s nulovým priemerom a odchýlkou rovnou jednej. Ukazuje nám ako metódy hodnotia náhodné (komplexnejšie) časové rady. Naopak  $1/f$  šum, inak nazývaný aj ružový šum, je jedným z najbežnejších signálov vyskytujúcich sa v biologických systémoch. Jeho spektrálna hustota je nepriamo

úmerná frekvencií signálu (spektrá šumov sa nachádzajú v Obrázku 11). V Grafe 5 sa nachádza metóda DFA aplikovaná na biely a 1/f šum.

### Fluktuácie okien v závislosti na ich veľkosti n pre biely a 1/f šum

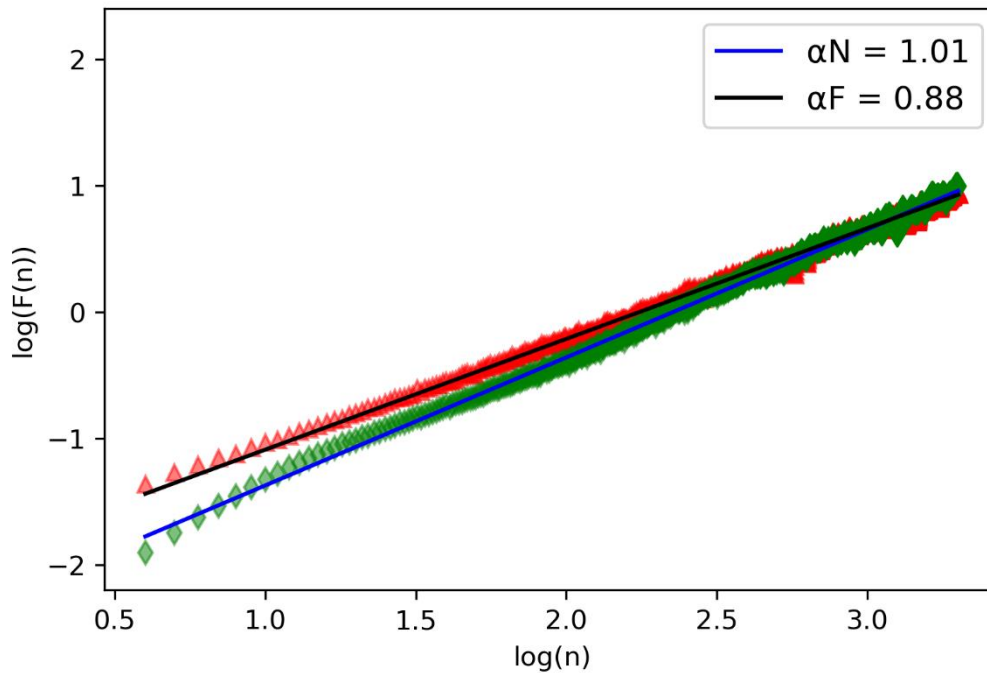


*Graf 5:* Metóda DFA aplikovaná na biely a 1/f šum. Hodnoty fialovej farby s čiernym fitom prislúchajú bielemu šumu a modré hodnoty s modrým fitom zas 1/f šumu.

Škálovací koeficient pre biely šum činí  $\alpha_b = 0,52$  a pre 1/f šum  $\alpha_{1/f} = 0,97$ . Z Grafu 5 je zjavné, že použitá DFA metóda vykazuje výsledky predpokladané v Tabuľke 1 ( $\alpha = 0,5$  pre biely šum a  $\alpha = 1$  pre ružový šum).

V Grafe 6 je metóda aplikovaná na intervaly subjektu 16265 (značené zelenými hodnotami s modrým fitom) a subjektu 04015 (značené červenou s čiernym fitom)

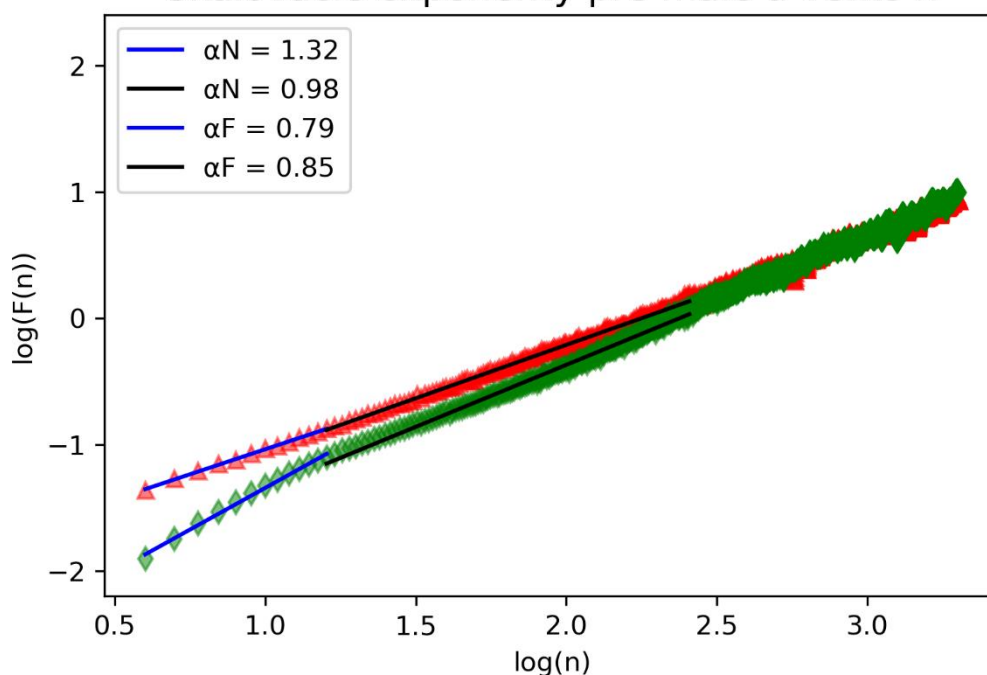
## Fluktuácie okien v závislosti na ich veľkosti $n$



Graf 6: Metóda DFA aplikovaná na intervaly subjektu 16265 (značené zelenými hodnotami s modrým fitom) a subjektu 04015 (značené červenou s čiernym fitom).

Hodnota škálovacieho exponentu pre subjekt 16265 zo skupiny normálneho sínusového rytmu je  $\alpha_{16265} = 1,01$ , kdežto pre subjekt 04015 zo skupiny fibrilácie predsieni to je  $\alpha_{04015} = 0,88$ . Exponent bol určený pre každý subjekt v oboch skupinách a jeho priemerná hodnota každej zo skupín sa nachádza v Tabuľke 2. V Grafe 6 dátových bodov subjektu  $N$  skupiny je možné všimnúť si tzv. „*crossover*“ fenomén. Fit fluktuácií v oknách s malým  $n$  má inú smernicu ako zvyšok hodnôt. Preto sa v Grafe 7 nachádza fitovanie na dvoch rôznych úsekoch, a to pre  $4 \leq n \leq 16$  a  $16 \leq n \leq 256$ . Exponent pre malé okná je označený ako  $\alpha_1$  a exponent pre veľké okná ako  $\alpha_2$ . Ich priemerné hodnoty a ich neistoty typu A pre obe skupiny sa znova nachádzajú v Tabuľke 2.

### Škálovacie exponenty pre malé a veľké n



Graf 7: Fitovanie fluktuácií na dvoch rôznych úsekoch pre  $4 \leq n \leq 16$  a  $16 \leq n \leq 256$ . Červené hodnoty prislúchajú k subjektu s fibriláciou a zelené k subjektu s normálnym rytmom. Fity určujúce exponenty  $\alpha_1$  pre oba subjekty sú znázornené modrou farbou a fity exponenty  $\alpha_2$  sú vyznačené čiernou farbou.

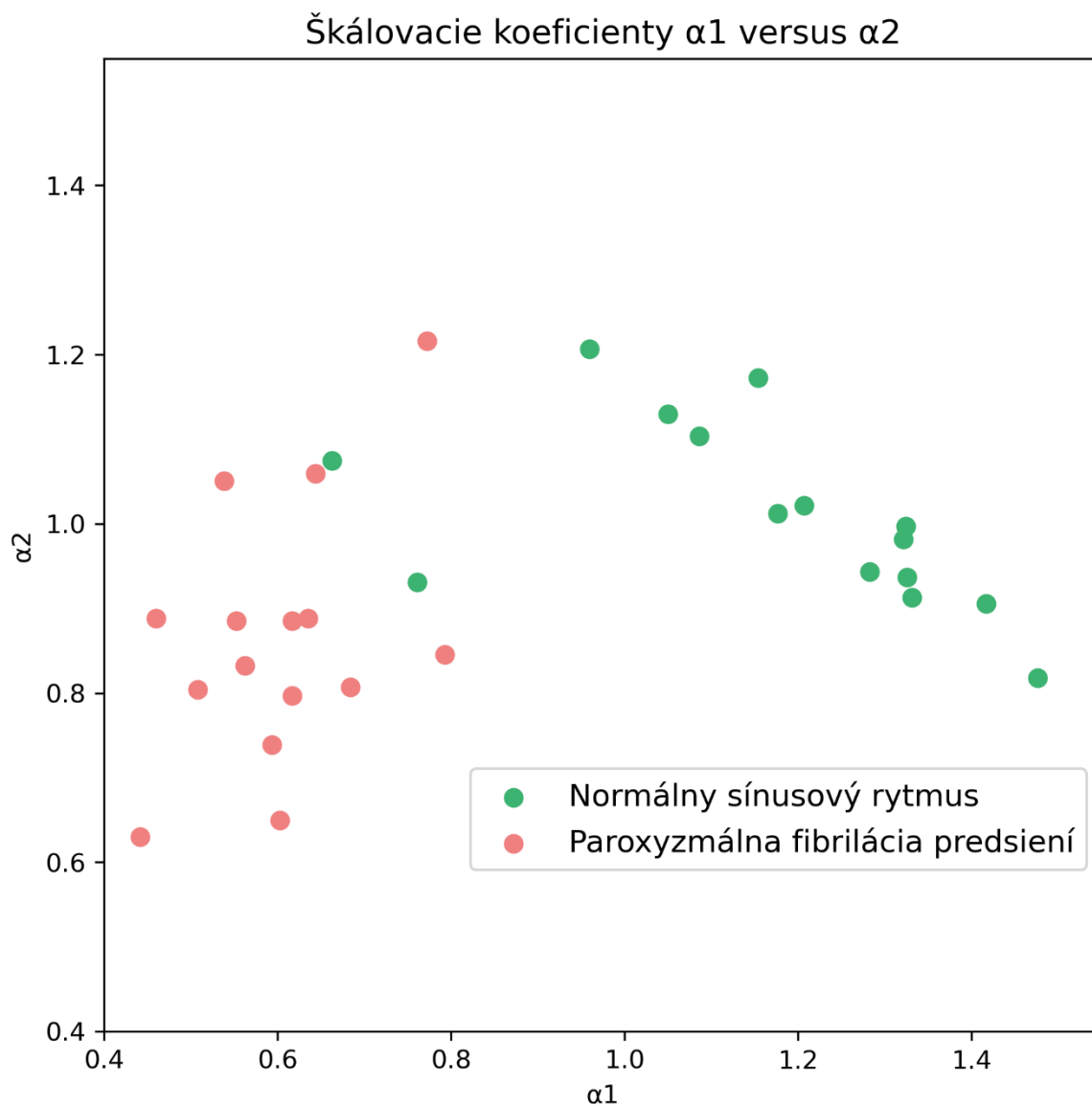
Tabuľka 2: Škálovacie koeficienty subjektov skupiny normálneho sínusového rytmu a skupiny fibrilácií predsieni pre okná  $4 \leq n \leq 2000$ ,  $4 \leq n \leq 16$  a  $16 \leq n \leq 256$ .

Škálovací koeficient	Normálny rytmus	Fibrilácie
$\alpha$	$1,07 \pm 0,02$	$1,02 \pm 0,03$
$\alpha_1$	$1,17 \pm 0,06$	$0,61 \pm 0,03$
$\alpha_2$	$1,02 \pm 0,03$	$0,87 \pm 0,04$

V Grafe 6 môžeme vidieť, že exponent pre subjekt 16265 je blízko hodnote exponentu  $1/f$  šumu. V Tabuľke 2 si možno všimnúť, že exponenty skupiny N  $\alpha_N = 1,07 \pm 0,02$  a AF  $\alpha_F = 1,02 \pm 0,03$  sú si blízke. Tu sa ukazuje fitovanie na dvoch rozličných intervaloch ako užitočné. Za „crossover“ fenomén je pravdepodobne zodpovedný fakt, že fluktuácie v oknách s malým  $n$  sú dominované osciláciami spojenými s dýchaním. V oknách s veľkým

$n$  fluktuácie odrážajú vnútornú dynamiku komplexného systému blížiacu sa správaním  $1/f$  šumu. U skupiny AF nie je tento fenomén zjavný.

Exponenty  $\alpha_1$  a  $\alpha_2$  pre každý subjekt z oboch skupín sú vynesené do Grafu 8.

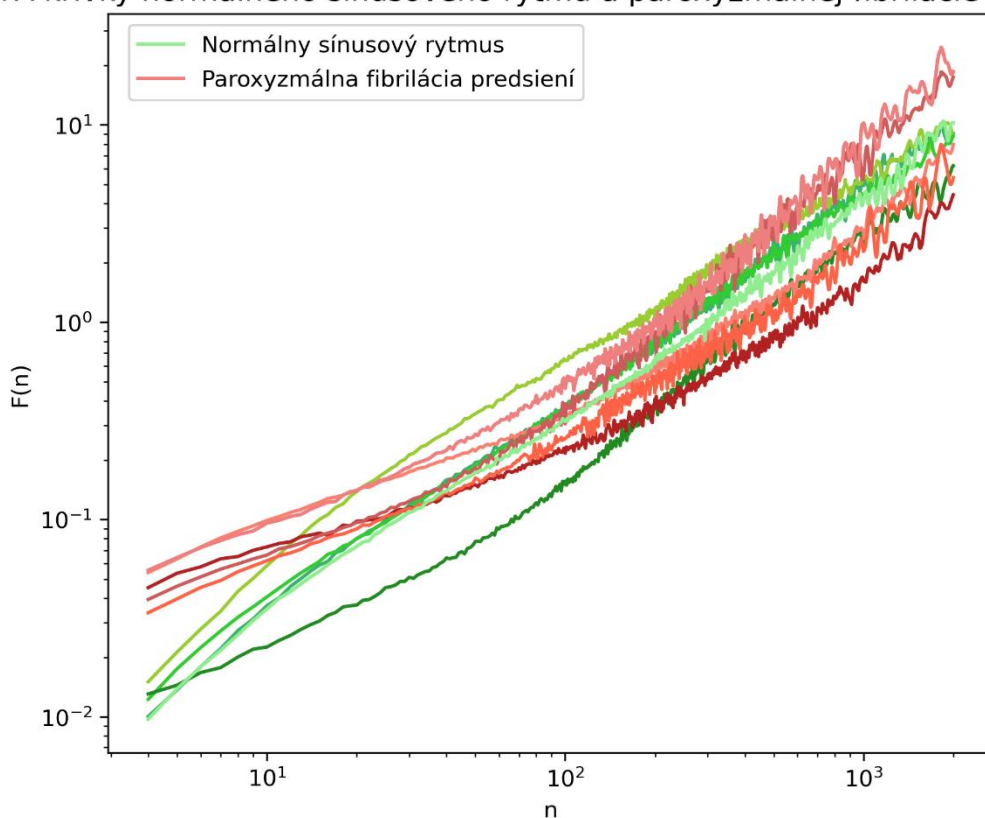


Graf 8: Exponenty  $\alpha_1$  a  $\alpha_2$  pre všetky subjekty zo skupiny s normálnym rytmom (zelená farba) a zo skupiny s fibriláciami (červená farba).

Aj napriek tomu, že si boli exponenty  $\alpha_N$  a  $\alpha_F$  blízke, po vynesení exponentov  $\alpha_1$  a  $\alpha_2$  pre jednotlivé subjekty oboch skupín do Grafu 8 sú skupiny jasne rozlíšiteľné. Dva subjekty z skupiny N v „mraku“ skupiny sú zľava žena vo veku 50 rokov a žena vo veku 35 rokov. Ku subjektom skupiny AF informácie o pohlaví a veku nie sú dostupné.

Do Grafu 9 boli vynesené flukтуаčné vektory (závislosť  $F(n)$  na  $n$ ) pre 5 subjektov z oboch skupín kvôli prehľadnosti bez fitovania.

DFA krivky normálneho sínusového rytmu a paroxyzmálnej fibrilácie predsiení



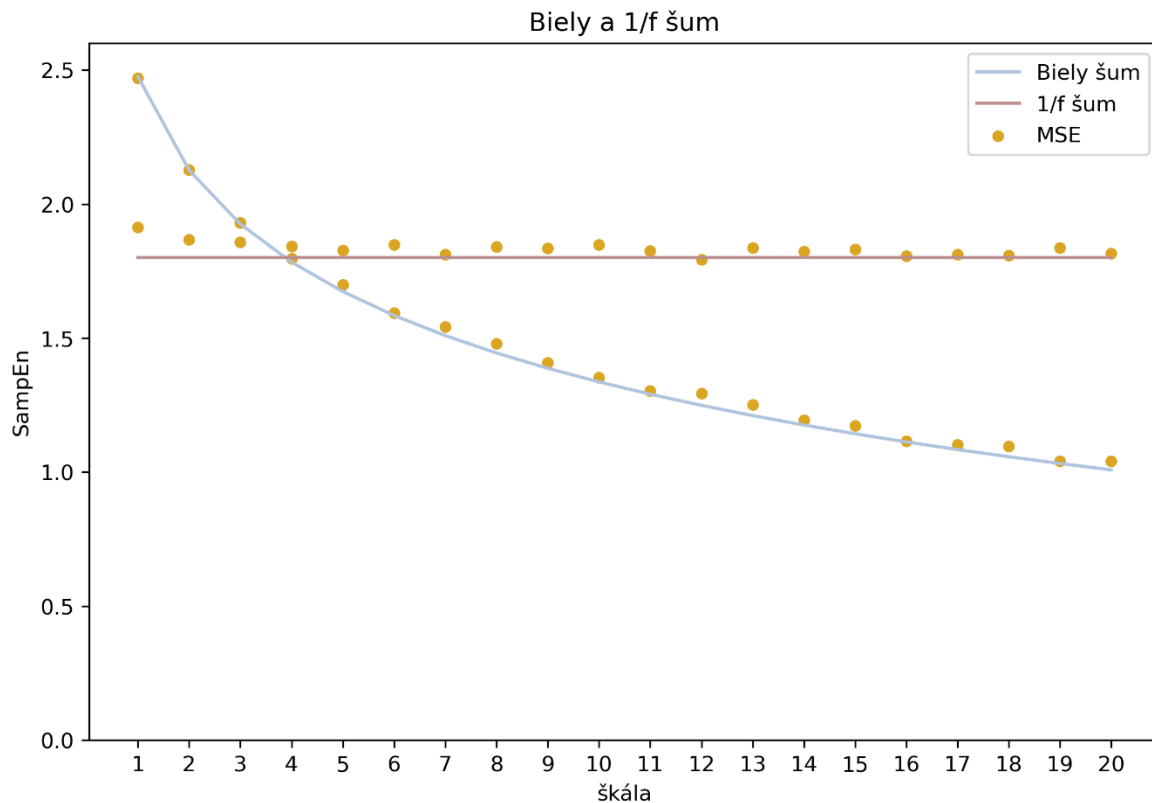
Graf 9: Flukтуаčné vektory pre 5 subjektov zo skupiny s normálnym rytmom (zelené krivky) a 5 subjektov zo skupiny s fibriláciou predsiení (červené krivky).

Po vynesení flukтуаčných vektorov v Grafe 9 je taktiež zjavný rozdiel medzi dvomi skupinami.

### 7.3 Metóda MSE

Ako v prípade DFA metódy, tak aj v prípade metódy MSE, bol algoritmus aplikovaný na šumy. Krivky v Grafe 10 reprezentujú výsledky analytického riešenia a body reprezentujú hodnoty SampEn algoritmom pridelené časovým radom. Biely a 1/f šum boli použité opäť z dôvodu reprezentácie nekorelovaných náhodných hodnôt a biologického signálu. Ďalším dôvodom je, že pre tieto dva šumy sa dá riešenie analyticky odvodiť.

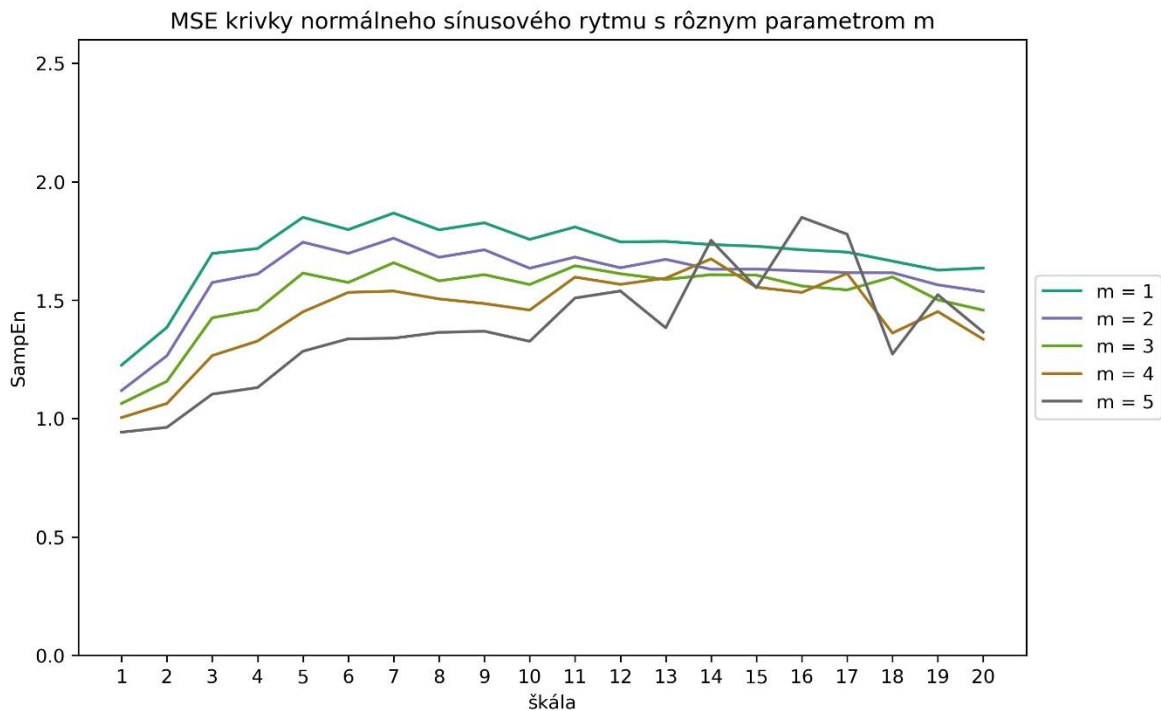
V Grafe 10 sa nachádza teda metóda MSE aplikovaná na biely a 1/f šum vyznačená žltou farbou. Bledomodrá a bledoružová krivka sú výsledkom analytického riešenia pre oba prípady. Analytické riešenie sa nachádza naprogramované v Prílohe 1 a jeho plné odvodenie vypracovali Costa a kol. (2005). Kým nie je uvedené inak, pri výpočtoch boli použité hodnoty parametrov MSE  $m = 2$  a  $r = 0,15 \cdot SD$  pôvodného časového radu.



Graf 10: Metóda MSE aplikovaná na biely a 1/f šum vyznačená žltou farbou. Bledomodrá a bledoružová krivka sú výsledkom analytického riešenia pre oba prípady.

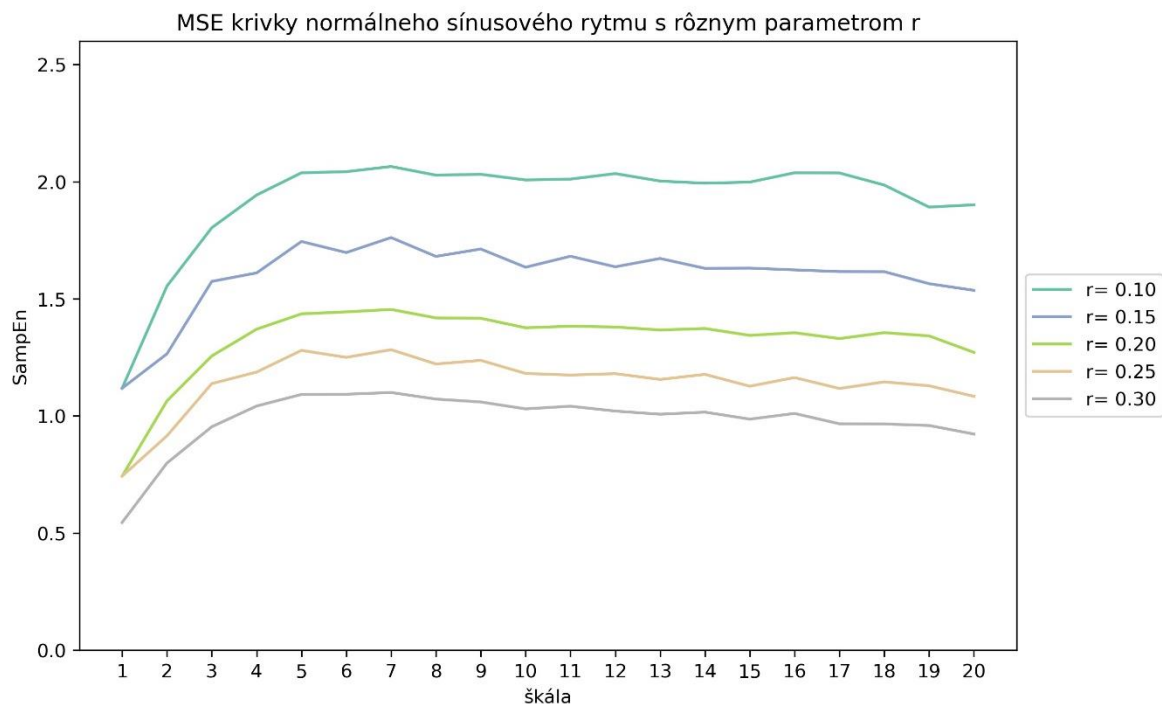


Taktiež bola preskúmaná závislosť metódy MSE na jej parametroch  $m$  (Graf 11) a  $r$  (Graf 12). Parameter  $m$  určuje veľkosť vektoru porovnávaného s ostatnými vektormi rovnakej veľkosti. Parameter  $r$  stanovuje ako „blízko“ musia byť vektory aby boli považované za zhodné. V literatúre (Costa a kol. 2002a, Costa a kol. 2002b, Costa a kol. 2005, Delgado-Bonal a Marshak 2019, Richman a Moorman 2000) sa odporúča používať hodnoty pre parameter  $m = 2$  a pre parameter  $r$  15-20% štandardnej odchýlky pôvodného časového radu.



*Graf 11:* Metóda MSE aplikovaná na subjekt 16265 zo skupiny normálneho sínusového rytmu. Jednotlivé hodnoty parametru  $m$  a k nim príslušné farby sa nachádzajú v legende.

Z Grafu 11 sa dá vypozaorovať, že pri delení radov na dlhšie vektory hodnota SampEn klesá a pre väčšie škály jej hodnota výrazne osciluje. Čím je väčší vektor  $m$ , tým menšia je pravdepodobnosť, že aj pre vektor  $m+1$  nájdeme zhodu. Je preto rozumné v tomto prípade pracovať s hodnotami 1, 2 alebo 3.

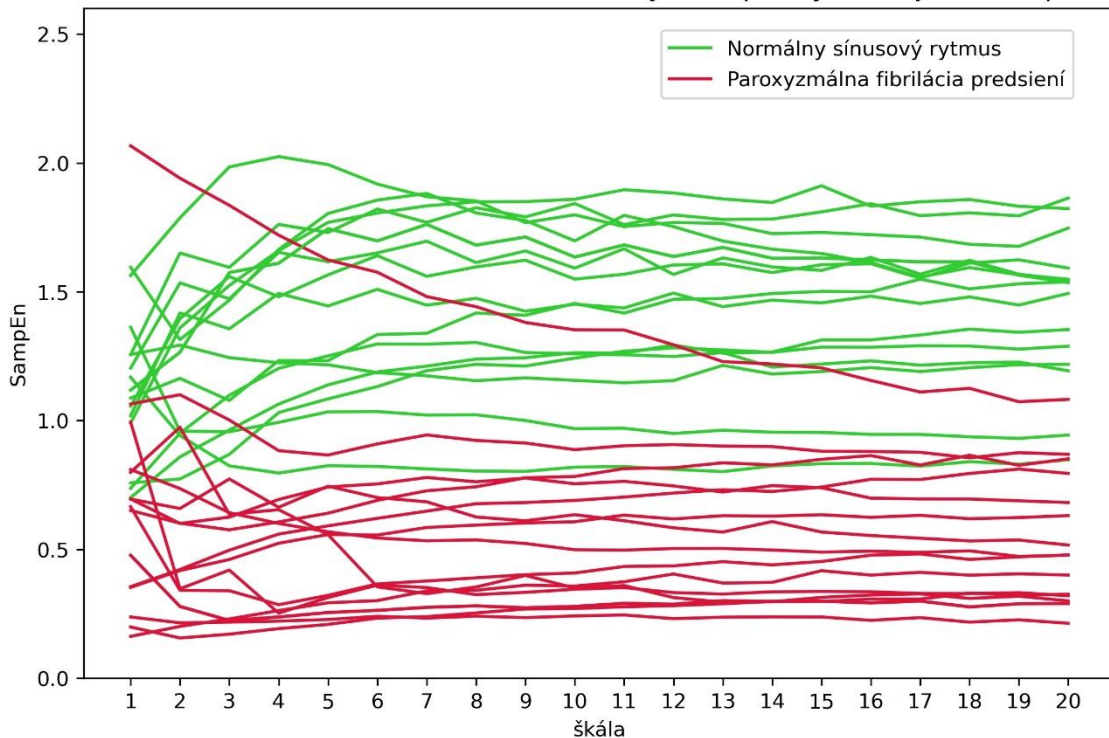


Graf 12: Metóda MSE aplikovaná na subjekt 16265 zo skupiny normálneho sínusového rytmu. Jednotlivé hodnoty parametru  $r$  a k nim príslušné farby sa nachádzajú v legende.

V Grafe 12 je zas znázornené ako parameter  $r$  ovplyvňuje výsledok. Čím sa zvyšuje tolerancia pre vzdialenosť, keď algoritmus hlási ešte zhodu, tým hodnota SampEn klesá.

Metóda MSE bola aplikovaná na všetkých 15 subjektov z oboch skupín a výsledky boli vynesené do Grafu 13. Subjekty zo skupiny s normálnym sínusovým rytmom sú vyznačené zelenou a subjekty zo skupiny fibrilácie predsiení zas červenou.

Porovnanie MSE kriviek normálneho sínusového rytmu a paroxyzmálnej fibrilácie predsiení

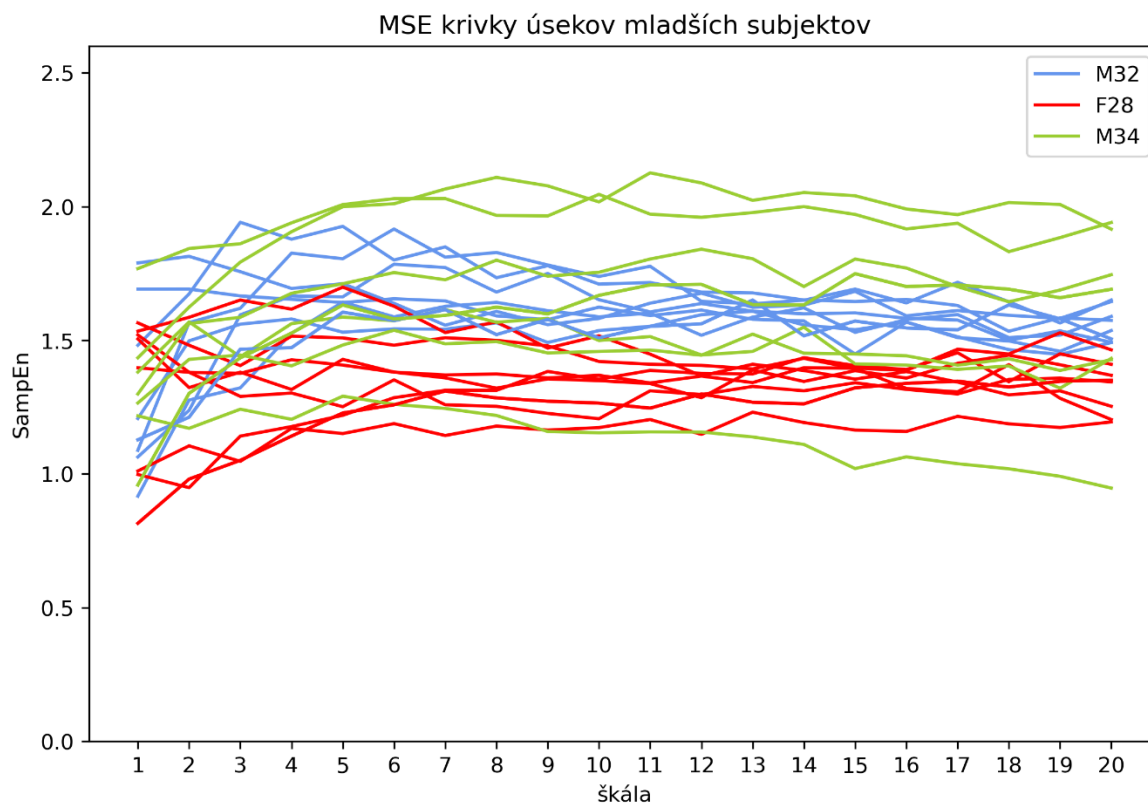


Graf 13: Metóda MSE aplikovaná na subjekty zo skupiny s normálnym sínusovým rytmom (krivky zelenej farby) a na subjekty zo skupiny fibrilácie predsiení (krivky červenej farby).

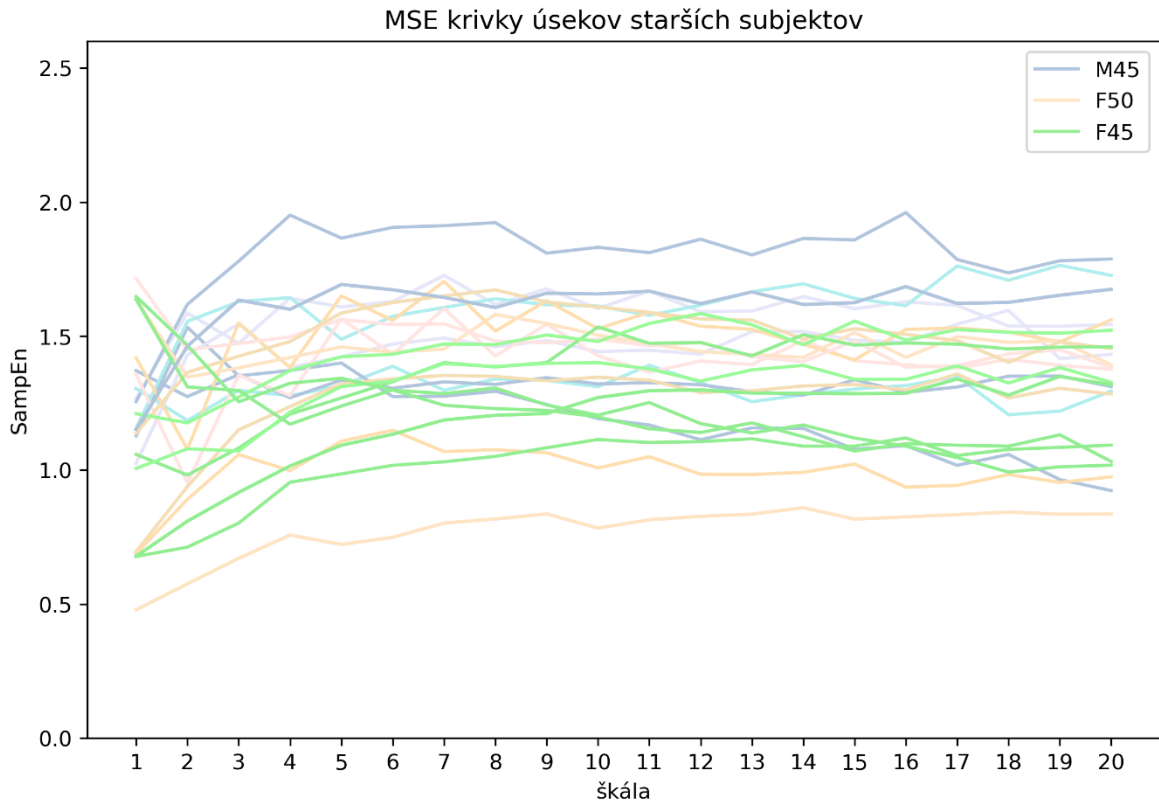
Graf 13 obsahuje porovnanie výsledkov v oboch skupinách. Skupiny sú oddelené, presah zelených kriviek do „červenej oblasti“ zodpovedajú subjektom zdola žena 45 rokov a žena 50 rokov. Práve subjekt zo skupiny N veku 50 rokov bol jedným zo zelených bodov nachádzajúcich sa v červenom „mraku“ metódy DFA (Graf 8). O subjekte zo skupiny AF pretínajúcom „zelenú oblasť“ nemáme informácie, no nejedná sa o vyčnievajúci subjekt metódy DFA. Pri porovnaní MSE kriviek subjektov v Grafe 13 s krivkami z Obrázku 13 vľavo zistujeme, že síce sú naše krivky pre skupiny N a AF jasne odlišiteľné, nie sú v zhode s výsledkami od Costa a kol. (2002b). Metóda MSE je citlivá na parametre (Graf 11 a 12, Delgado-Bonal a Marshak 2019), a taktiež na mieru šumu signálu (Costa a kol. 2005). Rozdiely tiež mohlo spôsobiť použitie Čebyševovej normy na určenie vzdialenosti porovnávaných vektorov namiesto euklidovskej. Pri výpočte SampEn bol použitý veľmi jednoduchý algoritmus bez akýchkoľvek filtrov artefaktov, čo tiež mohlo ovplyvniť výsledok našej analýzy. V neposlednom rade je potrebné zmieniť, že charakteristiky ako vek jednotlivých subjektov mohli spôsobiť tieto rozdiely.

## 7.4 Charakteristický DFA exponent a MSE krivka

Ďalej vzniká otázka, či má každé „srdce“ svoj charakteristický škálovací exponent a MSE krivku. Preto boli dlhodobé záznamy subjektov rozkúskované na neprekrývajúce sa úseky dĺžky R-R intervalov s počtom bodov  $N = 1 \cdot 10^4$  a na každý z nich bola aplikovaná MSE a DFA metóda. Za hodnotu MSE metódy bola vybraná asymptotická hodnota pre krivky priemerovaním hodnôt SampEn pre škály 8-20. V Grafe 14 sa nachádzajú MSE krivky troch mladších subjektov a v Grafe 15 troch starších subjektov.



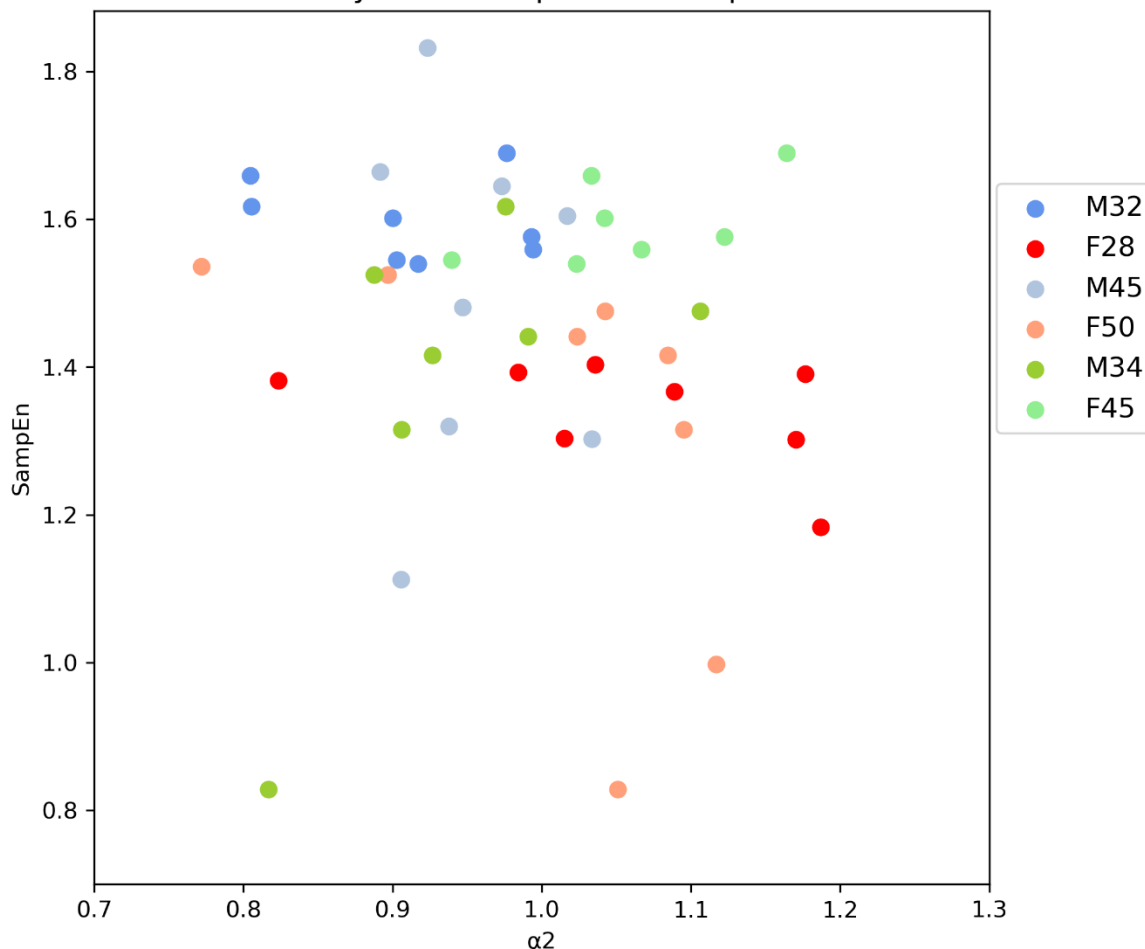
*Graf 14:* Metóda MSE aplikovaná na mladšie subjekty zo skupiny s normálnym sínusovým rytmom. Modrá farba – muž 32 rokov, červená farba – žena 28 rokov, zelená farba – muž 34 rokov.



*Graf 15:* Metóda MSE aplikovaná na staršie subjekty zo skupiny s normálnym sínusovým rytmom. Svetlomodrá farba – muž 45 rokov, ružová farba – žena 50 rokov, svetlozelená farba – žena 45 rokov.

Cieľom bolo zistiť, či je na základe DFA a MSE možné rozoznať jednotlivé subjekty ak sa budeme pozerat' na neprekrývajúce sa úseky časových radov dĺžky ich R-R intervalov. Je možné si všimnúť, že v prípade M32 s F28 v Grafe 14 sú krivky úsekov každého subjektu blízko, kdežto u starších subjektov v Grafe 15 sú ich krivky viac rozložené v priestore. Ako si však môžeme tiež všimnúť v Grafe 14 na subjekte M34, rozdielne hodnoty SampEn pre jednotlivé úseky nemusia byť charakteristické len pre staršie subjekty. Preto boli priemerné hodnoty SampEn 8. až 20. škály vynesené do Grafu 16 spolu s DFA exponentom  $\alpha_2$ .

Škálovacie koeficienty  $\alpha_2$  versus priemer SampEn hodnôt škál 8-20



Graf 16: SampEn priemer 8-20 škály a DFA exponent  $\alpha_2$ . Subjektom sú priradené farby nachádzajúce sa v legende.

V Grafe 16 je možné zahliadnuť zhľady hodnôt niektorých subjektov. Opäť nie je na mieste generalizovať na základe pohlavia alebo veku. V skutočnosti by bolo vhodné mať viac informácií o subjektoch (váha, zdravotný stav, predispozície v rodine a iné) a mať prístup k väčšiemu a rozmanitejšiemu datasetu (skupina subjektov  $N$  obsahuje napríklad len 4 mužov). Je však zjavné, že niektoré „srdcia“ si nesú svoje charakteristiky na rôznych škálach nielen pri skúmaní časových radov  $N = 2 \cdot 10^4$  (približne 5 hodín) ale aj  $N = 8 \cdot 10^4$  (približne 20 hodín). Otázkou však ostáva, čo robí tieto „srdcia“ špeciálne.

## 8 ZÁVER

Z matematického hľadiska sú fraktály objekty so striktne neceločíselnou dimenziou alebo objekty, ktorých geometrické časti sa opakujú v nich samých až do nekonečna. Fraktály nie sú len bezpredmetnou matematickou teóriou, ale v prírode sa tieto štruktúry objavujú veľmi často v podobe vzorov, tvarov, vzorcov správania alebo skrytých korelácií. Biologické objekty vykazujú fraktálne správanie. Ich fraktálna štruktúra je pri niektorých očividná, ako napríklad pri stromoch, pľúcach, cievnom systéme, lastúrach alebo vetvení riek. Naopak v iných prípadoch sú fraktálne vlastnosti viac skryté a objavujú sa až po dôkladnom štúdiu dát ako funkcie času alebo po vykreslení závislostí do grafu. Poznanie princípu fraktálneho fungovania prírody môže byť efektívnym nástrojom nie len pri skúmaní nových fenoménov, ale aj v otázkach skorej diagnostiky, udržateľnosti alebo výrobe biomateriálov zajtrajška.

V prírode je fraktálne správanie častým riešením problémov v štruktúrnom a aj časovom aspekte. Ľudské pľúca k efektívnej výmene dýchacích plynov využívajú jednu z výhod fraktálnych štruktúr, a to veľkú plochu v pomere k objemu objektu. To isté platí pre ľudský cievny systém či koreňový systém rastlín. Metódou kvantifikácie fraktálnych objektov a množín zameriavajúcou sa na tvar je výpočet ich fraktálnej dimenzie. V prípade uniformných sebepodobných fraktálov, akými sú Cantorova množina, Sierpiňského trojuholník alebo Kochova krivka, je dimenzia určená z definície. V prípade zložitejších štruktúr, neuniformných či prírodných fraktálov je najčastejšie používanou metódou počítania boxov. Fraktálne charakteristiky majú však aj niektoré časové rady tvorené postupnosťou hodnôt zaznamenaných v čase bez „tvaru“. Preto boli vyvinuté metódy analýzy fraktálnych časových radov, z ktorých dve boli použité v tejto práci. Detrendovaná fluktuálna analýza (DFA) sa používa na analýzu časových radov, v ktorých sa nachádzajú fenomény propagujúce sa radom na veľké vzdialenosti. Multiškálová entropia (MSE) používa algoritmus založený na myšlienke entropie asociovej s kvantifikovaním miery komplexity fyziologických signálov pomocou množstva informácie a stupňa náhodnosti systému.

Fyziologické systémy sú regulované pomocou interagujúcich mechanizmov, ktoré pôsobia na niekoľkých priestorových a časových škálach. Medzi tieto systémy patrí aj ľudské srdce. Výsledkom dynamiky týchto mechanizmov je fakt, že intervaly medzi údermi srdca, R-R

interval, fluktuujú komplexným spôsobom. Počas tisícok srdečných úderov nachádzame za použitia metódy DFA tzv. ďalekosahové mocninové korelácie. Takto so škálou nemenné správanie biologických systémov je nutné pre adekvátne funkčné odozvy systému a jeho celkovú adaptabilitu. Zabraňuje totižto organizmu prehnane ľpiet' na jednom režime obmedzujúcom jeho vlastné reakcie na podnety okolia. Za použitia metódy MSE sme boli schopní posúdiť komplexitu R-R intervalov cez určenie miery náhodnosti takéhoto časového radu. Nízke hodnoty štatistiky SampEn popisujú systém ako jednostajný a predpokladateľný s opakujúcimi sa úsekmi, zatiaľ čo jej vysoké hodnoty značia náhodnosť a nezávislosť dát.

V tejto práci boli EKG signály subjektov s normálnym sínusovým rytmom a subjektov s fibriláciou predsiení získané z platformy Physionet a spracované v programovacom jazyku Python. Z EKG časových radov boli odčítané R-R intervaly a na týchto radoch boli použité metódy DFA a MSE. Zo spracovaných výsledkov DFA metódy sme mohli vypožorovať, že časové rady subjektov s normálnym sínusovým rytmom (N) vykazujú prítomnosť ďalekosahových korelácií a vlastností  $1/f$  šumu, kdežto u subjektov s fibriláciou predsiení (AF) sú tieto korelácie zastúpené v menšej miere. Subjekty skupiny N tiež vykazujú „*crossover*“ fenomén, kým u tých zo skupiny AF tento fenomén nie je taký očividný. Metóda MSE zas ukázala, že dynamika za R-R intervalmi subjektov s normálnym sínusovým rytmom je komplexnejšia než u subjektov s fibriláciou predsiení. Entropia časových radov subjektov zo skupiny N je vyššia než u subjektov zo skupiny AF. Síce sú krivky jasne odlíšiteľné pre obe skupiny, ich hodnoty a tvary nie sú v úplnej zhode s literatúrou. Výsledky však mohli ovplyvniť faktory ako citlivosť metódy na parametre, počet dátových bodov či mieru šumu signálu, použitie inej normy, jednoduchosť použitého kódu, absencia filtrov artefaktov, či rozdielny vek subjektov tu a v literatúre. Obe metódy sa však ukázali ako schopné kvantifikátory komplexity a aj vzájomnej odlišnosti zdravého a patologického stavu.

Ďalej sme boli schopní ukázať, že u niektorých subjektov sa po analýze neprekrývajúcich sa kratších úsekov časového radu R-R intervalov vyskytujú podobné hodnoty entropie a škálovacieho exponentu. Zdá sa, že nejakú úlohu v tomto zaujímavom zistení hrá vek subjektov. Nie je však na mieste generalizovať, v skutočnosti by bolo vhodné získať prístup k rozmanitejšiemu datasetu spolu so širším okruhom informácií o subjektoch, ako napríklad váha, zdravotný stav, rodinné predispozície a iné.



## 9 ZOZNAM LITERATÚRY

- Azua-Bustos A., Vega-Martínez C. (2013) The potential for detecting ‘life as we don't know it’ by fractal complexity analysis. *International Journal of Astrobiology* **12**, 314-320.
- Barcellos A. (2008) Benoit Mandelbrot. In: *Mathematical People: Profiles and Interviews*. 2nd ed., (Albers D.J., Alexanderson G.L., eds.), A K Peters, Wellesley, MA, USA, 212-232.
- Barnsley M. (1988a) Metric Spaces; Equivalent Spaces; Classification of Subsets; and the Space of Fractals. In: *Fractals Everywhere*. Academic Press, Inc., San Diego, CA, USA, 6-42.
- Barnsley M. (1988b) Fractal Dimension. In: *Fractals Everywhere*. Academic Press, Inc., San Diego, CA, USA, 172-206.
- Bianchi S. (2020) fathon: A Python package for a fast computation of detrended fluctuation analysis and related algorithms. *The Journal of Open Source Software* **5**: 1828.
- Coornaert M. (2015) Topological Dimension. In: *Topological Dimension and Dynamical Systems*. 1st ed., (Axler S., Capasso V., Casacuberta C., MacIntyre A., Ribet K., Sabbah C., Süli E., Woyczyński W.A., eds.), Springer International Publishing, Switzerland, 3-25.
- Costa M., Goldberger A.L., Peng C.-K. (2002a) Multiscale Entropy to Distinguish Physiologic and Synthetic RR Time Series. *Computers in Cardiology* **29**, 137-140.
- Costa M., Goldberger A.L., Peng C.-K. (2002b) Multiscale Entropy to Distinguish Physiologic and Synthetic RR Time Series. *Physical Review Letters* **89**: 068102-3.
- Costa M., Goldberger A.L., Peng C.-K. (2005) Multiscale entropy analysis of biological signals. *Physical Review E* **71**: 021906.
- Coveney P., Highfield R. (2003) Tajné umění. In: *Mezi řádem a chaosem: Hranice komplexity: hledání řádu v chaotickém světě*. (Slanina F., Cejnarová A., Janata M., eds.), Mladá Fronta, Praha, ČR, 19-31.
- Delgado-Bonal A., Marshak A. (2019) Approximate Entropy and Sample Entropy: A Comprehensive Tutorial. *Entropy* **21**: 541.
- Dyson F. (1978) Characterizing Irregularity. *Science* **200**, 677-678.
- Goldberger A.L., Amaral L.A., Glass L., Hausdorff J.M., Ivanov P.C., Mark R.G., Mietus J.E., Moody G.B., Peng C.-K., Stanley H.E. (2000) PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation* **101**: E215-220.
- Havlin S., Buldyrev S.V., Goldberger A.L., Mantegna R.N., Ossadnik S.M., Peng C.-K., Simons M., Stanley H.E. (1995) Fractals in Biology and Medicine. *Chaos, Solitons & Fractals* **6**, 171-201.
- Hernández N.E., Hansen W.A., Zhu D., Shea M.E., Khalid M., Manichev V., Putnins M., Chen M., Dodge A.G., Yang L., Marrero-Berríos I., Banal M., Rechani P., Gustafsson T., Feldman L.C., Lee S.-H., Wackett L.P., Dai W., Khare S.D. (2019) Stimulus-responsive self-assembly of protein-based fractals by computational design. *Nature Chemistry* **11**, 605-614.
- Kempkes S.N., Slot M.R., Freeney S.E., Zevenhuizen S.J.M., Vanmaekelbergh D., Swart I., Morais Smiths C. (2019) Design and characterization of electrons in a fractal geometry. *Nature Physics* **15**, 127-131.

- Korolj A., Wu H.-T., Radisic M. (2019) A healthy dose of chaos: Using fractal framework for engineering higher-fidelity biomedical systems. *Biomaterials* **219**: 119363.
- Kucharski A. (2014) Math's beautiful monsters; how a destructive idea paved the way to modern math. *Nautilus Quarterly* **11**. <http://nautil.us/issue/11/light/math-beautiful-monsters>
- Liu J.Z., Zhang L.D., Yue G.H. (2003) Fractal Dimension in Human Cerebellum Measured by Magnetic Resonance Imaging. *Biophysical Journal* **85**, 4041-4046.
- Mandelbrot B.B. (1967) How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension. *Science* **156**, 636-638.
- Mandelbrot B.B. (1982) *The Fractal Geometry of Nature*. W.H. Freeman and Company, New York, NY, USA, 468 strán.
- Moody G.B., Mark R.G. (1983) A new method for detecting atrial fibrillation using R-R intervals. *Computers in Cardiology* **10**, 227-300.
- Peng C.-K., Buldyrev S.V., Goldberger A.L., Havlin S., Sciortino F., Simons M., Stanley H.E. (1992) Long-range correlations in nucleotide sequences. *Nature* **356**, 168-170.
- Peng C.-K., Buldyrev S.V., Havlin S., Simons M., Stanley H.E., Goldberger A.L. (1994) Mosaic organization of DNA nucleotides. *Physical Review E* **49**, 1685-1689.
- Peng C.-K., Havlin S., Hausdorff J.M., Mietus J.E., Stanley H.E., Goldberger A.L. (1995) Fractal mechanisms and heart rate dynamics: Long-range correlations and their breakdown with disease. *Journal of Electrocardiology* **28**, 59-65.
- Richman J.S., Moorman J.R. (2000) Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology - Heart and Circulatory Physiology* **278**: H2039-49.
- Sagan H. (1991) Some Reflections on the Emergence of Space-filling Curves: The Way it Could Have Happened and Should Have Happened, but did not Happen. *Journal of the Franklin Institute* **328**, 419-430.
- Zelinka I., Včelář F., Čandík M. (2006) *Fraktální geometrie – principy a aplikace*. BEN – technická literatura, Praha, Česká republika, 160 strán.

## 10 INTERNETOVÉ ZDROJE

MIT-BIH Normal Sinus Rhythm Database: <https://physionet.org/content/nsrdb/1.0.0/> (1.5.2021)

MIT-BIH Atrial Fibrillation Database: <https://physionet.org/content/afdb/1.0.0/> (1.5.2021)

PhysioNet: <https://physionet.org/> (1.5.2021)

Sample entropy: [https://en.wikipedia.org/wiki/Sample\\_entropy](https://en.wikipedia.org/wiki/Sample_entropy) (4.5.2021)

WFDB documentation: <https://wfdb.readthedocs.io/en/latest/processing.html#module-wfdb.processing> (6.5.2021)

## 11 ZDROJE OBRÁZKOV

Obrázok 1: Weierstrauss function. In: pngwing.com [online]. (n.d.) [vid. 2020-08-27]. Dostupné z: <https://www.pngwing.com/en/free-png-nkaow>

Obrázok 2: Three iteration of a Peano curve, whose limit is a space-filling curve. In: en.wikipedia.org [online]. 1. júna 2007 [vid. 2020-08-27]. Dostupné z: <https://commons.wikimedia.org/wiki/File:Peanocurve.svg#/media/File:Peanocurve.svg>

Obrázok 3: Measuring the coast of Great Britain. In: insider.com [online]. 28. februára 2019 [vid. 2021-04-27]. Dostupné z: <https://www.businessinsider.com/fractals-and-the-coast-of-great-britain-2019-2>

Obrázok 4: Self-Similarity. In: pages.cs.wisc.edu, Eric Green [online]. 1998 [vid. 2020-08-31]. Dostupné z: [http://pages.cs.wisc.edu/~ergreen/honors\\_thesis/similar.html](http://pages.cs.wisc.edu/~ergreen/honors_thesis/similar.html)

Obrázok 5: Self-Affine Sierpiński gasket. In: semanticscholar.org [online]. 2019 [vid. 2021-05-05]. Dostupné z: <https://www.semanticscholar.org/paper/A-converse-statement-to-Hutchinson's-theorem-and-a-Morris-Sert/ebd6a99025cd62fc664c29ea9f0658a946850b13>

Obrázok 6: The Cantor Set. In: georgcantorbyelithompson.blogspot.com [online]. 23. februára 2015 [vid. 2020-08-26]. Dostupné z: <http://georgcantorbyelithompson.blogspot.com/2015/02/the-cantor-set.html>

Obrázok 7: Generation of the Koch curve. In: ResearchGate.net [online]. Apríl 2006 [vid. 2020-08-26]. Dostupné z: [https://www.researchgate.net/figure/Generation-of-the-Koch-curve-produced-by-a-simple-iterative-transformation-beginning\\_fig4\\_47934714](https://www.researchgate.net/figure/Generation-of-the-Koch-curve-produced-by-a-simple-iterative-transformation-beginning_fig4_47934714)

Obrázok 8: Illustration of the box-counting method for estimating the fractal dimension of the ozolith contour. In: ResearchGate.net [online]. Január 2005 [vid. 2020-08-26]. Dostupné z: [https://www.researchgate.net/figure/Illustration-of-the-box-counting-method-for-estimating-the-fractal-dimension-of-the\\_fig2\\_228950126](https://www.researchgate.net/figure/Illustration-of-the-box-counting-method-for-estimating-the-fractal-dimension-of-the_fig2_228950126)

Obrázok 9: “Toe” (left) and “finger” (right) instabilities produced when a low-viscosity fluid invades a higher-viscosity fluid. In: physics.aps.org, S. Nagel, I. Bischofberger [online]. 26. mája 2017 [vid. 2021-05-05]. Dostupné z: <https://physics.aps.org/articles/v10/60>

Obrázok 10: The integrated time series  $y(k)$ . Prevzaté z Peng a kol. (1995)

Obrázok 11: White, Pink and Brown Noise. In: xpressocommunications.com [online]. (n.d.) [vid. 2021-05-05]. Dostupné z: <https://www.xpressocommunications.com/client-news/beyond-music-our-response-to-sound/attachment/blog-february-white-pink-brown-noise/>

Obrázok 12: Schematic illustration of the coarse-graining procedure. Prevzaté z Costa a kol. (2005)

Obrázok 13: *vľavo* MSE analysis of interbeat interval time series derived from healthy subjects, subjects with congestive heart failure (CHF), and subjects with atrial fibrillation (AF); *vpravo* MSE analysis of the cardiac interbeat time series derived from healthy young subjects and healthy elderly subjects. Prevzaté Costa a kol. (2002b)

# PRÍLOHA 1

Program použitý k analýze dát v programovacom jazyku Python.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import wfdb
5 from wfdb import processing
6 from matplotlib.colors import ListedColormap, LinearSegmentedColormap
7
8 Multiscale Entropy
9
10 # Vypočíta intervaly medzi R pikmi pomocou moving wave integration wavelets
11 def HRZ(name):
12     rec = wfdb.rdrecord(name, sampsto = 3000000)
13     r_loc = processing.xqrs_detect(rec.p_signal[:,0], fs=rec.fs)
14     hr = np.zeros(len(r_loc)-1)
15     hr = np.diff(r_loc)/rec.fs
16     return hr[:20000]
17
18 #vypočíta intervaly medzi anotovanými R pikmi, vráti 20000 hodnôt
19 def HRF(name):
20     ann = wfdb.rdann(name, 'qrs', return_label_elements = ('symbol', 'label_store',
21     'description'), summarize_labels = ['symbol', 'label_store', 'description',
22     'n_occurrences'])
23     r_loc = ann.sample
24     hdr = wfdb.rdheader(name)
25     hr = np.zeros(len(r_loc)-1)
26     hr = np.diff(r_loc)/hdr.fs
27     return hr[:20000]
28
29 # Vypočíta Sample Entropy
30 def sampen(L, m, r):
31     N = len(L)
32     B = 0.0
33     A = 0.0
34
35     # Tu xmj je JEDNA array vektorov ktoré odčítam od VŠETKÝCH xmi vektorkov
36     # Rozdelí časovú rad a uloží templates dĺžky m
37     xmi = np.array([L[i : i + m] for i in range(N - m)]) #tak aby bolo možné vytvoriť
38     m+1 vektor
39     xmj = np.array([L[i : i + m] for i in range(N - m + 1)])
40
41     # Uloží matches a vylúči self-match, vypočíta A
42     # np.abs+max(axis = 1) --> najväčšia vzdialenosť cez akúkoľvek súradnicu
43     # -1 vyhod' self-match
44     A = np.sum([np.sum(np.abs(xmii - xmj).max(axis=1) <= r) - 1 for xmii in xmi])
45
46     # Podobne pre B
47     m += 1
```

```

45     xm = np.array([L[i : i + m] for i in range(N - m + 1)])
46
47     B = np.sum([np.sum(np.abs(xmi - xm).max(axis=1) <= r) - 1 for xmi in xm])
48
49     # Vrátí SampEn
50     return -np.log(B/A)
51
52 # skonštruuje časové rady so škálou
53 def cgts(time_series, scale):
54     #počet hodnôt pôvodnej ts
55     n = len(time_series)
56     #fix zaokrúhľuje nadol, počet hodnôt vytvorenej cgts
57     b = int(np.fix(n / scale))
58     #zoberie 1d array, vyhodí posledný údaj (nepárny), a urobí z toho array bxscale
59     temp = np.reshape(time_series[0:b*scale], (b, scale))
60     #priemer po riadkoch
61     cts = np.mean(temp, axis = 1)
62     return cts
63
64 #počíta mse
65 def mse(time_series, m, r, scale):
66     ret = []
67     for i in range(1, scale+1):
68         cts = cgts(time_series, i)
69         std = np.std(time_series)
70         se = sampen(cts, m, r*std)
71         if se== 0:
72             ret.append((N))
73         else:
74             ret.append((se))
75     print("completed")
76     return ret
77
78 # normálny rytmus
79 # Hodnoty intervalov medzi R pikmi najde a vytiahne do arrays
80 n1 = HRZ('16265')
81 n2 = HRZ('17453')
82 n3 = HRZ('16273')
83 n4 = HRZ('16420')
84 n5 = HRZ('16483')
85 n6 = HRZ('16539')
86 n7 = HRZ('16773')
87 n8 = HRZ('16786')
88 n9 = HRZ('16795')
89 n10 = HRZ('17052')
90 n11 = HRZ('19088')
91 n12 = HRZ('19090')
92 n13 = HRZ('19093')
93 n14 = HRZ('19140')
94 n15 = HRZ('19830')
95
96 # fibrilácie

```

```
97 # Hodnoty intervalov medzi R pikmmi z anotácií vyťahne do arrays
98 f1 = HRF('04015')
99 f2 = HRF('04048')
100 f3 = HRF('04126')
101 f4 = HRF('04746')
102 f5 = HRF('04908')
103 f6 = HRF('04936')
104 f7 = HRF('05091')
105 f8 = HRF('05121')
106 f9 = HRF('05261')
107 f10 = HRF('06426')
108 f11 = HRF('07910')
109 f12 = HRF('08215')
110 f13 = HRF('08219')
111 f14 = HRF('08378')
112 f15 = HRF('08455')
113
114 # Vypočíta MSE pre jednotlivé dátové sady
115 #normálny rytmus
116 mse1n = mse(n1, 2, 0.15, 20)
117 mse2n = mse(n2, 2, 0.15, 20)
118 mse3n = mse(n3, 2, 0.15, 20)
119 mse4n = mse(n4, 2, 0.15, 20)
120 mse5n = mse(n5, 2, 0.15, 20)
121 mse6n = mse(n6, 2, 0.15, 20)
122 mse7n = mse(n7, 2, 0.15, 20)
123 mse8n = mse(n8, 2, 0.15, 20)
124 mse9n = mse(n9, 2, 0.15, 20)
125 mse10n = mse(n10, 2, 0.15, 20)
126 mse11n = mse(n11, 2, 0.15, 20)
127 mse12n = mse(n12, 2, 0.15, 20)
128 mse13n = mse(n13, 2, 0.15, 20)
129 mse14n = mse(n14, 2, 0.15, 20)
130 mse15n = mse(n15, 2, 0.15, 20)
131
132 # Vypočíta MSE pre jednotlivé dátové sady
133 #fibrilácie
134 mse1f = mse(f1, 2, 0.15, 20)
135 mse2f = mse(f2, 2, 0.15, 20)
136 mse3f = mse(f3, 2, 0.15, 20)
137 mse4f = mse(f4, 2, 0.15, 20)
138 mse5f = mse(f5, 2, 0.15, 20)
139 mse6f = mse(f6, 2, 0.15, 20)
140 mse7f = mse(f7, 2, 0.15, 20)
141 mse8f = mse(f8, 2, 0.15, 20)
142 mse9f = mse(f9, 2, 0.15, 20)
143 mse10f = mse(f10, 2, 0.15, 20)
144 mse11f = mse(f11, 2, 0.15, 20)
145 mse12f = mse(f12, 2, 0.15, 20)
146 mse13f = mse(f13, 2, 0.15, 20)
147 mse14f = mse(f14, 2, 0.15, 20)
148 mse15f = mse(f15, 2, 0.15, 20)
```

```

149 col_mse = ["n1", "n2", "n3", "n4", "n5", "n6", "n7", "n8", "n9", "n10", "n11", "n12",
150 "n13", "n14", "n15", "f1", "f2", "f3", "f4", "f5", "f6", "f7", "f8", "f9", "f10", "f11",
151 "f12", "f13", "f14", "f15"]

MSE = pd.DataFrame([mse1n, mse2n, mse3n, mse4n, mse5n, mse6n, mse7n, mse8n, mse9n,
152 mse10n, mse11n, mse12n, mse13n, mse14n, mse15n, mse1f, mse2f, mse3f, mse4f, mse5f,
153 mse6f, mse7f, mse8f, mse9f, mse10f, mse11f, mse12f, mse13f, mse14f, mse15f], index
    =col_mse)

MSE = MSE.T
154 MSE['škála'] = MSE.index+1
155 MSE.head()
156
157 norm = MSE.plot(x="škála", y=["n1", "n2", "n3", "n4", "n5", "n6", "n7", "n8", "n9",
158 "n10", "n11", "n12", "n13", "n14", "n15"], figsize = (9,6), legend = False, color
159 ="limegreen", title = "Porovnanie MSE kriviek normálneho sínusového rytmu a
    paroxyzmálnej fibrilácie predsiení")
    fibr = MSE.plot(x="škála", y=["f1", "f2", "f3", "f4", "f5", "f6", "f7", "f8", "f9",
    "f10", "f11", "f12", "f13", "f14", "f15"], legend = False, color="crimson", ax=norm)
160 leg = MSE.plot(x="škála", y= ["n1", "f1"], color = ["limegreen", "crimson"], label =
    ['Normálny sínusový rytmus', 'Paroxyzmálna fibrilácia predsiení'], ax = norm)
161 leg.get_legend().set_bbox_to_anchor((0.55, 0.85))

162 norm.set_ylabel("SampEn");
163 norm.set_xlabel("škála");
164 norm.set_xticks(MSE["škála"])
165 norm.set_ylim(0, 2.6)
166
167 plt.savefig('mse_norm+fibr15.jpg', dpi=300, bbox_inches='tight')
168
169 MSE pre biely a 1/f šum
170
171 from random import gauss
172 from random import seed
173 from pandas import Series
174 import colorednoise as cn
175 import math
176 from scipy.integrate import quad
177
178 # generuje náhodné čísla
179 seed(1)
180 # vytvorí white noise časový rad
181 white_noise = [gauss(0.0, 1.0) for i in range(20000)]
182 WN = np.array(white_noise)
183 mseWN = mse(white_noise, 2, 0.15, 20)
184
185 # vytvorí časový rad 1/f šumu
186 beta = 1 # exponent
187 samples = 20000 # počet generovaných hodnôt
188 pink_noise = cn.powerlaw_psd_gaussian(beta, samples)
189 PN = np.array(pink_noise)
190 msePN = mse(PN, 2, 0.15, 20)

```



```

191
192 df_noise = pd.DataFrame(np.array([mseWN, msePN]).T, columns = ('white', 'pink'))
193 df_noise['škála'] = df_noise.index+1
194 df_noise.head()
195
196 def white(x, tau):
197     return 1/2* np.sqrt(tau/(2*3.14)) * (math.erf((x+0.15)/np.sqrt(2/tau))-math.erf((x-
198 0.15)/np.sqrt(2/tau))) * np.exp(-1/2*x*x*tau)
199
200     def integrate(scale):
201         ret = []
202         for i in range(1, scale+1):
203             def white(x):
204                 return 1/2* np.sqrt(i/(2*3.14)) * (math.erf((x+0.15)/np.sqrt(2/i))-
205 math.erf((x-0.15)/np.sqrt(2/i))) * np.exp(-1/2*x*x*i)
206
207                 integral = -np.log(quad(white, -2,2))
208                 ret.append(integral[0])
209             return np.array(ret)
210 df_noise["num_white"] = integrate(20)
211 df_noise["num_pink"] = 1.8*np.ones(20)
212 df_noise.head()
213
214 ax = df_noise.plot.scatter(x=["škála", "škála"], y=["white", "pink"], c= "goldenrod",
215 figsize = (9,6), legend = False, title = "Biely a 1/f šum")
216 c = df_noise.plot(x= "škála", y = "num_white", color = 'lightsteelblue', legend = False,
217 ax=ax )
218 d = df_noise.plot( x = "škála", y = "num_pink", color='rosybrown', ax=ax)
219
220 ax.set_ylabel("SampEn");
221 ax.set_xlabel("škála");
222 ax.set_xticks(df_noise["škála"])
223 ax.legend(["Biely šum", "1/f šum", "MSE"])
224 ax.set_ylim(0, 2.6)
225
226 plt.savefig('white_pink_MSE', dpi=300., bbox_inches='tight')
227
228 MSE a vplyv parametrov m a r
229
230 # Vypočíta MSE pre rôzne r
231 mse10r = mse(n1, 2, 0.10, 20)
232 mse15r = mse(n1, 2, 0.15, 20)
233 mse20r = mse(n1, 2, 0.20, 20)
234 mse25r = mse(n1, 2, 0.25, 20)
235 mse30r = mse(n1, 2, 0.30, 20)
236
237 # Vypočíta MSE pre rôzne m
238 mse1m = mse(n1, 1, 0.15, 20)
239 mse2m = mse(n1, 2, 0.15, 20)
240 mse3m = mse(n1, 3, 0.15, 20)

```

```

239 mse4m = mse(n1, 4, 0.15, 20)
240 mse5m = mse(n1, 5, 0.15, 20)
241
242 dfR = pd.DataFrame()
243 dfR['R10'] = np.array(mse10r)
244 dfR['R15'] = np.array(mse15r)
245 dfR['R20'] = np.array(mse20r)
246 dfR['R25'] = np.array(mse25r)
247 dfR['R30'] = np.array(mse30r)
248
249 dfR['škála'] = dfR.index+1
250
251 # Vykreslí dáta do grafov, uloží ako obrázok
252 ax = dfR.plot(x="škála", y=["R10", "R15", "R20", "R25", "R30"], figsize = (9,6), legend
253 = False, colormap="Set2", title = "MSE krivky normálneho sinusového rytmu s rôznym
254 parametrom r")
    ax.legend(['r= 0.10', 'r= 0.15', 'r= 0.20', 'r= 0.25', 'r= 0.30'], loc='center left',
    bbox_to_anchor=(1.0, 0.5))
255 ax.set_ylabel("SampEn");
256 ax.set_xlabel("škála");
257 ax.set_xticks(dfR["škála"])
258 ax.set_ylim(0, 2.6)
259
260 plt.savefig('diff_r.jpg', dpi=300, bbox_inches='tight')
261
262 dfM = pd.DataFrame()
263 dfM['M1'] = np.array(mse1m)
264 dfM['M2'] = np.array(mse2m)
265 dfM['M3'] = np.array(mse3m)
266 dfM['M4'] = np.array(mse4m)
267 dfM['M5'] = np.array(mse5m)
268
269 dfM['škála'] = dfM.index+1
270
271 # Vykreslí dáta do grafov, uloží ako obrázok
272 m = dfM.plot(x="škála", y=["M1", "M2", "M3", "M4", "M5"], figsize = (9,6), legend =
273 False, colormap="Dark2", title = "MSE krivky normálneho sinusového rytmu s rôznym
274 parametrom m")
    m.legend(['m = 1', 'm = 2', 'm = 3', 'm = 4', 'm = 5'], loc='center left',
    bbox_to_anchor=(1.0, 0.5))
275 m.set_ylabel("SampEn");
276 m.set_xlabel("škála");
277 m.set_xticks(dfR["škála"])
278 m.set_ylim(0, 2.6)
279
280 plt.savefig('diff_m.jpg', dpi=300, bbox_inches='tight')
281
282
283 Detrended Fluctuation Analysis
284
285 import fathon
286 from fathon import fathonUtils as fu

```

```

287 def obj_dfa(series):
288     cs = fu.toAggregated(series)
289     dfa = fathon.DFA(cs)
290     return dfa
291
292 #dfa trieda - objekt, normalny rytmus
293 dfa1n = obj_dfa(n1)
294 dfa2n = obj_dfa(n2)
295 dfa3n = obj_dfa(n3)
296 dfa4n = obj_dfa(n4)
297 dfa5n = obj_dfa(n5)
298 dfa6n = obj_dfa(n6)
299 dfa7n = obj_dfa(n7)
300 dfa8n = obj_dfa(n8)
301 dfa9n = obj_dfa(n9)
302 dfa10n = obj_dfa(n10)
303 dfa11n = obj_dfa(n11)
304 dfa12n = obj_dfa(n12)
305 dfa13n = obj_dfa(n13)
306 dfa14n = obj_dfa(n14)
307 dfa15n = obj_dfa(n15)
308
309 #dfa trieda - objekt, fibrilácie
310 dfa1f = obj_dfa(f1)
311 dfa2f = obj_dfa(f2)
312 dfa3f = obj_dfa(f3)
313 dfa4f = obj_dfa(f4)
314 dfa5f = obj_dfa(f5)
315 dfa6f = obj_dfa(f6)
316 dfa7f = obj_dfa(f7)
317 dfa8f = obj_dfa(f8)
318 dfa9f = obj_dfa(f9)
319 dfa10f = obj_dfa(f10)
320 dfa11f = obj_dfa(f11)
321 dfa12f = obj_dfa(f12)
322 dfa13f = obj_dfa(f13)
323 dfa14f = obj_dfa(f14)
324 dfa15f = obj_dfa(f15)
325
326 #okná s 4 - 2000 dohnotami s krokom 1
327 winSizes = fu.linRangeByStep(4, 2000)
328
329 #výpočet fluktuácií F v okne s N 4 bodmi, 5 bodmi, 6....
330 nn, Fn = dfaln.computeFlucVec(winSizes, revSeg=True)
331 nf, Ff = dfalf.computeFlucVec(winSizes, revSeg=True)
332
333 #nafituje hodnoty fluktuácií F vypočítaných hore a intercept - kde to pretne y osu
334 Hn, Hn_intercept = dfaln.fitFlucVec(logBase = 10)
335 Hf, Hf_intercept = dfalf.fitFlucVec(logBase = 10)
336 ax = plt.plot(np.log10(nf), np.log10(Ff), 'r^', alpha = 0.5)
337 plt.plot(np.log10(nn), np.log10(Fn), 'gd', alpha = 0.5)
338

```

```

339 plt.plot(np.log10(nn), Hn_intercept+Hn*np.log10(nn), 'b-', lw = 1.5, label='αN =
340 {:.2f}'.format(Hn))
341 plt.plot(np.log10(nf), Hf_intercept+Hf*np.log10(nf), 'k-', lw = 1.5, label='αF =
342 {:.2f}'.format(Hf))
343 plt.xlabel('log(n)')
344 plt.ylabel('log(F(n))')
345 plt.title('Fluktuácie okien v závislosti na ich veľkosti n', fontsize=14)
346 plt.legend(loc=0, fontsize=12)
347 plt.ylim(-2.2, 2.4)
348 plt.savefig('F_na_n_1', dpi=300, bbox_inches='tight')
349
350 #pre výpočet L1 a L2
351 limits_list = np.array([[4,16], [16, 256]], dtype=int)
352
353 # nafituje fluktuácie na rôznych intervaloch osobitne
354 list_Hn, list_Hn_intercept = dfaln.multiFitFlucVec(limits_list, logBase = 10)
355 list_Hf, list_Hf_intercept = dfalf.multiFitFlucVec(limits_list, logBase = 10)
356
357 clr = ['b', 'k', 'm', 'c', 'y']
358 stls = ['-', '--', '-.-']
359 plt.plot(np.log10(nf), np.log10(Ff), 'r^', alpha = 0.5)
360 plt.plot(np.log10(nn), np.log10(Fn), 'gd', alpha = 0.5)
361
362
363 for i in range(len(list_Hn)):
364     n_rng = np.arange(limits_list[i][0], limits_list[i][1]+1)
365     plt.plot(np.log10(n_rng), list_Hn_intercept[i]+list_Hn[i]*np.log10(n_rng),
366             clr[i%len(clr)]+stls[(i//len(clr))%len(stls)], label='αN =
367 {:.2f}'.format(list_Hn[i]))
368     for i in range(len(list_Hf)):
369         n_rng = np.arange(limits_list[i][0], limits_list[i][1]+1)
370         plt.plot(np.log10(n_rng), list_Hf_intercept[i]+list_Hf[i]*np.log10(n_rng),
371                 clr[i%len(clr)]+stls[(i//len(clr))%len(stls)], label='αF =
372 {:.2f}'.format(list_Hf[i]))
373
374     plt.xlabel('log(n)')
375     plt.ylabel('log(F(n))')
376 plt.title('Škálovacie exponenty pre malé a veľké n', fontsize=14)
377 plt.legend(loc=0, fontsize=10)
378 plt.ylim(-2.2, 2.4)
379
380 plt.savefig('male_velke_n_1', dpi=300, bbox_inches='tight')
381
382 def L(dfaobj, limits_list, winSizes):
383     fv = dfaobj.computeFlucVec(winSizes, revSeg=True)
384     alpha = dfaobj.multiFitFlucVec(limits_list, logBase = 10)[0]
385     return alpha
386
387 columns_n = ["L1n", "L2n"]
388 Ln = pd.DataFrame([L(dfaobj, limits_list, winSizes),

```

```

385         L(dfa2n, limits_list, winSizes),
386         L(dfa3n, limits_list, winSizes),
387         L(dfa4n, limits_list, winSizes),
388         L(dfa5n, limits_list, winSizes),
389         L(dfa6n, limits_list, winSizes),
390         L(dfa7n, limits_list, winSizes),
391         L(dfa8n, limits_list, winSizes),
392         L(dfa9n, limits_list, winSizes),
393         L(dfa10n, limits_list, winSizes),
394         L(dfa11n, limits_list, winSizes),
395         L(dfa12n, limits_list, winSizes),
396         L(dfa13n, limits_list, winSizes),
397         L(dfa14n, limits_list, winSizes),
398         L(dfa15n, limits_list, winSizes)], columns = columns_n)
399
400 meanN = Ln.mean(axis = 0)
401 steN = Ln.std(axis = 0)/np.sqrt(14)
402 meanN, steN
403
404 columns_f = ["L1f", "L2f"]
405
406 Lf = pd.DataFrame([L(dfa1f, limits_list, winSizes),
407                    L(dfa2f, limits_list, winSizes),
408                    L(dfa3f, limits_list, winSizes),
409                    L(dfa4f, limits_list, winSizes),
410                    L(dfa5f, limits_list, winSizes),
411                    L(dfa6f, limits_list, winSizes),
412                    L(dfa7f, limits_list, winSizes),
413                    L(dfa8f, limits_list, winSizes),
414                    L(dfa9f, limits_list, winSizes),
415                    L(dfa10f, limits_list, winSizes),
416                    L(dfa11f, limits_list, winSizes),
417                    L(dfa12f, limits_list, winSizes),
418                    L(dfa13f, limits_list, winSizes),
419                    L(dfa14f, limits_list, winSizes),
420                    L(dfa15f, limits_list, winSizes)], columns =columns_f)
421
422 meanF = Lf.mean(axis = 0)
423 steF = Lf.std(axis = 0)/np.sqrt(14)
424 meanF, steF
425
426 ax = Ln.plot.scatter(x = "L1n", y = "L2n", s = 45, figsize = (7,7), c =
427 "mediumseagreen")
428 f = Lf.plot.scatter(x="L1f", y="L2f", s = 45, figsize = (7,7), ax=ax, c = "lightcoral")
429 f.legend(['Normálny sinusový rytmus', 'Paroxyzmálna fibrilácia predsieni'], loc='lower
429 right', bbox_to_anchor=(1.0, 0.15), fontsize = 12)
430 plt.title("Škálovacie koeficienty  $\alpha_1$  versus  $\alpha_2$ ", fontsize = 13)
431 ax.set_xlabel(" $\alpha_1$ ");
432 ax.set_ylabel(" $\alpha_2$ ")
433 ax.set_xlim(0.4, 1.55)
434 ax.set_ylim(0.4, 1.55)
435

```

```

435 plt.savefig('L1L2', dpi=300, bbox_inches='tight')
436 plt.show()
437
438 def H(dfaobj, winSizes):
439     fv = dfaobj.computeFlucVec(winSizes, revSeg=True)
440     H = dfaobj.fitFlucVec(logBase = 10)
441     return H
442
443 columns_nh = ["Hn", "H_intn"]
444 HuN = pd.DataFrame([H(dfa1n, winSizes),
445                     H(dfa2n, winSizes),
446                     H(dfa3n, winSizes),
447                     H(dfa4n, winSizes),
448                     H(dfa5n, winSizes),
449                     H(dfa6n, winSizes),
450                     H(dfa7n, winSizes),
451                     H(dfa8n, winSizes),
452                     H(dfa9n, winSizes),
453                     H(dfa10n, winSizes),
454                     H(dfa11n, winSizes),
455                     H(dfa12n, winSizes),
456                     H(dfa13n, winSizes),
457                     H(dfa14n, winSizes),
458                     H(dfa15n, winSizes)], columns = columns_nh)
459 HuN.head()
460
461 meanNh = HuN.mean(axis = 0)
462 steNh = HuN.std(axis = 0)/np.sqrt(14)
463 meanNh, steNh
464
465 columns_fh = ["Hf", "H_intf"]
466 HuF = pd.DataFrame([H(dfa1f, winSizes),
467                     H(dfa2f, winSizes),
468                     H(dfa3f, winSizes),
469                     H(dfa4f, winSizes),
470                     H(dfa5f, winSizes),
471                     H(dfa6f, winSizes),
472                     H(dfa7f, winSizes),
473                     H(dfa8f, winSizes),
474                     H(dfa9f, winSizes),
475                     H(dfa10f, winSizes),
476                     H(dfa11f, winSizes),
477                     H(dfa12f, winSizes),
478                     H(dfa13f, winSizes),
479                     H(dfa14f, winSizes),
480                     H(dfa15f, winSizes)], columns = columns_fh)
481 HuF.head()
482
483 meanFh = HuF.mean(axis = 0)
484 steFh = HuF.std(axis = 0)/np.sqrt(14)
485 meanFh, steFh
486

```

```

487 columns_FV = ["window", "n11", "n12", "n13", "n14", "n15", "f11", "f12", "f13", "f14",
488 "f15"]
489 fluctVec = pd.DataFrame([dfa1n.computeFlucVec(winSizes, revSeg=True)[0],
                           dfa11n.computeFlucVec(winSizes, revSeg=True)[1],
490                           dfa12n.computeFlucVec(winSizes, revSeg=True)[1],
491                           dfa13n.computeFlucVec(winSizes, revSeg=True)[1],
492                           dfa14n.computeFlucVec(winSizes, revSeg=True)[1],
493                           dfa15n.computeFlucVec(winSizes, revSeg=True)[1],
494                           dfa11f.computeFlucVec(winSizes, revSeg=True)[1],
495                           dfa12f.computeFlucVec(winSizes, revSeg=True)[1],
496                           dfa13f.computeFlucVec(winSizes, revSeg=True)[1],
497                           dfa14f.computeFlucVec(winSizes, revSeg=True)[1],
498                           dfa15f.computeFlucVec(winSizes, revSeg=True)[1]], index
499 =columns_FV)
500 fluctVec.head()

501 Vec = fluctVec.T
502
503 norm = ListedColormap(["lightgreen", "mediumseagreen", "yellowgreen", "limegreen",
504 "forestgreen"])
505 fibr = ListedColormap(["lightcoral", "salmon", "firebrick", "indianred", "tomato"])

506 no = Vec.plot(x="window", y=["n11", "n12", "n13", "n14", "n15"], loglog = True,
507 figsize = (7,6), legend = False, colormap= norm)
508 fi = Vec.plot(x="window", y=["f11", "f12", "f13", "f14", "f15"], loglog = True,
                    legend = False, colormap = fibr, ax=no)
509
                    leg = Vec.plot(x="window", y= ["n11", "f11"], color = ["lightgreen", "lightcoral"],
510 label = ['Normálny sínusový rytmus', 'Paroxyzmálna fibrilácia predsieni'], ax = no)
511 leg.get_legend().set_bbox_to_anchor((0.55, 1))

512
513 plt.title("DFA krivky normálneho sínusového rytmu a paroxyzmálnej fibrilácie
514 predsieni", fontsize = 13)
515 no.set_ylabel("F(n)");
                    no.set_xlabel("n");
516 plt.savefig('DFA_krivky_5v5', dpi=300, bbox_inches='tight')
517 plt.show()
518
519
520 DFA pre biely a 1/f šum
521
522 dfaWN = obj_dfa(WN)
523 dfaPN = obj_dfa(PN)
524
525 #výpočet fluktuácií F v okne s N 4 bodmi, 5 bodmi, 6...
526 nW, FW = dfaWN.computeFlucVec(winSizes, revSeg=True)
527 nP, FP = dfaPN.computeFlucVec(winSizes, revSeg=True)
528
529 #nafituje hodnoty fluktuácií F vypočítaných hore a intercept - kde to pretne y osu
530 HWN, HWN_intercept = dfaWN.fitFlucVec(logBase = 10)
531 HPN, HPN_intercept = dfaPN.fitFlucVec(logBase = 10)

```

```
532 ax = plt.plot(np.log10(nP), np.log10(FP), 'm^', alpha = 0.4)
533 plt.plot(np.log10(nW), np.log10(FW), 'd', alpha = 0.4)
534
535 plt.plot(np.log10(nW), HWN_intercept+HWN*np.log10(nW), 'b-', lw = 1.5, label='α biely
    šum = {:.2f}'.format(HWN))
536 plt.plot(np.log10(nP), HPN_intercept+HPN*np.log10(nP), 'k-', lw = 1.5, label='α 1/f šum
    = {:.2f}'.format(HPN))
537
538 plt.xlabel('log(n)')
539 plt.ylabel('log(F(n))')
540 plt.title('Fluktuácie okien v závislosti na ich veľkosti n pre biely a 1/f šum',
    fontsize=14)
541 plt.legend(loc=0, fontsize=12)
542 plt.ylim(-2.2, 2.4)
543
544 plt.savefig('F_na_n_WN-PN', dpi=300, bbox_inches='tight')
```



## PRÍLOHA 2

Program použitý k analýze dát v programovacom jazyku Python.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import wfdb
5 from wfdb import processing
6
7 # Vypočíta intervaly medzi R píkmi pomocou moving wave integration wavelets
8 def HRZ(name):
9     rec = wfdb.rdrecord(name, sampsto = 8800000)
10    r_loc = []
11    r_loc = processing.xqrs_detect(rec.p_signal[:,0], fs=rec.fs)
12    hr = np.zeros(len(r_loc)-1)
13    hr = np.diff(r_loc)/rec.fs
14    return hr
15
16 # Vypočíta Sample Entropy
17 def sampen(L, m, r):
18    N = len(L)
19    B = 0.0
20    A = 0.0
21
22    # Tu xmj je JEDNA array vektorov ktoré odčítam od VŠETKÝCH xmi vektorkov
23    # Rozdelí časový rad a uloží templates dĺžky m
24    xmi = np.array([L[i : i + m] for i in range(N - m)]) #tak aby bolo možné vytvoriť
        m+1 vektor
25    xmj = np.array([L[i : i + m] for i in range(N - m + 1)])
26
27    # Uloží matches a vylúči self-match, vypočíta A
28    # np.abs+max(axis = 1) --> najväčšia vzdialenosť cez akúkoľvek súradnicu
29    # -1 vyhodí self-match
30    A = np.sum([np.sum(np.abs(xmii - xmj).max(axis=1) <= r) - 1 for xmii in xmi])
31
32    # Podobne pre B
33    m += 1
34    xm = np.array([L[i : i + m] for i in range(N - m + 1)])
35
36    B = np.sum([np.sum(np.abs(xmi - xm).max(axis=1) <= r) - 1 for xmi in xm])
37
38    # Vráti SampEn
39    return -np.log(B/A)
40
41 # skonštruuje časové rady so škálou
42 def cgts(time_series, scale):
43    #počet hodnôt pôvodnej ts
44    n = len(time_series)
45    #fix zaokrúhľuje nadol, počet hodnôt vytvorenej cgts
46    b = int(np.fix(n / scale))
```

```

47     #zoberie 1d array, vyhodí posledný údaj (nepárny), a urobí z toho array bxscale
48     temp = np.reshape(time_series[0:b*scale], (b, scale))
49     #priemer po riadkoch
50     cts = np.mean(temp, axis = 1)
51     return cts
52
53 #počíta mse
54 def mse(time_series, m, r, scale):
55     ret = []
56     for i in range(1, scale+1):
57         cts = cgts(time_series, i)
58         std = np.std(time_series)
59         se = sampen(cts, m, r*std)
60         if se== 0:
61             ret.append(( N))
62         else:
63             ret.append((se ))
64     print("completed")
65     return ret
66
67 n1 = HRZ('16265')
68 n3 = HRZ('16273')
69 n12 = HRZ('19090')
70 n15 = HRZ('19830')
71 n10 = HRZ('17052')
72 n13 = HRZ('19093')
73
74 len(n1), len(n3), len(n12), len(n15)
75 (89068, 84061, 81029, 101605)
76 len(n10), len(n13)
77 (76898, 71666)
78
79 def trim(series):
80     series = np.reshape(series[0:80000], (8, 10000))
81     a = series[0,:]
82     b = series[1,:]
83     c = series[2,:]
84     d = series[3,:]
85     e = series[4,:]
86     f = series[5,:]
87     g = series[6,:]
88     h = series[7,:]
89     return a, b, c, d, e, f, g, h
90 def trim2(series):
91     series = np.reshape(series[0:70000], (7, 10000))
92     a = series[0,:]
93     b = series[1,:]
94     c = series[2,:]
95     d = series[3,:]
96     e = series[4,:]
97     f = series[5,:]
98     g = series[6,:]

```

```
99
100     return a, b, c, d, e, f, g
101
102 n1a, n1b, n1c, n1d, n1e, n1f, n1g, n1h = trim(n1)
103 n3a, n3b, n3c, n3d, n3e, n3f, n3g, n3h = trim(n3)
104 n12a, n12b, n12c, n12d, n12e, n12f, n12g, n12h = trim(n12)
105 n15a, n15b, n15c, n15d, n15e, n15f, n15g, n15h = trim(n15)
106 n10a, n10b, n10c, n10d, n10e, n10f, n10g = trim2(n10)
107 n13a, n13b, n13c, n13d, n13e, n13f, n13g = trim2(n13)
108
109 mse3a = mse(n3a, 2, 0.15, 20)
110 mse3b = mse(n3b, 2, 0.15, 20)
111 mse3c = mse(n3c, 2, 0.15, 20)
112 mse3d = mse(n3d, 2, 0.15, 20)
113 mse3e = mse(n3e, 2, 0.15, 20)
114 mse3f = mse(n3f, 2, 0.15, 20)
115 mse3g = mse(n3g, 2, 0.15, 20)
116 mse3h = mse(n3h, 2, 0.15, 20)
117 mse1a = mse(n1a, 2, 0.15, 20)
118 mse1b = mse(n1b, 2, 0.15, 20)
119 mse1c = mse(n1c, 2, 0.15, 20)
120 mse1d = mse(n1d, 2, 0.15, 20)
121 mse1e = mse(n1e, 2, 0.15, 20)
122 mse1f = mse(n1f, 2, 0.15, 20)
123 mse1g = mse(n1g, 2, 0.15, 20)
124 mse1h = mse(n1h, 2, 0.15, 20)
125 mse15a = mse(n15a, 2, 0.15, 20)
126 mse15b = mse(n15b, 2, 0.15, 20)
127 mse15c = mse(n15c, 2, 0.15, 20)
128 mse15d = mse(n15d, 2, 0.15, 20)
129 mse15e = mse(n15e, 2, 0.15, 20)
130 mse15f = mse(n15f, 2, 0.15, 20)
131 mse15g = mse(n15g, 2, 0.15, 20)
132 mse15h = mse(n15h, 2, 0.15, 20)
133 mse12a = mse(n12a, 2, 0.15, 20)
134 mse12b = mse(n12b, 2, 0.15, 20)
135 mse12c = mse(n12c, 2, 0.15, 20)
136 mse12d = mse(n12d, 2, 0.15, 20)
137 mse12e = mse(n12e, 2, 0.15, 20)
138 mse12f = mse(n12f, 2, 0.15, 20)
139 mse12g = mse(n12g, 2, 0.15, 20)
140 mse12h = mse(n12h, 2, 0.15, 20)
141 mse10a = mse(n10a, 2, 0.15, 20)
142 mse10b = mse(n10b, 2, 0.15, 20)
143 mse10c = mse(n10c, 2, 0.15, 20)
144 mse10d = mse(n10d, 2, 0.15, 20)
145 mse10e = mse(n10e, 2, 0.15, 20)
146 mse10f = mse(n10f, 2, 0.15, 20)
147 mse10g = mse(n10g, 2, 0.15, 20)
148 mse13a = mse(n13a, 2, 0.15, 20)
149 mse13b = mse(n13b, 2, 0.15, 20)
150 mse13c = mse(n13c, 2, 0.15, 20)
```

```
151 mse13d = mse(n13d, 2, 0.15, 20)
152 mse13e = mse(n13e, 2, 0.15, 20)
153 mse13f = mse(n13f, 2, 0.15, 20)
154 mse13g = mse(n13g, 2, 0.15, 20)
155
156 MSE = pd.DataFrame()
157 MSE["1a"] = mse1a
158 MSE["1b"] = mse1b
159 MSE["1c"] = mse1c
160 MSE["1d"] = mse1d
161 MSE["1e"] = mse1e
162 MSE["1f"] = mse1f
163 MSE["1g"] = mse1g
164 MSE["1h"] = mse1h
165 MSE["15a"] = mse15a
166 MSE["15b"] = mse15b
167 MSE["15c"] = mse15c
168 MSE["15d"] = mse15d
169 MSE["15e"] = mse15e
170 MSE["15f"] = mse15f
171 MSE["15g"] = mse15g
172 MSE["15h"] = mse15h
173 MSE["3a"] = mse3a
174 MSE["3b"] = mse3b
175 MSE["3c"] = mse3c
176 MSE["3d"] = mse3d
177 MSE["3e"] = mse3e
178 MSE["3f"] = mse3f
179 MSE["3g"] = mse3g
180 MSE["3h"] = mse3h
181 MSE["12a"] = mse12a
182 MSE["12b"] = mse12b
183 MSE["12c"] = mse12c
184 MSE["12d"] = mse12d
185 MSE["12e"] = mse12e
186 MSE["12f"] = mse12f
187 MSE["12g"] = mse12g
188 MSE["12h"] = mse12h
189 MSE["10a"] = mse10a
190 MSE["10b"] = mse10b
191 MSE["10c"] = mse10c
192 MSE["10d"] = mse10d
193 MSE["10e"] = mse10e
194 MSE["10f"] = mse10f
195 MSE["10g"] = mse10g
196 MSE["13a"] = mse13a
197 MSE["13b"] = mse13b
198 MSE["13c"] = mse13c
199 MSE["13d"] = mse13d
200 MSE["13e"] = mse13e
201 MSE["13f"] = mse13f
202 MSE["13g"] = mse13g
```

```

203
204 MSE['škála'] = MSE.index+1
205 MSE.head()
206
207 M45 = MSE.plot(x="škála", y=["12a","12b", "12c", "12d", "12e","12f", "12g", "12h"],
    figsize = (9,6), legend = False, color=["lightsteelblue", "lavender","lightsteelblue",
    "paleturquoise", "lightsteelblue", "lightsteelblue", "lavender", "paleturquoise"],
    title = "MSE krivky úseků starších subjektů")
208 F50 = MSE.plot(x="škála", y=["15a","15b", "15c", "15d", "15e","15f", "15g", "15h"],
    legend = False, color=["bisque", "mistyrose", "navajowhite", "wheat", "bisque",
    "mistyrose", "navajowhite", "wheat"], ax=M45)
209 F45 = MSE.plot(x="škála", y=["10a","10b", "10c", "10d", "10e","10f", "10g"], legend =
    False, color=["palegreen", "lightgreen", "palegreen", "lightgreen", "palegreen",
    "lightgreen", "lightgreen"], ax=M45 )
210
211 leg = MSE.plot(x="škála", y= ["12a", "15a", "10a"], color = ["lightsteelblue",
    "bisque", "lightgreen"], label = ['M45','F50', 'F45'], ax = M45)
212 leg.get_legend().set_bbox_to_anchor((1, 1))
213
214 M45.set_ylabel("SampEn");
215 M45.set_xlabel("škála");
216 M45.set_xticks(MSE["škála"])
217 M45.set_ylim(0,2.6)
218
219 plt.savefig('stari2.png', dpi=300, bbox_inches='tight')
220
221 M32 = MSE.plot(x="škála", y=["1a", "1b", "1c", "1d", "1e", "1f", "1g", "1h"], figsize =
    (9,6), label = False, legend = False, color =["cornflowerblue" ], title = "MSE krivky
    úseků mladších subjektů")
222 F28 = MSE.plot(x="škála", y=["3a","3b", "3c", "3d", "3e","3f", "3g", "3h"], figsize =
    (9,6), label = False, legend = False, color=["red"], ax= M32)
223 M34 = MSE.plot(x="škála", y=["13a","13b", "13c", "13d", "13e","13f", "13g"], label =
    False, legend = False, color=["yellowgreen"], ax=M32 )
224
225 leg = MSE.plot(x="škála", y= ["1a", "3a", "13a"], color =["cornflowerblue", "red",
    "yellowgreen"], label = ['M32','F28', 'M34'], ax = M32)
226 leg.get_legend().set_bbox_to_anchor((1, 1))
227
228
229 M32.set_ylabel("SampEn");
230 M32.set_xlabel("škála");
231 M32.set_xticks(MSE["škála"])
232 M32.set_ylim(0,2.6)
233
234 plt.savefig('mladi2.png', dpi=300, bbox_inches='tight')
235
236 import fathon
237 from fathon import fathonUtils as fu
238
239 def hurst(series):
240     series = np.reshape(series[0:80000], (8, 10000))
241     a = series[0,:]

```

```

242     b = series[1,:]
243     c = series[2,:]
244     d = series[3,:]
245     e = series[4,:]
246     f = series[5,:]
247     g = series[6,:]
248     h = series[7,:]
249     csa = fu.toAggregated(a)
250     csb = fu.toAggregated(b)
251     csc = fu.toAggregated(c)
252     csd = fu.toAggregated(d)
253     cse = fu.toAggregated(e)
254     csf = fu.toAggregated(f)
255     csg = fu.toAggregated(g)
256     csh = fu.toAggregated(h)
257     dfa = fathon.DFA(csa)
258     dfb = fathon.DFA(csb)
259     dfc = fathon.DFA(csc)
260     dfd = fathon.DFA(csd)
261     dfe = fathon.DFA(cse)
262     dff = fathon.DFA(csf)
263     dfg = fathon.DFA(csg)
264     dfh = fathon.DFA(csh)
265
266     winSizes = fu.linRangeByStep(4, 2000)
267     limits_list = np.array([[4,16], [16, 256]], dtype=int)
268
269     fva = dfa.computeFlucVec(winSizes, revSeg=True)
270     fvb = dfb.computeFlucVec(winSizes, revSeg=True)
271     fvc = dfc.computeFlucVec(winSizes, revSeg=True)
272     fvd = dfd.computeFlucVec(winSizes, revSeg=True)
273     fve = dfe.computeFlucVec(winSizes, revSeg=True)
274     fvf = dff.computeFlucVec(winSizes, revSeg=True)
275     fvg = dfg.computeFlucVec(winSizes, revSeg=True)
276     fvh = dfh.computeFlucVec(winSizes, revSeg=True)
277
278     list_a, list_a_intercept = dfa.multiFitFlucVec(limits_list, logBase = 10)
279     list_b, list_b_intercept = dfb.multiFitFlucVec(limits_list, logBase = 10)
280     list_c, list_c_intercept = dfc.multiFitFlucVec(limits_list, logBase = 10)
281     list_d, list_d_intercept = dfd.multiFitFlucVec(limits_list, logBase = 10)
282     list_e, list_e_intercept = dfe.multiFitFlucVec(limits_list, logBase = 10)
283     list_f, list_f_intercept = dff.multiFitFlucVec(limits_list, logBase = 10)
284     list_g, list_g_intercept = dfg.multiFitFlucVec(limits_list, logBase = 10)
285     list_h, list_h_intercept = dfh.multiFitFlucVec(limits_list, logBase = 10)
286
287     H = [list_a[1], list_b[1], list_c[1], list_d[1], list_e[1], list_f[1], list_g[1],
list_h[1]]
288     return H
289
290 def hurst2(series):
291     series = np.reshape(series[0:70000], (7, 10000))
292     a = series[0,:]

```

```

293     b = series[1,:]
294     c = series[2,:]
295     d = series[3,:]
296     e = series[4,:]
297     f = series[5,:]
298     g = series[6,:]
299     csa = fu.toAggregated(a)
300     csb = fu.toAggregated(b)
301     csc = fu.toAggregated(c)
302     csd = fu.toAggregated(d)
303     cse = fu.toAggregated(e)
304     csf = fu.toAggregated(f)
305     csg = fu.toAggregated(g)
306     dfa = fathon.DFA(csa)
307     dfb = fathon.DFA(csb)
308     dfc = fathon.DFA(csc)
309     dfd = fathon.DFA(csd)
310     dfe = fathon.DFA(cse)
311     dff = fathon.DFA(csf)
312     dfg = fathon.DFA(csg)
313
314     winSizes = fu.linRangeByStep(4, 2000)
315     limits_list = np.array([[4,16], [16, 256]], dtype=int)
316
317     fva = dfa.computeFlucVec(winSizes, revSeg=True)
318     fvb = dfb.computeFlucVec(winSizes, revSeg=True)
319     fvc = dfc.computeFlucVec(winSizes, revSeg=True)
320     fvd = dfd.computeFlucVec(winSizes, revSeg=True)
321     fve = dfe.computeFlucVec(winSizes, revSeg=True)
322     fvf = dff.computeFlucVec(winSizes, revSeg=True)
323     fvg = dfg.computeFlucVec(winSizes, revSeg=True)
324
325     list_a, list_a_intercept = dfa.multiFitFlucVec(limits_list, logBase = 10)
326     list_b, list_b_intercept = dfb.multiFitFlucVec(limits_list, logBase = 10)
327     list_c, list_c_intercept = dfc.multiFitFlucVec(limits_list, logBase = 10)
328     list_d, list_d_intercept = dfd.multiFitFlucVec(limits_list, logBase = 10)
329     list_e, list_e_intercept = dfe.multiFitFlucVec(limits_list, logBase = 10)
330     list_f, list_f_intercept = dff.multiFitFlucVec(limits_list, logBase = 10)
331     list_g, list_g_intercept = dfg.multiFitFlucVec(limits_list, logBase = 10)
332
333     H = [list_a[1], list_b[1], list_c[1], list_d[1], list_e[1], list_f[1], list_g[1]]
334     return H
335
336 h1 = hurst(n1)
337 h3 = hurst(n3)
338 h12 = hurst(n12)
339 h15 = hurst(n15)
340 h10 = hurst2(n10)
341 h13 = hurst2(n13)
342
343 HU = pd.DataFrame()
344 HU["h1"] = h1

```

```

345 HU["h3"] = h3
346 HU["h12"] = h12
347 HU["h15"] = h15
348
349 HU2 = pd.DataFrame()
350 HU2["h10"] = h10
351 HU2["h13"] = h13
352
353 MSE8 = MSE.drop([0, 1, 2, 3, 4, 5, 6], axis = 0)
354
355 mean2 = np.mean(MSE8, axis = 0)
356 MSE88 = pd.DataFrame()
357 MSE88["M32"] = mean2[0], mean2[1], mean2[2], mean2[3], mean2[4], mean2[5], mean2[6],
    mean2[7]
358 MSE88["F50"] = mean2[8], mean2[9], mean2[10], mean2[11], mean2[12], mean2[13],
    mean2[14], mean2[15]
359 MSE88["F28"] = mean2[16], mean2[17], mean2[18], mean2[19], mean2[20], mean2[21],
    mean2[22], mean2[23]
360 MSE88["M45"] = mean2[24], mean2[25], mean2[26], mean2[27], mean2[28], mean2[29],
    mean2[30], mean2[31]
361
362 MSE88_2 = pd.DataFrame()
363 MSE88_2["F45"] = mean2[0], mean2[1], mean2[2], mean2[3], mean2[4], mean2[5], mean2[6]
364 MSE88_2["M34"] = mean2[7], mean2[8], mean2[9], mean2[10], mean2[11], mean2[12],
    mean2[13]
365
366 HFP_2 = pd.concat([MSE88_2, HU2], axis=1)
367
368 M32 = HFP.plot.scatter(x = "h1", y = "M32", s = 45, figsize = (7,7), c =
    "cornflowerblue")
369 F28 = HFP.plot.scatter(x = "h3", y = "F28", s = 45, figsize = (7,7), ax=M32, c = "red")
370 M45 = HFP.plot.scatter(x = "h12", y = "M45", s = 45, figsize = (7,7), ax=M32, c =
    "lightsteelblue")
371 F50 = HFP.plot.scatter(x = "h15", y = "F50", s = 45, figsize = (7,7), ax=M32, c =
    "lightsalmon")
372 M34 = HFP_2.plot.scatter(x = "h13", y = "M34", s = 45, figsize = (7,7), ax=M32, c =
    "yellowgreen")
373 F45 = HFP_2.plot.scatter(x = "h10", y = "F45", s = 45, figsize = (7,7), ax=M32, c =
    "lightgreen")
374
375 M32.legend(["M32", "F28", "M45", "F50", "M34", "F45"], loc='lower right',
376 bbox_to_anchor=(1.2, 0.5), fontsize = 12)
377 plt.title("Škálovacie koeficienty  $\alpha_2$  versus priemer SampEn hodnôt škál 8-20", fontsize
    = 13)
378 M32.set_xlabel(" $\alpha$ ");
379 M32.set_ylabel("SampEn")
380 M32.set_xlim(0.7, 1.3)
381 M32.set_ylim(0.7, )
382
383 plt.savefig('L2vs8-20_subj2.png', dpi=300, bbox_inches='tight')
384 plt.show()

```