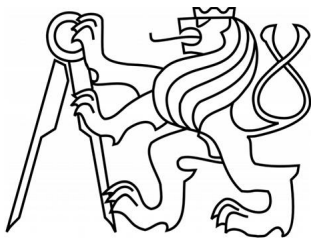
Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Cybernetics



Bachelor thesis

Processing of Neurophysiological Data

Filip Albert

Supervisor: Ing. Daniel Novák, Ph.D.

Study Programme: Cybernetics and Robotics

Specialisation: Robotics

May 25, 2012

Acknowledgements

I would like to thank to my supervisor Ing. Daniel Novák, Ph.D. for his unselfish help, valuable advices, and for very interesting topic of my thesis. I would also like to thank to my parents and grandparents for their unflagging support.

Prohlášení autora práce

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

V Praze dne 20.5.2012 ..... Filip Albert .....

Podpis autora práce

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Filip Albert

Studijní program: Kybernetika a robotika (bakalářský)

Obor: Robotika

Název tématu: Zpracování neurofyziologických dat

Pokyny pro vypracování:

1. Porovnejte různé softwarové implementace pro paralelní programování.
2. Vybraný software použijte pro optimalizaci výpočtů pomocí vybrané metody komplexnosti signálu.
3. Porovnejte metody pro výpočet komplexnosti signálu na neurofyziologických signálech.

Seznam odborné literatury:

[1] Hsiao-Lung Chan, Ming-An Lin, Shih-Tseng Lee, Yu-Tai Tsai, Pei-Kuang Chao, Tony Wu: Complexity analysis of neuronal spike trains of deep brain nuclei in patients with Parkinson's disease, Brain Research Bulletin, n.81,vol 6, p.534-42,2003
[2] Steven M. Pincus: Approximate entropy as a measure of system complexity: Proc Natl Acad Sci U S A. ,v88., no. 6, p.2297-2301,1991
[3] Yi Zheng, Jianbo Gao, Justin C. Sanchez, Jose C. Principe, Michael S. Okun: Multiplicative multifractal modeling and discrimination of human neuronal activity,vol.344,no2-4,p.253-264, Physics Letters A 344, 2005

Vedoucí bakalářské práce: Ing. Daniel Novák, Ph.D.

Platnost zadání: do konce zimního semestru 2012/2013

L.S.

prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 18. 1. 2012

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Filip Albert

Study programme: Cybernetics a Robotics

Specialisation: Robotics

Title of Bachelor Project: Processing of Neurophysiological Data

Guidelines:

1. Compare various software solutions for parallel programming.
2. Use chosen framework for optimalization of complexity measure.
3. Compare various complexity measure methods for neurophysiological data.

Bibliography/Sources:

[1] Hsiao-Lung Chan, Ming-An Lin, Shih-Tseng Lee, Yu-Tai Tsai, Pei-Kuang Chao, Tony Wu: Complexity analysis of neuronal spike trains of deep brain nuclei in patients with Parkinson's disease, Brain Research Bulletin, n.81,vol 6, p.534-42,2003

[2] Steven M. Pincus: Approximate entropy as a measure of system complexity: Proc Natl Acad Sci U S A. ,v88., no. 6, p.2297-2301,1991

[3] Yi Zheng, Jianbo Gao, Justin C. Sanchez, Jose C. Principe, Michael S. Okun: Multiplicative multifractal modeling and discrimination of human neuronal activity,vol.344,no2-4,p.253-264, Physics Letters A 344, 2005

Bachelor Project Supervisor: Ing. Daniel Novák, Ph.D.

Valid until: the end of the winter semester of academic year 2012/2013

L.S.

prof. Ing. Vladimír Mařík, DrSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 18, 2012

Abstract

This bachelor project deals with analysis of neurological data using regularity measures. The signals are microelectrode recordings acquired during deep brain stimulation surgery. This project's aim is find suitable measure for classification of basal ganglia's cores, namely subthalamic nucleus, globus pallidus and substantia nigra, and adjacent parts of thalamus. Capability of an automatic signal classification would be a considerable benefit during operations, where it is necessary to locate cores precisely. All measures passed ANOVA/Kruskal-Wallis test on significance level $\alpha = 0.05$ while the best discrimination in term of p-value was achieved for sample entropy. Two used regularity measures, Lempel-Ziv complexity and sample entropy, provided good results to discriminate different basal ganglia nuclei.

Keywords: Parkinson's disease, signal processing, core classification, complexity measure

Abstrakt

Tato bakalářská práce se zabývá analýzou neurologických dat pomocí metod meření komplexity signálu. Těmito signály jsou naměřené akční potenciály získané při operaci pomocí hloubkové mozkové stimulace. Cílem této práce je najít vhodnou metodu pro klasifikaci jader bazálních ganglií, konkrétně subthalamického jádra, globu pallidu a černé substance, a přilehlých částí thalamu. Schopnost automatické klasifikace těchto signálů by byla značným přínosem při operacích, u kterých je nutné přesně daná jádra zaměřit. Všechny metody prošly ANOVA/Kruskal-Wallisovým testem na hladině významnosti $\alpha = 0.05$. Samplovací entropie byla z hlediska p-hodnoty nejlepší mírou použitou k diskriminaci jader. Dvě použité míry komplexity signálu, Lempel-Zivova komplexita a samplovací entropie, poskytly dobré výsledky pro rozlišení rozdílných jader basálních ganglií.

Klíčová slova: Parkinsonova nemoc, zpracování signálu, klasifikace jader, komplexita signálu

Contents

Chapter 1

Introduction

As human average age increases, the prevalence of Parkinson's disease increases too. This places higher demandes on a healthcare system. A treatment of the Parkinson's disease can be divided into two main parts, the treatment with medicaments and a surgery. In recent years, the surgery has been becoming more and more popular. In the Parkinson's disease surgery, there is necessary to lead thin electrode through a brain to the structures, which are only a few milimeters small [39]. Very precise targeting of the chosen structure is important. Unfortunately, the surgeon's possibilities how to do it are limited, so many scientists has been interested in how to find a method, which could help surgeon in the targeting. Dozens methods based on processing signals taken during the surgical operation were proposed. In this work I'll focus on the comparison of these methods and their application to real data sets.

This work was divided into six chapters. The first chapter is this introduction.

In the second chapter I'll focus on a description of parallel computing methods. There are methods, especially those that process raw data, which are very time-consuming. The aim of the parallel computing is to reduce the time consumed. I'll describe properties of three most used softwares.

There is needfull to describe some theoretical issues, so in the third chapter I'll give some informations about Parkinson's disease and related topics, such as a treatment of Parkinson's disease, participated brain structures, a deep brain stimulation and a measuring action potentiales and the data sets that I'll work with.

In the fourth chapter I'll describe and explain all used complexity measures such as entropy based algorithms and chaos based algorithms that have lot of representatives on the both sides.

The fifth chapter is devoted to results. I'll refer about a practical usage of a chosen parallel computing method and announce results and comparison of complexity measures.

In the sixth chapter I'll summarize the complexity measure results and refer about their usage to our data sets.

Chapter 2

Methods of parallel computing

2.1 Introduction to parallel computing

It is only few decades ago when computer was privilege of big corporations, important goverment projects and army. How the development of computers goes ahead it pushes costs of elder and simplier computers to the ground. The high cost and technological complexity caused that computers began to link together and processors began to split in cores. Subsequently, the parallel programming began to develop. This approach enabled to develop new demanding signal processing algorithms.

Since part of my thesis deals with such demanding algorithms, it was very far-sighted from my supervisor to start with survey of parallel programming methods. I got the assignment to compare various approaches, which were able to parallelize the Matlab code. Then I applied the chosen approach to the assigned problem. In the following text, there are described three various approaches of the parallel computing, namely Matlab Parallel Computing Toolbox, Condor and Sun Grid Engine. The choice of the Condor and the Sun Grid Engine was based on the fact, that these programs are one of the most frequently used open source environments. The choice of the Matlab Parallel Computing Toolboxu came naturally because our faculty has the license for it.

2.2 Condor

Condor project starded at University of Wisconsin-Madison in 1988. It has been proposed by Miron Livny in his doctoral thesis [48]. Since then, it is still under development. It has been used for example for rendering of 3D movies, analysis of stock markets, processing of biological data or simulations. There are two main products: The Condor high-throughput computing system (Condor), and the Condor-G agent for grid computing.

Condor is a high-throughput distributed batch computing system. Like other batch systems,

Condor provides a job management mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Users submit their jobs to Condor, and Condor subsequently chooses when and where to run them based upon a policy, monitors their progress, and ultimately informs the user upon completion [49].

There are several useful tools provided by Condor. Users can put demands to remote machine (minimum RAM capacity, faster floating point performance) for some group of jobs or vice versa. For this purpose there is ClassAds framework. It matches jobs with appropriate machines.

Condor provides also tool for checkpointing and migration. This allows Condor to move job to another machine if certain computer, where job runs, is going to be used by user. Thanks to checkpointing, job that moved to another machine is performed where it had been interrupted. Unfortunately, this advantege works under the Linux systems only and many conditions has to be complied there.

Condor runs jobs on the remote machine on the background. Remote system calls is one of Condor's mobile sandbox mechanisms for redirecting all of a jobs I/O related system calls back to the machine which submitted the job. Users do not need to make data files available on remote workstations before Condor executes their programs there, even in the absence of a shared filesystem [47, 49].

Condor runs under the Linux, Mac OS X, FreeBSD and with some restrictions also under the Windows operating system.

To run a parallel computation, user has to do several steps. First the user's program and input data have to be prepared. Input data must be divided for all jobs and program must receive data as an input. Communication has to be available through input, output and error terminals only, because jobs will run on the background. Then the Condor's universe has to be chosen. Each universe has certain functions and abilities. The most common universes are Standard universe, which allows checkpointing, migration and remote system call, and Vanilla universe, which doesn't provide these advantages but also isn't limited. For Java Virtual Machine there is a Java universe, for grid computing there is a Grid universe and so on. The third step is to write a submit description file. It establishes how will Condor manage jobs. There is important to tell Condor which program will be executed, in which universe, if there are any requirements to jobs or machines, where are input data and where will be stored output data and so on. Finally the computation is started by `condor_submit` command. Also there are many commands for managing the computation. The example of the submit description file is in the Algorithm 2.1 [47].

Algorithm 2.1 Example of the submit description file [47]

```
Executable = matlabparallel.m
Universe = vanilla
Requirements = Memory >= 32 && OpSys == \
“LINUX” && Arch == “INTEL”
Rank = Memory >= 64
Image_Size = 28 Meg

Error = err.$(Process)
Input = in.$(Process)
Output = out.$(Process)
Log = mathematica.log

Queue 150
```

2.3 Sun Grid Engine

Sun grid engine (SGE) is an open source batch-queuing system and is the open source version of the newer Oracle grid engine. SGE was developed in 2001 from the previous environment CODINE.

As you can see in the Figure 2.1, SGE cluster consists of QMaster which runs on the master host and manages a parallel computation, execution deamons, which run on execution hosts, shadow master, which take over the role of master host if crashes, submit hosts, which enable users to submit and control batch jobs. In the Figure 2.1, there is also ARCo box, which represents Accounting and Reporting Console, which allows SGE to store informations to SQL database and user to work with. The last component is DRMAA, which provides a programmatic interface for applications to submit, monitor, and control jobs.

User specifies the requirements and submits the job to the SGE. SGE puts the job to the queue and then send it to the execution device. When the job are finished, SGE show logs about it. It contains QMON, the grafical user's interface, which simplifies the work with SGE.

Like Condor's universes, SGE has policies that are customized according to whatever is appropriate for the site. Urgency policy defines priority for each job. This priority is value which is derived from job's resource requirements, the job's deadline specification, and how long the job waits before it is run. Using the Functional policy, an administrator can provide special treatment because of a user's or a job's affiliation with a certain user group, project, and so forth. In Share-based policy, the level of service depends on an assigned share entitlement, the corresponding shares of other users and user groups, the past usage of resources by all users, and the current presence of users within the system. The last policy in SGE is Override policy. It requires manual intervention by the cluster administrator, who modifies the implementation. Like the policies, there are also four users categories with different accesses to the SGE commands.

Job is submited by the `qsub` command from the terminal or QMON GUI and can be controled

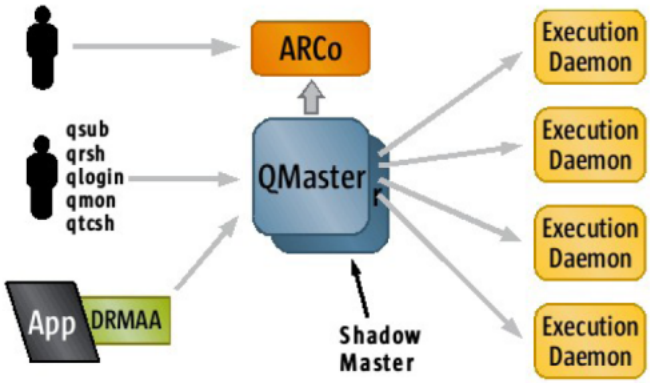


Figure 2.1: Oracle Grid Engine Component Architecture [53]

by `qstat` command. When the job is finished, a log can be sent through email.

All these informations were obtained from Sun N1 Grid Engine 6.1 User's guide [52] and from Beginner's Guide to Oracle Grid Engine 6.2 [53].

2.4 Matlab Parallel Computing Toolbox

Matlab Parallel Computing Toolbox (MPCT) is the official Matlab toolbox for parallelization of Matlab and Simulink programs. In its standard version, it provides twelve workers (MATLAB computational engines) locally on a multicore desktop or eight workes for computing on a remote cluster. There are three main possibilities how to use MPCT. It can be used for parallelization of for loop by `parfor` statement, to executing batch jobs in parallel and for partitioning of large data sets.

Parfor is exactly parallel for loop. It devides loop by its iterations. Each iteration is sent to certain worker in any order so there is a condition for iteration independence. Parfor loop requires either many iterations or long iterations. With a small amout of short iterations it could be counterproductive. As you can see in the Algorithm 2.2, a Matlab pool (cluster) has to be opened. Then `parfor` statement can be used and in the end Matlab pool has to be closed.

Batch jobs in parallel are useful for example when user wants to execute a time-consuming job, but he still wants to work in the same Matlab session. For this purpose, there is a `batch` command. It starts to run a task on the worker as a batch job. This job is asynchronous, so the client Matlab session is not blocked. Nevertheless, if the user wants to wait to the job, there is a `wait` command. To retrieve the results, classical `load` statement can be used. In the Algorithm 2.3, there is an example of the batch jobs in parallel. Naturally, the batch parallel loop is possible to use also.

If a user has an array that is too large for computer's memory, it cannot be easily handled in a single MATLAB session. By `distributed` command used inside `matlabpool` statement, a user converts classical array to distributed array. This distributed array is divided by columns

to smaller arrays and each small array is sent to different worker. User works with distributed array as a classical array and workers automatically transfer data between themselves when necessary. An example of using distributed array is in the Algorithm 2.4.

All these informations were obtained from Parallel Computing Toolbox User's Guide [50].

Algorithm 2.2 Example of using parfor loop [50]

```
matlabpool open local 3

parfor i=1:1024
  A(i) = sin(i*2*pi/1024);
end
plot(A);

matlabpool close
```

Algorithm 2.3 Example of using batch jobs in parallel [50]

```
edit mywave

for i=1:1024
  A(i) = sin(i*2*pi/1024);
end

job = batch('mywave');
wait(job);
load(job, 'A');
plot(A);
destroy(job);
```

Algorithm 2.4 Example of using distributed array [50]

```
Matlabpool open
  spmd
    MM=distributed(M);
    % there are operations with MM array
    al=getLocalPart(MM);
  end
Matlabpool close

al{1} % in the first cell of the cell array al, there is the part of the
resultant MM array
```

Chapter 3

Parkinson's disease

Parkinson's desease (PD) was described by James Parkinson in 1817. It is a progressive neurological disorder characterised by a large number of motor and non-motor features that can impact on function to a variable degree. PD is characterized by four major features include tremor, rigidity, akinesia (or bradykinesia) and postural instability [26].

Although the cause of PD is unknown, there are some factors that contribute to the pathogenesis of PD. This factors include for instance genetic factors, environmental toxins and excessive iron deposition [29]. The main feature occurs on the cellular scope is a loss of neurons in the caudal and anterolateral parts of the substantia nigra pars compacta, which causes deficiency of dopamine in the striatum, whose extent is directly correlated with the severity of PD [19].

3.1 Treatment

PD is incurable, but there are several possibilities how to lessen symptomps. In first line of treatment, Levodopa is in general better solution for patients aged more then 60 years [30]. For younger patients there is dopamine agonists. In later stage of the disease, combinations of these approaches are necessary. If medicaments doesn't work well or if side effects are unbearable, there is possibility in surgery, in deep brain stimulation. But on the other hand, no treatment is required if the quality of patient's life is satisfactory [29].

3.1.1 Levodopa treatment

Levodopa has been highly used in treatment of Parkinson's disease for 30 years. It apears better to start with lower dose of levodopa at the beginning of the treatment, because it may lessen the severity. But after ten years of treatment, there is no significant difference between treatment regimens. There are many problems with levodopa use as well. As disease progresses, levodopa causes dyskinesia and dose failure [28]. The most common side effects of levodopa treatment

are alternating between "on" and "off" phases [31], behavioral changes [32], nausea, postural hypotension, distonia, pain and sweating [29].

3.1.2 Dopamine agonists treatment

Against levodopa, this treatment is in general less well tolerated, less effective and more expensive. It causes nausea and dizziness as well, but until no levodopa is used, there is no dyskinesia and motor fluctuations. Other side effects of dopamine agonists treatment include hallucinations, confussion, pleuropulmonary and retroperitoneal fibrosis, postural hypotension, erythromelalgia. So the one of the main benefits in the comparison with levodopa is delay of the levodopa use itself [28].

3.1.3 Surgery

In general, surgical treatment is used to improve "off" phases and dyskinesia mentioned above, when medicamental treatment is no longer effective or side effects of treatment is worse then PD symptoms. But there are many contraindications force surgeons and patients to consider all possibilities. To these contraindications belong for instance pre-morbid cognitive or psychiatric disturbance, problems with puncturing dura mater (brain shifting, infection) and severe cerebral atrophy [29]. Also the surgery may not be beneficial to patients with atypital parkinsonian syndromes [33].

In according to [34], there are various possibilities how to do operation. The surgeon must make decision about ablative brain surgery (lesioning) or deep brain stimulation (DBS), about pallidal or subthalamic surgery and about unilateral or bilateral surgery. Among these possibilities, there are several differencies. For instance bilateral pallidotomy are more risky than unilateral, because there is higher chance to cause speech, swallowing or cognitive disturbancies, but the benefit can be higher as well [35]. The lesioning in comparison with DBS is more economical, there is negligible risk of infection and there is no need for adjustment stimulation parameters. Nevertheless, DBS is more effective and reversible, so it is more acceptable for younger patients [29].

3.2 Deep brain stimulation

The first deep brain stimulation (DBS) was realized in January 1987 by A. L. Benabid to improve essential tremor [38]. Later, the DBS was used for treatment of Parkinson's disease, dystonia, clinical depression and chronic pain [40]. In the treatment of PD, there are two possibilities of the surgical operation mentioned above, the DBS and the ablative brain surgery. In both cases, physician has to decide about the core that will be targeted. There are three

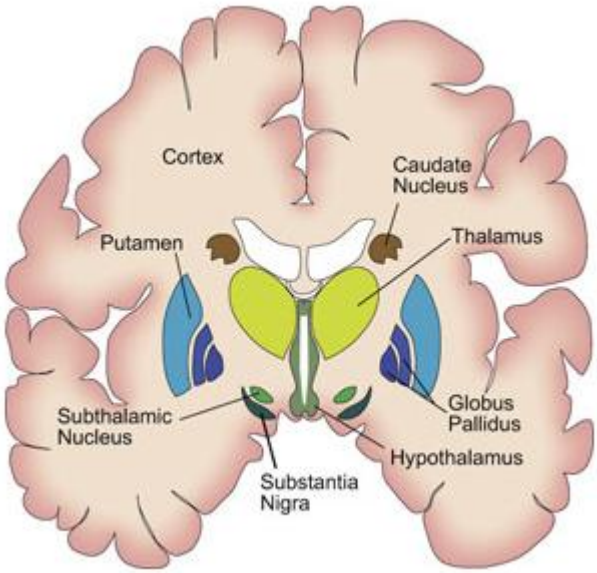


Figure 3.1: Coronal cut of the brain with visualization of the thalamus and the basal ganglia [54]

possibilities in DBS treatment, STN that is most common target, GPi or Th that is rather used for treatment of essential tremor. Just before operation, Magnetic Resonance Imaging (MRI) tooks the entire picture of the brain. Surgeon then pinpoints the trajectory of the microelectrode. This microelectrode, which is only tens of micrometer thin, is applied. Because of brain shifting, surgeon's MRI image is not so precise, therefore electrophysiological exploration must be carried out. Surgeon listens the neuronal signal that is measured through implanted electrode during microelectrode recording (MER) and by his experience he desides about reached core. This can be very complicated, so reliable automatic classification could be useful [43].

3.3 Cores structures

3.3.1 Basal ganglia

The basal ganglia (BG) are a set of subcortical nuclei located in the midbrain, around the thalamus. The major nuclei of basal ganglia are the striatum, which is composed of the caudate nucleus and the putamen; the internal and external parts of the globus pallidus; the pars reticulata and the pars compacta of the substantia nigra; and the subthalamic nucleus [22]. Although for a long time, BG were thought to support motor functions exclusively, it is now recognized that BG have a role in cognitive, affective and autonomous function also. BG circuitry, shown in the Figure 3.3, is involved in a great range of functions including: reward based learning, exploratory/navigational behavior, goal-oriented behavior, motor preparation, working memory, timing, action gating, action selection, fatigue and apathy. In spite of the

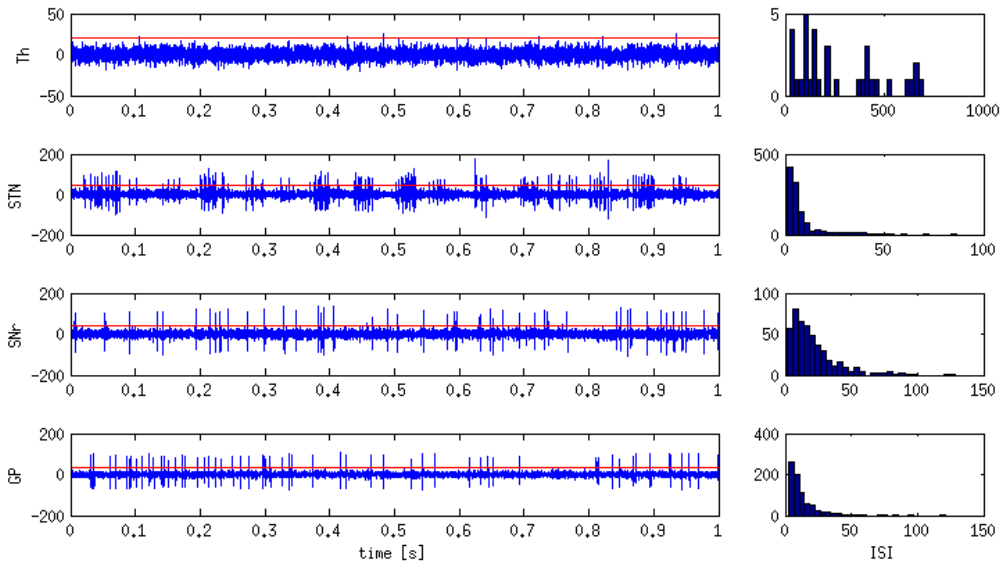


Figure 3.2: Typical signal shapes of the main measured cores with their ISI histograms

significant progress in our knowledge of BG at several levels, it is still not clear how such an overwhelming range of functions is supported by the same subcortical circuit [23].

3.3.1.1 Striatum

The Striatum is made from caudate nucleus and putamen and is the major input nuclei of BG. These two parts are divided by a white matter tract called internal capsule. Striatum receives inputs from a number of cortical areas and the thalamus [23]. In the case of PD, dopaminergic innervation from substantia nigra pars compacta is affected, which causes PD symptoms. Huntington's disease, chorea and dyskinesia causes striatum atrophy [46].

3.3.1.2 Subthalamic nucleus

Subthalamic nucleus (STN) is another input of BG, and like the striatum, also receives inputs from cortex, but the exact function of the STN is still unknown [23]. STN is one of the major targets of lesioning or DBS. Bilateral subthalamic surgery is used to improve appendicular and axial akinesia, rigidity [29] and severe Parkinsonian tremor [36].

3.3.1.3 Substantia nigra

The substantia nigra (SN) is portion of the basal ganglia and consists of the pars compacta (SNc) and the pars reticulata (SNr). The SNr (ventral portion of SN) contains small amounts of dopamine and iron, giving it reddish color, while the SNc (dorsal portion) contains large quantities of dopamine and melanin, making it black (whence the name, substantia nigra) [19].

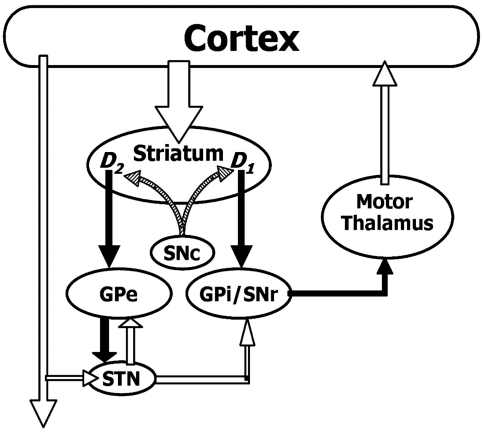


Figure 3.3: Basal ganglia circuitry [41]

With GPi, SNr constitutes the output nuclei of the BG [22]. During MER of Parkinson patients, the SNr is characterized by more regularly firing units [20].

3.3.1.4 Globus pallidus

The globus pallidus (GP) consists of internal (GPi) and external (GPe) part. With SNr, GPi constitutes the output nuclei of the BG [22]. As well as STN, GP is also one of the major tergets of lesioning and DBS. Unilateral pallidotomy is ofted used for improving contralateral akinetic/rigid symptoms of PD [29] and severe tremor [37].

3.3.2 Thalamus

The human thalamus (Th) is a nuclear complex located in the diencephalon and comprising of four parts (the hypothalamus, the epythalamus, the ventral thalamus, and the dorsal thalamus). The Th is a relay centre subserving both sensory and motor mechanisms [27]. During MER of Parkinson patients, the Th is characterized by relatively slowly firing and bursting activity patterns [20]. In former treatments of PD, Th was lesioned or stimulated to improve unilateral tremor. Bilateral surgery caused often severe speech disturbances. But it was found that pallidal and subthalamic surgery has better results, so there is no need to target Th for this reason [35].

3.4 DBS Toolbox

The DBS Toolbox provides processing of MER signals such as spike detection and sorting, automatic data access or feature extraction. All informations about this environment is in [43, 44, 45].

3.5 Data

Data were recorded using 1 – 5 tungsten microelectrodes with an exposed tip size of 15 – 25 μm. They were recorded for ten seconds at sampling rate 24 kHz and annotated by a neurologist using visual and audio inspection of the MER recordings [43]. This annotation cosists of ten groups of signals, exactly STN, STN_rb, STN_like, Th, SNr, SNr_like, GPi, GPe and Unknown. Unknown, STN_like and SNr_like signals were removed from analysis. Later GPe has been put together with GPi to form only GP annotation and STN_rb has been connected with STN. Data structure is provided in Table 3.1. In the latter data analysis, there was usefull to know number of signals for certain range of spike frequencies. See Table 3.2 shows amount of signals for these frequencies.

This set of data is extension of the date from previous paragraph. If the certain neuron is measured, signal from close neurons are added to the signal of measured neuron. There are methods [24, 25] how to extract these signals, so we can work with only one neuron if needed. The number of signals increased after neuron extraction, so I had to rewrite the Table 3.2 to the Table 3.3.

Table 3.1: Number of signals from new data annotation

Th	STN	STN_like	STN_rb	SNr	SNr_like	GPi	GPe	Unknown	Total
530	2803	79	298	349	24	155	222	7710	12170

Table 3.2: Number of signals for certain spike frequency threshold

>spikes	0	100	200	300	400	500	600	700	800	900	1000
Th	530	219	111	62	43	34	27	23	20	16	13
STN	3101	1937	1462	1083	754	522	348	217	127	81	55
SNr	349	214	169	141	117	86	64	48	37	27	19
GP	377	169	130	108	82	68	56	46	38	32	24
Total	4357	2539	1872	1394	996	710	495	334	222	156	111

Table 3.3: Number of signals with separated neurons for certain spike frequency threshold

>spikes	0	100	200	300	400	500	600	700	800	900	1000
Th	997	270	103	39	21	13	9	8	7	4	3
STN	6946	2766	1094	503	229	109	58	39	21	16	11
SNr	798	343	157	84	48	34	21	12	8	6	5
GP	787	300	116	57	37	24	15	10	10	6	4
Total	9528	3679	1470	683	335	180	103	69	46	32	23

Chapter 4

Complexity measures

As the title indicates, the complexity measures are used to estimate a complexity of a signal. These measures are based on information or chaos theory and reveal hidden properities of a signal. These properities are used to classify a groups of signals, such as RR-interbeats [4, 10], neuronal spike trains [14] or DNA sequences [15]. In this work I used interspike intervals (ISIs), so before a complexity measure is applied, a raw signal must be transformed. At first, raw signal is measured. Then threshold

$$\text{Threshold} = 4 \cdot \text{median}\left(\frac{|x|}{0.6745}\right) \tag{4.1}$$

is estimated to detect spikes [42]. Then times between spikes are computed to create an ISI. Finally, the complexity measures are estimated. Functions provided the transformation are built in DBS Toolbox. See Figure 4.1 for shallow notion.

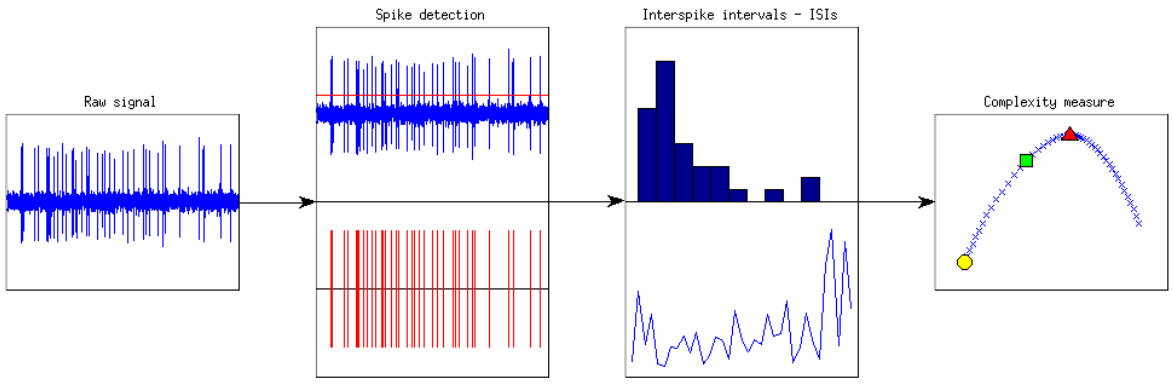


Figure 4.1: Diagram of signal transformation

4.1 Lempel-Ziv complexity

The Lempel-Ziv complexity (LZC), a new approach of evaluating the complexity of finite sequences, was proposed by Abraham Lempel and Jacob Ziv in 1976 [1]. LZC has been primarily used to solve information theoretic problems and applications, but in recent years, it has been applied extensively in biomedical signal analysis [13].

The signal must be converted into a finite symbol sequence first. In the context of biomedical signal analysis, typically the discrete-time biomedical signal is converted into a binary sequence [13]. A neurological signal with T duration is divided into N bins with a size of $\Delta\tau$ ($N = T/\Delta\tau$). Each bin is annotaded as "1" if it contains at least one spike belonging to major aggregated cluster; "0", otherwise [14]. Principle of the signal transformation to a binary string is shown in the Figure 4.2.

Independent explanation of Lempel-Ziv algorithm in accordance with [13] is in the following. Consider binary strings $P = s(1)\, s(2) ... s(n)$, $S = s(1)$, $Q = s(2)$, integers $r = 1$, $c(n) = 1$ (n is equal to the number of bins) and operation π, which removes last character of the string, so using further string $SQ\pi$ consists of $s(1)$ only. It was inicialization. Then algorithm is provided as following. Is string Q substring of $SQ\pi$? If so, make new string Q by $Qs(r+2)$, increase r by 1 and go to the begining. If not increase r and $c(n)$ by 1, make new string S by $s(1) ... s(r)$ and Q by $s(r+1)$ and go to the begining. If r is equal to n, algorithm will finish. Finally the Lempel-Ziv complexity is counted

$$C(n) = \frac{c(n) \log_2(n)}{n}. \tag{4.2}$$

Example: Consider string $P(57) = s(1) ... s(57) = 001001001010100100010111010110010101100010001001011011001010100$. Make string $S = s(1) = 0$, where s means substring and number in brackets is position in the string P, and string $Q = s(2) = 0$. Then make string $SQ\pi = 0$. Look if Q is substring of the $SQ\pi$. It is, so make new string by adding next character, so $Q = 01$. Now Q isn't substring of $SQ\pi = 00$, so $c(57) = 2$. Make new string $S = 001$ and $Q = 0$ and do the same for all other characters. For humans the method is very easy, so we can split the string to unique substrings as $S(57) = 0|01|00100101|01001000|1011|0101100|101011000|10001011011|$ $|0010100$. We have 9 vocabulary words and 57 characters, so the Lempel-Ziv complexity is given by

$$C(57) = \frac{9 \log_2(57)}{57} \doteq 0.9210.$$

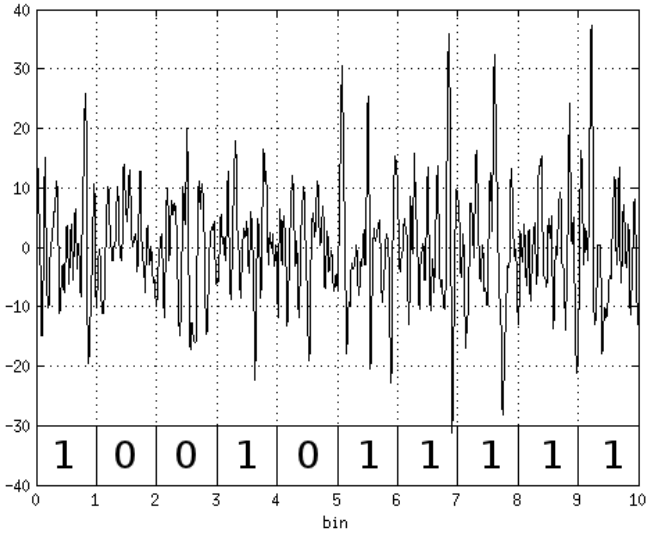


Figure 4.2: Principle of the signal transformation to a binary string

4.2 Burst index, pause index and pause ratio

The burst index (BI), pause index (PI) and pause ratio (PR) have been proposed by Jacques Favre et al. in 1999 and can by used to quantify burst activities of deep brain nuclei. The BI was defined as the number of ISI < 10 ms divided by the number of ISI > 10 ms, the PI was defined as the number of ISI > 50 ms divided by the number of ISI < 50 ms and the PR was defined as the total duration of ISI > 50 ms divided by the total duration of ISI < 50 ms [21].

4.3 Detrended fluctuation analysis

The detrended fluctuation analysis (DFA) has been proposed by C. K. Peng et al. in 1994 to estimate long-range power-law correlations for DNA sequences [15], heartbeat time series [16] or stride intervals of human gait [17].

First, the original time series is integrated, and then divided into boxes of equal lenght, n. For each box of length n, a least squares line (representing the trend in that box) is fit to the data. For a given box size n, the characteristic size of the fluctuations, denoted by $F(n)$, is then calculated as the rms deviation between $y(k)$ and its trend in each box $y_n(k)$ as shows Eqn 4.3. This computation is repeated over all time scales (box sizes). Typically, $F(n)$ will increase with box size n. A linear relationship on a log-log graph indicates presence of scaling (self-similarity), such that fluctuations in larger boxes in a power-law fashion. The slope of the line relating $\log F(n)$ to $\log n$ determines the fractal scaling exponent, α [18].

$$F(n) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y(k) - y_n(k)\right)^2} \tag{4.3}$$

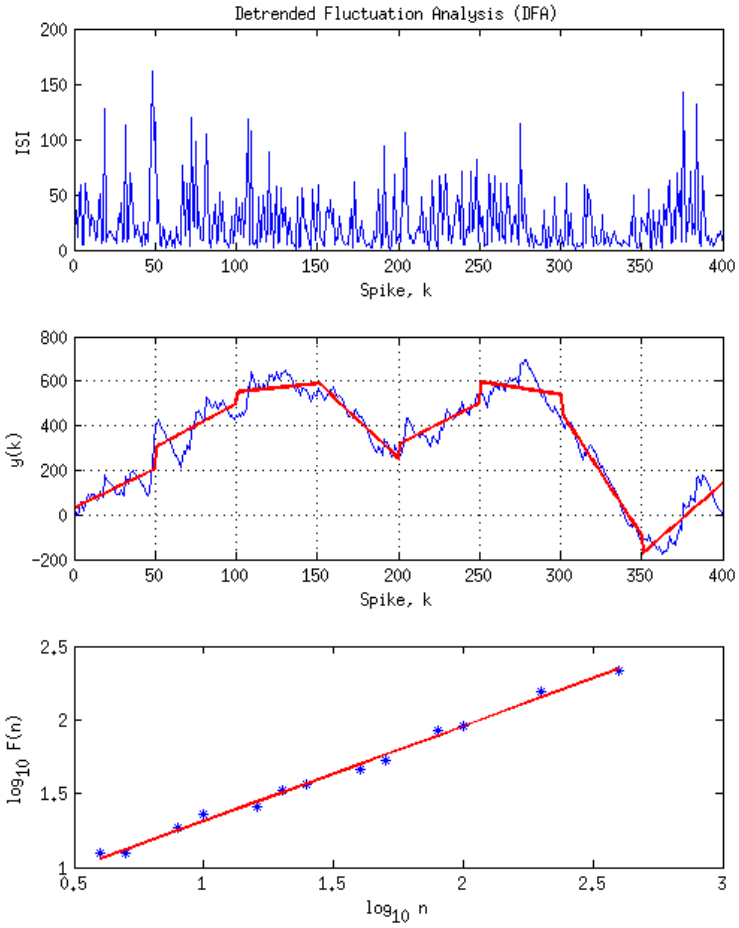


Figure 4.3: Detrended Fluctuation Analysis. First graph shows input signal (ISIs of one neuron). As you can see on the second graph, input signal is integrated (blue) and divided to the boxes (50 samples for this case). For each box the trend line is computed (red). Then the fluctuation $F(n)$ is computed and ploted as one of the blue points on third graph. This method is repeated for all chosen boxes. Finally, the trend line of all fluctuations is computed. The slope of the line is the measure required.

4.4 Approximate entropy

Approximate entropy (ApEn) was proposed by Steven M. Pincus in 1990. It was suggested to classify complex systems with a relatively small amount of data, such as heart rate [5]. Later, ApEn has been used in finance [6], psychology [7], and human factors engineering [8].

In accordance with Pincus in [5], the algorithm is in the following. Consider a positive integer m and positive real number r. Note time-series of data as $u(1), u(2), \ldots, u(N)$, where N is the signal length, then form vectors $x(1), x(2), \ldots, x(N-m+1)$ in $\mathbb{R}^m$ defined by $x(i) = [u(i), u(i+1), \ldots, u(i+m-1)]$. Next define for each i, $1 \leq i \leq N-m+1$,

$$C_i^m(r) = \frac{\text{number of } j \text{ such that } d[x(i), x(j)] \leq r}{N-m+1}, \tag{4.4}$$

where $d[x(i), x(j)] = \max_{k=1,2,\ldots,m} (|u(i+k-1) - u(j+k-1)|)$. Then define

$$\Phi^m(r) = \frac{1}{N-m+1} \sum_{i=1}^{N-m+1} \ln C_i^m(r). \tag{4.5}$$

Finally, the ApEn is counted as

$$\text{ApEn}(m, r, N) = \Phi^m(r) - \Phi^{m+1}(r). \tag{4.6}$$

Example: Consider signal $u = [2, 3, 5, 1, 0]$, $m = 2$ and $r = 3$. Then form vectors

$$\begin{aligned} x(1) &= [2, 3], \\ x(2) &= [3, 5], \\ x(3) &= [5, 1], \\ x(4) &= [1, 0]. \end{aligned}$$

In the first step $i = 1$. Take $j = 1$ and quantify d function with $x(1)$ and $x(1)$ as follows:

$$d[x(1), x(1)] = \max(|u(1) - u(1)|, |u(2) - u(2)|) = \max(0, 0) = 0.$$

Is $d[x(1), x(1)] \leq r$, is $0 \leq 3$? It is, so increase a numerator of $C_1^2(3)$ by 1. Do this procesude for each i and each j. Then count

$$\begin{aligned} C_{i=1}^2(3) &= \frac{3}{4}, \\ C_{i=2}^2(3) &= \frac{3}{4}, \\ C_{i=3}^2(3) &= \frac{4}{4}, \\ C_{i=4}^2(3) &= \frac{4}{4}. \end{aligned}$$

Subsequently count

$$\Phi^2(3) = \frac{1}{4}\sum_{i=1}^{4} \ln C_i^2(3) \doteq -0.1438.$$

Do the whole procedure again for $m = 3$. You should get

$$\begin{aligned} x(1) &= [2,\, 3,\, 5], \\ x(2) &= [3,\, 5,\, 1], \\ x(3) &= [5,\, 1,\, 0], \end{aligned}$$

$$C_1^3(3) = C_2^3(3) = C_3^3(3) = \frac{1}{3},$$

and

$$\Phi^3(3) = \frac{1}{3}\sum_{i=1}^{3} \ln C_i^3(3) \doteq -1.0986.$$

Finally ApEn is estimated as

$$\mathrm{ApEn}(2,\, 3,\, 5) = \Phi^2(3) - \Phi^3(3) \doteq -0.9548.$$

4.5 Sample entropy

A sample entropy (SampEn) was proposed by Douglas E. Lake in 2000 as a measure of neonatal heart rate variability. The SampEn was designed to reduce a bias of ApEn and has closer agreement with theory for datasets with known probabilistic content [9]. The SampEn is largely independent of a record length and displays relative consistency under circumstances where ApEn does not [11].

The SampEn is very similar to the ApEn. It differs in two cases only. The first case is that SampEn does not count self-matches. In acordance to the notation from Section 4.4, it means that match is not taken into account if $j \neq i$. The second case is that SampEn does not use a template-wise approach [14].

A good explanation of the SampEn is for instance in [14], so I will follow very freely the explanation provided in [14]. Consider a positive integer m and positive real number r. Note time-series of data as $u(1), u(2), \ldots, u(N)$, where N is the signal length, then form vectors $x_m(1), x_m(2), \ldots, x_m(N-m+1)$ in $\mathbb{R}^m$ defined by $x_m(i) = [u(i), u(i+1), \ldots, u(i+m-1)]$. Next define for each i, $1 \leq i \leq N-m+1$,

$$B_i^m(r) = \frac{\text{number of } j \text{ such that } d[x_m(i),\, x_m(j)] \leq r}{N-m-1}, \tag{4.7}$$

where $d[x_m(i),\, x_m(j)] = \max_{k=1,2,\ldots,m} (|u(i+k-1) - u(j+k-1)|)$ and j ranges from 1 to $N-m$,

and $j \neq i$. Then define

$$B^m(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} B_i^m(r). \tag{4.8}$$

Similarly define

$$A_i^m(r) = \frac{\text{number of } j \text{ such that } d[x_{m+1}(i),\, x_{m+1}(j)] \leq r}{N-m-1}, \tag{4.9}$$

and

$$A^m(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} A_i^m(r). \tag{4.10}$$

Set

$$B = \frac{(N-m-1)(N-m)}{2} B^m(r) \tag{4.11}$$

and

$$A = \frac{(N-m-1)(N-m)}{2} A^m(r). \tag{4.12}$$

B is the total number of template matches of length m and A is the total number of forward matches of length $m+1$, then the SampEn is defined as

$$\text{SampEn}(m,\, r,\, N) = -\ln\left(\frac{A}{B}\right). \tag{4.13}$$

4.6 Multifractal analysis

A theory of fractals and multifractals describes self-similar and complex scaling properties observed in various physical systems. Especially in biomedicine, most of signals behave like fractals. Fractals are geometric objects that exhibit some degree of similarity in a wide range of scales. In the 1991, Muzy, Bacry, and Arneodo [3] proposed a wavelet transform modulus maxima (WTMM) method, which is based on wavelet analysis, which is also called a mathematical microscope due to its ability to preserve good resolution on multiple scale [2].

Because of high complexity of this method, only a brief description follows. This description is an encapsulated version of Pavlov's explanation in [2]. There is fractal-like signal denoted as $g(x)$. The wavelet transform is applied to the signal as

$$W(a,\, x_0) = \frac{1}{a} \int_{-\infty}^{\infty} \psi\left(\frac{x - x_0}{a}\right) g(x)\, \mathrm{d}x, \tag{4.14}$$

where ψ is a wavelet function, a is a scaling exponent and b is a space or time coordinate. All wavelet coeficients $W(a,\, x_0)$ form a surface in a three dimensional space. This surface contains

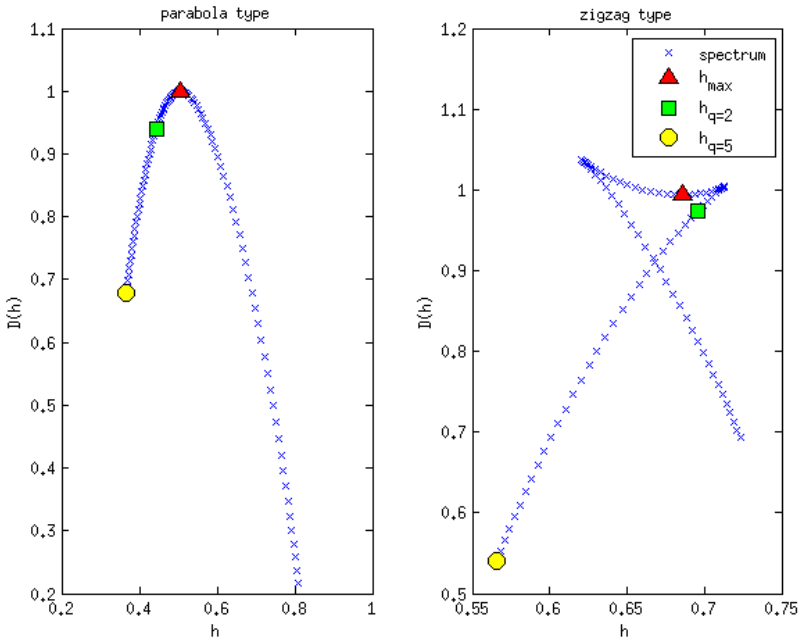


Figure 4.4: Typical multifractal spectrums of a neuronal signal, regular spectrum, so called parabola type (left), and irregular spectrum, so called zigzag type (right)

so-called local maxima lines. These lines are removed to create partition function defined as

$$Z(q,\, a) = \sum_{l \in L(a)} |W(a,\, x_l(a))|^q , \tag{4.15}$$

where $L(a)$ is the set of all lines of local maxima that exist on scale a, and $x_l(a)$ characterizes the position of the maximum belonging to the line l. Then we can say that

$$Z(q,\, a) \sim a^{\tau(q)},\ a \to 0^+, \tag{4.16}$$

where $\tau(q)$, the scaling exponent, is determined for some q and it pose the slope of $\ln Z(q,\, \ln a)$. Now we can count the Hölder exponent by the Legendre transform

$$h = \frac{\mathrm{d}\tau}{\mathrm{d}q}. \tag{4.17}$$

Finally the singularity spectrum is determined by

$$D(h) = \min_q (qh - \tau(q)) . \tag{4.18}$$

There is a question how to read this singularity spectrum. There are various methods. One method, which is used in this work, has been provided by Makowiec et al. in [4]. Makowiec

used four measures:

1. maximum of the spectrum h_{max} defined as $h\,(q = 0)$,
2. Hurst exponent which is a measure of the long term memory of time series,
3. rare events h_l where $h\,(q \to \infty)$ and h_r where $h\,(q \to -\infty)$,
4. a spectrum width $\Delta = h_r - h_l$.

Makowiec redefined these measures for practical usage, so only the h_l rare event has been used and defined as $h\,(q = 5)$ and the spectrum width defined as $\Delta = |h\,(q = 0) - h\,(q = 2)|$. Makowiec used these measures to RR-series and integrated RR-series so he could estimate $\Delta_{max} = h^{int}_{max} - h_{max}$.

An example of the singularity spectrums is in the Figure 4.4.

Chapter 5

Results

5.1 Optimalization of time-consuming complexity measures

For an optimalization of the complexity measures, I naturally choose the Matlab Parallel Computing Toolbox. There were many reasons for this choice. The most significant were easy implementation and the fact, that most of used programs were written for Matlab. I used Matlab parallel approach to find out features of parallelization in three different cases.

In the first case, there was a code from the DBS Toolbox with `parfor` cycle that performed 107 very long and memory intensive iterations. The time of serial computation was 5087 s on average. In the Figure B.1 in Apendix B, there is a dependence of a speedup coefficient to each worker. The speedup coefficient is a ratio where a consumed time of each worker divides a consumed time of a serial calculation. The best result is teoretically equal to the number of workers. In accordance with previous, the result which I proposed in this case is not sufficient and is discussed in Chapter 6. Repeated calculations gave the same results. In the Figure 5.1, there is a load of the server which I used to compute on. An explanation of the picture parameters are in the following:

- %us is the CPU load with the user's processes, which have default priority,
- %sy is the CPU load with a core and its processes,
- %ni is the CPU load with the user's processes, of which priority has been changed by `nice` command,
- %id is the CPU idleness.

In the second case, there was the code that I used to count the multifractal spectrum where input signals was white noise $N = 10000$ samples long in the first case and $N = 50000$ samples long in the second case. The `parfor` cycle, which has been used for paralelization, was consisted

of 100 iteration. The speedup coefficient, which provide good result in both cases, and the server load are shown in the Figures B.3 and B.4 in Apendix B.

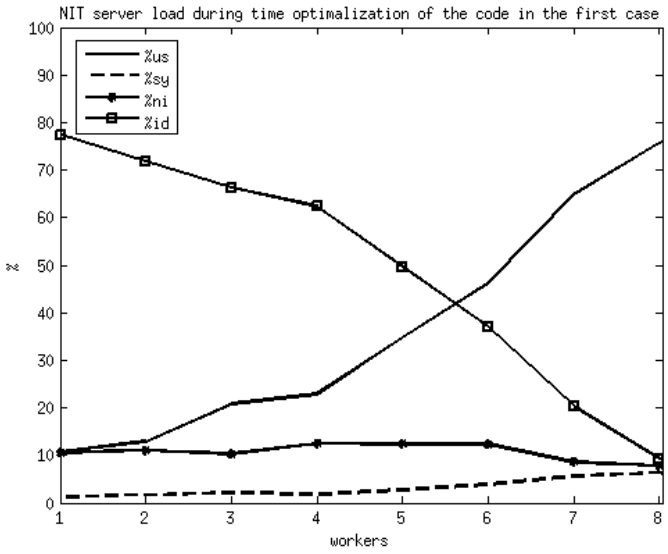


Figure 5.1: NIT server load during usage of the parallelization of the code from DBS Toolbox

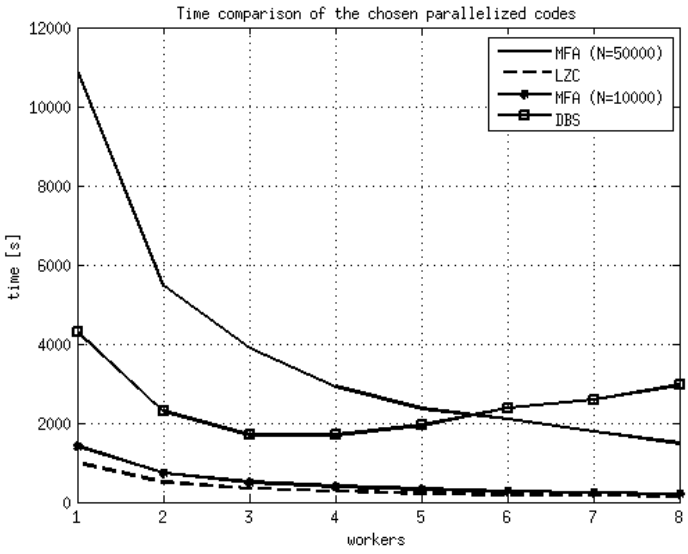


Figure 5.2: Time comparison of the chosen parallelized codes

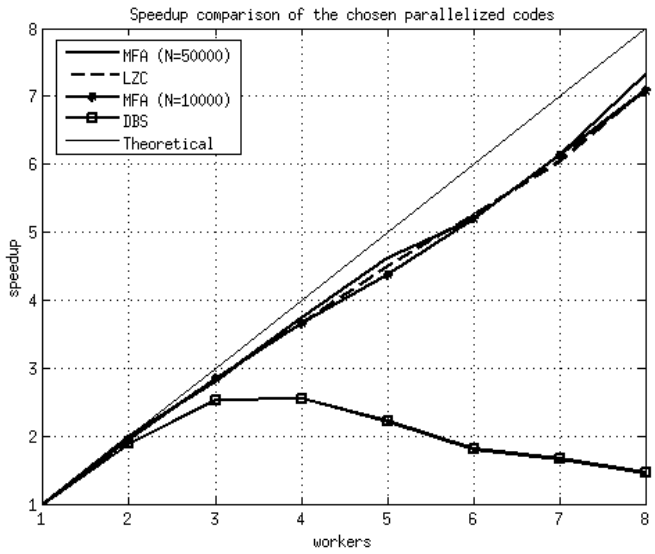


Figure 5.3: Speedup comparison of the chosen parallelized codes

A very similar result was obtained in the case of parallelization of the Lempel-Ziv complexity calculation. The `parfor` cycle consisted of 1000 short iterations. The speedup coefficient and the server load are shown in the Figures B.5 and B.6 in Apendix B.

All these calculations are compared in following figures. In the Figure 5.2, there is a comparison of the times required to compute proposed problems for each worker, where MFA means multifractal analysis, LZC means Lempel-Ziv algorithm and DBS means the code gained from DBS Toolbox. In the Figure 5.3, there is an comparison of the speedup coefficients for all these computations and theoretical speedup, which is the absolutely best result that can be theoretically reached.

5.2 Complexity measures

Only most promising results are presented in this chapter. All methods were applied to the same data. There were two main groups of the data mentioned in Section 3.5, ISIs without neuron sorting and ISIs where the neuron sorting method was applied. Results were subsequently divided in accordance to their annotation. For purposes of the statistical testing, four groups have been created. In the first group, there are all six cores tested separately. In the second group, the STN_rb core was unified with STN core and GPi core was unified with GPe core to create only GP core. The third group separates STN core from the others and the fourth group contains GPe and GPi cores only. All the results are summarized in the Appendix B.

For statistical testing, the Kolmogorov-Smirnov test has been used to decide about distribution normality, and then the Kruskal-Wallis or the analysis of variance test has been used to find the best input parameters that differentiate cores.

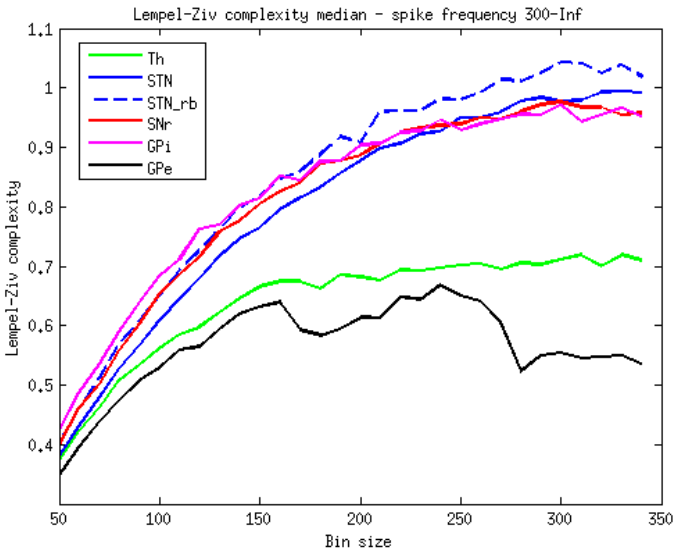


Figure 5.4: LZC medians through all bin sizes for signals with neuron sorting

The three most promising results are compared by mean with standard deviation and by median and median absolute deviation in the form of confidence interval. Median absolute deviation is defined as

$$\sigma_\mu = \text{median}\left(\left|\alpha\left(n\right) - \text{median}\left(\alpha\left(n\right)\right)\right|\right), \tag{5.1}$$

where α is a measure and n is a number of signals for each group of annotation. It forms a confidence interval as

$$\left[\mu - \sigma_\mu,\, \mu + \sigma_\mu\right]. \tag{5.2}$$

5.2.1 Lempel-Ziv complexity

Lempel-Ziv complexity (LZC) was counted for bin length from 50 to 340 samples with step by 10. Shorter bin length caused identical mean through all cores and longer bin length increased variance. This method is not so time-consuming. Average time to count LZC for one signal was $0.6366\,s$ for bin length 50 samples and $0.0641\,s$ for bin length 340 samples. See dependence of average consumed time to bin length in the Figure 5.6. The consumed time is higher for shorter bins because LZ algorithm has to work with longer strings.

In the Figure 5.4 you can see how the median is changing with the bin length for all cores separately. One can see the GPe and the Th are separated from other cores. This is a promising result, so the statistical testing has been applied to find the best input parameters (bin length and spike frequency range) that could separate cores as much as possible. The box plot for the best result for six cores annotation is in the Figure 5.5.

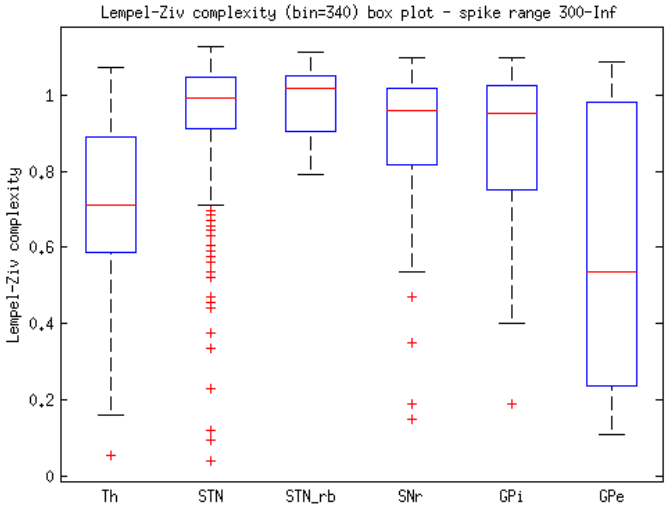


Figure 5.5: LZC box plot for signals with neuron sorting for bin lenght 340

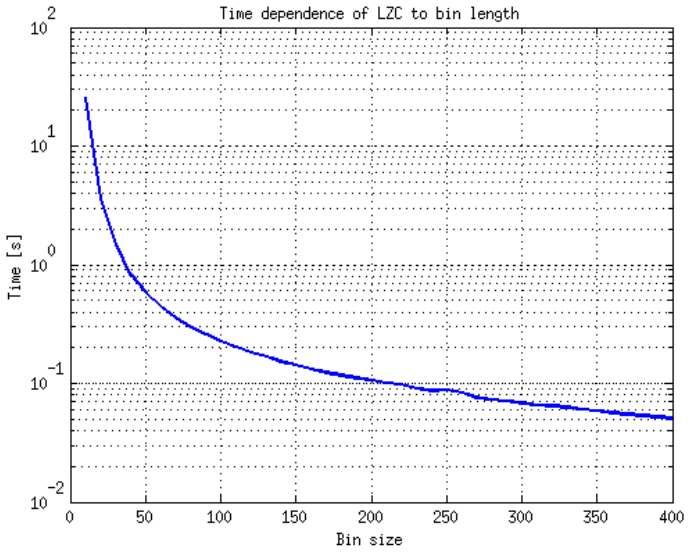


Figure 5.6: Lempel-Ziv algorithm dependence of averege consumed time to bin length

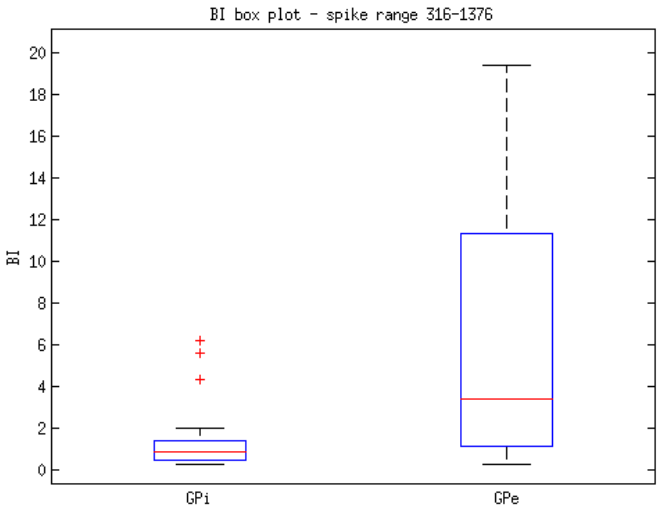


Figure 5.7: The best differentiation of GP cores by burst index

5.2.2 Burst index, pause index and pause ratio

As mentioned in Section 4.2, Favre used this measures to automatic classification GPi signals from GPe [21]. Despite Favre's conclusion, in our case the best of these measures was burst index. See Figure 5.7 where can be seen the box graph of the best configuration. The input data in this case were ISIs without neuron sorting. These measures weren't successfully used to classify other data sets. This method is the easiest to compute, therefore it is fatest too. The consumed time to compute all three measures for an average signal ($N = 317$) was only $6.04 \cdot 10^{-5}\, s$.

5.2.3 Detrended fluctuation analysis

Because of requirement of the minimal length of the input data, only ISIs with more then 100 spikes have been used. Unfortunately, the results of the coefficient α were very bad. In the figure 5.8, there is the result where p-value of the Kruskal-Wallis test was the smallest. To verify the DFA software, the α were counted for the white noise and the Brownian noise which is integrated white noise. The α coefficient were 0.5 for white noise and 1.5 for Brownian noise which corresponds to the Golberger's explanation in [18]. Time to count α coefficient of an average ISI with more then 100 spikes ($N = 840$) was $0.1439\, s$. The conclusion is that the DFA is not useful to differenciate short ISIs.

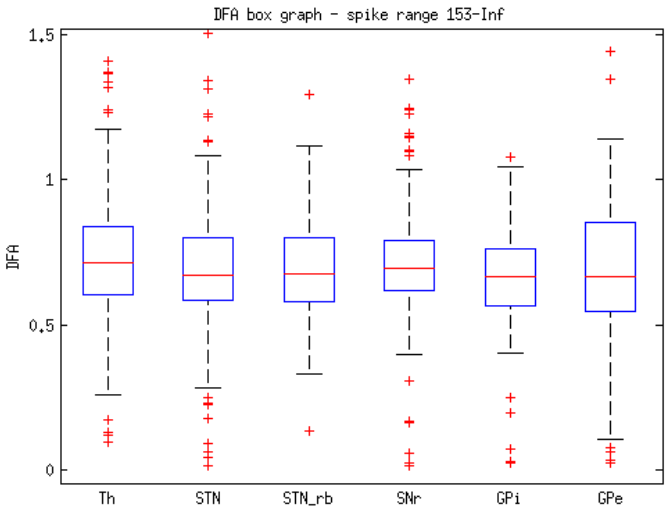


Figure 5.8: The best result for α coefficient of the DFA

5.2.4 Approximate entropy

The best results of the ApEn for 6 core, 4 core and STN/others annotation has been moving around spike range 20–Inf where the variance of the data is high. However for GP annotation, the most significant result was for the spike range 304–Inf where the variance is relatively small and amount of ISIs are still high enough. The quartiles of the box graph in the Figure 5.9 do not overlap which is very interesting result.

5.2.5 Sample entropy

SampEn has been giving the best results among all used methods. I decided to establish the best result as the result with the lowest p-value of the statistical testing mentioned above. Firstly, there was a problem how to select the m and r coefficients. For instance in [9], there is a method how to choose m and r parameters. I counted the SampEn for wide range of r and m, and took the most significant value that medians of data sets are different. The best result is in the Figure 5.10. In comparison to the other methods, SampEn is relatively fast to compute. For average ISI ($N = 317$) it took $0.009\,s$ only. The advantage is that the computation time does not rely on the input parameters, because the complexity of the method is the same.

5.2.6 Multifractal analysis

The singularity spectrum for each signal has been counted through software implementation from Physionet [55]. The input parameters were set to MaxScale = 20, MinScale = 2, $q_{min} = -5$, $q_{max} = 5$, dq = 0.1, ord = 3, a = 0, b = 2.53, where scale is the a parameter of the wavelet trans-

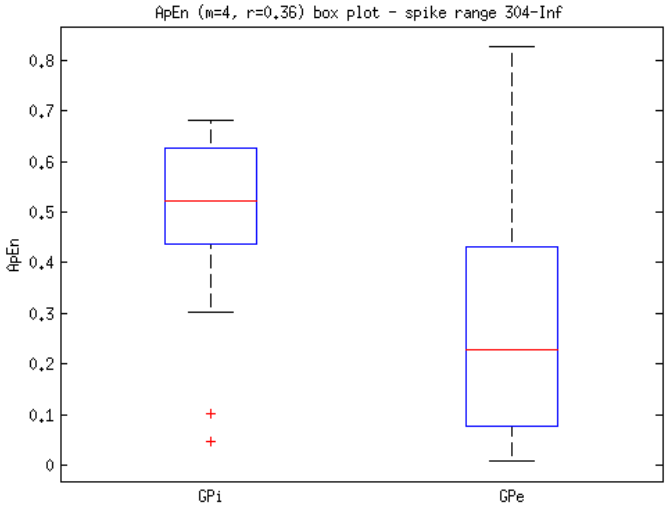


Figure 5.9: The best result of ApEn

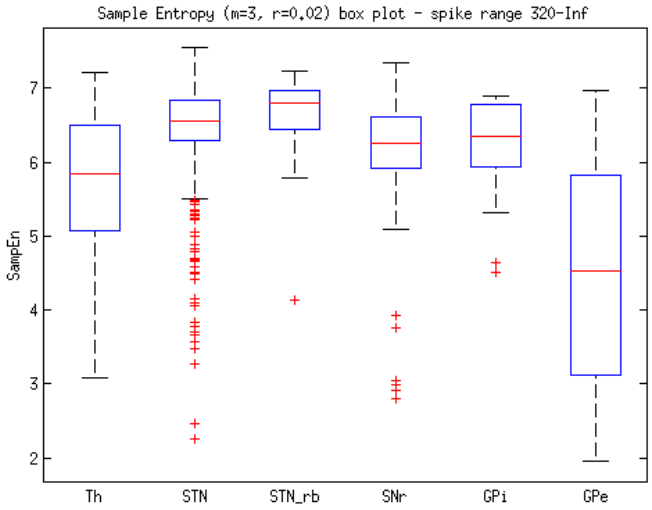


Figure 5.10: The best result of SampEn

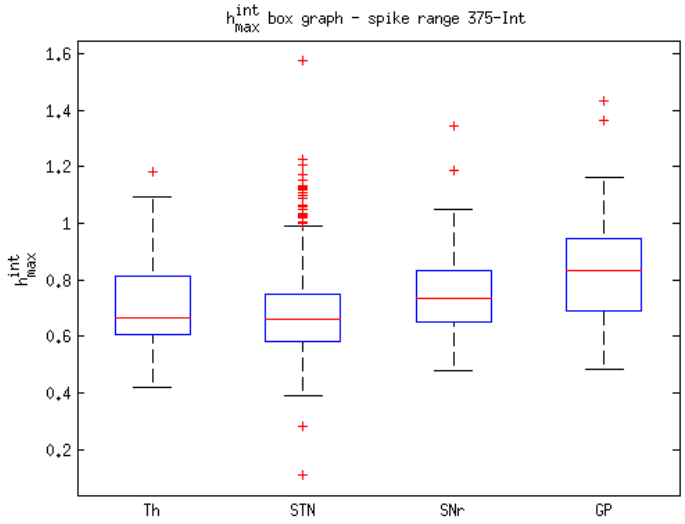


Figure 5.11: The most significant result for h_{max}^{int} and ISIs without neuron sorting

Table 5.1: Comparison of multifractal measures for ISI without neuron sorting (mean±std)

Core	Th	STN	SNr	GP
h_{max}	0.78 ± 0.21	0.86 ± 0.15	0.88 ± 0.13	0.89 ± 0.17
h_{max}^{int}	0.73 ± 0.16	0.68 ± 0.14	0.75 ± 0.14	0.83 ± 0.17
Δ_{max}	-0.06 ± 0.25	-0.19 ± 0.20	-0.13 ± 0.20	-0.07 ± 0.20
h_l	0.44 ± 0.26	0.50 ± 0.20	0.56 ± 0.22	0.45 ± 0.32
h_l^{int}	0.33 ± 0.22	0.27 ± 0.22	0.36 ± 0.26	0.41 ± 0.40
Δ	0.22 ± 0.18	0.21 ± 0.15	0.18 ± 0.18	0.36 ± 0.28
Δ^{int}	0.24 ± 0.23	0.24 ± 0.17	0.23 ± 0.20	0.34 ± 0.27

form, q is mentioned in section 4.6, dq is the step of q, ord is an derivative order of gaussian wavelet function, and a and b are the borders for obtaining spectrum through partition function defined at 4.15. Then the measures that used Makowiec in [4] has been obtained. The most significant result were obtained for h_{max}^{int} and ISIs without neuron sorting method applied. The box graph is displayed in the Figure 5.11. Tables 5.1 and 5.2 show the mean and standard deviation of obtained parameters through cores. Applications of this method to ISIs weren't convenient because software implementation required at least 300 samples to count the spectrum, so the number of data was quite low. Because of this, it was appropriate to use four cores annotation only. This method is very time-consuming for raw signals, but for short ISIs is fast enough and it took only $0.182\,s$ an average. In the Figure 5.12, there is a time dependence of the algorithm to the signal length. For this purposes, white noise signal were used.

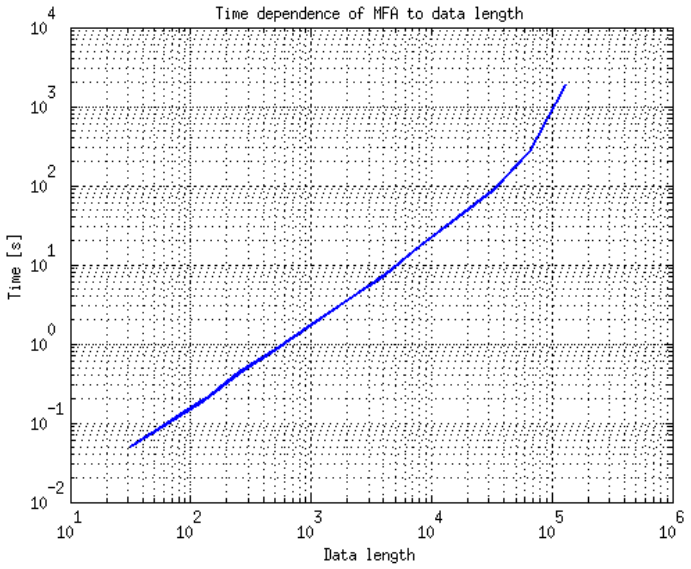


Figure 5.12: MFA algorithm dependence of average consumed time to data length

Table 5.2: Comparison of multifractal measures for ISI with neuron sorting (mean±std)

Core	Th	STN	SNr	GP
h_{max}	0.81 ± 0.19	0.83 ± 0.16	0.87 ± 0.13	0.87 ± 0.15
h_{max}^{int}	0.65 ± 0.13	0.65 ± 0.14	0.72 ± 0.14	0.75 ± 0.15
Δ_{max}	-0.21 ± 0.14	-0.17 ± 0.21	-0.15 ± 0.19	-0.07 ± 0.22
h_l	0.48 ± 0.22	0.47 ± 0.19	0.54 ± 0.21	0.39 ± 0.23
h_l^{int}	0.35 ± 0.26	0.22 ± 0.21	0.28 ± 0.24	0.30 ± 0.26
Δ	0.21 ± 0.19	0.21 ± 0.14	0.20 ± 0.20	0.30 ± 0.21
Δ^{int}	0.19 ± 0.16	0.26 ± 0.17	0.26 ± 0.21	0.32 ± 0.25

5.3 Classification

Only the three most promising results are compared in this section. In the Table 5.3, $\text{SampEn}_{3,0.02}$, LZC_{270} and h_{max}^{int} are compared by their means with variation and medians (confidence intervals). The ** symbol means the medians through all cores were significantly ($p<0.001$) different. For purpose of classification, 1-NN classifier has been applied to classify STN core from the others. Data were divided to training and testing sets in the ratio 4 to 1. Results of the classification with sensitivity and specificity could be seen in the Table 5.4.

Table 5.3: Comparison of the best three methods. The first value is mean ± standard deviation and the second value is range of median absolute deviation described in Chapter 4.

Core	Th	STN	SNr	GP
$\text{SampEn}^{**}_{3,0.02}$	5.68 ± 1.09 [4.46, 7.21]	6.50 ± 0.57 [6.03, 7.11]	6.16 ± 0.78 [5.56, 6.95]	5.04 ± 1.51 [3.05, 7.92]
$\text{LZC}^{**}{}_{270}$	0.67 ± 0.31 [0.33, 1.14]	0.93 ± 0.15 [0.87, 1.09]	0.83 ± 0.24 [0.67, 1.16]	0.68 ± 0.29 [0.25, 1.18]
h_{max}^{int**}	0.73 ± 0.16 [0.49, 0.92]	0.68 ± 0.14 [0.50, 0.82]	0.75 ± 0.14 [0.57, 0.90]	0.83 ± 0.17 [0.58, 1.10]

Table 5.4: 1-NN classification

Measure	TP	FP	TN	FN	Sensitivity	Specificity
$\text{SampEn}_{3,0.02}$	50	14	53	212	19%	79%
LZC_{270}	220	48	28	51	81%	37%
h_{max}^{int}	136	49	16	68	67%	25%

Chapter 6

Conclusion

The work has been divided into two parts. In the first part I described properities of three parallel computing frameworks – Condor, Sun Grid Engine and Matlab Parallel Computing Toolbox. Then I chose the Matlab parallel approach to speed up a computation of chosen complexity measure algorithms. In the case of the complexity measures, the speedup coefficient followed approximately the function

$$speedup = \frac{6}{7}w + \frac{1}{7}, \tag{6.1}$$

where w is the number of workers used in the parallel approach. But in the first case, where no complexity algorithm was used, the speedup coefficient behaves very strange. I suppose the problem is in the working memory demands, but because the parallelization of the complexity measures had been working well, I decided not to looking for a solution.

In the second part of the work I focused on the complexity measures. I presented brief description of all chosen algorithms and where it was possible I added an example of the method applied to very easy input signal. These methods have been used to count complexity measures and the best result for each measure has been presented. The main goal was to find a method used to separate STN ISIs from the others. 1-NN classifier has been used for sample entropy, Lempel-Ziv complexity and h_{max}^{int} parameter. An interpretation of the Table 5.4 could be that LZC_{270} is the best method to classify ISIs from STN while $\mathrm{SampEn}_{3,0.02}$ is the safest method for STN detection. The most of the methods had promising results to separate GP cores. Unfortunately, these methods worked well for ISIs with more then 300 samples, where amount of GP ISIs were small.

Because of input signals have been short ISIs, a time to compute one measure for a signal was negligible. Lempel-Ziv algorithm was the slowest to compute in the case of ISIs, but in the case of raw signals, multifractal analysis appears to be the slowest. Compared to raw signals, ISIs carry not so much information. Also DFA and MFA needed minimum of samples in a signal. I concluded that ISI signals without neuron sorting method applied is better to use,

because of higher number of samples in the ISI.

For future work there is a place for more thorough parameter choosing, sophisticated classifiers and new complexity measures. Also raw signals should be included to the analysis.

Bibliography

[1] A. Lempel and J. Ziv, "On the complexity of finite sequences," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 1, pp.75–81, 1976

[2] A. N. Pavlov, V. S. Anishchenko, "Multifractal analysis of complex signals," *Uspekhi Fizicheskikh Nauk*, vol. 177, pp. 859–876, 2007.

[3] J. F. Muzy, E. Bacry, A. Arneodo, "Wavelets and multifractal formalism for singular signals: Application to turbulence data," *Phys. Rev. Lett*, vol. 67, pp. 3515–3518, 1991.

[4] D. Makowiec, A. Rynkiewicz, R. Galaska, J. Wdowczyk-Szulc, and M. Zarczynska-Buchowiecka, "Reading multifractal spectra: Aging by multifractal analysis of heart rate," *EPL*, vol. 94, pp. 68005–68010, 2011.

[5] S. M. Pincus, "Approximate entropy as a measure of system complexity," *Proc. Natl. Acad. Sci.*, vol. 88, pp. 2297–2301, 1991.

[6] S. M. Pincus, E. K. Kalman, "Irregularity, volatility, risk, and financial market time series," *Proc. Natl. Acad. Sci.*, vol. 101, pp. 13709–13714, 2004.

[7] S. M. Pincus, A. L. Goldberger, "Physiological time-series analysis: what does regularity quantify?," *The American journal of physiology*, vol. 266, pp. 1643–1656, 1994.

[8] R. A. McKinley, L. K. McIntire, R. Schmidt, D. W. Repperger, and J. A. Caldwell, "Evaluation of Eye Metrics as a Detector of Fatigue," *Human Factors*, vol. 53, pp. 403–414, 2011.

[9] D. E. Lake, J. S. Richman, M. P. Griffin, and J. R. Moorman, "Sample entropy analysis of neonatal heart rate variability," *Am J Physiol Regul Integr Comp Physiol*, vol. 283, pp. 789–797, 2002.

[10] D. E. Lake, J. R. Moorman, "Accurate estimation of entropy in very short physiological time series: the problem of atrial fibrillation detection in implanted

ventricular devices," *Am J Physiol Heart Circ Physiol*, vol. 300, pp. 319–325, 2011.

[11] J. S. Richman, J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *Am J Physiol Heart Circ Physiol*, vol. 278, pp. 2039–2049, 2000.

[12] W. Chen, J. Zhuang, W. Yu, and Z. Wang, "Measuring complexity using FuzzyEn, ApEn, and SampEn," *Med. Eng. Phy.*, vol. 31, pp. 61–68, 2009.

[13] M. Aboy, R. Hornero, D. Abasolo, D. Alvarez, "Interpretation of the Lempel-Ziv Complexity Measure in the Context of Biomedical Signal Analysis," *IEEE Trans. on Biomed. Engineering*, vol. 53, pp. 2282–2288, 2006.

[14] H. L. Chan, M. A. Lin, S. T. Lee, Y. T. Tsai, P. K. Chao, T. Wu, "Complex analysis of neuronal spike trains of deep brain nuclei in patients with Parkinson's disease," *Brain Research Bulletin*, vol. 81, pp. 534–542, 2010.

[15] C. K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger, "Mosaic organization of DNA nucleotides," *Phys. Rev. E*, vol. 49, no. 2, pp. 1685–1689, 1994.

[16] C. K. Peng, S. Havlin, H. E. Stanley, A. L. Goldberger, "Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series," *Chaos*, vol. 5, pp. 82–87, 1995.

[17] J. M. Hausdorff, C. K. Peng, Z. Ladin, J. Y. Wei, A. L. Goldberger, "Is walking a random walk? Evidence for long-range correlations in the stride interval of human gait," *J Appl Physiol*, vol. 78, pp. 349–358, 1995.

[18] A. L. Goldberger, L. A. N. Amaral, J. M. Hausdorff, P. Ch. Ivanov, C. K. Peng, H. E. Stanley, "Fractal dynamics in physiology: Alternations with disease and aging," *PNAS*, vol. 99, pp. 2466–2472, 2002.

[19] R. Rohkamm, "Parkinson Disease: Pathogenesis," in *Color Atlas of Neurology*, 2nd ed. New York: Thieme, 2004, pp. 210-211.

[20] H. Cagnan, K. Dolan, X. He, M. F. Contarino, R. Schuurman, P. van den Munckhof, W. J. Wadman, L. Bour, H. C. F. Martens, "Automatic subthalamic nucleus detection from microelectrode recordings based on noise level and neuronal activity," *J. Neural Eng*, vol. 8, pp. 1–9, 2011.

[21] J. Favre, J. M. Taha, T. Baumann, K. J. Burchiel, "Computer Analysis of the Tonic, Phasic, and Kinesthetic Activity of Pallidal Discharges in Parkinson Patients," *Surg. Neurol*, vol. 51, pp. 665–673, 1999.

[22] A. Stocco, Ch. Lebiere, J. R. Anderson, "Conditional Routing of Information to the Cortex: A Model of the Basal Ganglia's Role in Cognitive Coordination," *Psychol Rev*, vol. 117, pp. 541–574, 2010.

[23] K. N. Magdoom, D. Subramanian, V. S. Chakravarthy, B. Ravindran, S. Amari, N. Meenakshisundaram, "Modeling Basal Ganglia for understanding Parkinsonian Reaching Movements," unpublished.

[24] T. Takekawa, Y. Isomura, T. Fukai, "Accurate spike sorting for multi-unit recordings," *European Journal of Neuroscience*, vol. 31, pp. 263–272, 2010.

[25] J. Wild, Z. Prekopcsak, T. Sieger, D. Novák, R. Jech, "Performance comparison of spike sorting algorithms for single-channel recordings," unpublished.

[26] J. Jankovic, "Parkinson's disease: clinical features and diagnosis," *J Neurol Neurosurg Psychiatry*, vol. 79, pp. 368–376, 2008.

[27] M. T. Herrero, C. Barcia, J. M. Navarro, "Functional anatomy of thalamus and basal ganglia," *Child's Nerv Syst*, vol. 18, pp. 386–404, 2002.

[28] M. A. Hely, J. G. L. Morris, R. Trafficente *et al.*, "The Sydney multicentre study of Parkinson's disease progression and mortality at 10 years," *J Neurol Neurosurg Psychiatry*, vol. 67, pp. 300–307, 1999.

[29] M. A. Hely, V. S. C. Fung, J. G. L. Morris, "Treatment of Parkinson's disease," *Journal of Clinical Neuroscience*, vol. 7, pp. 484–494, 2000.

[30] M. M. Hoehn, M. D. Yahr, "Parkinsonism: onset, progression and mortality," *Neurology*, vol. 17, pp. 427–442, 1967.

[31] E. Melamed, J. Zoldan, R. Galili-Mosberg, I. Ziv, R. Djaldetti, "Current management of motor fluctuations in patients with advanced Parkinson's disease treated chronically with levodopa," *J. Neural. Transm. Suppl.*, vol. 56, pp. 173–183, 1999.

[32] D. Merims, N. Giladi, "Dopamine dysregulation syndrome, addiction and behavioral changes in Parkinson's disease," *Parkinsonism and Related Disorders*, vol. 14, pp. 273–280, 2008.

[33] A. E. Lang, A. M. Lozano, E. Montgomery, J. Duff, R. Tasker, W. Hutchinson, "Posteroventral medial pallidotomy in advanced Parkinson's disease," *N Engl J Med*, vol. 337, pp. 1036–1042, 1997.

[34] M. I. Hariz, "Current contraversies in pallidal surgery," *Adv Neurol*, vol. 80, pp. 593–602, 1999.

[35] P. Krack, W. Hamel, H. M. Mehdorn, G. Deuschl, "Surgical treatment of Parkinson's disease," *Curr Opin Neurol*, vol. 12, no. 4, pp. 417–425, 1999.

[36] P. Krack, A. Benazzouz, P. Pollak *et al*, "Treatment of tremor in Parkinson's disease by subthalamic nucleus stimulation," *Mov Disord*, vol. 13, pp. 907–914, 1998.

[37] A. M. Lozano, A. E. Lang, W. D. Hutchison, "Pallidotomy for tremor" *Mov Disord*, vol. 13, pp. 107–110, 1998.

[38] A. L. Benabid, P. Pollak, E. Seigneuret, D. Hoffmann, E. Gay, and J. Perret, "Chronic VIM thalamic stimulation in Parkinson's disease, essential tremor and extra-pyramidal dyskinesias," *Acta Neurochir Suppl*, vol. 58, pp. 39–44, 1993.

[39] P. Limousin, P. Krack, P. Pollak, A. Benazzouz, C. Ardouin, D. Hoffman, A. L. Benabid, "Electrical stimulation of the subthalamic nucleus in advanced Parkinson's disease," *N Engl J Med*, vol. 339(16), pp. 1105–1111, 1998.

[40] S. Farris and M. Giroux, "Deep brain stimulation: a review of the procedure and the complications," *JAAPA*, vol. 24, pp. 39–40, 2011.

[41] L. L. Rubchinsky, N. Kopell, and K. Sigvardt, "Modeling facilitation and inhibition of competing motor programs in basal ganglia subthalamic nucleus–pallidal circuits," *PNAS*, vol. 100(24), pp. 14427–14432, 2003.

[42] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering," *Neural Computation*, vol. 16, pp. 1661–1687, 2004.

[43] E. Bakštein, "Navigation system for deep brain surgery," unpublished, 2012.

[44] J. Wild, "Micro-EEG data analysis," unpublished, 2010.

[45] T. Sieger, "Statistical Analysis of Single-Cell Recordings," unpublished, 2010.

[46] F. O. Walker, "Huntington's disease," *Lancet*, vol. 369, pp. 218–228, 2007.

[47] Official manual of the Condor, ver. 7.6.7, http://research.cs.wisc.edu/condor/manual/v7.6/

[48] M. Livny, "The Study of Load Balancing Algorithms for Decentralized Distributed Processing Systems," *PhD thesis*, Weizmann Institute of Science, 1983.

[49] D. Thain, T. Tannenbaum, and M. Livny, "Distributed Computing in Practice: The Condor Experience," *Concurrency and Computation: Practice and Experience*, vol. 17, pp. 323–356, 2005.

[50] "Parallel Computing Toolbox User's Guide," The MathWorks, Inc., 2011.

[51] "Matlab R2012a Documentation," http://www.mathworks.com/help/

[52] "Sun N1 Grid Engine 6.1 User's guide," http://docs.oracle.com/cd/E19957-01/820-0699/

[53] "Beginner's Guide to Oracle Grid Engine 6.2," http://www.oracle.com/technetwork/oem/grid-engine-166852.html

[54] [Untitled coronal cut of the brain]. Retrieved April 24, 2012, from: http://science-naturalphenomena.blogspot.com/2009/05/basal-ganglia.html

[55] "Software for analysis of multifractal time series" http://www.physionet.org/physiotools/multifractal/

List of Figures

List of Tables

List of Algorithms

Appendix A

CD attachment

Data structure of the attached CD:

- Bachelor_thesis - tex codes and other files for bachelor thesis compilation
- Matlab_files - data structure of all source codes, MAT files and pictures
- Bachelor_thesis.pdf

Appendix B

Graphs

B.1 Parallel computations

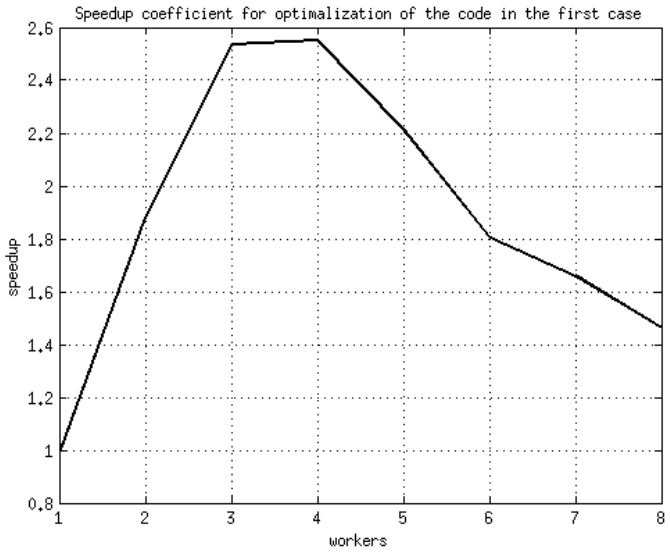


Figure B.1: Speedup coefficient for optimalization of the code from DBS Toolbox

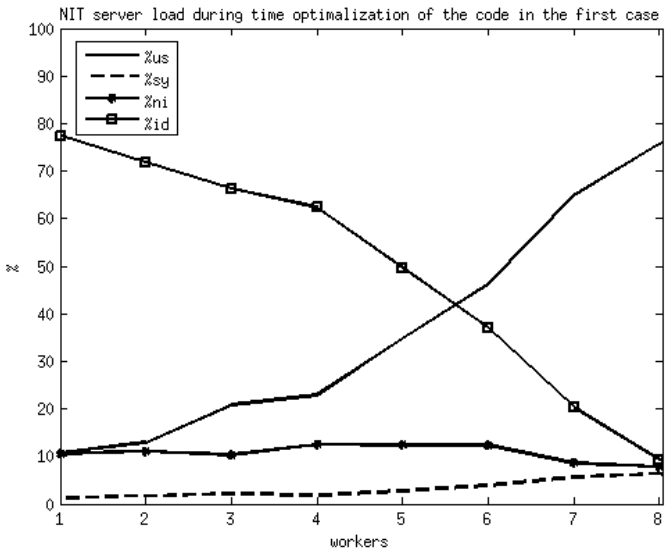


Figure B.2: NIT server load during usage of the parallelization of the code from DBS Toolbox

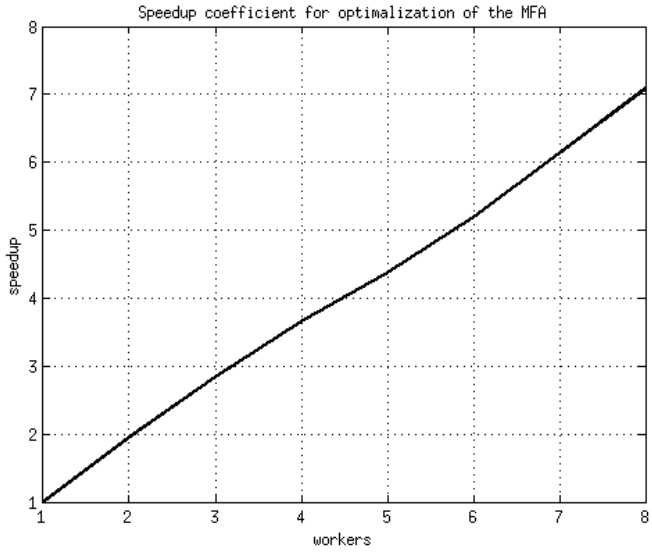


Figure B.3: Speedup coefficient for optimalization of the MFA (N=10000) code

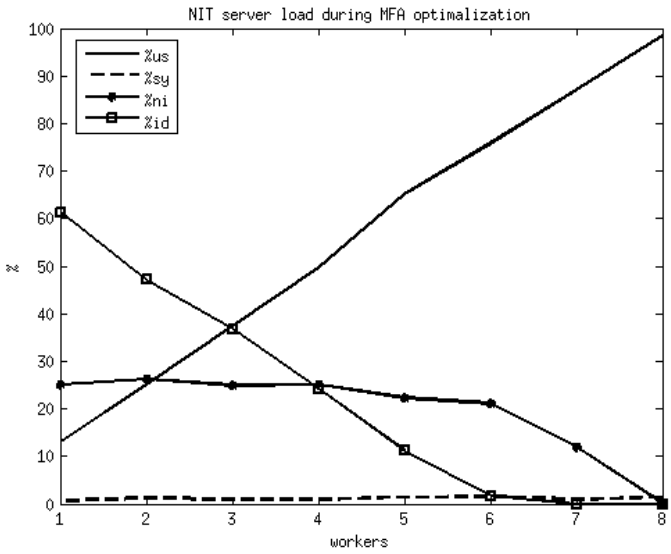


Figure B.4: NIT server load during usage of the parallelization of the MFA (N=10000) code

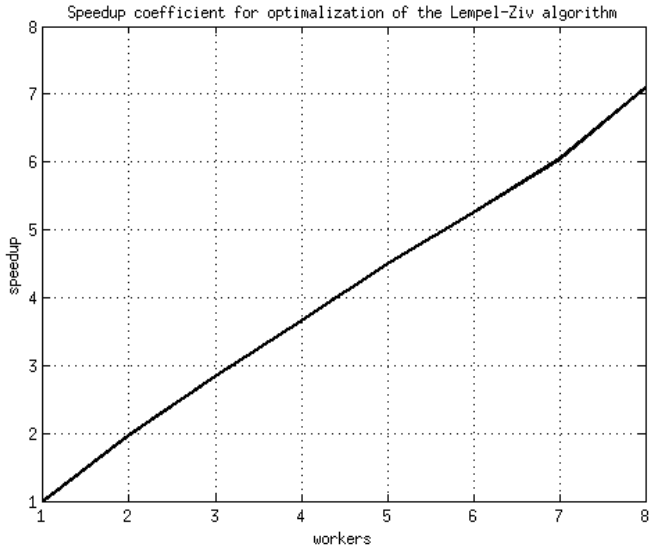


Figure B.5: Speedup coefficient for optimalization of the LZC code

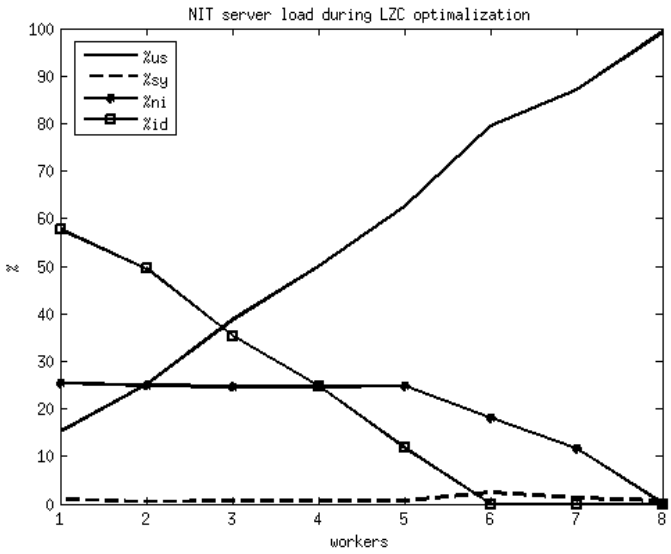


Figure B.6: NIT server load during usage of the parallelization of the LZC code

B.2 Lempel-Ziv complexity

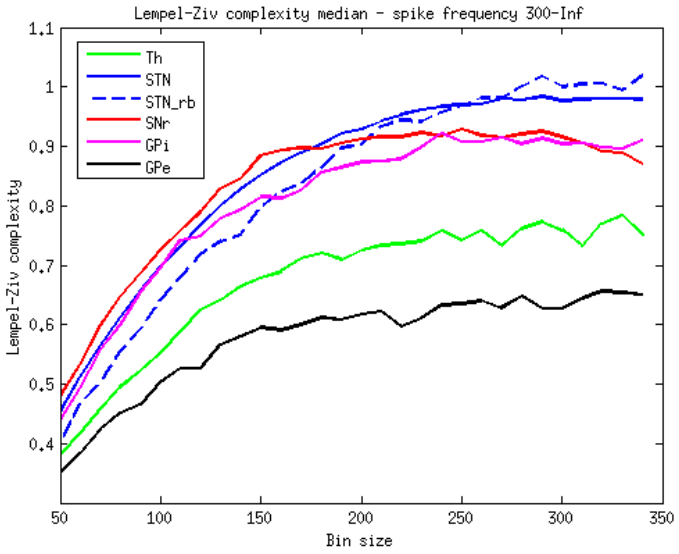


Figure B.7: LZC medians through all bin sizes for signals without neuron sorting

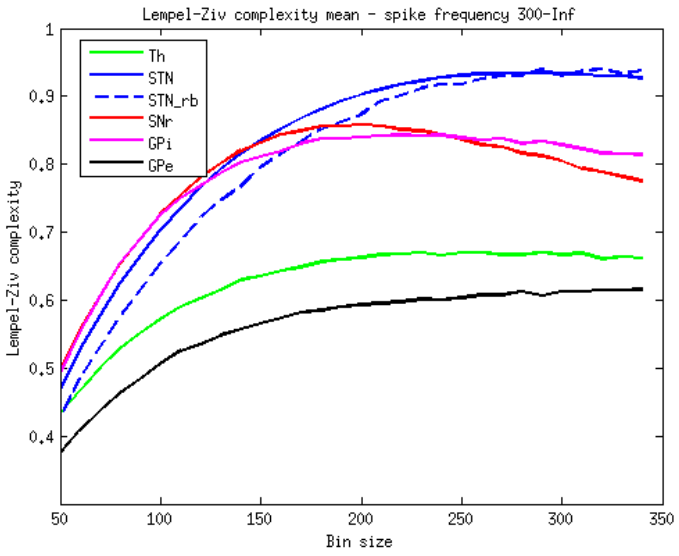


Figure B.8: LZC means through all bin sizes for signals without neuron sorting

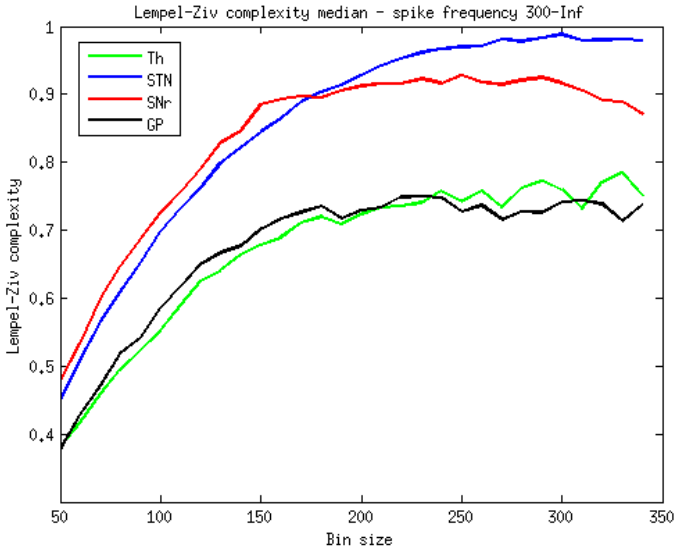


Figure B.9: LZC medians through all bin sizes for signals without neuron sorting

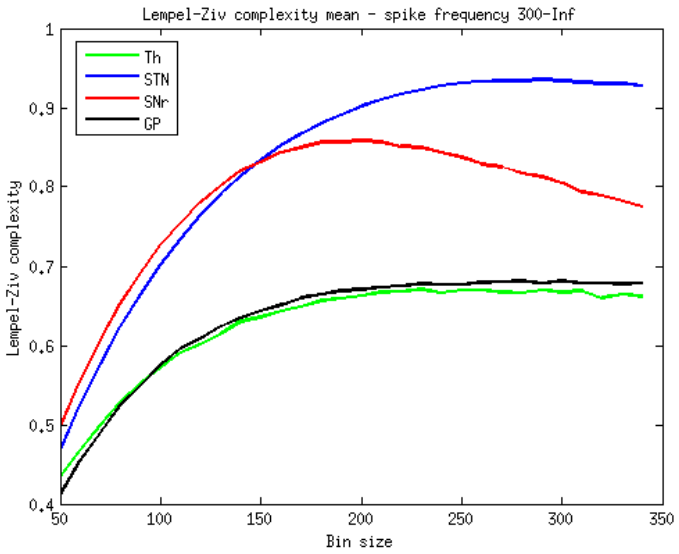


Figure B.10: LZC means through all bin sizes for signals without neuron sorting

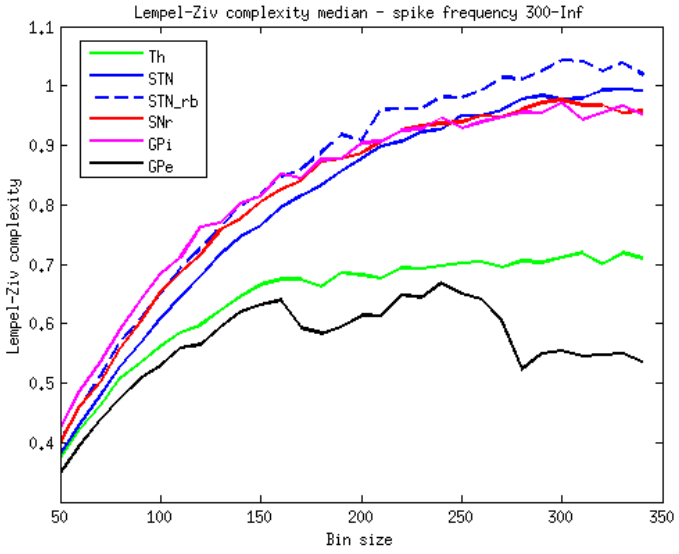


Figure B.11: LZC medians through all bin sizes for signals with neuron sorting

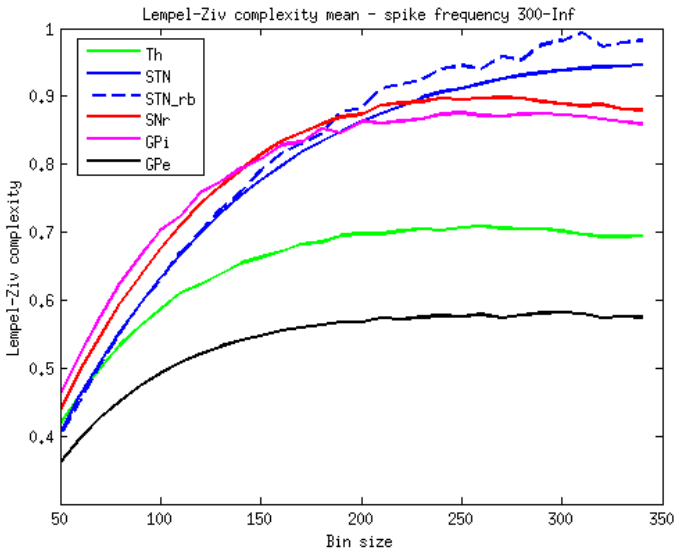


Figure B.12: LZC means through all bin sizes for signals with neuron sorting

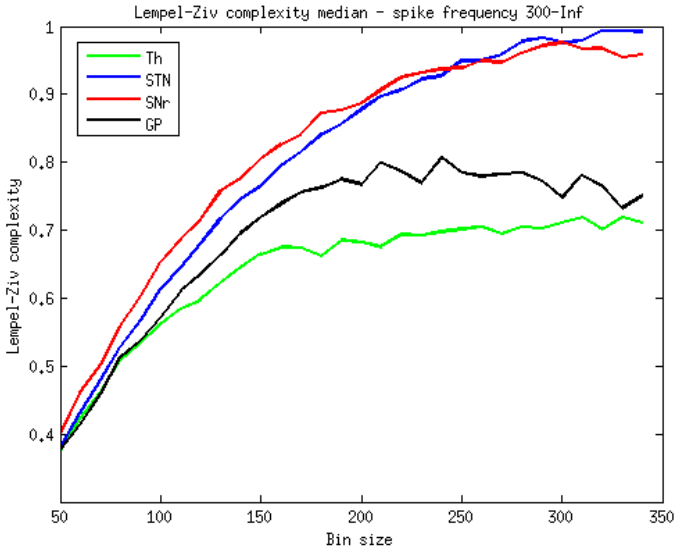


Figure B.13: LZC medians through all bin sizes for signals with neuron sorting

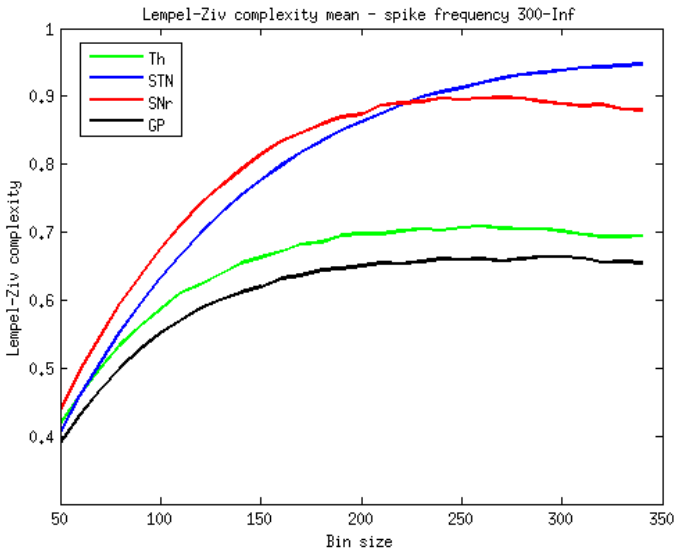


Figure B.14: LZC means through all bin sizes for signals with neuron sorting

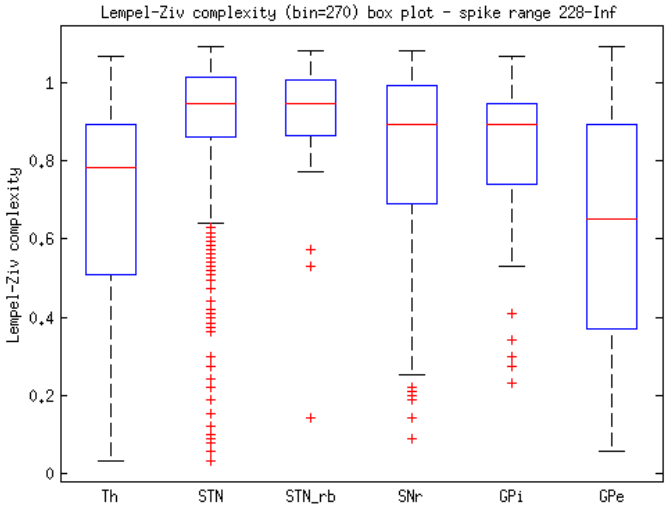


Figure B.15: LZC box graph for signals without neuron sorting

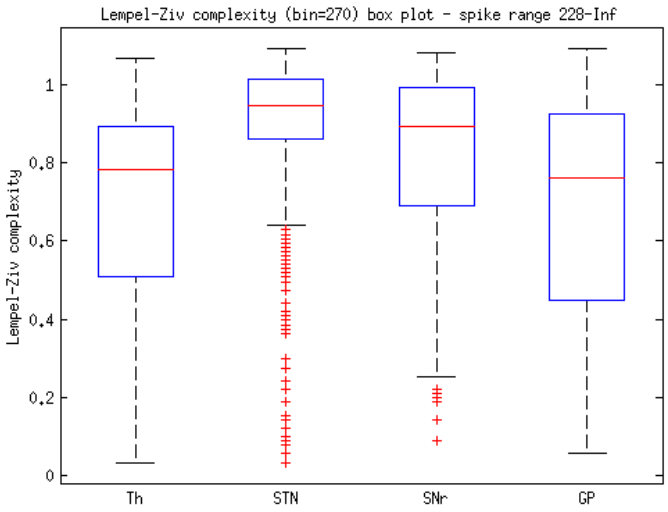


Figure B.16: LZC box graph for signals without neuron sorting

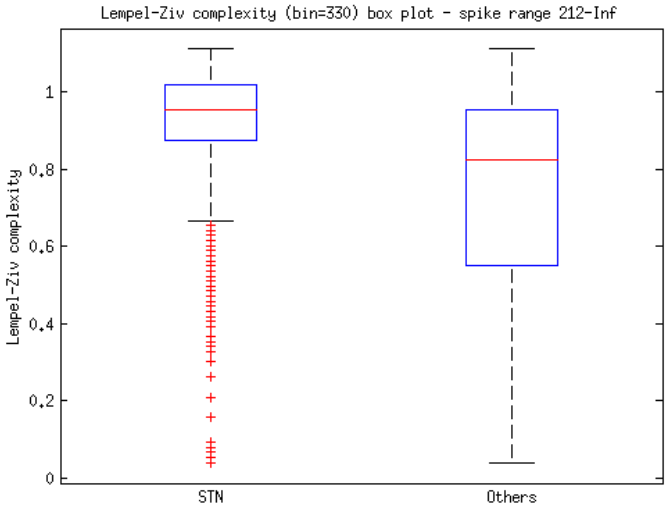


Figure B.17: LZC box graph for signals without neuron sorting

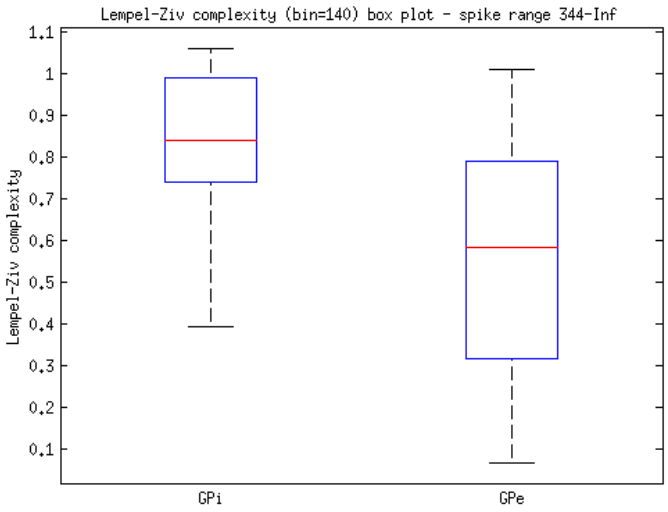


Figure B.18: LZC box graph for signals without neuron sorting

B.3 Burst index, pause index and pause ratio

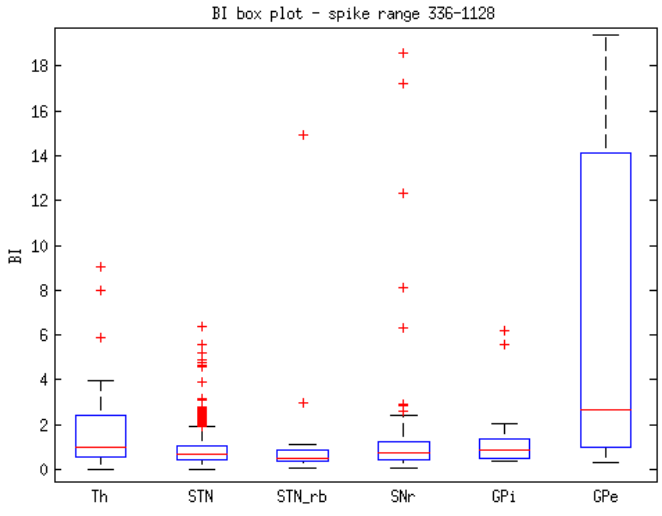


Figure B.19: Burst index box graph for 6 cores annotation and ISIs without neuron sorting

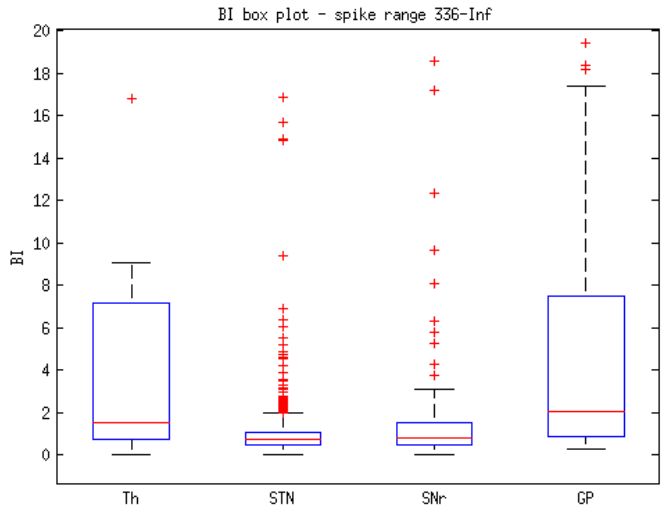


Figure B.20: Burst index box graph for 4 cores annotation and ISIs without neuron sorting

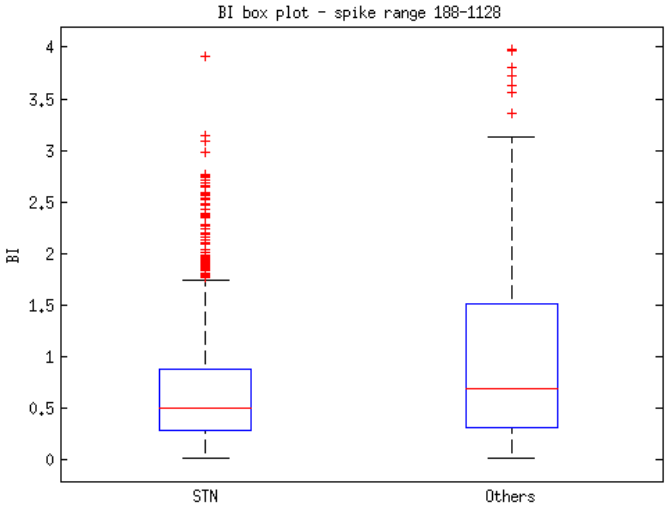


Figure B.21: Burst index box graph for STN/others annotation and ISIs without neuron sorting

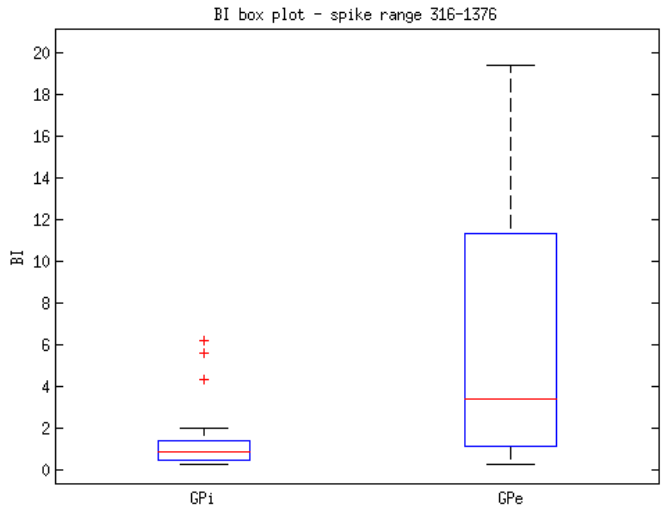


Figure B.22: Burst index box graph for GP annotation and ISIs without neuron sorting

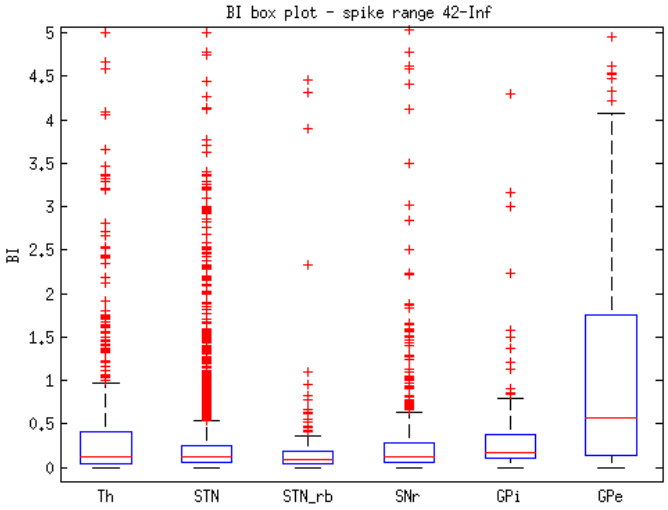


Figure B.23: Burst index box graph for 6 cores annotation and ISIs with neuron sorting

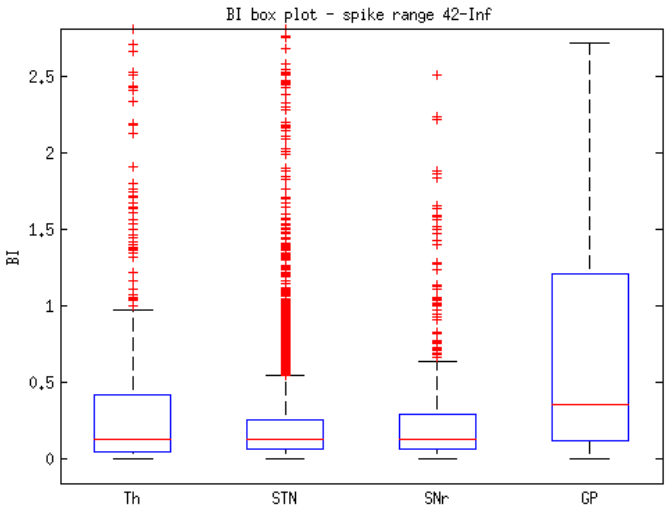


Figure B.24: Burst index box graph for 4 cores annotation and ISIs with neuron sorting

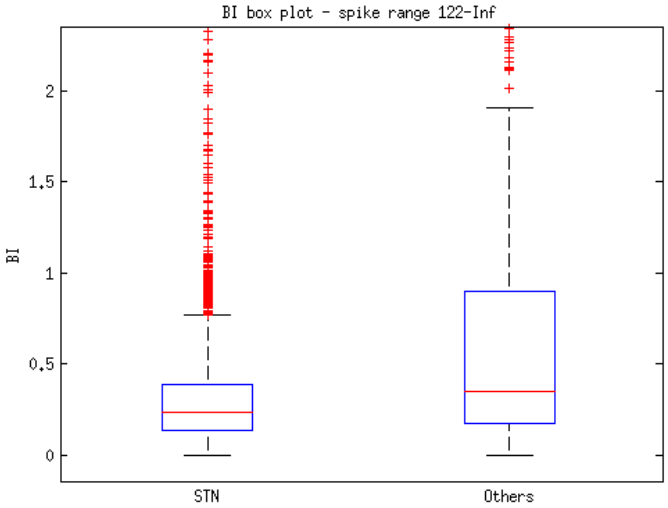


Figure B.25: Burst index box graph for STN/others annotation and ISIs with neuron sorting

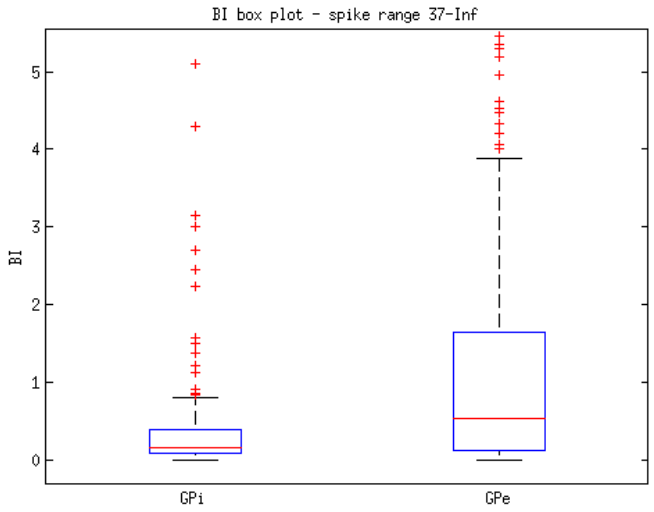


Figure B.26: Burst index box graph for GP annotation and ISIs with neuron sorting

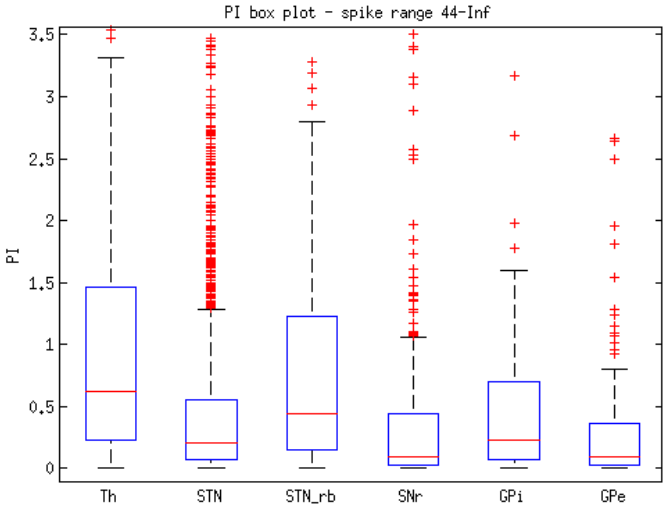


Figure B.27: Pause index box graph for 6 cores annotation and ISIs without neuron sorting

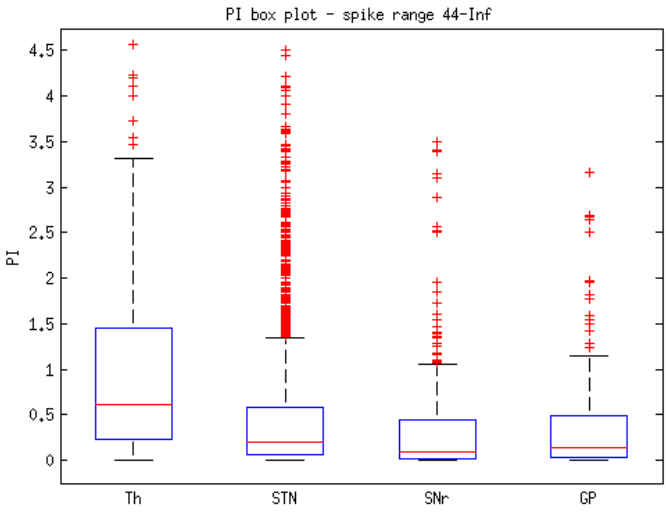


Figure B.28: Pause index box graph for 4 cores annotation and ISIs without neuron sorting

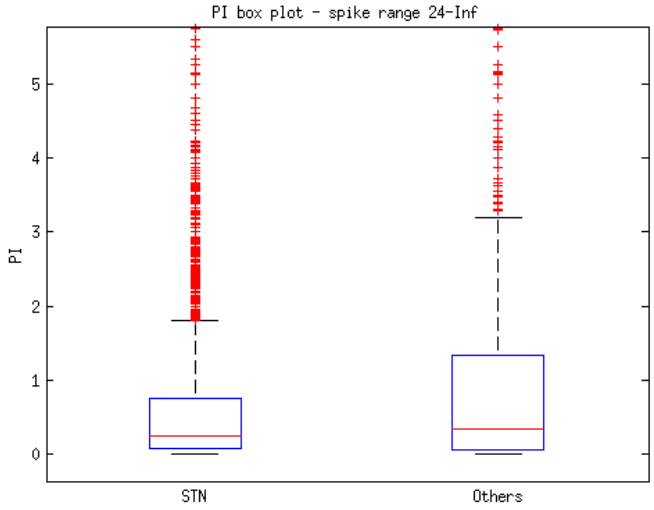


Figure B.29: Pause index box graph for STN/others annotation and ISIs without neuron sorting

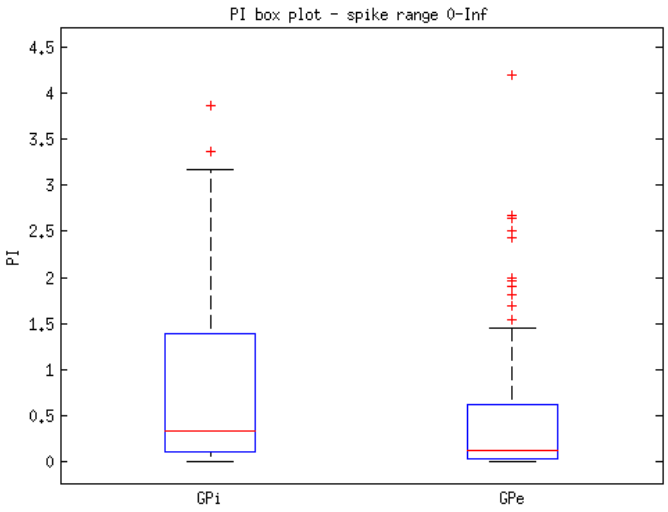


Figure B.30: Pause index box graph for GP annotation and ISIs without neuron sorting

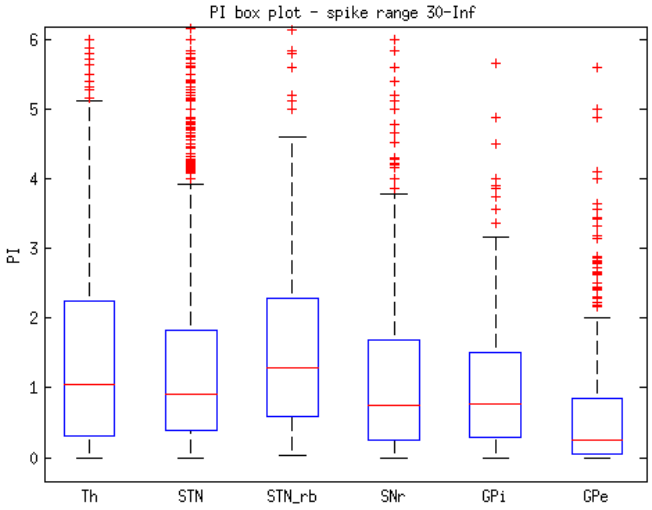


Figure B.31: Pause index box graph for 6 cores annotation and ISIs with neuron sorting

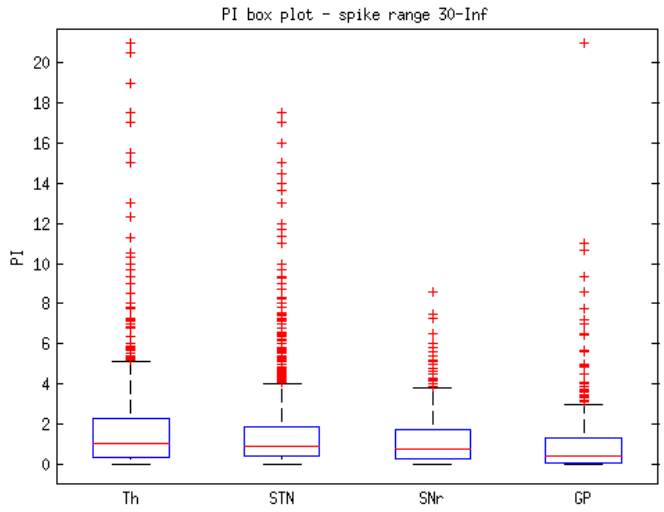


Figure B.32: Pause index box graph for 4 cores annotation and ISIs with neuron sorting

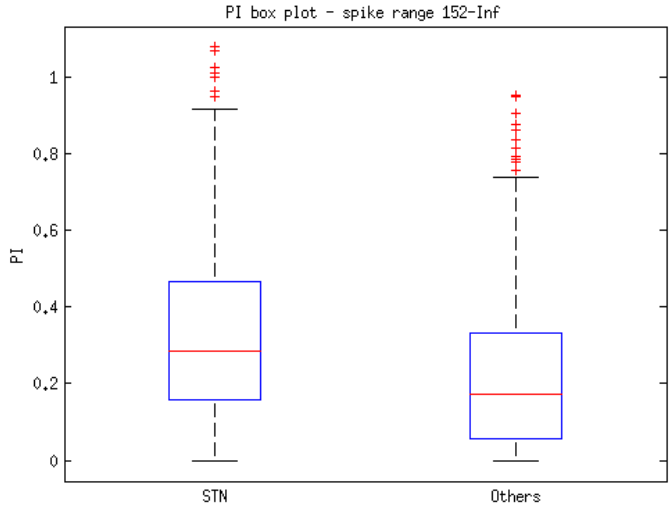


Figure B.33: Pause index box graph for STN/others annotation and ISIs with neuron sorting

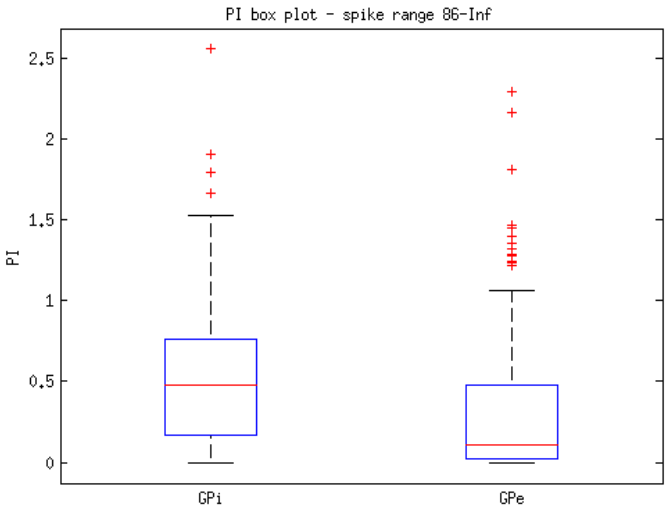


Figure B.34: Pause index box graph for GP cores annotation and ISIs with neuron sorting

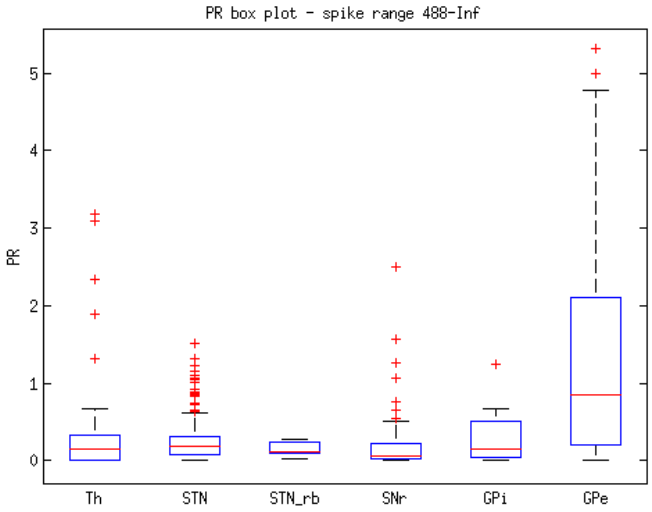


Figure B.35: Pause ratio box graph for 6 cores annotation and ISIs without neuron sorting

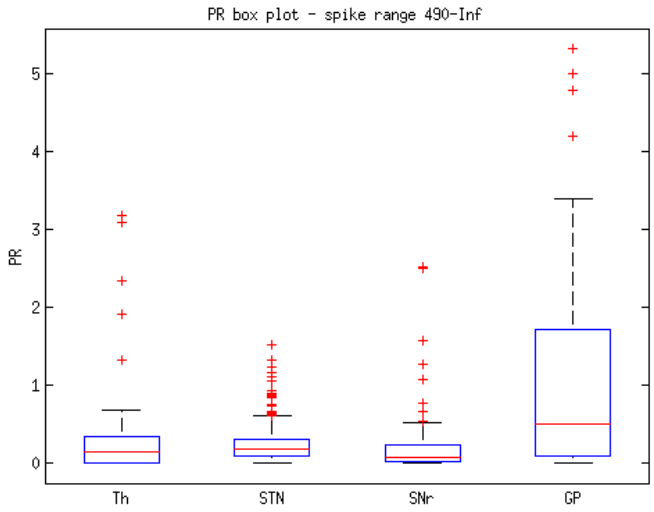


Figure B.36: Pause ratio box graph for 4 cores annotation and ISIs without neuron sorting

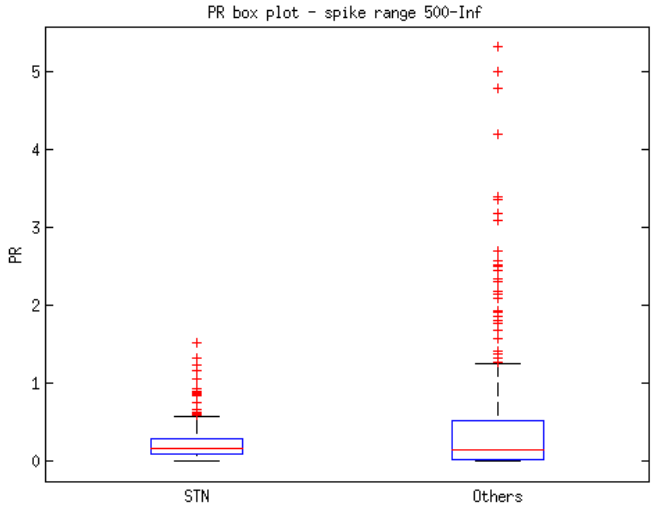


Figure B.37: Pause ratio box graph for STN/others annotation and ISIs without neuron sorting

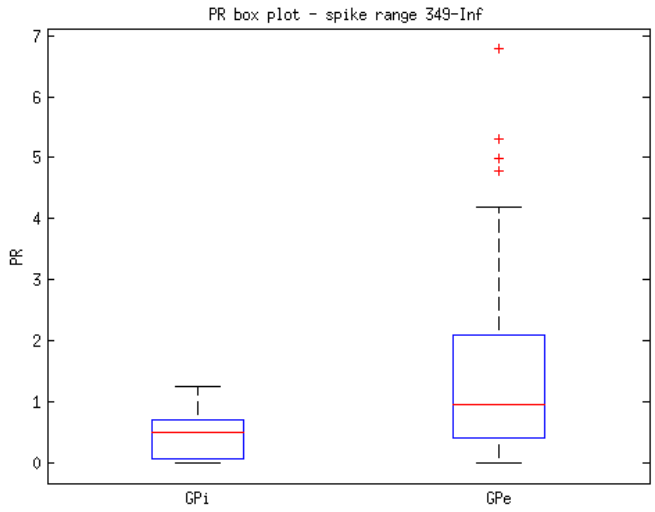


Figure B.38: Pause ratio box graph for GP annotation and ISIs without neuron sorting

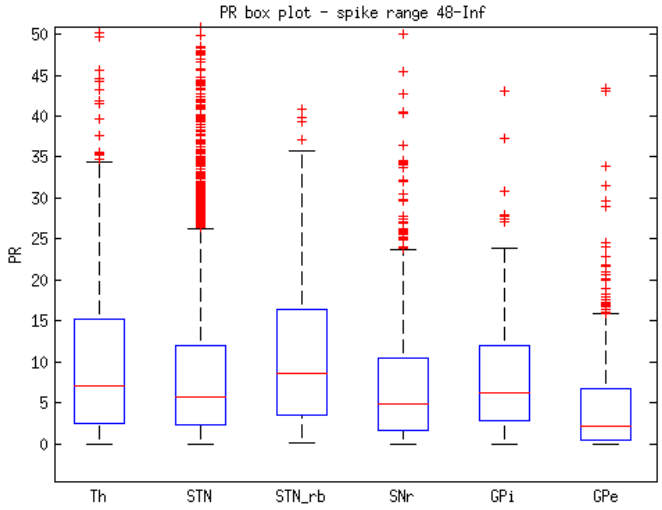


Figure B.39: Pause ratio box graph for 6 cores annotation and ISIs with neuron sorting

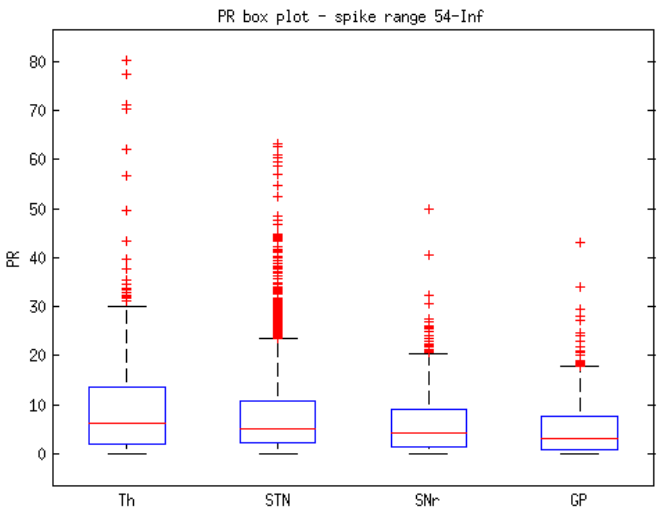


Figure B.40: Pause ratio box graph for 4 cores annotation and ISIs with neuron sorting

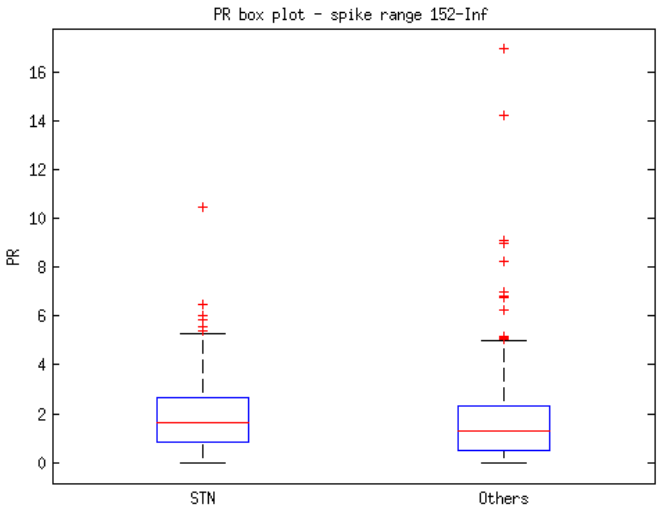


Figure B.41: Pause ratio box graph for STN/others annotation and ISIs with neuron sorting

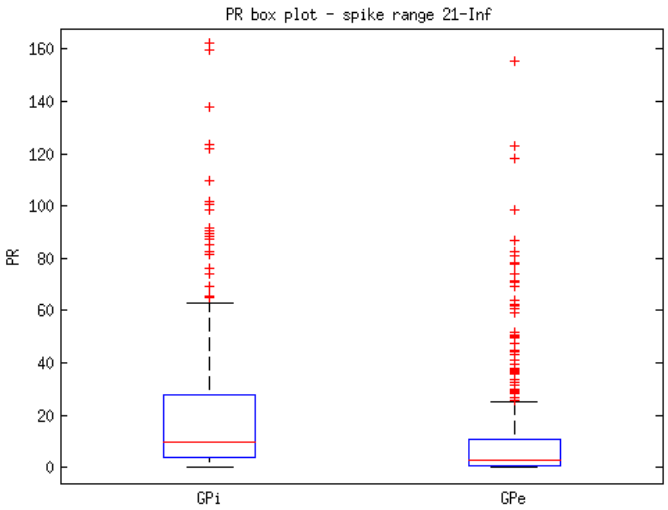


Figure B.42: Pause ratio box graph for GP annotation and ISIs with neuron sorting

B.4 Detrended fluctuation analysis

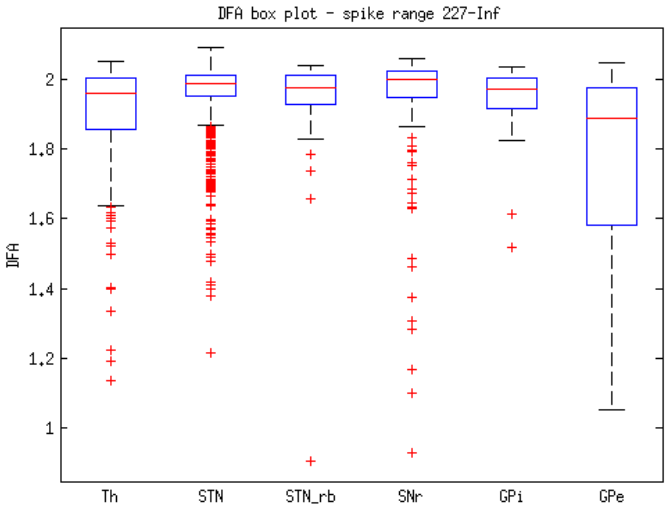


Figure B.43: DFA box graph for 6 cores annotation and ISIs without neuron sorting

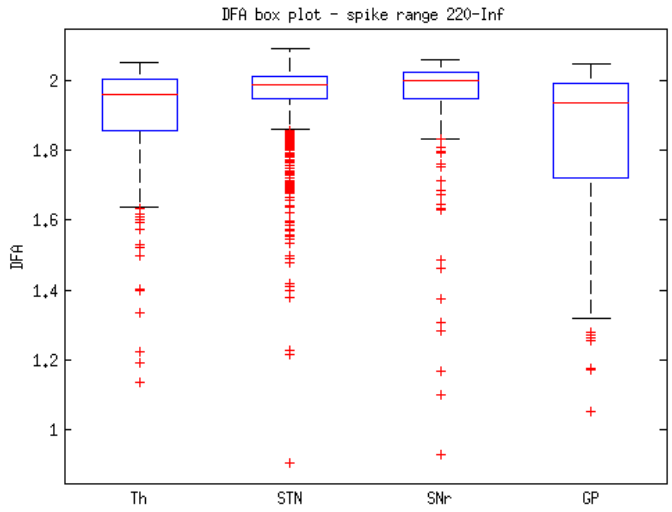


Figure B.44: DFA box graph for 4 cores annotation and ISIs without neuron sorting

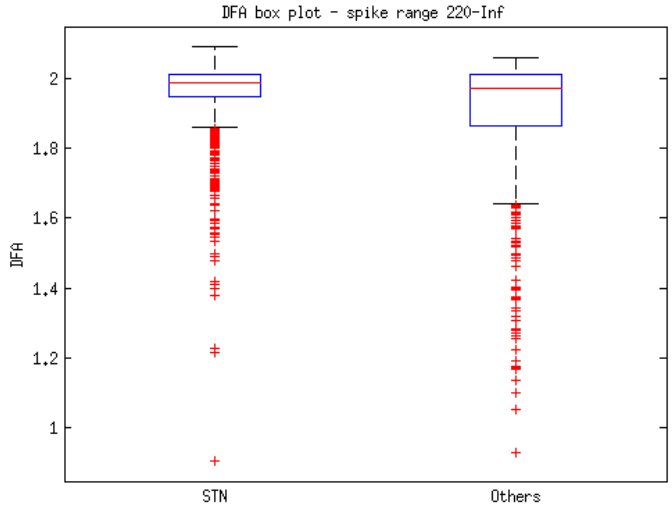


Figure B.45: DFA box graph for STN/others annotation and ISIs without neuron sorting

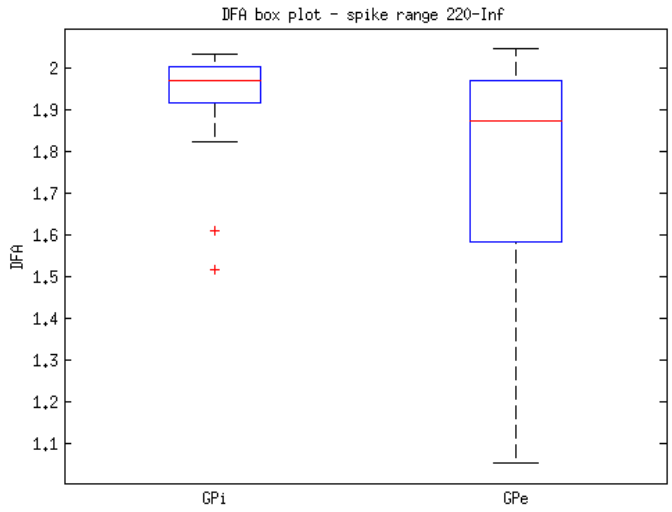


Figure B.46: DFA box graph for GP annotation and ISIs without neuron sorting

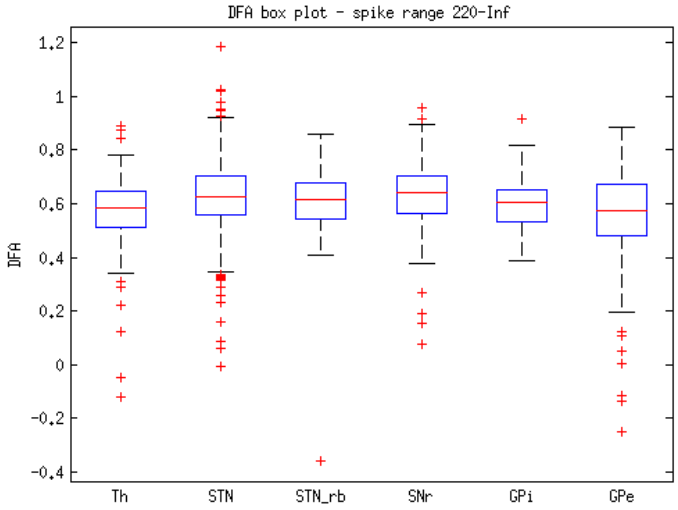


Figure B.47: DFA box graph for 6 cores annotation and ISIs with neuron sorting

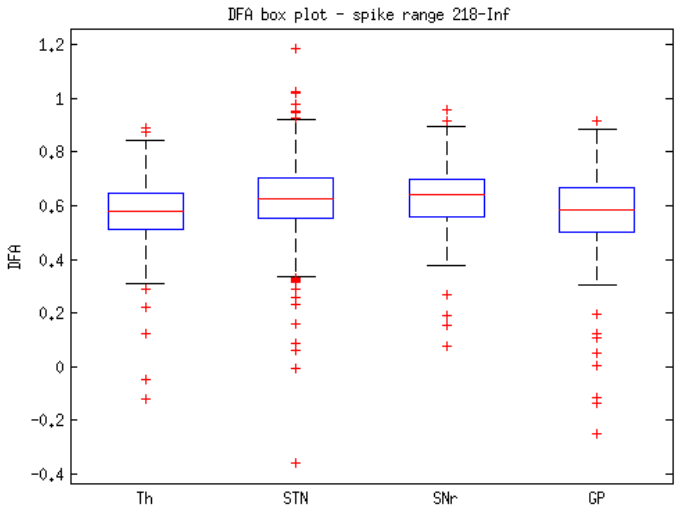


Figure B.48: DFA box graph for 4 cores annotation and ISIs with neuron sorting

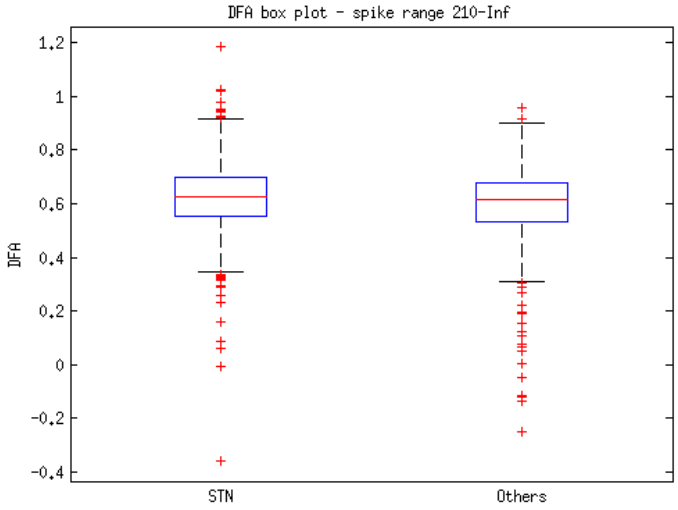


Figure B.49: DFA box graph for STN/others annotation and ISIs with neuron sorting

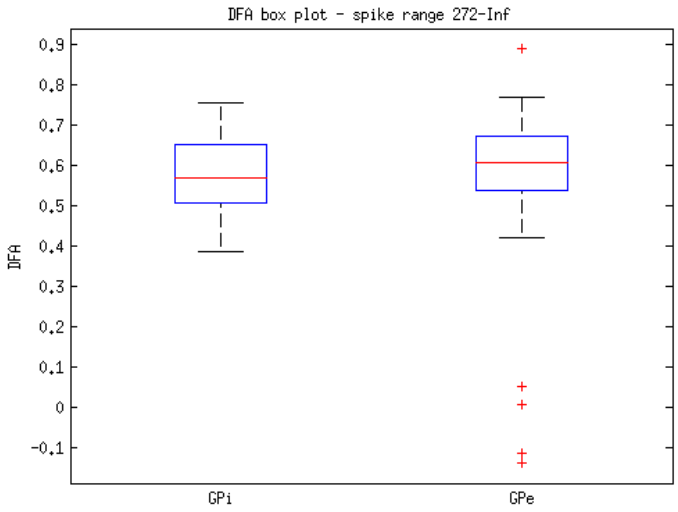


Figure B.50: DFA box graph for GP annotation and ISIs with neuron sorting

B.5 Approximate entropy

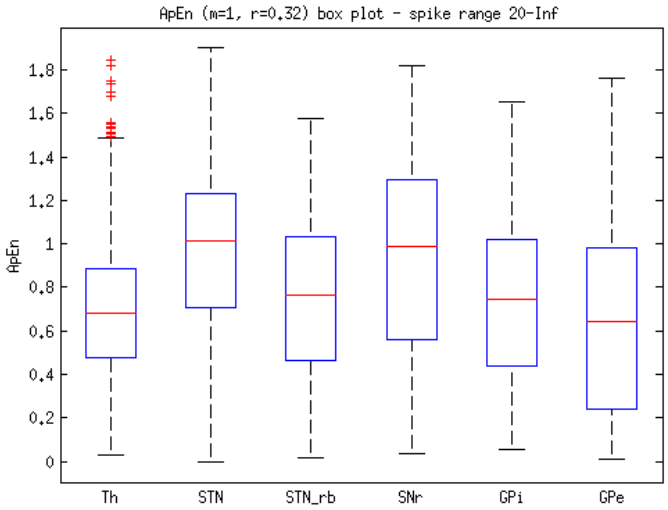


Figure B.51: ApEn box graph for 6 cores annotation and ISIs without neuron sorting

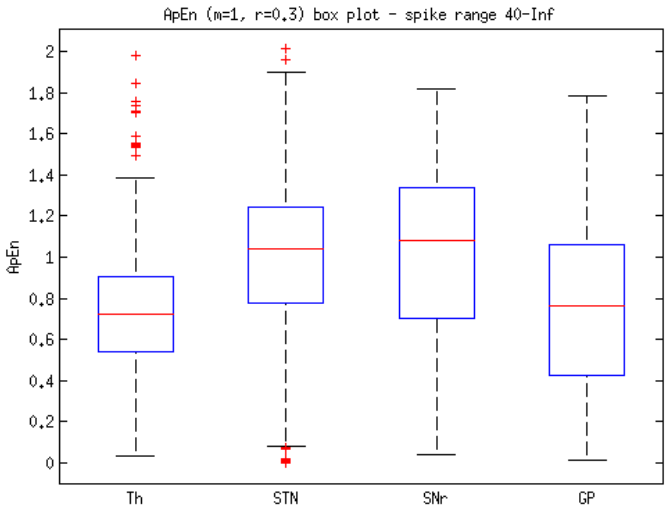


Figure B.52: ApEn box graph for 4 cores annotation and ISIs without neuron sorting

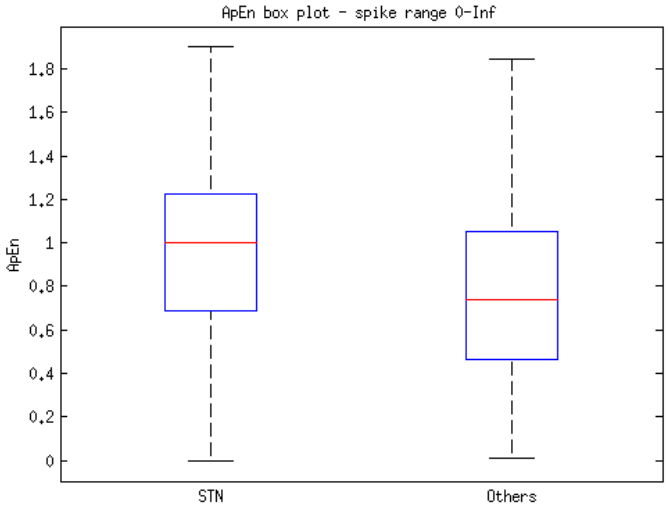


Figure B.53: ApEn box graph for STN/others annotation and ISIs without neuron sorting

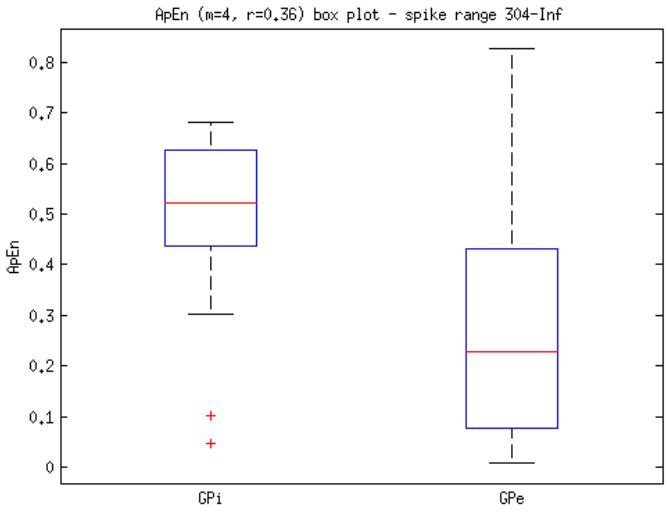


Figure B.54: ApEn box graph for GP annotation and ISIs without neuron sorting

B.6 Sample entropy

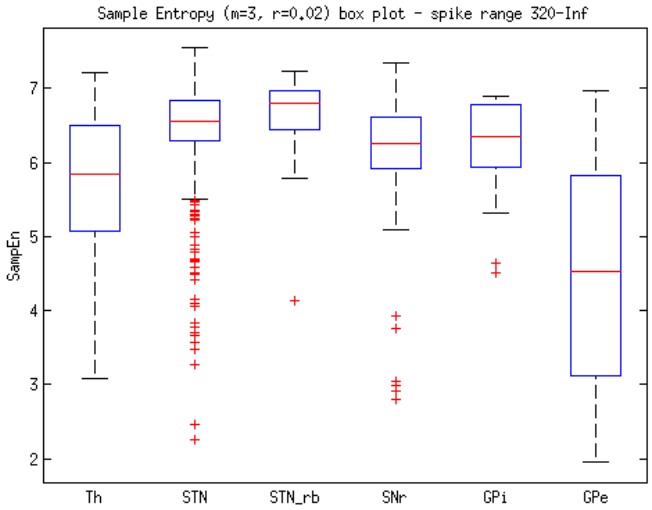


Figure B.55: The best result of SampEn for 6 cores annotation and ISIs without neuron sorting

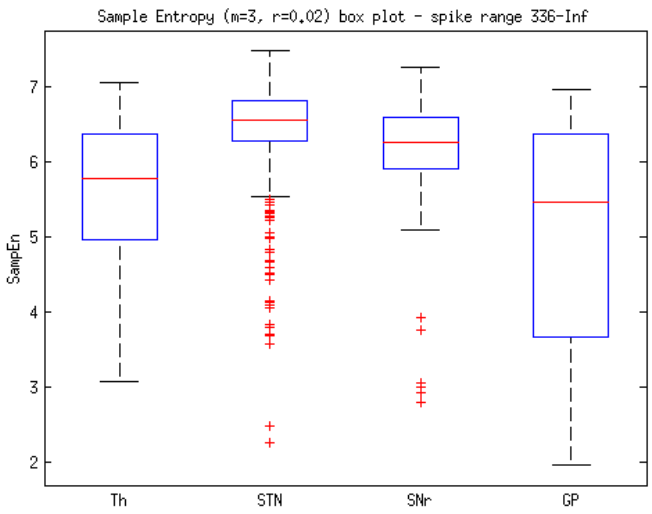


Figure B.56: The best result of SampEn for 4 cores annotation and ISIs without neuron sorting

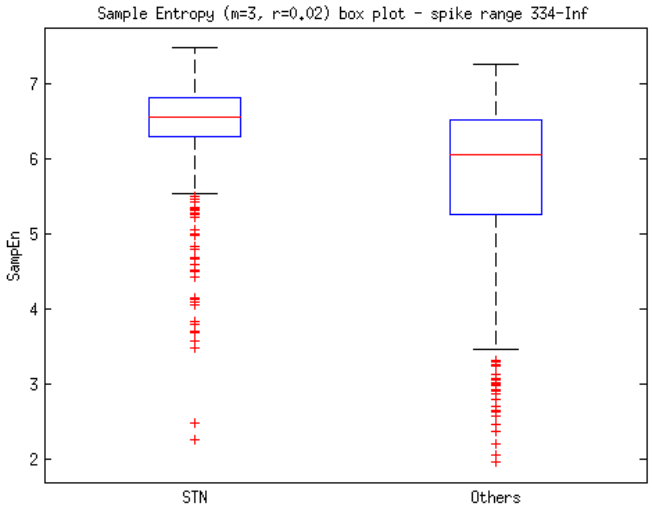


Figure B.57: The best result of SampEn for STN/others annotation and ISIs without neuron sorting

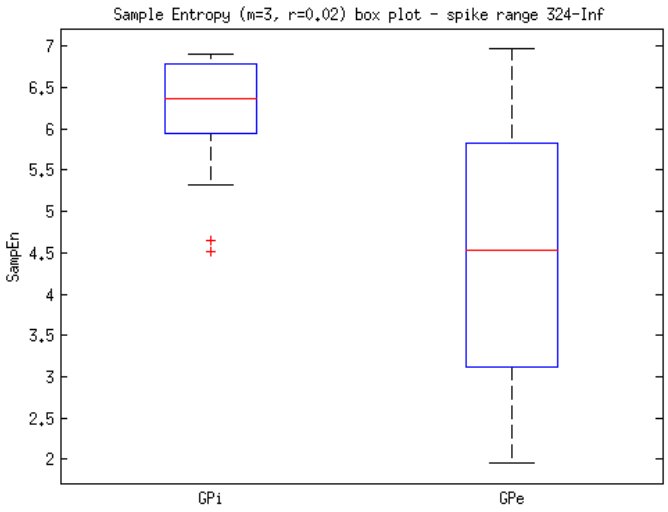


Figure B.58: The best result of SampEn for GP annotation and ISIs without neuron sorting

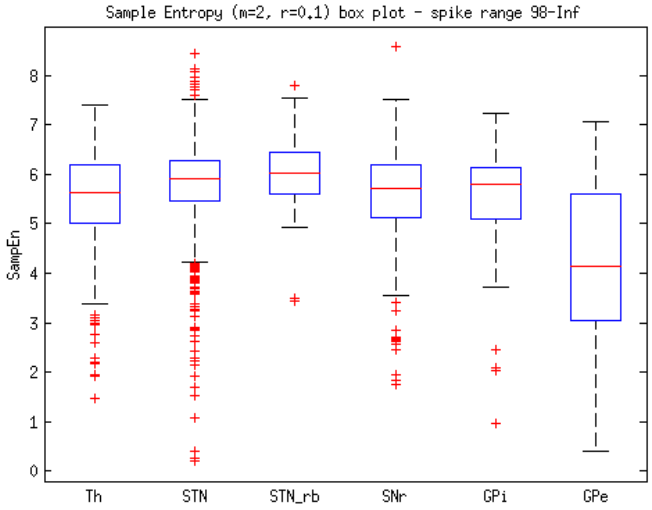


Figure B.59: The best result of SampEn for 6 cores annotation and ISIs with neuron sorting

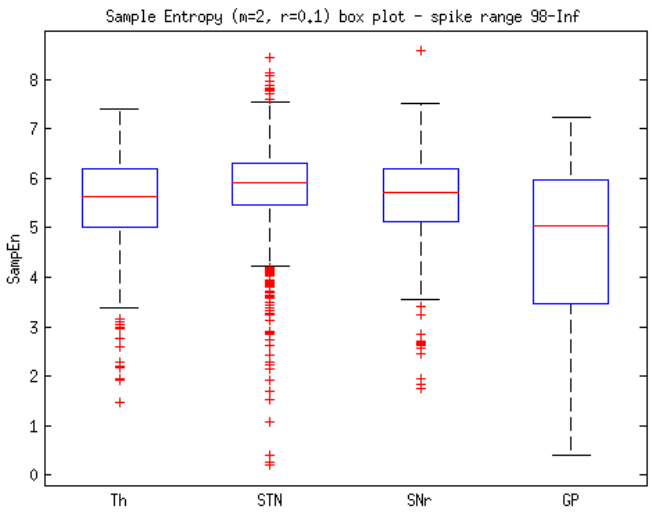


Figure B.60: The best result of SampEn for 4 cores annotation and ISIs with neuron sorting

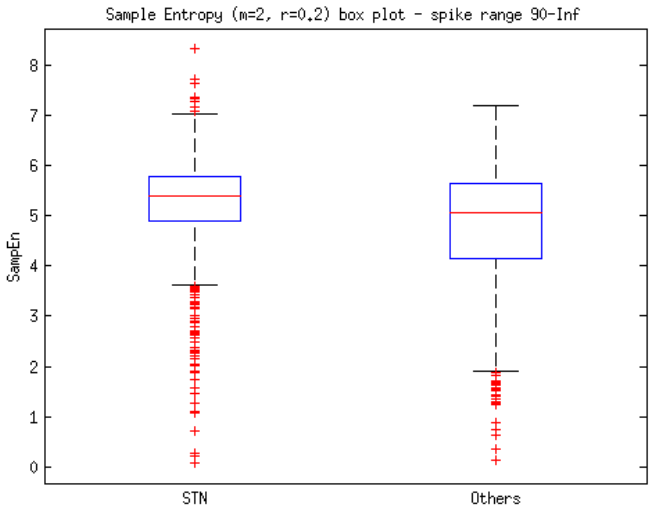


Figure B.61: The best result of SampEn for STN/others annotation and ISIs with neuron sorting

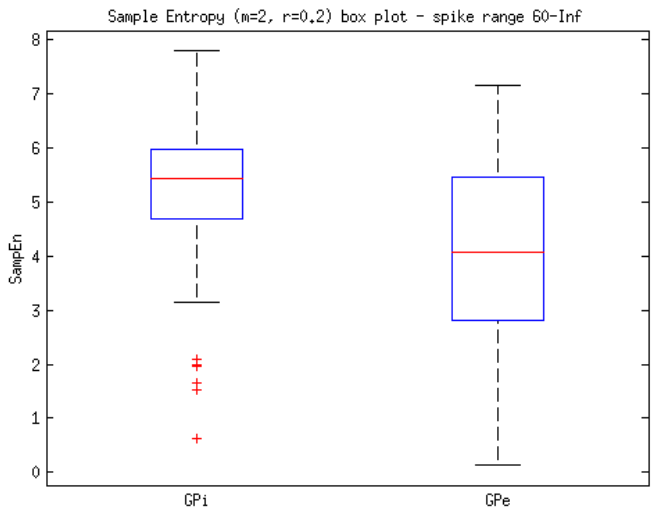


Figure B.62: The best result of SampEn for GP annotation and ISIs with neuron sorting

B.7 Multifractal analysis

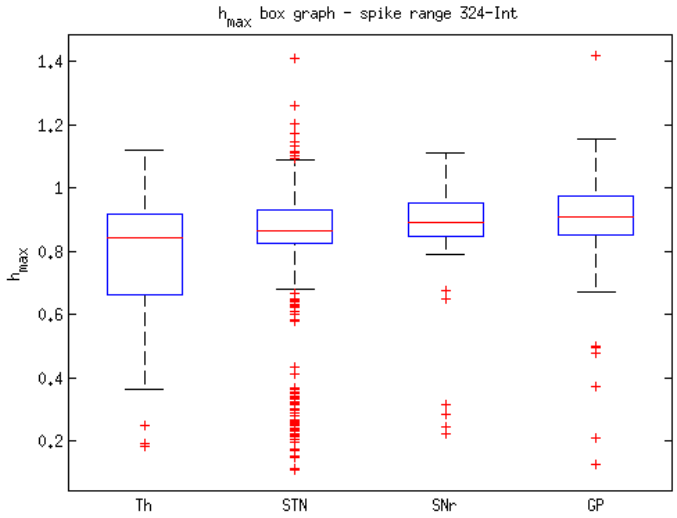


Figure B.63: The most significant result for h_{max} and ISIs without neuron sorting

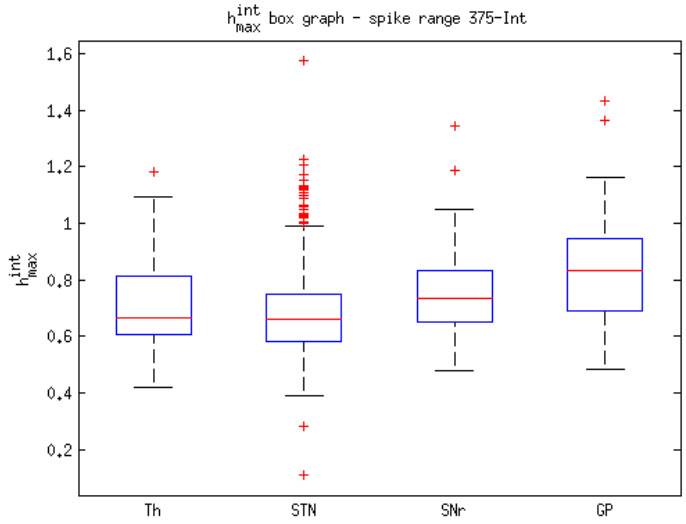


Figure B.64: The most significant result for h^{int}_{max} and ISIs without neuron sorting

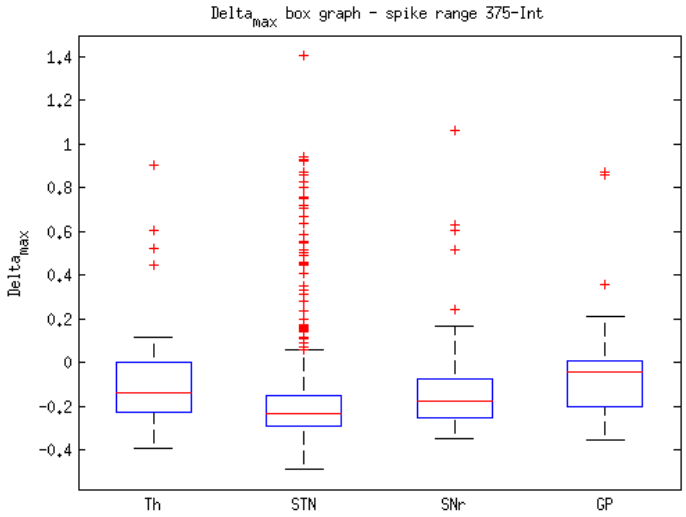


Figure B.65: The most significant result for Δ_{max} and ISIs without neuron sorting

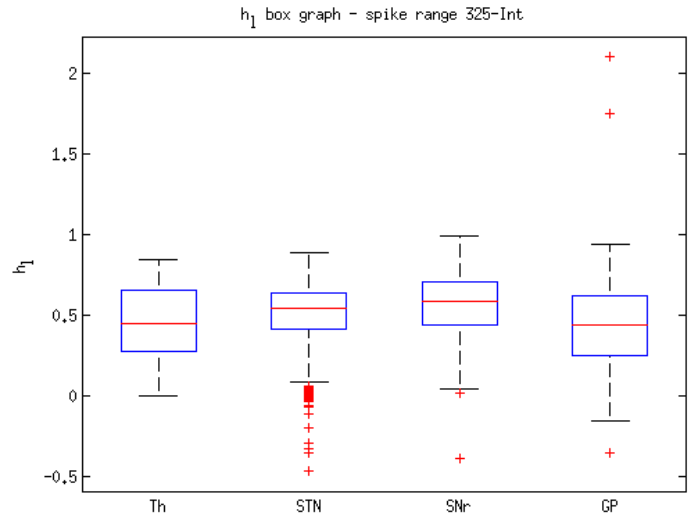


Figure B.66: The most significant result for h_l and ISIs without neuron sorting

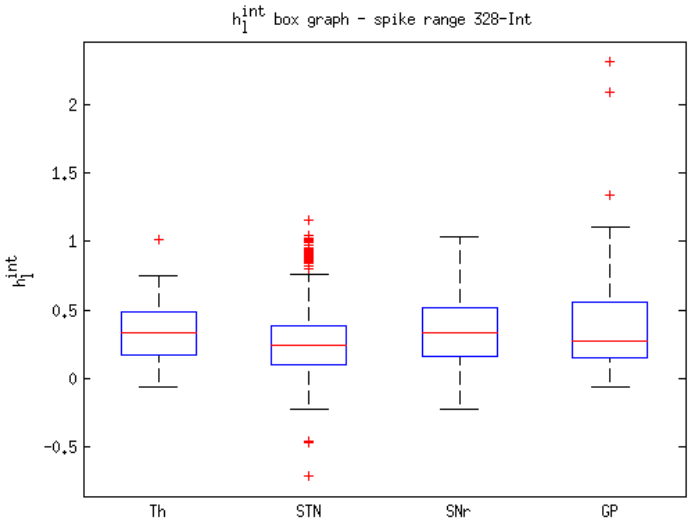


Figure B.67: The most significant result for h_l^{int} and ISIs without neuron sorting

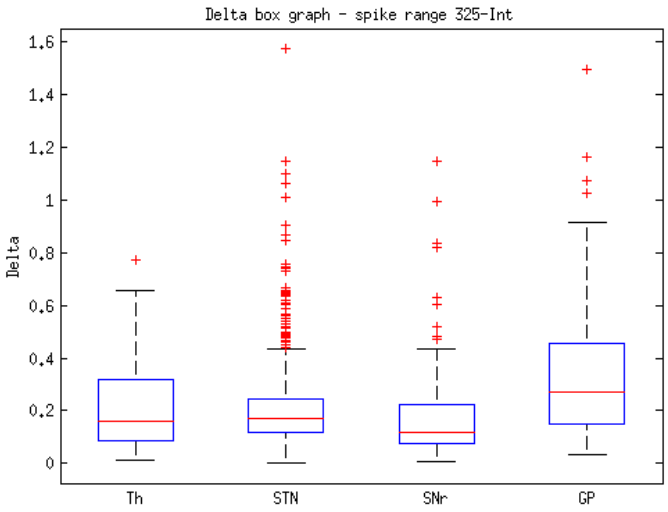


Figure B.68: The most significant result for Δ and ISIs without neuron sorting

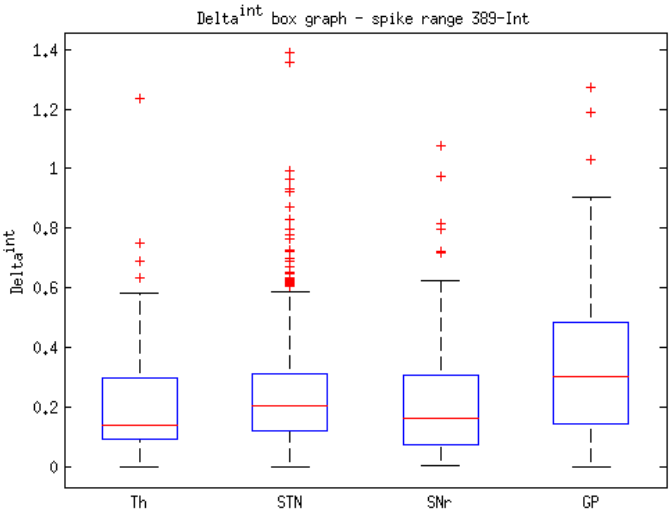


Figure B.69: The most significant result for Δ^{int} and ISIs without neuron sorting

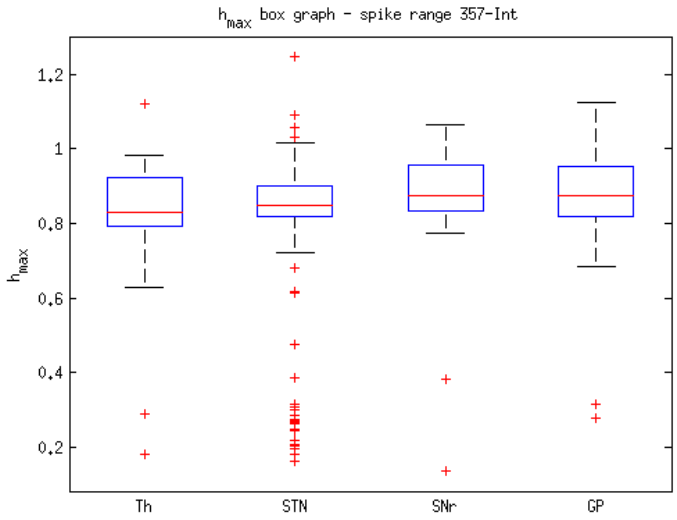


Figure B.70: The most significant result for h_{max} and ISIs with neuron sorting

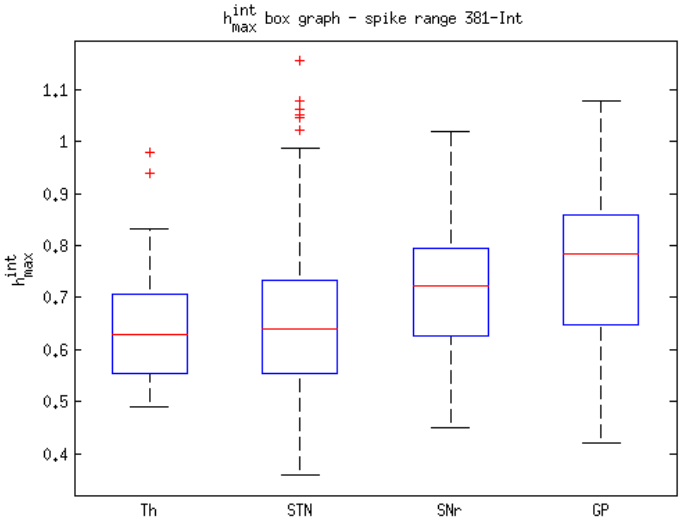


Figure B.71: The most significant result for h^{int}_{max} and ISIs with neuron sorting

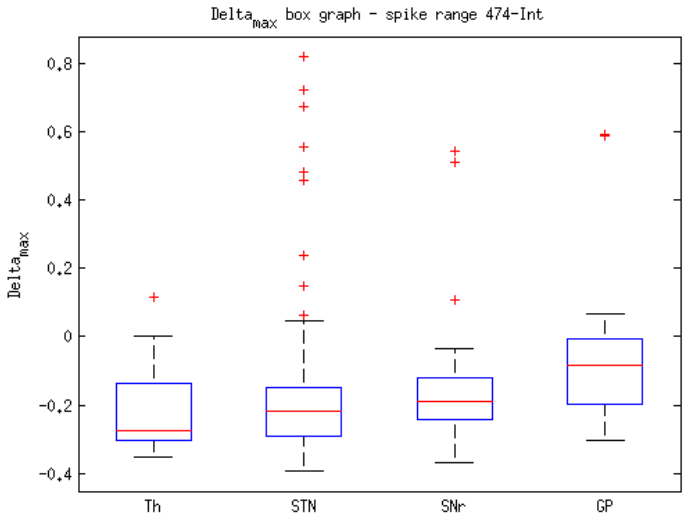


Figure B.72: The most significant result for Δ_{max} and ISIs with neuron sorting

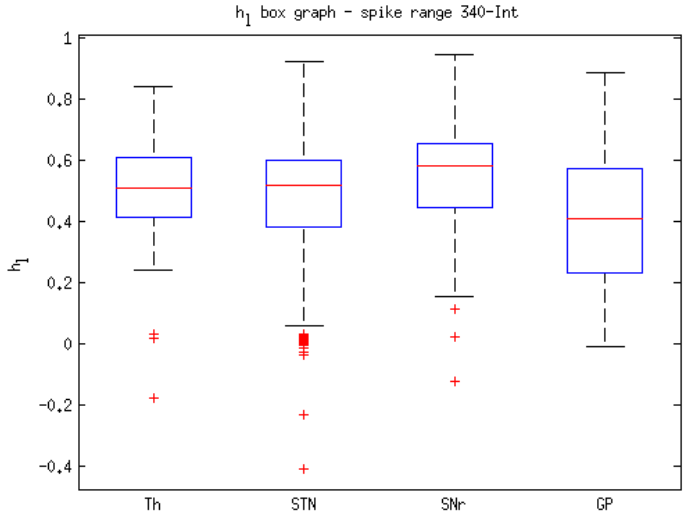


Figure B.73: The most significant result for h_l and ISIs with neuron sorting

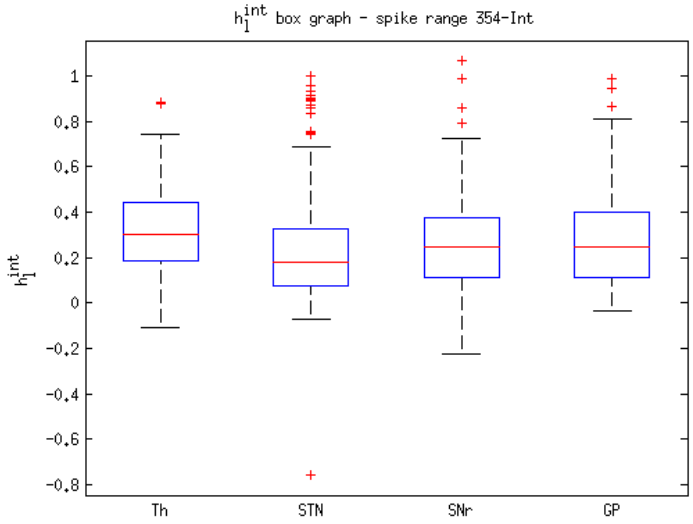


Figure B.74: The most significant result for h_l^{int} and ISIs with neuron sorting

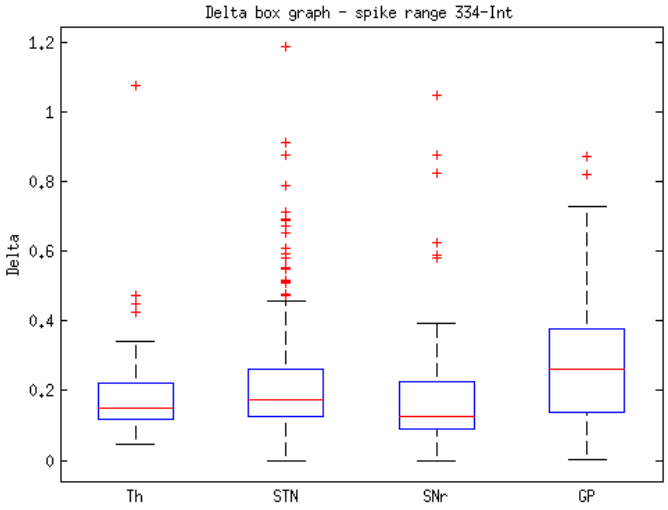


Figure B.75: The most significant result for Δ and ISIs with neuron sorting

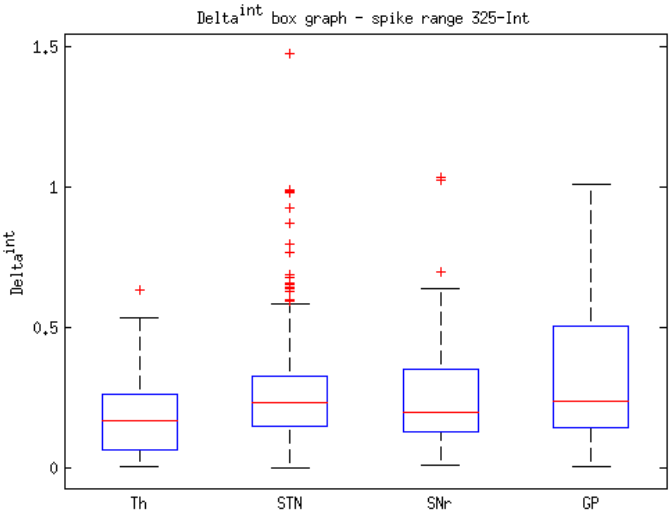


Figure B.76: The most significant result for Δ^{int} and ISIs with neuron sorting