DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL

Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

PhD THESIS

Statistical and optimization methods for spatial data analysis applied to neuroscience

Author

Laura Anton-Sanchez
MS Mathematics
MS Artificial Intelligence

PhD supervisors

Pedro Larrañaga
PhD Computer Science

Concha Bielza
PhD Computer Science

2017

Thesis Committee

President: Victor Maojo

External Member: Stephen J. Eglen

Member: Basilio Sierra

Member: Virgilio Gómez-Rubio

Secretary: Angel Merchán-Pérez

A mis padres, Asun y Antonio,
por enseñarme a pensar
despacio para andar deprisa

Acknowledgements

Many people deserve a grateful acknowledge after the years dedicated to this thesis:

My supervisors, Concha Bielza and Pedro Larrañaga, for their support and guidance.

All the people working in the Cajal Blue Brain project, with special acknowledge to Javier DeFelipe, Ruth Benavides-Piccione and Isabel Fernaud-Espinosa for their patience and help.

Hermann Cuntz, Felix Effenberger and all the members of the Ernst Strüngmann Institute for Neuroscience in Frankfurt, for their hospitality during the three months I was working with them.

My colleagues at the Computational Intelligence Group, where I include Martín Gutiérrez because he has been like one of the group, for their help and for a friendly work environment.

Juan Francisco Monge and José Fernández for guiding me in my first contacts with research. Filip Radulovic, for his trust and advice. Pedro L. López-Cruz for his hospitality and help from the first moment I arrived in Madrid. Rubén Armañanzas for being, despite the distance, the best mentor I could imagine.

I also want to thank the financial support of the following projects which has made this work possible: Cajal Blue Brain (C080020-09), TIN2010-20900-C04-04, TIN2013-41592-P and TIN2016-79684-P projects, S2013/ICE-2845-CASI-CAM-CM project, European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 604102 (Human Brain Project) and European Union's Horizon 2020 research and innovation programme under grant agreement No. 720270.

I thankfully acknowledge the computer resources, technical expertise and assistance provided by the Supercomputing and Visualization Center of Madrid (CeSViMa).

I acknowledge support from the Spanish MINECO scholarship at the Residencia de Estudiantes. I am grateful to all the people with whom I had the immense luck to share a year in that magical place.

Last but not least, I want to thank my family and friends for their support and encouragement during these years. Especially my parents, my sister and Antonio, because the result of this work is as much theirs as mine. This work is dedicated to them.

Abstract

Neuroscience has undergone great development in recent decades, making it one of the most relevant biomedical disciplines today. The development of new technologies and in particular the recent technical advances in microscopy make it possible to have a great amount of data that collect the nature and the spatial distribution of some neuronal elements that form the brain. In the current state of development of neuroscience, the need of new computational techniques is becoming more evident, and in this thesis it is carried out developing statistical and optimization methods for data analysis giving explicit consideration to spatial characteristics such as location, spatial organization or distance between elements.

The work developed in this thesis is mainly applied to the study of neuronal morphology. Despite the numerous efforts to better understand the brain, current knowledge about the neuron structure is still incomplete. Neuronal morphology reflects the organization of synaptic inputs and the way in which a neuron expands plays an important role in its functional and computational characteristics. Therefore, taking into account the inherent spatiality in neuronal morphology, key features can be revealed in the design of brain circuits.

This thesis focuses on the modeling of the spatial distribution of different neuronal structures in order to discover specific patterns and rules in their spatial organizations. To do this, we develop spatial point process methods for 3D spatial modeling, in particular, using replicated point patterns. In addition, considering neuronal arborizations as networks connecting the points where the synapses are located, we use graph theory and evolutionary computational techniques with a reverse engineering approach, to analyze if these networks follow principles of optimality in their design.

Regarding spatial point processes, the 3D spatial distribution of synapses is modeled in the six layers of the rat somatosensory cortex. Because several samples are available from each layer, replicated spatial patterns are used to detect similarities and differences between layers. Then, the existing 2D methodology for network spatial analysis is extended to 3D space. In addition, replicated spatial patterns are applied for the first time in this context. These methods are used to model the distribution of spines along the dendritic arborizations of human pyramidal neurons in both basal and apical dendrites. Next, the hypothesis of optimal wiring in neuronal circuits is used in conjunction with the analysis of the spatial distribution of branching and terminal points of dendritic arbors, using a measure related to the distance to the nearest neighbour to quantify how a set of points are distributed in space.

Regarding network optimization, a new way of representing and solving the structural constraints that commonly limit network design problems is proposed, namely, restrictions on the maximum number of edges incident on a node and establishing a priori the roles of the nodes in the network (root, intermediate or leaf node). Then, using graph theory and the proposed representation it is analyzed if individual neurons optimize brain connectivity in terms of wiring length. The analysis is carried out to the dendritic and axonal wiring of interneurons with very different morphology and to the dendritic wiring of a homogeneous population of pyramidal neurons, also studying in the latter case if there are differences between cortical layers.

Resumen

La neurociencia ha experimentado un gran desarrollo en las últimas décadas, convirtiéndola en una de las disciplinas biomédicas de mayor relevancia. El desarrollo de nuevas tecnologías y en concreto los recientes avances en microscopía permiten disponer de gran cantidad de datos que recogen la naturaleza y la distribución espacial de algunos elementos neuronales que forman el cerebro. En el estado actual de desarrollo de la neurociencia resulta cada vez más evidente la necesidad de nuevas técnicas computacionales, y en esta tesis se lleva a cabo desarrollando métodos estadísticos y de optimización para el análisis de datos dando consideración explícita a características espaciales como localización, organización espacial o distancia entre elementos.

El trabajo desarrollado en esta tesis se aplica principalmente al estudio de la morfología neuronal. Pese a los numerosos esfuerzos para comprender mejor el cerebro, el conocimiento actual sobre la estructura de la neurona es todavía incompleto. La morfología neuronal refleja la organización de las entradas sinápticas y la forma en la que una neurona se expande juega un papel importante en sus características funcionales y computacionales. Por ello, teniendo en cuenta la espacialidad inherente en la morfología neuronal se pueden revelar características clave en el diseño de los circuitos cerebrales.

Esta tesis se centra en el modelado de la distribución espacial de diferentes estructuras neuronales con el objetivo de descubrir patrones y reglas específicas en sus organizaciones espaciales. Para ello, se desarrollan métodos de procesos puntuales espaciales para el modelado espacial en 3D, en particular, utilizando patrones espaciales replicados. Además, considerando las arborizaciones neuronales como redes conectando los puntos donde se encuentran las sinapsis, se utilizan teoría de grafos y técnicas de computación evolutiva con un enfoque de ingeniería inversa, para analizar si estas redes siguen principios de optimalidad en su diseño.

En relación a los procesos puntuales espaciales, se modela la distribución espacial en 3D de sinapsis en las seis capas de la corteza somatosensorial del cerebro de rata. Al disponer de varias muestras de cada capa, se hace uso de patrones espaciales replicados para detectar similitudes y diferencias entre capas. Después, la metodología existente en 2D para el análisis espacial en redes se extiende al espacio 3D. Además, se aplican patrones espaciales replicados por primera vez en este contexto. Estos métodos se utilizan para modelar la distribución de las espinas a lo largo de las arborizaciones dendríticas de neuronas piramidales humanas, tanto en dendritas basales como apicales. A continuación, se trabaja con la hipótesis de un cableado óptimo en los circuitos neuronales junto con el análisis de la distribución espacial de los puntos de bifurcación y los puntos terminales de las arborizaciones dendríticas, haciendo uso de una medida relacionada con la distancia al vecino más cercano para cuantificar cómo se distribuyen un conjunto de puntos en el espacio.

En cuanto a la optimización de redes, se propone una nueva forma de representar y resolver las restricciones estructurales que comúnmente limitan los problemas de diseño de redes, en concreto, restricciones de número máximo de aristas incidentes en un nodo y el establecimiento a priori de los roles que deben tener los nodos en la red (nodo raíz, intermedio

u hoja). Después, utilizando teoría de grafos y la representación propuesta, se analiza si las neuronas individualmente optimizan la conectividad del cerebro en términos de longitud de cableado. Se analiza el cableado de dendritas y axones de interneuronas con muy diversa morfología, y el cableado dendrítico de una población homogénea de neuronas piramidales, estudiando también en este último caso si existen diferencias entre capas corticales.

Contents

List of Figures

List of Tables

Acronyms

BAM Brain Activity Map

BBP Blue Brain Project

BRAIN Brain Research through Advancing Innovative Neurotechnologies

CB Common Basket cell

CBBP Cajal Blue Brain Project

CeSViMa Supercomputing and Visualization Center of Madrid

CH Chandelier cell

CSIC Consejo Superior de Investigaciones Científicas

CSR Complete Spatial Randomness

CT Common Type cell

DCMST Degree-Constrained Minimum Spanning Tree

DRCMST Degree- and Role-Constrained Minimum Spanning Tree

ecdf empirical cumulative distribution function

EDA Estimation of Distribution Algorithm

EPFL École Polytechnique Fédérale de Lausanne

FIB Focused Ion Beam

GA Genetic Algorithm

gGA generational Genetic Algorithm

HBP Human Brain Project

HT Horse Tail cell

jMetal Metaheuristic Algorithms in Java

LB Large Basket cell

MA Martinotti cell

MC Monte Carlo

MKEDA Mallows Kernel EDA

MST Minimum Spanning Tree

NHBSA Node Histogram Based Sampling Algorithm

NN Nearest Neighbour

PMX Partially Matched Crossover

RSA Random Sequential Adsorption

SEM Scanning Electron Microscopy

ssGA steady-state Genetic Algorithm

UML Unified Modeling Language

UPM Universidad Politécnica de Madrid

Part I

INTRODUCTION

Chapter 1

Introduction

A major goal of neuroscience is that in the next few years our knowledge about the structure and function of the brain is much deeper than it is now. In this way we will better understand some fundamental aspects of the brain, for example, the alterations that some diseases produce in it, how it forms, develops and ages, or the mechanisms by which we learn and improve our intellectual capacities.

Guided by this ambitious objective, computational neuroscience studies the brain function in terms of the information processing properties of structures that make up the nervous system [Churchland et al., 1993]. Within the field of computational neuroscience, in this dissertation we focus on computational neuroanatomy that intends to create anatomically accurate models of neuronal structures through the application of computational techniques, such as analysis, visualization, modeling and simulation [Berzhanskaya and Ascoli, 2008]. Great efforts to better understand the brain are taking place in several fields. Within them, computational neuroanatomy is an emerging science that requires advances from statistical analysis. Here we work at the cellular level, carrying out quantitative descriptions of the structure of single neurons and the density of neuronal elements within specific areas of the brain.

The Spanish neuroscientist Santiago Ramón y Cajal, designated by many as the father of modern neuroscience, already suggested more than a hundred years ago to interpret the construction planes of the brain by observing individual neuron morphology. Nevertheless, current knowledge about neuron structure is still incomplete. In this dissertation we propose to develop Cajal's idea through the use of spatial statistics techniques, in particular spatial point processes, and also optimization methods, specifically evolutionary computation techniques for network design optimization.

Spatial point processes may be applied in many areas of research to infer information on underlying processes that are reflected in the spatial structure. The evolution of new technologies and, particularly, the recent technical advances in microscopy allow the use of large databases that collect the nature and spatial distribution of some neuronal elements that form the brain (e.g., synapses, spines). Thus, neuroscience is one of the fields in which the application of spatial point processes as a modeling tool may be useful.

In addition to the development of spatial point processes to reveal spatial patterns in different brain structures, the study of the existence of an optimal neuronal design in individual cell morphologies is also a central part of this dissertation. We consider neuronal arborizations as networks connecting all points where synapses are located. Using graph theory and evolutionary computation techniques with a reverse engineering approach, we analyze if these neuronal networks follow optimality principles.

Chapter outline

This chapter is organized as follows. Section 1.1 details the main hypotheses and objectives of this dissertation. Then, Section 1.2 presents the complete organization of this manuscript.

1.1 Hypotheses and objectives

As mentioned above, the research of this dissertation is centered on spatial point processes and network design optimization, with the particular interest of extracting valuable knowledge in the field of neuroscience. Based on this, this dissertation has the following two main hypotheses:

- Spatial statistics methods can provide a deeper understanding of the spatial organization of different neuronal structures, discovering specific patterns and rules in their spatial distributions that may reveal key features in brain design.

- Individual neurons optimize brain connectivity. In particular, we hypothesize that by imposing constraints that provide realistic neuronal arborizations, we can for the most part explain wiring economy in single neurons considering only wiring length.

Based on these hypotheses, the main objectives of this dissertation can be stated as follows:

- To model the spatial distribution of different neuronal structures (synapses, spines and branching and terminal points of branching structures) from microscopy images. In particular, using replicated spatial point patterns taking advantage of the availability of several samples.

- To apply spatial statistics methods specifically designed for events that occur along networks to neuronal structures for which this constraint may be more appropriate than traditional spatial statistics techniques, e.g., spines along dendritic trees.

- To develop methods for performing spatial modeling in a 3D space, in cases where the existing methodology is available mainly for two dimensions (both in the whole space and along networks).

- To study the structural constraints that commonly limit network design problems and to propose an adequate representation to collect such constraints and optimize the network design.
- To develop optimization algorithms for the wiring optimality analysis of dendritic trees and axons in different types of neurons (pyramidal neurons and interneurons).

1.2 Document organization

The manuscript is grouped into five parts, each one divided into chapters as follows:

- **Part I. Introduction**

  This part introduces the dissertation and is the current part.

  - Chapter 1 details the research hypotheses and objectives and the manuscript organization.

- **Part II. Background**

  This part is divided into three chapters that introduce the theory and basic concepts used throughout the following parts of this manuscript. The state-of-the-art of the research areas related to this dissertation is discussed in these chapters.

  - Chapter 2 is an introduction to the theory and notation of spatial statistics, and more specifically, point process statistics. The chapter focuses on the basic concepts of spatial point processes, network spatial analysis and replicated point patterns. The theory presented in this chapter is essential to Part III of this dissertation.
  - Chapter 3 presents the NP-hard problem of finding the degree-constrained minimum spanning tree (DCMST) of a graph, and one of its most important applications: network design optimization. The chapter also introduces the evolutionary computation algorithms chosen in this work to solve network design optimization problems.
  - Chapter 4 provides a basic introduction to neuroscience and some biological concepts related to the applications developed in this dissertation. The most important neuroscience projects of the last decade are included. The chapter describes the two main types of neurons in the cerebral cortex and reviews some important studies about wiring economy principle.

- **Part III. Contributions to point process statistics**

  This part of the dissertation includes our proposals on point process statistics and the application to the modeling of the 3D spatial distribution of synapses, dendritic spines and branching and terminal points of dendritic arborizations.

- Chapter 5 presents a complete 3D spatial analysis in the context of replicated point patterns, illustrated with the study of the 3D distribution of synapses in the six layers of the cerebral cortex. Taking advantage that we have several samples from each layer, first, the intensity in each group (layer) is analyzed, examining whether there are significant differences between groups. Then, a model is fitted for each replicate (sample) independently. Finally, replicated point patterns are used to analyze differences and similarities between groups of replicates. To honestly estimate the goodness-of-fit of the resulting models, this chapter proposes a cross-validation technique for models within each group of replicates. The tool developed to process and analyze the 3D spatial distribution of synapses is also presented.
- Chapter 6 expands the existing 2D computational techniques for network spatial analysis to perform a spatial analysis along 3D networks. Spines can only lie on the dendritic shaft. Therefore, in this chapter we perform a 3D network spatial modeling of the spine distribution along the dendritic networks of pyramidal neurons in both basal and apical dendrites. To search for significant differences in the spine distribution of basal dendrites between different cells and between all the basal and apical dendrites, we use replicated point patterns together with a recent variant of Ripley's K function defined to work along networks.
- Chapter 7 introduces the average nearest neighbour ratio R, a measure that captures the degree of clustering of the points in a point cloud. To obtain an accurate estimate of R it is required to estimate the supporting volume of the point cloud as well as dealing with edge effects. Both problems are covered in this chapter. We illustrate the utility of measure R by analyzing the spatial distribution of branching points and terminal points of dendritic structures in both real and synthetic dendritic trees.

- **Part IV. Contributions to network design optimization**

  This part of the dissertation includes our proposals on network design optimization and the application to the study of neuronal wiring economy.

  - Chapter 8 deals with a new variant of the DCMST problem presented in Chapter 3, which consists of finding not only the degree- but also the role-constrained minimum spanning tree (DRCMST), i.e., we add constraints to restrict the role of the nodes in the tree to root, intermediate or leaf node. Furthermore, we do not limit the number of root nodes to one, thereby, generally, building a forest of DRCMSTs. This chapter also proposes a novel permutation-based representation to encode these forests. We use the jMetal framework in order to compare the performance of genetic algorithms (GAs) and estimation of distribution algorithms (EDAs) to solve DRCMST problems. To illustrate the applicability of our approach and that it is easy to add constraints depending on the specific char-

acteristics of the problem, we formulate the trans-European transport network consisting of nine transport corridors as a DRCMST problem.

- In Chapter 9 we hypothesize that both axonal and dendritic networks of individual cortical neurons optimize brain connectivity in terms of wiring length. We test this optimization problem using the DRCMST problem introduced in Chapter 8. In addition, we introduce a parallelization method for large DRCMST problems. The chapter also presents the software developed to analyze the optimality of the dendritic and axonal wiring of a 3D neuronal reconstruction.

- **Part V. Conclusions**

  This part summarizes and concludes the dissertation.

  - Chapter 10 summarizes the most important contributions obtained throughout this research and describes some lines of future work and open issues. The chapter also includes the list of publications and submissions produced in this dissertation.

Part II

BACKGROUND

Chapter 2

Point process statistics

2.1 Introduction

Spatial statistics analyzes data which have a spatial location, giving explicit consideration to spatial properties such as location, distance or spatial arrangement. Banerjee et al. [2004] classify the spatial data sets into three basic categories:

- Point-referenced data, often referred to as *geostatistical* data: the location index varies continuously over a region W, a fixed subset of $\mathbb{R}^D$.
- Areal data, often referred to as *lattice* data: W is again a fixed subset but now partitioned into a finite number of areal units with well-defined boundaries and observations are associated with the areal units.
- Point pattern data: unordered collection of n objects/points, where $n \geq 0$ is not fixed in advance, located in some specified region W. The points may have extra information called marks attached to them (marked point pattern).

A good introduction to the analysis of the above spatial data categories with R[1] software is provided by Bivand et al. [2013]. Here, we focus on analyzing the third type of spatial data through point process statistics. "Point process statistics is perhaps the most developed and beautiful branch of the modern field of spatial statistics" [Illian et al., 2008]. Its aim is to analyze the spatial structure of patterns formed by objects that are distributed in the plane or in space and that can be modeled as discrete points. These patterns are analyzed in many scientific disciplines [Baddeley et al., 2006]. Besides the classical fields of application, such as forestry (locations of trees), particle physics (locations of particles in material samples) and astronomy (galaxies or stars), today other fields such as ecology (animal nests), geography (positions of towns or facilities) or neuroscience (organization of neuronal structures such as synapses, spines...), also apply methods of point process statistics.

[1] http://www.r-project.org

Chapter outline

This chapter is an introduction to the notation and theory of point processes statistics. Section 2.2 provides the basic concepts in the theory of spatial point processes used in the next chapters. Section 2.3 describes the basic concepts in network spatial analysis, i.e., spatial statistics techniques to analyze events occurring on or along networks. Finally, Section 2.4 focuses on the theory of replicated spatial point patterns.

2.2 Spatial point processes

Within the field of spatial statistics, spatial point processes are mathematical models that describe the arrangement of elements randomly or irregularly distributed in space forming patterns. Illian et al. [2008] provide a good introduction to the topic; Daley and Vere-Jones [2003, 2008], Møller and Waagepetersen [2004] and Stoyan et al. [1995] include more complex mathematical introductions to the fundamental theory; and Baddeley [2010] and Baddeley et al. [2006] present a more applied text. Illian et al. [2008] and Baddeley [2010] represent the main sources of information used for the development of this section.

2.2.1 Fundamentals

A spatial point process $\boldsymbol{X}$ is a random set of points, with a random number of points and random locations in an abstract space S. We only consider scenarios in which $S \equiv \mathbb{R}^D, D \leq 3$. An observed point pattern $\boldsymbol{x}$ is a realization or sample of the point process $\boldsymbol{X}$, and is formally defined as an unordered set of points located in some known subset W (the sampling window) of S:

$$\boldsymbol{x} = \{x_1, ..., x_n\}, x_i \in W,$$

where $n \geq 0$ is not fixed in advance.

A covariate is any data that we treat as an explanatory variable. It is a spatial function $Z(u)$ defined at all spatial locations $u \in W$ (or at least at all x_i and some other locations). Examples of covariates are altitude, soil pH, etc. in a forestry problem.

Points in a spatial point pattern may have also extra information called 'marks' attached to them. Marks represent an 'attribute' of the point, like an additional coordinate. The mark attached to each point can be either continuous (for example, each point is a tree location and its attached marks are its height or/and diameter), categorical (for example, points which are classified into two or more different types, as case/control, color, etc.) or even more complex. Spatial point patterns with categorical marks are usually called *multitype point patterns*.

A marked point process of points in space S with marks belonging to a set M is mathematically defined as a point process in $S \times M$. A marked point pattern is an unordered set of points:

$$\boldsymbol{y} = \{(x_1, m_1), ..., (x_n, m_n)\},\ n \geq 0,\ x_i \in W,\ m_i \in M,$$

where x_i are the locations of the points and m_i their marks.

In marked point patterns there are different null hypotheses that can be tested. For example, we might be interested in analyzing if given the locations of the points, the marks are conditionally independent and identically distributed, if the sub-processes $\boldsymbol{Y}_m$ of points of each mark m, are independent point processes, etc. The main objective in spatial point process statistics is to characterize spatial point patterns in terms of the dependency of the objects on their position, covariates, marks and interaction with other objects.

In general, point patterns are expected to show some properties. Point patterns are assumed to fulfill the *simplicity* property, i.e., there is no more than one point lying on each location. There is another property which is essential: *stationarity*. A process is considered stationary if the overall point distribution is invariant to pattern translations. In practical terms, stationarity implies that the expectation of observing some point configuration is independent of the particular location, in or outside the sampling window.

The nature of the sampling area W is not such an obvious matter and it usually depends on the problem to solve. This can be naturally imposed by the physical limits of the environment in which objects exist (for example, the distribution of trees in a city park are limited by its boundaries). In those cases, the process is considered a *finite point process*. Alternatively, W may be arbitrarily chosen to reflect that patterns are part of a much larger (supposedly infinite) structure, in which points are distributed according to the same laws (for example, distribution of stars in a particular small region of the galaxy). In those cases, the process is known as *infinite point process*.

2.2.2 Summary characteristics

Spatial point process statistics provides the tools to characterize patterns in terms of the number and distribution of the elements. To do this, two aspects are mainly analyzed: intensity and inter-point interactions, closely related to distances between points.

2.2.2.1 Intensity

The most important numerical summary characteristic for a point process is the intensity or average density of points. Point intensity is the simplest distributional property and is similar to the use of the sample mean in classical statistics. The intensity function $\lambda(u)$, $u \in W$ of a point process $\boldsymbol{X}$, is the expected number of points per unit area or volume in the vicinity of u. $\lambda(u)$ is a first-order (mean) property of the point process, describing the expected density of points in any location, i.e., $\lambda(u)$ is proportional to the point density around a location u.

The intensity of a point process may be constant (uniform or homogeneous) or may vary from location to location (non-uniform or inhomogeneous):

- If it is constant, $\lambda(u) \equiv \lambda$, an unbiased estimator of the true intensity λ is $\hat{\lambda} = n/|W|$, where n is the number of points in the point pattern and $|W|$ is the area or volume of W.

- If it is suspected that the intensity may be inhomogeneous, $\lambda(u)$ can be estimated non-parametrically by techniques such as quadrat counting and kernel smoothing.

  In quadrat counting, W is divided into q quadrats (subregions) $B_1, ..., B_q$ of equal area, and the number of points falling in each quadrat n_j for $j = 1, ..., q$ is counted. $n_j/|B_j|$ are unbiased estimators of the corresponding intensity values in each quadrat. Counts should be approximately equal if there is a uniform density.

  In kernel smoothing, the usual kernel estimator of the intensity function is $\hat{\lambda}(u) = e(u)\sum_{i=1}^{n}\kappa(u - x_i)$, where $\kappa(u)$ is an arbitrary probability density (the kernel) and $e(u)$ is an edge effect bias correction (see below).

2.2.2.2 Distance methods

Apart from intensity analysis, one of the first steps often performed to explore the spatial distribution of a spatial pattern is to obtain the distances between points. Distance methods are the main classical techniques for investigating interpoint interaction (dependence). The dependence is assumed to be stronger for points which are close to each other. The following are usually considered ($||\cdot||$ denotes the Euclidean distance):

- Empty space distances: $d(u, \boldsymbol{x}) = \min_i\{||u - x_i|| : x_i \in \boldsymbol{x}\}$, distance from a fixed location u in W to the nearest point in a point pattern $\boldsymbol{x}$.
- Nearest neighbour distances: $t_i = \min_{j\neq i}||x_i - x_j||$, distance from each point x_i to its nearest neighbour in a point pattern.

When dealing with infinite point process, W introduces a sampling bias. Limiting observations to W implies that the observed empty space distance $d(u, \boldsymbol{x}) = d(u, W \cap \boldsymbol{X})$ to the nearest data point in W may be greater than the true distance $d(u, \boldsymbol{X})$ in the complete point process $\boldsymbol{X}$. For nearest neighbour distances $t_i = min_{j\neq i}||x_i - x_j||$ we encounter similar problems. Limiting observations to W implies that, in general, the observed nearest neighbour distances are larger than the true nearest neighbour distances of points in the complete point process $\boldsymbol{X}$ (Fig. 2.1).

Two well-known edge-correction techniques are the so-called toroidal edge correction and the border area edge correction [Yamada, 2009]. The first one maps a finite rectangular study region into a torus by identifying opposing edges (Fig. 2.2). The second one specifies a buffer zone inside the boundary of the study region and uses the inner part as the new study region. Points in the buffer zone are used only to take measurements of points (e.g., nearest neighbour distances) that are within the new study region and are further discarded (Fig. 2.3).

To develop formal statistical analysis, we typically use the empirical cumulative distribution function (ecdf) of the previously described distances. They are defined for stationary processes and their estimators are edge-corrected. As mentioned, some possibilities of edge correction are to consider the window W as a continuous medium (toroidal) or to discard the

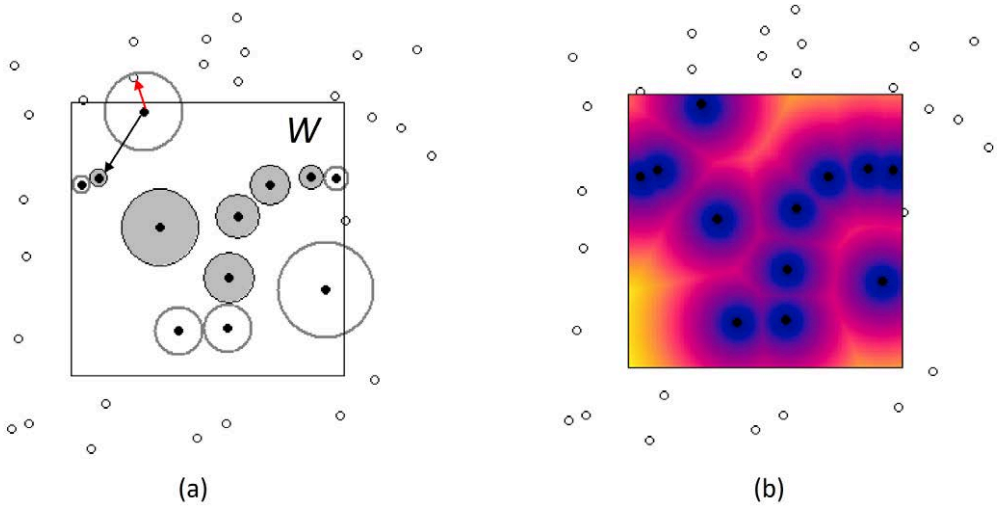


Figure 2.1: Example of edge effects. The point process $\boldsymbol{X}$ is observed only inside a region W, so the observed distances in W are, in general, larger than the true distances in the entire point process $\boldsymbol{X}$. (a) *Stienen diagram* obtained by drawing a circle around each data point of diameter equal to its nearest neighbour distance in W. Circles are grey if the observed nearest neighbour distance is observed without edge effects, i.e., it is shorter than the distance to the window boundary. Top left it is shown a point where the distance to its nearest neighbour in W (black arrow) is much larger than the nearest neighbour distance in the entire point process (red arrow). (b) Empty space distances for each point in W. As in (a), the observed empty space distances to the nearest data points in W are greater than the true distances in the complete point process

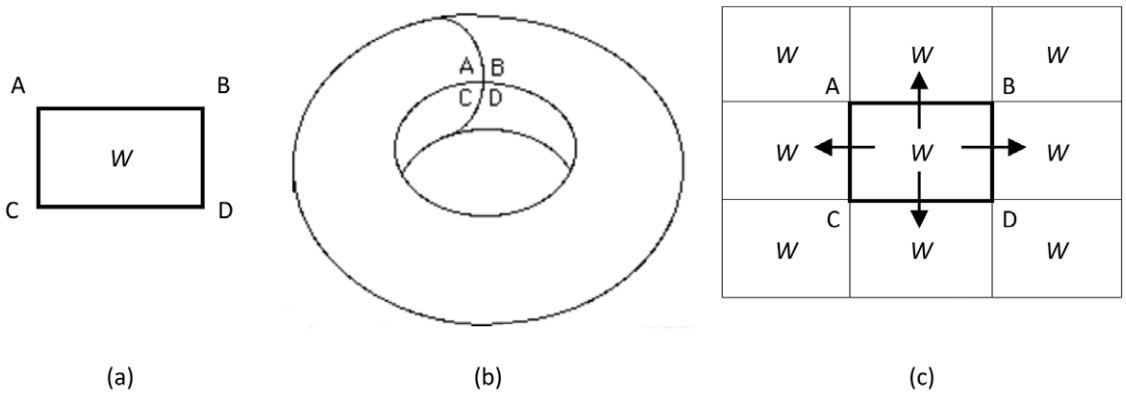


Figure 2.2: Toroidal edge correction. (a) Original study region W. (b) Study region W transformed into a torus by identifying opposing edges. (c) Another interpretation of the toroidal edge correction technique is that the study region W is replicated as to form a 3×3 grid of nine identical rectangles (also known as *periodic continuation*)

points in a buffer zone specified inside the study region (border area). Yet, both techniques have their drawbacks: the toroidal edge correction cannot be used for non-rectangular regions and the border area edge correction discards a large number of available points. The most

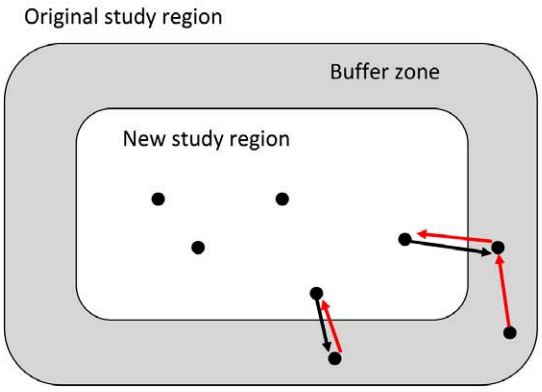


Figure 2.3: Border area edge correction. As an example, the application of the border area edge correction to nearest neighbour (NN) distances analysis is shown. Black arrows show some examples of NN distances to be measured. Red arrows show NN distances not to be measured

commonly used strategy is to use a weighted version of the ecdf and there are many types of edge correction weights.

The most common summary functions are:

- Empty-space function F

  Assuming $\boldsymbol{X}$ is stationary, we can define the cumulative distribution function of the empty space distance as

  $$F(d) = \mathbb{P}\{d(u, \boldsymbol{X}) \leq d\}, \tag{2.1}$$

  where u is an arbitrary location. If the process $\boldsymbol{X}$ is stationary, the F function does not depend on u.

  For the explained reasons, the ecdf of the observed empty space distances is a negatively biased estimator of $F(d)$. Estimators are typically weighted versions of the ecdf:

  $$\hat{F}(d) = \sum_{j=1}^{r} e(u_j, d)\mathbf{1}\{d(u_j, \boldsymbol{x}) \leq d\},$$

  defined on a grid of locations $u_j, j = 1, ..., r$, where $\mathbf{1}\{\cdot\}$ denotes the indicator function and $e(u, d)$ is an edge correction weight so that $\hat{F}(d)$ is unbiased and the influence of the window is reduced. Assuming that the point process is homogeneous, the estimation of the empty-space function defined by Eq. (2.1) can be unbiased and reasonably accurate.

- Nearest-neighbour distance function G

  Assuming $\boldsymbol{X}$ is stationary, we can define the cumulative distribution function of the nearest-neighbour distance for a typical point in the pattern as:

$$G(d) = \mathbb{P}\{d(u, \boldsymbol{X}\backslash\{u\}) \leq d | u \in \boldsymbol{X}\}, \tag{2.2}$$

where u is a typical point in the pattern, and $d(u, \boldsymbol{X}\backslash\{u\})$ is the shortest distance from u to the point pattern $\boldsymbol{X}$ excluding u itself. If the process is stationary then this definition does not depend on u. As before, the ecdf of the observed nearest-neighbour distances is a negatively biased estimator of $G(d)$. Many edge corrections, typically weighted versions of the ecdf, are available:

$$\hat{G}(d) = \sum_{i=1}^{n} e(x_i, d)\mathbf{1}\{t_i \leq d\},$$

where $e(x_i, d)$ is an edge correction weight so that $\hat{G}(d)$ is approximately unbiased.

- J function

  The J function [van Lieshout and Baddeley, 1996] is a useful combination of F and G functions:

$$J(d) = \frac{1 - G(d)}{1 - F(d)}, \tag{2.3}$$

  defined for all $d \geq 0$ such as $F(d) < 1$.

- Ripley's K function

  For a stationary process, Ripley's K function for a distance d [Ripley, 1977], $K(d)$, is defined as the expected number of other points of the process within a distance d of a typical point of the process divided by the intensity. Formally:

$$K(d) = \frac{1}{\lambda}\mathbb{E}[N(\boldsymbol{X} \cap b(u, d)\backslash\{u\}) | u \in \boldsymbol{X}], \tag{2.4}$$

  where $N(\boldsymbol{X} \cap B)$ counts the number of points from $\boldsymbol{X}$ falling in a region B and $b(u, d)$ is the neighbourhood of radius d centred on u.

  It has been shown that specifying $K(d)$ for all d is equivalent to specifying the variance of the number of points occurring in a subregion B for any B. This is why $K(d)$ is associated with second-order properties of the point process.

  Numerous estimators of $K(d)$ have been proposed, typically like:

$$\hat{K}(d) = \frac{1}{\hat{\lambda}^2|W|}\sum_i \sum_{j \neq i} e(x_i, x_j, d)\mathbf{1}\{||x_i - x_j|| \leq d\},$$

  where $e(x_i, x_j, d)$ is an edge correction weight and $|W|$ is the area or volume of W. The choice of the estimator does not seem to be very important, as long as some edge correction is applied [Baddeley, 2010].

- Besag's L function

  Besag's L function [Besag, 1977] is a commonly used transformation of the K function:

$$L(d) = \left(\frac{K(d)}{|W_D|}\right)^{1/D}, \tag{2.5}$$

  where D is the dimensionality and $|W_D|$ is the volume of the unit ball in $\mathbb{R}^D$. $|W_D| = \frac{\pi^{D/2}}{\Gamma(1+\frac{D}{2})}$ and $\Gamma(\cdot)$ is the gamma function ($|W_D|$=2 if D=1, $|W_D| = \pi$ if D=2, $|W_D| = 4\pi/3$ if D=3).

- Pair-correlation function g

  The pair correlation function for a distance d, $g(d)$, is another usual transformation of the Ripley's K function. Roughly speaking, it is the probability of observing a pair of points separated by a distance d, divided by the corresponding probability for a Poisson process (see below). This is a non-centred correlation, always non-negative:

$$g(d) = \frac{K'(d)}{D|W_D|d^{D-1}}, \tag{2.6}$$

  where $K'(d)$ is the derivative of K and $|W_D|$ is defined as before.

2.2.3 Point process models

2.2.3.1 Homogeneous Poisson process

The homogeneous spatial Poisson point process, also known as complete spatial randomness (CSR), is considered as the reference model in spatial point process statistics, since it represents a boundary condition between regular and clustered patterns. A random pattern, where a point is equally likely to occur at any location regardless of the locations of other points, follows a CSR process. The patterns known as regular patterns show repulsion, i.e., the distances between points are larger than expected in a random pattern of the same intensity. Furthermore, patterns where points tend to be closer than they should be for a given intensity are known as clustered patterns (Fig. 2.4).

The homogeneous Poisson process has constant intensity $\lambda(u) \equiv \lambda$. The basic properties of a CSR process with intensity $\lambda > 0$ are:

- P1: The number of points $N(\boldsymbol{X} \cap B)$ falling in any region B has a Poisson distribution.
- P2: The mean is given by $\lambda \cdot |B|$ points falling in B.
- P3: For any B_1, B_2 disjoint sets, then $N(\boldsymbol{X} \cap B_1)$ and $N(\boldsymbol{X} \cap B_2)$ are independent random variables.
- P4: Given n points inside region B, their locations are independent, identically and uniformly distributed in B.

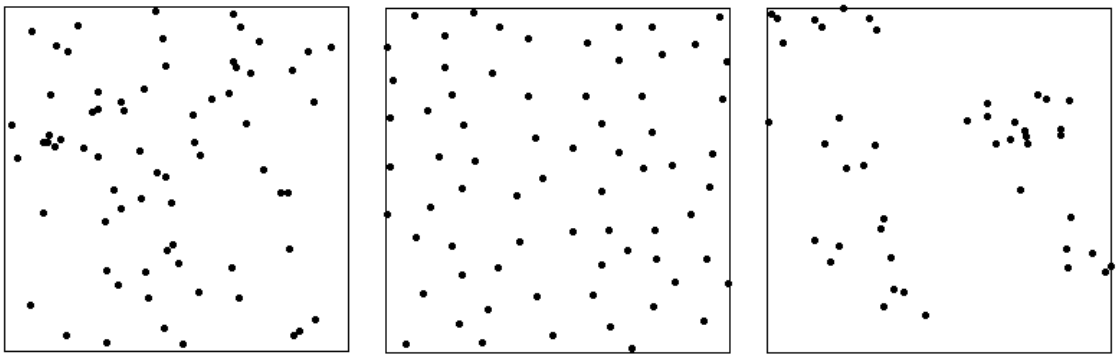


Figure 2.4: Three simulated point patterns: (left) random, (middle) regular, (right) clustered

P1 and P2 introduce the idea of an intensity λ representing the number of points per unit area or volume (constant but unknown). P4 represents the general concept of CSR, points uniformly distributed across the study area and independent of each other (with the same propensity to be found at any location regardless of those of other points).

The CSR process has the following expression for the summary functions presented in the previous section:

- Empty-space function: $d(u, \boldsymbol{X}) > d$ if and only if there are no points of $\boldsymbol{X}$ in the disc $b(u, d)$ of radius d centred on u. For a CSR process of intensity λ, the number of points falling in $b(u, d)$ follows a Poisson distribution with mean $\mu = \lambda|b(u, d)| = \lambda|W_D|d^D$, so the probability that there are no points in this region is $exp(-\mu)$ and for a Poisson process we have that:

$$F_{CSR}(d) = 1 - exp(-\lambda|W_D|d^D),$$

  where, as before, $|W_D|$ depends on the dimensionality D, and equals to 2 (D=1), π (D=2), $4\pi/3$ (D=3).

  Values $\hat{F}(d) < F_{CSR}(d)$ suggest a clustered pattern because empty space distances in the point pattern are larger than for a CSR process with the same intensity. Values $\hat{F}(d) > F_{CSR}(d)$ suggest a regularly spaced pattern.

- Nearest-neighbour distance function: for a CSR process of intensity λ it is known that:

$$G_{CSR}(d) = 1 - exp(-\lambda|W_D|d^D).$$

  $G_{CSR}(d)$ is identical to $F_{CSR}(d)$ since, due to independence, knowing that u is a point of $\boldsymbol{X}$ does not affect any other points of the process.

Unlike the F function, values $\hat{G}(d) < G_{CSR}(d)$ suggest a regular pattern because nearest neighbour distances in the point pattern are larger than those for a CSR process with the same intensity. Values $\hat{G}(d) > G_{CSR}(d)$ suggest a clustered spatial pattern.

- J function: $J_{CSR}(d) \equiv 1$ since $F_{CSR}(d) = G_{CSR}(d)$. Values $J(d) < 1$ suggest clustering, while values $J(d) > 1$ suggest regularity.

- Ripley's K function: for a CSR process, the knowledge that u is a point of $\boldsymbol{X}$ does not affect the other points of the process, so $\boldsymbol{X}\backslash\{u\}$ is conditionally a Poisson process. The expected number of points falling in $b(u,d)$ is $\lambda|b(u,d)| = \lambda|W_D|d^D$. Thus,

$$K_{CSR}(d) = |W_D|d^D$$

  regardless of the intensity. Values $\hat{K}(d) < K_{CSR}(d)$ suggest a regular pattern because we expect fewer points within a distance d of an arbitrary point than under a CSR process. Values $\hat{K}(d) > K_{CSR}(d)$ suggest clustering.

- Besag's L function: Eq. (2.5) converts the CSR K function to the straight line $L_{CSR}(d) = d$, making the plots much easier to assess visually. Values $\hat{L}(d) < d$ suggest regular spacing, while values $\hat{L}(d) > d$ suggest spatial clustering.

- Pair-correlation function: for a CSR process $g_{CSR}(d) \equiv 1$. Values $\hat{g}(d) < 1$ suggest regularity, while values $\hat{g}(d) > 1$ suggest clustering.

The use of summary functions for analyzing point patterns has become established. Neither function is considered to outperform the rest, although Ripley's K function and its derivations are often used extensively. It is important to note that the F, G and K functions are defined and estimated under the assumption that the point process $\boldsymbol{X}$ is stationary (homogeneous). If the process is not stationary, deviations between the empirical and theoretical functions (for example, between $\hat{K}(d)$ and $K_{CSR}(d)$) are not necessarily evidence of interpoint interaction, since they may also be attributable to variations in intensity. Other important considerations are that these summary functions do not completely characterize the process; further, as d increases, edge-effects are more important.

2.2.3.2 Inhomogeneous Poisson process

Spatial point processes methodology starts by testing the simplest hypothesis of CSR and if rejected it tries with inhomogeneous Poisson point processes which are a straightforward generalization of the homogeneous Poisson introducing inhomogeneity but no interaction, i.e., the intensity function $\lambda(u)$ depends on the position of the points in the region of interest.

The inhomogeneous Poisson process with intensity function $\lambda(u)$ modifies basic properties P2 and P4:

- P2': The mean is $\mathbb{E}[N(\boldsymbol{X} \cap B)] = \int_B \lambda(u)du$ points falling in B.

- P4': Given n points inside region B, their locations are independent and identically distributed, with density $f(u) = \lambda(u)/\int_B \lambda(u)du$ (we expect more (fewer) points in areas/volumes with higher (lower) values of $\lambda(u)$).

Points are independent of one another, but clusters appear in areas of high intensity ($\lambda(u)$ describes the expected density of points in any location).

Baddeley et al. [2000] proposed a modification of the K function that applies to inhomogeneous processes. The inhomogeneous K function is defined as:

$$K_{inhom}(d) = \mathbb{E}\left[\sum_{x_j \in \boldsymbol{X}} \frac{1}{\lambda(x_j)} \mathbf{1}\{0 < ||u - x_j|| \leq d\} \middle| u \in \boldsymbol{X}\right]. \tag{2.7}$$

If $\lambda(u)$ is the true intensity function of the point process $\boldsymbol{X}$, $\lambda(u)K(d)$ is the expected total 'weight' of all random points within a distance d of the point u, where the 'weight' of a point x_i is $1/\lambda(x_i)$. If the process is homogeneous, Eq. (2.7) reduces to Eq. (2.4).

The estimators of the K function can be extended to the inhomogeneous case:

$$\hat{K}_{inhom}(d) = \frac{1}{\sum_i 1/\hat{\lambda}(x_i)} \sum_i \sum_{j \neq i} e(x_i, x_j, d) \frac{\mathbf{1}\{||x_i - x_j|| \leq d\}}{\hat{\lambda}(x_i)\hat{\lambda}(x_j)}, \tag{2.8}$$

where $\hat{\lambda}(u)$ is an intensity function estimate and $e(x_i, x_j, d)$ is an edge correction weight.

For an inhomogeneous Poisson process with intensity function $\lambda(u)$:

$$K_{inhomCSR}(d) = K_{CSR}(d) = |W_D|d^D.$$

The inhomogeneous Besag's $L(d)$ function and the inhomogeneous pair correlation function $g(d)$, are defined analogously to the homogeneous case using Eq. (2.5) and Eq. (2.6), respectively, but substituting $K(d)$ for $K_{inhom}(d)$. For an inhomogeneous Poisson process, $L_{inhomCSR}(d) = d$ and $g_{inhomCSR}(d) \equiv 1$.

van Lieshout [2011] introduced the inhomogeneous versions of the F, G and J functions subject to special conditions (it is assumed that the 'k-point correlation functions' for all $k \geq 2$ are invariant under translation, see van Lieshout [2011] for definitions and details).

The inhomogeneous F function is defined as:

$$F_{inhom}(d) = 1 - \mathbb{E}\left[\prod_{x_i \in \boldsymbol{X} \cap b(u,d)} \left(1 - \frac{\lambda_{\min}}{\lambda(x_i)}\right)\right], \tag{2.9}$$

and the inhomogeneous G function as:

$$G_{inhom}(d) = 1 - \mathbb{E}\left[\prod_{x_i \in \boldsymbol{X} \cap b(u,d)} \left(1 - \frac{\lambda_{\min}}{\lambda(x_i)}\right) \middle| \boldsymbol{X} \text{ has a point at } u\right], \tag{2.10}$$

where u is an arbitrary location, $\lambda(u)$ is the true intensity function of the point process $\boldsymbol{X}$ and $\lambda(u) \geq \lambda_{min} > 0$ for all u.

The inhomogeneous J function is then defined as:

$$J_{inhom}(d) = \frac{1 - G_{inhom}(d)}{1 - F_{Finhom}(d)}. \tag{2.11}$$

For an inhomogeneous Poisson process, $J_{inhomCSR}(d) \equiv 1$.

2.2.3.3 Non-Poisson processes

A point process that is not Poisson is said to exhibit interaction or dependence between points. Briefly, some models derived from the Poisson process, that retain some of the tractable characteristics of the Poisson model, are described below:

- Poisson cluster processes: we start with a Poisson process $\boldsymbol{Y}$ of 'parent' points. Then, each point $y_i \in \boldsymbol{Y}$ gives rise to a finite set of 'offspring' points according to some stochastic mechanism. $\boldsymbol{X}$ comprising all the offspring points is a cluster process (parent points are not observed). Fig 2.5 shows an example.

- Cox processes: let $\Delta(u)$ be a random function with non-negative values. Conditional on Δ, let $\boldsymbol{X}$ be a Poisson process with intensity function Δ. Then $\boldsymbol{X}$ is a Cox process. The intensity function of $\boldsymbol{X}$ is $\lambda(u) = \mathbb{E}[\Delta(u)]$. Cox processes are always overdispersed relative to a Poisson process, i.e., the variance of the number of points falling in a region is greater than the mean.

- Thinned processes: thinning means deleting some of the points from a point pattern. If independent thinning is applied to a Poisson process, the resulting process of the retained points is again Poisson. To get a non-Poisson process we need some kind of dependent thinning mechanism.

- Sequential models: we start with an empty window, and the points are placed into the window one-at-a-time, according to some criterion. For example, in random sequential adsorption models [Evans, 1993], also known as simple sequential inhibition, each new point is generated uniformly in W and independently of preceding points. If the new point lies closer than a minimum distance from an existing point, it is rejected and another random point is generated. The minimum distance can be fixed or obtained according to a probability density function. The process terminates when a certain number of points are reached or no further points can be added.

Some of these processes have analytic expressions for some summary functions in terms of the model parameters. For example, suppose that the expression of the K function of the process with parameters θ, $K_\theta(d)$, is known; then, θ is estimated minimizing the discrepancy between $\hat{K}(d)$ estimated from data and $K_\theta(d)$ over some range $[a, b]$ (method of minimum contrast [Diggle, 2003]):

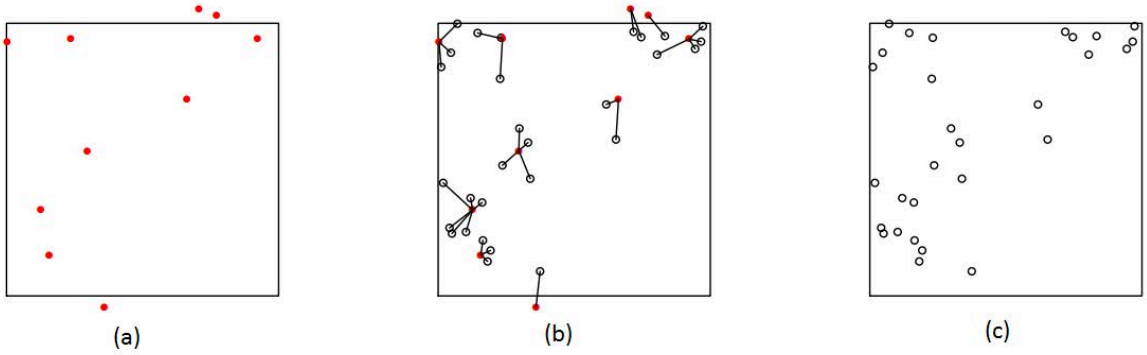


Figure 2.5: Example of Poisson cluster process. (a) Parents. (b) Clusters. (c) Offspring

$$D(\theta) = \int_a^b |\hat{K}(d)^q - K_\theta(d)^q|^p \mathrm{d}d,$$

where $0 \leq a < b$ and where $p, q > 0$ are exponents.

When the true summary function $T(d)$ is not known analytically, we can use Monte Carlo simulation to approximate it for any given θ. That is, we generate many realizations of the process with parameter θ, compute $\hat{T}(d)$ for each simulation and take the pointwise sample average.

2.2.4 Monte Carlo tests and envelopes

Beyond exploratory purposes, summary functions can be used as a basis for statistical inference. Because of random variability, never perfect agreement between the empirical and theoretical functions will be found, even with a completely random pattern. In point process statistics, tests are usually based on simulations. Thus, to test the null hypothesis that some specific model fits the data we can use Monte Carlo tests whose principle was originated by Dwass [1957] and Barnard [1963]. A Monte Carlo test is based on simulations from the null hypothesis and it can be applied to any point process model serving as a null hypothesis. Suppose that the reference curve is the summary function $T(d)$, then:

1. We generate M independent simulations from the null model of interest using the estimated parameters inside the study region W.

2. We compute the estimated T functions for each of these realizations, $\hat{T}_j(d)$ for $j = 1, ..., M$.

3. We obtain the pointwise minimum and maximum of these M simulated curves that define the envelope: $T_{min}(d) = \min_j \hat{T}_j(d)$ and $T_{max}(d) = \max_j \hat{T}_j(d)$.

4. We draw three curves $T_{min}(d)$, $\hat{T}(d)$ estimated from the dataset, and $T_{max}(d)$ (see Fig. 2.6).

5. For a fixed d chosen prior to simulation, the probability that $\hat{T}(d)$ lies outside the envelope for the simulated curves (type I error=α) is equal to $2/(1+M)$. Instead of the pointwise minimum and maximum, one could use the pointwise order statistics (the pointwise k-th smallest and k-th largest values) giving a test with significance level $\alpha = 2k/(1+M)$.

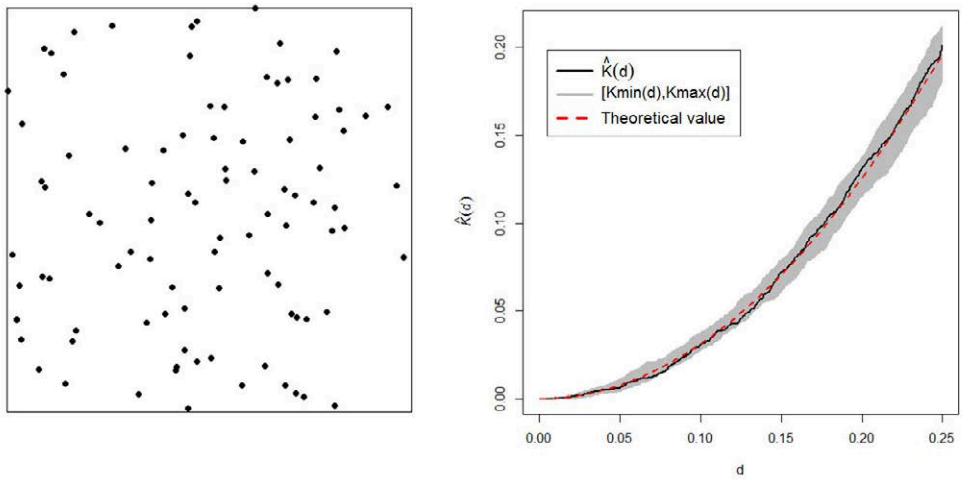


Figure 2.6: Pointwise envelope example for a random point pattern. (Left) Dataset of n=100 independent uniform random points in a square window $[0,1]\times[0,1]$. (Right) Envelopes from M=39 CSR simulations inside the same window and with the same intensity using Ripley's K function (grey), K function estimated from the dataset (black) and theoretical K function of that CSR process (red). This corresponds to a Monte Carlo test with significance level 2/(1+39)=0.05

Note that the previous pointwise envelopes specify the critical points for a Monte Carlo test [Ripley, 1981] but they are not 'confidence intervals' for the true value of the function. The test is constructed by choosing a fixed value of d, and rejecting the null hypothesis if the observed function value lies outside the envelope at this value of d. If we draw the pointwise envelope as presented above and check whether the empirical summary function $\hat{T}(d)$ is ever outside the envelope for all d, this is equivalent to choosing the value of d in a data-dependent manner, and the true significance level is higher, i.e., less 'significant'. To avoid this problem if we have no prior information about the range of spatial interaction, we can use *global envelopes*, also called *simultaneous critical envelopes*, as follows:

1. We generate M independent simulations from the null model of interest using the estimated parameters inside the study region W.

2. We compute the estimated T functions for each of these realizations, $\hat{T}_j(d)$ for $j = 1, ..., M$.

3. We obtain the theoretical value of the summary function, $T(d)$. If we are testing CSR, the theoretical value is known. Otherwise we generate a separate set of M' simulations,

compute the average of the estimated T functions of all these M' simulations and take this as an estimate of the theoretical value.

4. For each simulation of the first step, we compare its estimated $\hat{T}_j(d)$ function to the theoretical curve, and compute the maximum absolute difference between them (over the interval of d values in which we are interested). This gives a deviation value $w_j = max_d|\hat{T}_j(d) - T(d)|$ for each of the M simulations, and we take the largest of the deviation values (w_{max}). Then, the upper and lower limits that define the envelope are $T_{min}(d) = T(d) - w_{max}$ and $T_{max}(d) = T(d) + w_{max}$, i.e., global envelopes have constant width $2w_{max}$.

5. We draw three curves $T_{min}(d)$, $\hat{T}(d)$ estimated from the dataset, and $T_{max}(d)$.

6. The test rejects the null hypothesis if $\hat{T}(d)$ lies outside the envelope at any value of d in the analyzed interval. This test has significance level $\alpha = 1/(1 + M)$. As before, instead of the largest deviation, one could use the k-th largest deviation values giving a test with significance level $\alpha = k/(1 + M)$.

2.3 Network spatial analysis

Many types of real-world events are constrained by networks, such as stores located alongside streets, traffic accidents on roads, street crime sites, etc. These events are called *network events* (Fig. 2.7). Network spatial analysis refers to statistical and computational methods for analyzing events occurring on or along networks. Most of these methods have been developed by Okabe and collaborators [Okabe and Sugihara, 2012] and include techniques similar to the methods used in traditional spatial analysis but taking into account the network topology. The main difference from traditional spatial analysis using Euclidean distances is that network spatial analysis measures shortest path distances. Shortest path distances are much harder to calculate because they require network topology management. If traditional spatial analysis assuming a plane with Euclidean distances [Illian et al., 2008] is applied to network events, then we are likely to draw false conclusions due to short-range clustering (due to the concentration of events, for example, on a road) and/or long-range regularity (for example, due to the separation of different roads).

A linear network L in $\mathbb{R}^3$ is defined as the union of a finite collection of line segments l_i in $\mathbb{R}^3$ $(i = 1, ..., l)$, where a line segment with endpoints $u \in \mathbb{R}^3$ and $v \in \mathbb{R}^3$ is defined as $[u, v] = \{su + (1 - s)v : 0 \leq s \leq 1\}$. The shortest path distance between two points u and v located in L, $d_L(u, v)$, is the minimum length of all paths along the network from u to v. If there are no paths from u to v (the network is not connected), then $d_L(u, v) = \infty$. A network that has no cycles is called acyclic network or tree.

Let $\boldsymbol{X}$ be a point process on a linear network L. A realization of $\boldsymbol{X}$ is a finite set $\boldsymbol{x} = \{x_1, ..., x_n\}$ of distinct points x_i located in L, where $n \geq 0$ is not fixed in advance. Each point x_i is called *network event*. The intensity function $\lambda(u)$, $u \in L$ of a point process $\boldsymbol{X}$ on a linear network L, is the expected number of points per unit length in the network in the

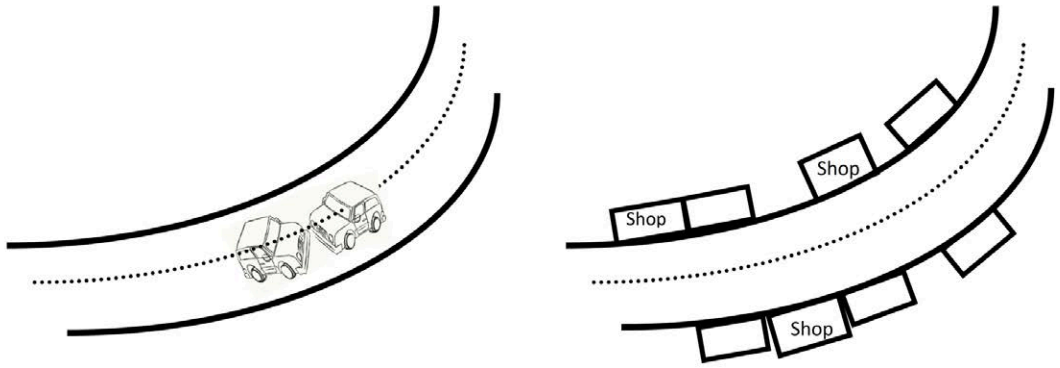


Figure 2.7: Examples of network events, i.e., events that occur on a network (car accidents on a road, left) or events that occur along a network (shops located along a street, right)

vicinity of u. The intensity of the homogeneous Poisson process or CSR is constant $\lambda(u) \equiv \lambda$, where $\hat{\lambda} = n/|L|$ is an unbiased estimator of the intensity, n being the number of points in $\boldsymbol{x}$ and $|L|$ being the total length of all line segments in L. The general intensity function of a point process $\boldsymbol{X}$ on a linear network can be estimated using kernel smoothing estimators [Okabe et al., 2009].

As previously explained, one of the most commonly used summary functions in spatial point pattern analysis is Ripley's K function [Ripley, 1977]. Let L be a linear network with events at locations $x_1, ..., x_n$. Okabe and Yamada [2001] developed a network K function analogous to Ripley's K function, where the shortest path distances in the network $d_L(x_i, x_j)$ replace the Euclidean distances. This function is estimated as:

$$\hat{K}_{net}(d) = \frac{|L|}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} 1\{d_L(x_i, x_j) \leq d\}. \tag{2.12}$$

As shown in Ang et al. [2012], the estimated value of the network K function depends on the geometry of the network. Therefore, the network K functions of different networks are not directly comparable.

The solution proposed in Ang et al. [2012] was a geometrically corrected version of the network K function, K_L, that compensated for the geometry of the network. The empirical estimator of K_L is intrinsically corrected for edge effects, and its variance is approximately stabilized. The geometrically corrected empirical K function for a distance d is defined as:

$$\hat{K}_L(d) = \frac{|L|}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} \frac{1\{d_L(x_i, x_j) \leq d\}}{m(x_i, d_L(x_i, x_j))} \tag{2.13}$$

for $0 \leq d \leq R$, where $m(u,t) = \#\{v \in L : d_L(u,v) = t\}$ is the number of points located in L lying at the exact distance t from the point u measured by the shortest path, and $R = \sup\{t : m(u,t) > 0 \text{ for all } u \in L\}$ is the *circumradius* of the network, i.e., the radius of the smallest disc that contains the entire network, as explained in Ang et al. [2012].

For a homogeneous Poisson process on L, $K_L(d) = d$ for all $0 \leq d \leq R$. This provides

a simple benchmark for completely spatial random point patterns on a linear network and also allows comparison between geometrically corrected K functions obtained from different point patterns in different networks.

For non-constant intensity spatial point processes, Baddeley et al. [2000] introduced the inhomogeneous version of Ripley's K function. The contribution to the inhomogeneous K function of each pair of points x_i and x_j is weighted by $1/(\lambda(x_i)\lambda(x_j))$ (Eq. (2.8)). Consequently, the properties of the inhomogeneous K function are very similar to the original version of Ripley's K function. For a spatial point process on a linear network, Ang et al. [2012] similarly defined the inhomogeneous network K function, estimated as:

$$\hat{K}_{LI}(d) = \frac{1}{\sum_i 1/\hat{\lambda}(x_i)} \sum_{i=1}^{n} \sum_{j \neq i} \frac{1\{d_L(x_i, x_j) \leq d\}}{\hat{\lambda}(x_i)\hat{\lambda}(x_j) m(x_i, d_L(x_i, x_j))}, \tag{2.14}$$

where $\hat{\lambda}(\,\cdot\,)$ is the estimated intensity function.

2.4 Replicated spatial point patterns

Replicated spatial point patterns are a particular situation in the spatial point processes field where different patterns are considered as instances of the same process and are said to form a group. Historically, spatial point processes have been more related to applications in which data collection tended to be costly (e.g. forestry). For this reason, the study of several independent samples as realizations of the same process was not usually considered. Recently, the field of replicated point patterns is growing strongly since technological advances have simplified sampling, particularly 3D sampling.

Pooling or combining several datasets into a single dataset is a common statistical procedure and it may also be applied to summary functions [Baddeley et al., 2015]. In general, a natural pooled estimate is the *ratio-of-sums* estimator, i.e., the weighted average of individual ratios with proportional weights to each dataset.

Let g be the number of different experimental groups. In group i $(i = 1, ..., g)$, we observe m_i point patterns that can be regarded as independent replicates within this group. Let n_{ij} $(j = 1, ..., m_i)$ be the number of points for the jth pattern within the ith group $\boldsymbol{x}_{ij}$ $(i = 1, ..., g)$. Given an estimate of the summary function T of pattern $\boldsymbol{x}_{ij}$, $\hat{T}_{ij}(d)$, the estimated mean function for each group i is defined as

$$\bar{T}_i(d) = \frac{\sum_{j=1}^{m_i} w_{ij} \hat{T}_{ij}(d)}{\sum_{j=1}^{m_i} w_{ij}}, \; i = 1, ..., g. \tag{2.15}$$

Different weights w_{ij} have been proposed in the literature for function aggregation. Myllymäki et al. [2012] chose to use $w_{ij} = n_{ij}^2$ to aggregate K functions together with linear mixed models to investigate the spatial structure of epidermal nerve fiber. Jafari-Mamaghani et al. [2010] used $w_{ij} = n_{ij}$ to study the 3D distribution of pyramidal neurons in the mouse barrel

cortex. The weight $w_{ij} = n_{ij}$ was also recommended by Diggle [2003]. See Pawlas [2011] for a review.

The main objective in replicated point pattern analysis is to test whether the differences between groups are statistically significant. The null hypothesis of no difference between groups establishes that the observed point patterns are independent and identically distributed patterns. Two proposed tests for testing differences between groups are detailed below.

- Diggle test

  Diggle and collaborators proposed a bootstrap Monte Carlo test for difference between group means of independent replicates of empirical K functions [Diggle et al., 1991, 2000]. The idea is to generate bootstrap samples, $\hat{K}^*_{ij}$, from the original sample $\hat{K}_{ij}$ through the following steps.

  Residual functions $\hat{R}_{ij}(d)$ are obtained from the empirical summary functions $\hat{K}_{ij}(d)$:

  $$\hat{R}_{ij}(d) = n_{ij}^{1/2}(\hat{K}_{ij}(d) - \bar{K}_i(d)),$$

  where $\bar{K}_i(d) = (1/n_i)\sum_{j=1}^{m_i} n_{ij}\hat{K}_{ij}(d)$ is the mean in group i, n_{ij} is the number of points in the pattern $\boldsymbol{x}_{ij}$ and $n_i = \sum_{j=1}^{m_i} n_{ij}$ is the number of points in group i.

  Then, the bootstrap samples are calculated as:

  $$\hat{K}^*_{ij}(d) = \bar{K}(d) + n_{ij}^{-1/2}\hat{R}^*_{ij}(d),$$

  where $\hat{R}^*_{ij}$ is a random sample from the set of rescaled residual functions $\hat{R}_{ij}$, $\bar{K}(d) = (1/n)\sum_{i=1}^{g} n_i\bar{K}_i(d)$ is the overall mean and $n = \sum_{i=1}^{g} n_i$.

  For the choice of the weight $n_{ij}^{1/2}$ the authors used the assumption that the variance of Ripley's K function is inversely proportional to the number of points in the point pattern. Under this assumption, the residuals functions $\hat{R}_{ij}$ are approximately identically distributed and the distribution of the bootstrap samples $\hat{K}^*_{ij}$ approximates the distribution of $\hat{K}_{ij}$ under the null hypothesis of no difference between groups.

  Diggle et al. [1991] proposed the following test statistic on an interval $[d_0, d_1]$ (they advise drawing the bootstrap residuals $\hat{R}^*_{ij}$ with replacement from all $\hat{R}_{ij}$):

  $$D = \sum_{i=i}^{g}\int_{d_0}^{d_1}\left(\sqrt{\bar{K}_i(d)} - \sqrt{\bar{K}(d)}\right)^2 \mathrm{d}d. \tag{2.16}$$

  Later, Diggle et al. [2000] changed the test statistic in favor of sampling without replacement, i.e., performing a permutation test with:

  $$D = \sum_{i=i}^{g} n_i \int_{d_0}^{d_1} \frac{1}{d^2}\left(\bar{K}_i(d) - \bar{K}(d)\right)^2 \mathrm{d}d. \tag{2.17}$$

To determine the p-value, the observed value of the test statistic is ranked among the corresponding bootstrap values of the test statistic.

- Studentized permutation test

  The idea of using permutations to test for differences between groups comes from Fisher [1935] and Pitman [1937]. Hahn [2012] proposed the studentized permutation test. Suppose we have g groups of point patterns, with $m_1, ..., m_g$ point patterns each. Hahn [2012] proposed comparing the means of groups corresponding to estimates $\hat{T}_{ij}(d)$, where T is the summary function of pattern $\boldsymbol{x}_{ij}$ in an interval $[d_0, d_1]$ for d, with the test statistic

$$H = \sum_{1 \le i \le j \le g} \int_{d_0}^{d_1} \frac{(\bar{T}_i(d) - \bar{T}_j(d))^2}{\frac{1}{m_i} s_i^2(d) + \frac{1}{m_j} s_j^2(d)} \mathrm{d}d, \tag{2.18}$$

  where $\bar{T}_i(d) = (1/m_i) \sum_{j=1}^{m_i} \hat{T}_{ij}(d)$ is the mean in group i and

$$s_i^2(d) = \frac{1}{m_i - 1} \sum_{j=1}^{m_i} (\hat{T}_{ij}(d) - \bar{T}_i(d))^2$$

  are the estimated within-group variances of the estimates for a distance d.

  If we use a summary function that stabilizes the variance, the denominator of Eq. (2.18) can be improved by pooling over all values of d, and the test statistic would be:

$$H = \sum_{1 \le i \le j \le g} \int_{d_0}^{d_1} \frac{(\bar{T}_i(d) - \bar{T}_j(d))^2}{\frac{1}{m_i} \bar{s_i^2} + \frac{1}{m_j} \bar{s_j^2}} \mathrm{d}d, \tag{2.19}$$

  where

$$\bar{s_i^2} = \frac{1}{d_1 - d_0} \int_{d_0}^{d_1} \frac{1}{m_i - 1} \sum_{j=1}^{m_i} (\hat{T}_{ij}(d) - \bar{T}_i(d))^2 \mathrm{d}d.$$

  The test is performed by calculating the H statistic for the observed data and for a large number of random permutations of the set of point patterns, and then computing the p-value ranking the observed value of the test statistic among the corresponding permutation values of the test statistic.

Chapter 3

Network design optimization

3.1 Introduction

In network design problems such as transportation, telecommunications and distribution systems a basic topological structure is the spanning tree. Well-known-classical algorithms exist for building a minimum spanning tree (MST) [Kruskal, 1956, Prim, 1957] but, in practice, a more realistic representation for network design is a degree-constrained minimum spanning tree (DCMST), i.e., a MST with constraints on the number of edges incident to each node. Fig. 3.1(a) shows an example of the MST of a graph with 7 nodes. The cost of each connection is indicated on the edges that join the nodes. Fig. 3.1(b) shows the DCMST of the same graph, where each node can have a maximum of three incident edges.

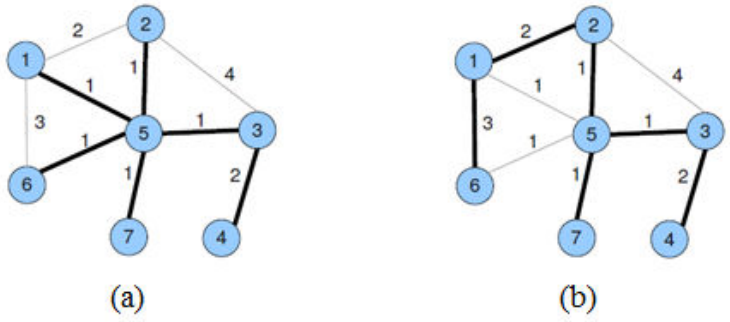


Figure 3.1: Example of MST (a) and DCMST (b) of a graph. Each edge has its cost annotated. In (b) the maximum degree of each node is fixed to 3

The DCMST problem can be applied in a transportation system, such as wires, pipes or canals, where the length of the connections of m nodes should be minimum. The handling capacity of each node imposes a constraint on the number of edges that can be connected to that node. In communication networks, the degree constraint limits network vulnerability if a node fails. The DCMST problem could also be applied to the design of a computer network

or a road network with a maximum number of roads at a crossing [Krishnamoorthy et al., 2001].

Chapter outline

Section 3.2 formally describes the DCMST problem, whose resolution requires heuristic techniques due to its complexity. Section 3.3 introduces the evolutionary computation algorithms chosen for this purpose and the framework used to compare their performance.

3.2 Degree-constrained minimum spanning tree

A DCMST is a minimum spanning tree where we assume that there is a degree constraint on each node such that, at node v, its degree value $\deg(v)$ (i.e., its number of incident edges) is at most a given value $d_v \in \mathbb{N}$.

Formally, let $G = (V, E)$ be an undirected complete graph with a set of vertices (nodes) V and a set of edges E. A spanning tree of G is a subgraph $T = (V, E_T)$, $E_T \subset E$ that contains all vertices in V which it connects with exactly $|V| - 1$ edges. Let $c_{uv} \geq 0$ be the cost of each edge $(u, v) \in E$, $u, v \in V$. The DCMST problem consists of finding a minimum spanning tree $T^* = (V, E_{T^*}), E_{T^*} \subset E$ such that

$$T^* = \operatorname*{argmin}_T \sum_{(u,v)\in E_T} c_{uv},$$

subject to

$$\deg(v) \leq d_v \quad \text{for all } v \in V.$$

The DCMST problem is NP-hard (this can be shown by reduction of the Hamiltonian path problem, Garey and Johnson [1979]). The problem of finding the DCMST of a graph, and particularly finding a good representation of the tree, has been widely studied in the literature. For example, Knowles et al. [2000] introduce the randomized primal method, a novel tree construction algorithm for stochastic iterative search techniques. This method builds low-cost degree-constrained trees. Krishnamoorthy et al. [2001] compare three heuristics (simulated annealing, a genetic algorithm and a method based on problem space search) and two exact algorithms (Lagrangian relaxation and branch-and-bound) for the DCMST problem. Further, they propose alternative tree representations to facilitate the genetic algorithm neighbourhood searches. Raidl and Julstrom [2003] propose representing spanning trees for network design problems directly as sets of their edges. They demonstrate the usefulness of their encoding for the DCMST problem. Soak et al. [2004] develop another effective encoding method for use by black-box optimization methods when addressing tree-based combinatorial problems.

The above representations are based on the construction of a single tree. Some more recent studies consider building a forest. This extension is not straightforward. In Delbem

et al. [2004], the proposed forest representation, named node-depth encoding, is composed of the union of the encodings of all trees of the forest. The union is implemented using an array of pointers, where each pointer indicates a tree consisting of linear lists containing the tree nodes and their depths. The proposed approach is evaluated for the DCMST problem. Some years later, Delbem et al. [2012] propose a new data structure to generate and manipulate a set of spanning forests, called node-depth-degree representation. This structure improves the average running time of their previous node-depth encoding (the forest is again composed of the union of the trees). Also working with a group of trees, Czajko and Wojciechowski [2009] take a different approach. They formulate the hop- and degree-constrained minimum spanning forest problem with minimization of the number of trees (this problem is defined as part of an access network topology design).

3.3 Evolutionary computation techniques

Evolutionary computation is a branch of artificial intelligence inspired by biological evolution mechanisms to solve optimization problems. Evolutionary algorithms can be considered as metaheuristic or stochastic techniques of global optimization, distinguished by the use of a population of candidate solutions that are evolved through iterative processes inspired by Charles Darwin principles [Darwin, 1859].

We used genetic algorithms (GAs) [Holland, 1975] and estimation of distribution algorithms (EDAs) [Larrañaga and Lozano, 2002] to solve and compare a variety of network design optimization problems. We opted for two GAs and two EDAs. Specifically, we used the generational genetic algorithm (gGA) [Cobb and Grefenstette, 1993], the steady-state genetic algorithm (ssGA) [Syswerda, 1991], the node histogram based sampling algorithm (NHBSA) [Tsutsui, 2006] and the Mallows kernel EDA (MKEDA) [Ceberio et al., 2015].

The gGA and the ssGA are two of the best-known families of GAs. Alg. 3.1 shows the pseudo-code of a generic GA. Briefly, these algorithms evolve a population of individuals until a specified stop condition is met (line 4). The main steps are as follows: select the parents to be crossed (line 6), recombine them (line 7), and mutate the resulting children (line 8). Usually, the fittest individuals survive and the least fit individuals are discarded. In the particular case of the gGA, in each generation, two parents are selected from the whole population and recombined, generating two children that are then mutated. The resulting children are placed in an auxiliary population which will replace the current population when the auxiliary population is completely filled. In our case, the size of the auxiliary population is the same as the size of the current one. Note that this strategy could remove the best solution in the current population. By contrast, the ssGA is an elitist strategy because the best solution is always retained. In each generation of the ssGA, one of the resulting children is mutated and evaluated and then inserted back into the population if the new individual is better than the worst individual in the current population.

EDAs are stochastic optimization methods that guide the search for the optimum with the estimation of probabilistic models of promising individuals and sampling from these models.

Algorithm 3.1 Generic GA pseudo-code

```
1: t ⟵ 0
2: P(t) ⟵ GenerateInitialPopulation()
3: Evaluate(P(t))
4: while ! StopCondition() do
5:    t ⟵ t + 1
6:    P_p ⟵ SelectParents(P(t − 1))
7:    P_c ⟵ Crossover(P_p,CrossProb)
8:    Mutate(P_c, MutProb)
9:    Evaluate(P_c)
10:   P_t ⟵ BuildNextGeneration(P_c, P(t − 1))
11: end while
```

Both EDAs and GAs are heuristic optimization algorithms based on the stochastic nature of the search itself and both are based on evolving populations. However, while in GAs the population evolution is carried out by crossover and mutation operators, in EDAs the new population of individuals is sampled from a probability distribution. While in EDAs the interrelations between the variables representing the individuals are expressed explicitly through the probabilistic model associated with the selected individuals in each generation, in GAs these interrelationships are taken into account implicitly.

Alg. 3.2 shows the pseudo-code of a generic EDA. We start from an initial population of M individuals (line 2). In the main loop, until the stop condition is met (line 4), a number N ($N \leq M$) of individuals are selected (usually the individuals with best fitness) (line 6). Next, the probabilistic model of the selected individuals is estimated (line 7), and the new population of size M is generated by sampling the learned model (line 8). With regard to the EDAs used in this work, the NHBSA models frequencies at each absolute position in the permutation. The other EDA, the MKEDA, calculates as many Mallows models (distance-based exponential probability models over permutation spaces [Mallows, 1957]) as individuals in the selected population of solutions. Particularly, we analyzed the MKEDA under the Cayley distance [Irurozki et al., 2016] because the algorithm performs better with this distance [Ceberio et al., 2015].

Algorithm 3.2 Generic EDA pseudo-code

```
1: t ⟵ 0
2: P(t) ⟵ GenerateInitialPopulation(M)
3: Evaluate(P(t))
4: while ! StopCondition() do
5:    t ⟵ t + 1
6:    P_sel(t − 1) ⟵ Select(N, P(t − 1))
7:    Model(t) ⟵ EstimateModel(P_sel(t − 1))
8:    P(t) ⟵ Sample(M, Model(t))
9:    Evaluate(P(t))
10: end while
```

In order to compare the performance of these algorithms we used the jMetal framework [Durillo and Nebro, 2011, Durillo et al., 2010]. jMetal stands for Metaheuristic Algorithms in Java, and it is an object-oriented Java-based framework for single and multi-objective optimization with a variety of metaheuristics techniques. It is licensed under the GNU Lesser General Public License[1] and can be freely obtained from http://jmetal.sourceforge.net. jMetal provides a rich set of classes that can be used as building blocks of metaheuristics; making use of code reuse, algorithms share the same basic components, which facilitates not only the development of new techniques but also carry out different types of studies. Fig. 3.2 shows an Unified Modeling Language (UML) diagram describing the jMetal architecture with the main components and their relationships. The diagram is a simplified version in order to make it understandable. The basic architecture of jMetal is based on an *Algorithm* that solves a *Problem* using one or more solution sets (*SolutionSet*) and a set of *Operator* objects. jMetal uses a generic terminology to name classes in order to make them useful for any metaheuristic. In the context of evolutionary algorithms, populations and individuals correspond to *SolutionSet* and *Solution* jMetal classes, respectively. jMetal already contained the GAs in which we were interested and we plugged the implementation of EDAs into jMetal framework.

[1]LGPL License: http://creativecommons.org/licenses/LGPL/2.1/

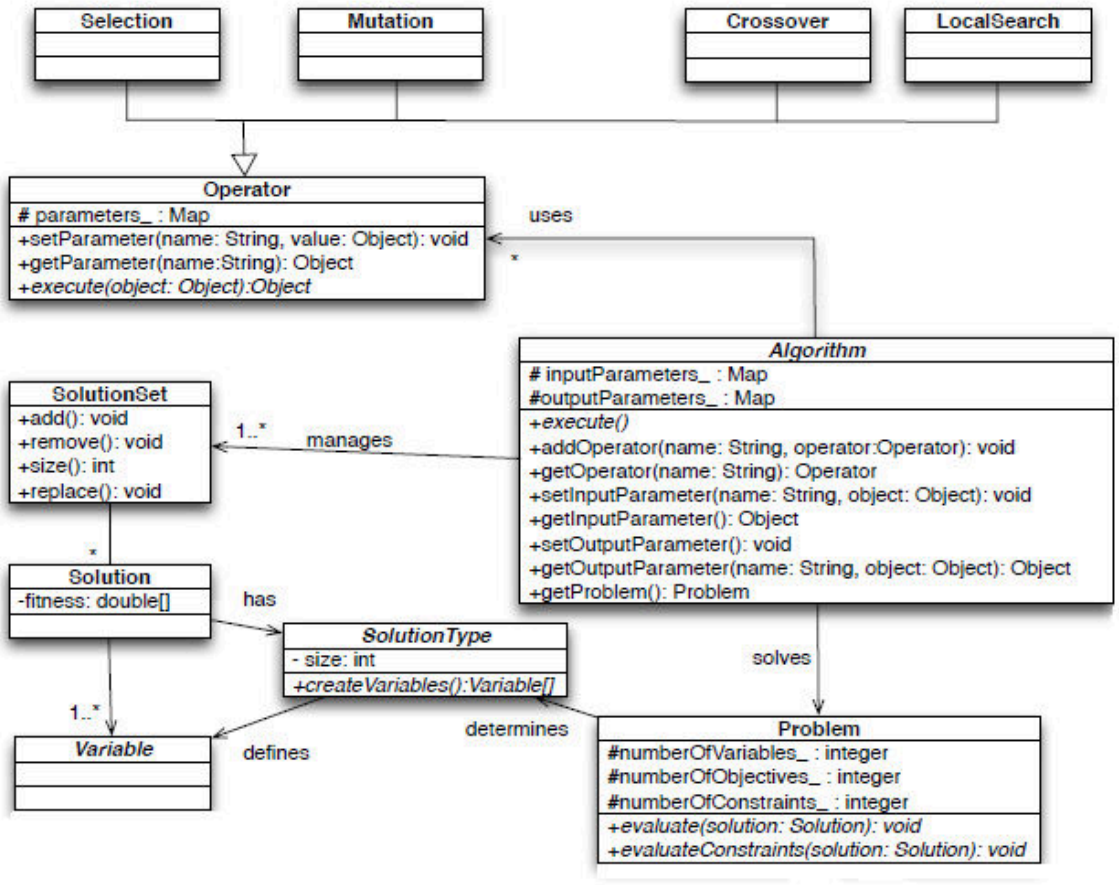


Figure 3.2: UML diagram including the main classes of jMetal and their relationships. Diagram extracted from the jMetal user manual available at http://jmetal.sourceforge.net

Chapter 4

Neuroscience

4.1 Introduction

Neuroscience can be defined as the scientific study of the nervous system. The biological study of the brain is a multidisciplinary area that includes many levels of study, from the purely molecular to the specifically behavioral and cognitive. Neuroscience has had a great development in the last decades and it has become one of the most important biomedical disciplines today. This is partly due, among other factors, to the growing impact of nervous system diseases in Western societies. The increase in patients suffering from stroke, neurodegenerative diseases (such as Alzheimer's disease or Parkinson's disease) or psychiatric disorders (such as depression or schizophrenia), have caused an increase in material resources devoted to the research of the brain and its disorders.

One of the fundamental objectives of neuroscience is to understand the biological mechanisms responsible for human mental activity. The study of the brain and, in particular, of the cerebral cortex (nervous tissue that covers the surface of the cerebral hemispheres) is one of the greatest challenges of science. It is believed that the cerebral cortex is the part of the brain responsible for conscious thinking and that cerebral cortex activity is related to the ability to perform extremely complex tasks that distinguish humans from other mammals. Anatomically, in the cerebral cortex there is a stratification in six horizontal layers, labeled from the most superficial (layer I) to the innermost (layer VI). Each layer is characterized by the predominance of a type of nerve cell and the axon destination of these cells within the brain. The hypothesis of columnar organization is currently the most widely adopted to explain cortical processing of information [DeFelipe et al., 2012]. According to this hypothesis, neurons are arranged in structures called cortical columns, considered the basic functional unit of the brain (Fig. 4.1).

More than a century ago, Santiago Ramón y Cajal suggested interpreting the brain by observing the morphology of individual neurons. The development of this idea is materialized in this dissertation through the use of spatial point processes methods (whose basic concepts have been described in Chapter 2) and network design optimization techniques (introduced in Chapter 3). We study and develop methods to analyze the spatial distribution of different

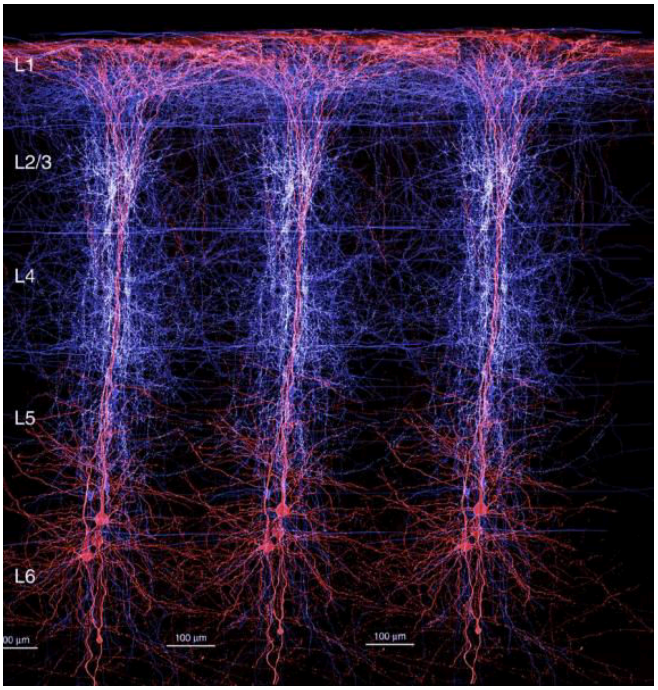


Figure 4.1: Cortical column development in mammals. Source: Blue Brain Project, École Polytechnique Fédérale de Lausanne

neuronal structures (synapses, spines, branching points, etc.), with the aim of obtaining useful results in the field of neuroscience. We also develop optimization methods for wiring analysis of different neuronal arborizations (basal and apical dendrites, and axons) in different types of neurons.

Chapter outline

This chapter provides a basic introduction to some biological concepts, useful to understand the applications developed in the next chapters. Section 4.2 presents the neuron doctrine introduced by Ramón y Cajal in which modern neuroscience continues to be supported, as well as the most important neuroscience projects of the last decade. Section 4.3 describes the two main groups of neurons in the cerebral cortex, pyramidal neurons and interneurons, both involved in the studies carried out in this dissertation. Section 4.4 details the wiring economy principle and some of the many recent studies related to this topic. In this thesis we study the existence of optimal neuronal wiring in both pyramidal neurons and interneurons.

4.2 Neuron doctrine and modern neuroscience

At the end of the XIX century, cells were known to be autonomous entities that related to the rest of body's cells. However, it was thought that this did not occur in the brain, where neurons would form a continuous network. In 1888, Santiago Ramón y Cajal (1852-1934) was able to demonstrate that neurons were also independent cells, what has been called

the *neuron doctrine* [Ramón y Cajal, 1888]. He was able to reach this conclusion, which represented an authentic paradigm shift in the neurological science of the time, thanks to the help of a microscope and a staining technique designed by the Italian researcher Camillo Golgi (1843-1926). By their technique and the discovery, respectively, Golgi and Ramón y Cajal shared the Nobel Prize of Medicine of 1906.

Ramón y Cajal made numerous important contributions to the knowledge of the structure and function of the nervous system in general, and the microanatomy of the cerebral cortex in particular. His research contributed decisively in the creation of the scientific atmosphere necessary for the birth of modern neuroscience and his ideas are still present today. Neuron doctrine states that the nervous system is composed of independent cells, neurons, whose interaction, mediated by synapses, leads to the appearance of complex responses.

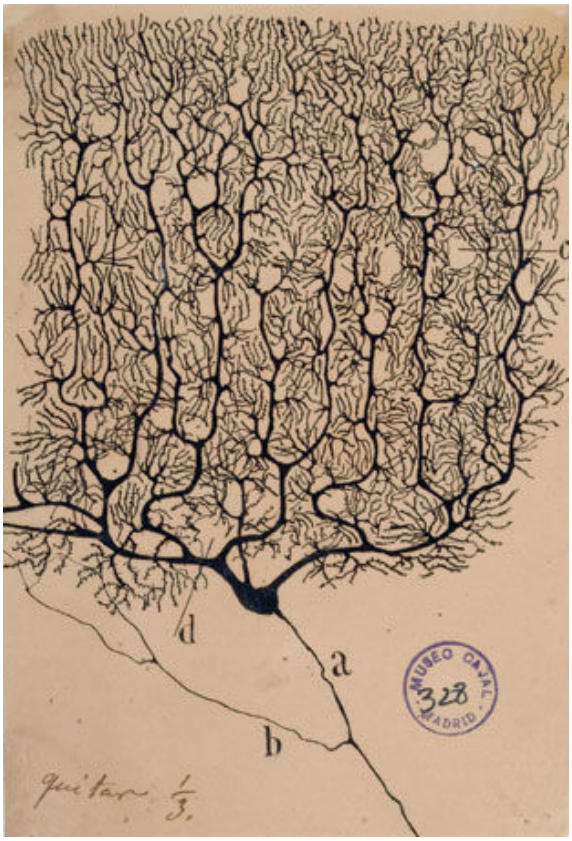


Figure 4.2: Drawing of Purkinje cell in the human cerebellum by Santiago Ramón y Cajal. He depicted the thick dendritic forest of the neuron (c and d) that branches off from the cell body, the axon (a), and the collateral axon (b). Source: Instituto Cajal (CSIC)

In general, the structure of a neuron is composed of the following parts: soma, axon and dendrites (Fig. 4.2):

- Soma: is the body of the neuron. It is a compact structure that contains the cell nucleus and stores the genetic information of the cell. From it two types of extensions or neurites arise: the axon and the dendrites.

- Axon: is a thin cellular extension that arises from the soma at the axon hillock and extends up to even tens of thousands of times the diameter of the soma in length. The axon carries exit nerve signals. In most connections between neurons, called synapses, the signals are sent from the axon of a neuron (presynaptic) to the dendrite of another neuron (postsynaptic), although there are exceptions. Axons can also grow collateral branches to connect with close neurons.

- Dendrites: are cellular extensions with many branches that arise from the soma, forming complex 'dendritic trees'. Most neuron inputs occur via dendritic spines (described for the first time by Ramón y Cajal [1888]). Dendritic spines are small membranous protrusions of the dendrite of a neuron that typically receive input from an axon at a synapse. Spines are known to be critical in learning, memory and cognition; moreover, loss or alteration of these structures has been described in the pathogenesis of major neurological disorders such as Alzheimer's disease [Fiala et al., 2002]. Recent studies also suggest selective alterations in spines with aging in humans [Benavides-Piccione et al., 2013].

4.2.1 Current projects

In the last decade important and ambitious projects related to the study of the brain have arisen. The Blue Brain Project[1] (BBP) [Markram, 2006], headed by the founding director Henry Markram, began in 2005 by the Brain Mind Institute at the École Polytechnique Fédérale de Lausanne (EPFL) and IBM. The goal of the BBP is to build biologically detailed digital reconstructions and simulations of the rodent, and ultimately the human brain by means of reverse engineering, using the BlueGene supercomputer from IBM. The project represents the world's first comprehensive attempt of reverse engineering the mammalian brain, with the objective of knowing their functioning and dysfunctions through detailed simulations, helping to explore solutions to health mental problems and neurological diseases. At the end of 2006, the initial objective of the project was completed: the simulation of a rat neocortical column. The BBP website specifies that the final result of the project will be a facility with the capability to model and simulate:

- The brain or any region of the brain of any species, at any stage in its development.

- Specific pathologies of the brain.

- Diagnostic tools and treatments for these pathologies. The geometric and computational models of the brain produced by the facility will reproduce the structural and functional features of the biological brain with electron microscopic and molecular dynamic level accuracy.

The Neocortical Microcircuit Collaboration Portal[2] provides an online public resource of the Blue Brain Project's first release of a digital reconstruction of the microcircuitry of

[1] http://bluebrain.epfl.ch/
[2] https://bbp.epfl.ch/nmc-portal/

juvenile rat somatosensory cortex, access to experimental data sets used in the reconstruction, and the resulting models [Markram et al., 2015, Ramaswamy et al., 2015, Reimann et al., 2015] (Fig. 4.3).

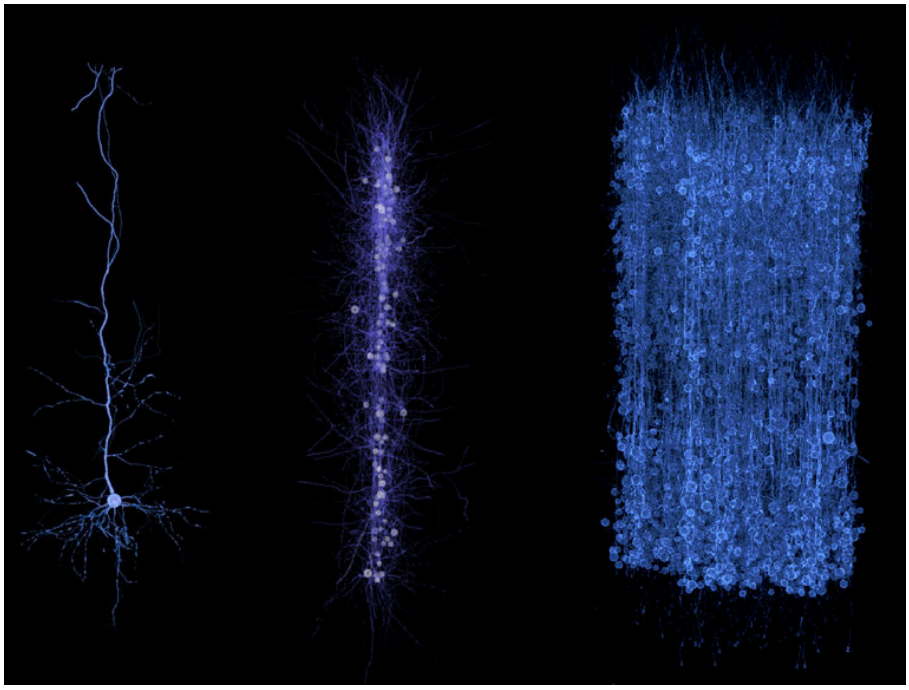


Figure 4.3: Single neuron (left), microcircuit consisting of several neurons (middle) and cortical column composed of multiple nerve cells (right). Source: Blue Brain Project, École Polytechnique Fédérale de Lausanne

On January 2009, the Spanish participation within the BBP called Cajal Blue Brain Project[3] (CBBP) was presented. This project is led by the Universidad Politécnica de Madrid (UPM) and the Instituto Cajal from Consejo Superior de Investigaciones Científicas (CSIC). The project uses the resources provided by the Magerit supercomputer installed in the Supercomputing and Visualization Center of Madrid (CeSViMa). The CBBP has the following key long-term objectives specified on its website:

- To decode the synaptome or detailed map of the synaptic connections of the cortical column and, as a result, reconstruct all its components.
- To give a strong boost to research on the cortical column, exploring in depth current hypotheses about its normal function and dysfunctions (especially Alzheimer's disease).
- To devise new methods to process and analyze the experimental data obtained in the aforementioned research studies.
- To develop computer technology to study neuronal functions using graphics tools and visualization methods.

[3] http://cajalbbp.cesvima.upm.es/

In 2013, two major international initiatives in Europe and the United States began almost simultaneously, with the provocative goal of grouping information and acting as an exchange platform to unravel the mysteries of brain function.

In Europe, the Human Brain Project[4] (HBP) [Markram, 2012] is a ten-year scientific research project whose objective is to understand the human brain and its diseases and, ultimately, to emulate its computational capabilities. A key objective is to reconstruct and simulate the whole human brain. The project has three main areas that are medicine, neuroscience and computing. The HBP has the following main objectives:

- To create and operate a European Scientific Research Infrastructure for brain research, cognitive neuroscience, and other brain-inspired sciences.
- To gather, organize and disseminate data describing the brain and its diseases.
- To simulate the brain.
- To build multi-scale scaffold theory and models for the brain.
- To develop brain-inspired computing, data analytics and robotics.
- To ensure that the HBP's work is undertaken responsibly and that it benefits society.

The Brain Research through Advancing Innovative Neurotechnologies[5] (BRAIN) Initiative [Alivisatos et al., 2012, 2013] or Brain Activity Map (BAM) project is a similar but completely separate project in the United States. President Obama launched the BRAIN Initiative to "accelerate the development and application of new technologies that will enable researchers to produce dynamic pictures of the brain that show how individual brain cells and complex neural circuits interact at the speed of thought." The three main goals of the project are specified in Alivisatos et al. [2013]:

- To build new classes of tools that can simultaneously image or record the individual activity of most, or even all, neurons in a brain circuit, including those containing millions of neurons.
- To create tools to control the activity of every neuron individually in these circuits, because testing function requires intervention.
- To understand circuit function.

The results achieved in these ambitious projects will depend on the progress of computer science and statistics, as well as the ability to pool a huge amount of data in order to extract patterns describing their organization. Iteratively, the available information must be incorporated, models must be programmed according to the known biological rules, and simulations must be executed, which must be compared with the experimental data, to adjust the models if necessary.

[4] https://www.humanbrainproject.eu/
[5] https://www.braininitiative.nih.gov/

4.3 Neurons in the cerebral cortex

Although there have been numerous efforts to classify different types of neurons [Armañanzas and Ascoli, 2015, Ascoli et al., 2008, Bota and Swanson, 2007, DeFelipe et al., 2013] a global consensus has not yet been achieved. In general, at least there is a consensus that neurons in the cerebral cortex can be classified into two major groups according to their morphology: pyramidal neurons and interneurons (Fig. 4.4). Differences in the identification and nomenclature of subgroups of this basic classification do not allow at this time to have a rigorous and complete neuronal classification.

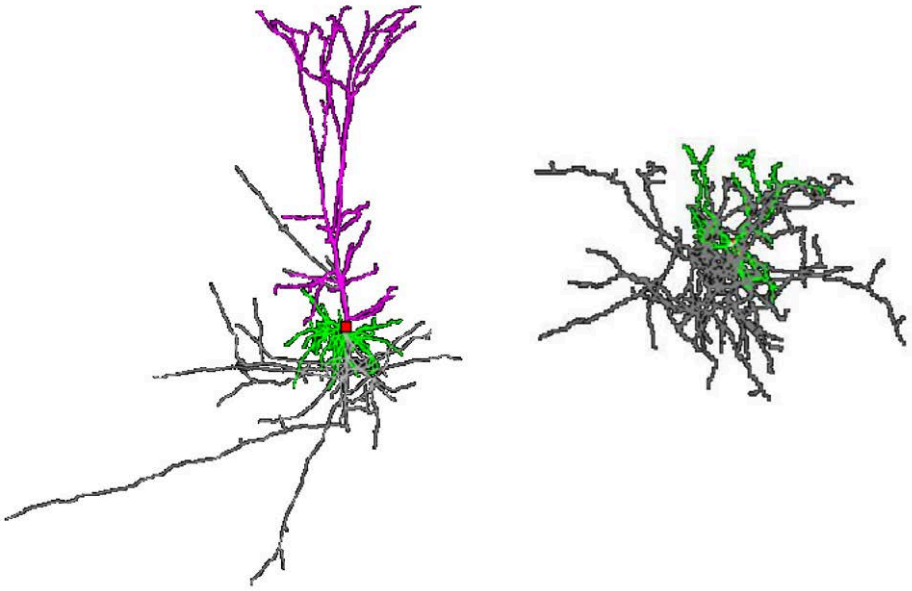


Figure 4.4: 3D reconstructions of a pyramidal cell (left) and an interneuron (right) of rat neocortex. The location of the soma is shown in red (in the interneuron it is difficult to see because it is behind the neurites). Axon is shown in gray and dendrites in green (in the pyramidal neuron basal dendrites are shown in green and apical dendrite in pink). Source: NeuroMorpho.Org [Ascoli et al., 2007]

The main features of the two major groups mentioned above can be summarized as follows:

- Projection neurons also called **pyramidal neurons** are so named because of the triangular shape of their cell body. The morphology of these neurons is characterized by a single axon, an apical dendrite and by multiple branched basal dendrites arising from its triangular cell body. Pyramidal cells have spines in their dendrites and are usually excitatory, using glutamate as neurotransmitter. They are the most abundant cortical neurons (70-80%). Pyramidal neurons are considered the main neuronal building blocks of the cerebral cortex because dendritic spines of these cells are the main postsynaptic target of excitatory synapses in the cerebral cortex [DeFelipe, 2011]. They are neurons with long axons that communicate separate and distant regions within the nervous tissue.

- **Interneurons** are short axon cells that innervate neighbouring regions with few or no spines in their dendrites. These cells show a great morphological variability across brain areas and animal species. Most interneurons are inhibitory and use gamma-aminobutyric acid (GABA) as the main neurotransmitter. GABAergic interneurons represent the vast majority of smooth or sparsely spiny non-pyramidal neurons (estimated as 15-30% of the total population of all cortical areas), which together with pyramidal cells and spiny non-pyramidal cells, represent the major classes of cortical neurons [DeFelipe, 2011].

4.4 Neuronal wiring

Fig. 4.5 shows an example of the variability that may exist in different wiring configurations passing through the same target points. In order to be considered as possible neuronal trees, in all configurations the maximum number of branches at each point is limited to two because, in general, neuronal branching nodes give rise to two branches.

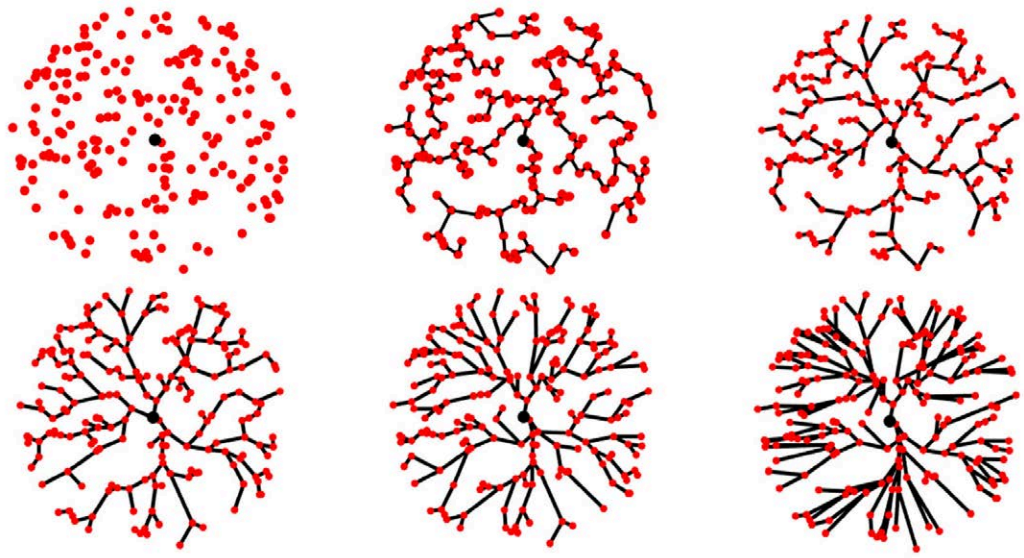


Figure 4.5: Example of trees with different wiring configurations passing through the same 200 target points randomly distributed on a circular surface starting from a root located in the center

Santiago Ramón y Cajal proposed the wiring economy principle. This principle states that neurons are arranged in such a way as to minimize the wiring cost where the structure of axons and dendrites is designed to save space, time and matter [Ramón y Cajal, 1899]. Wiring economy has been widely used in the literature to explain neuron placement in different brain areas and species, as well as morphological properties in single neurons.

Regarding placement, some authors consider the minimization of wiring costs in order to explain neuron placement in simple nervous systems such as *Caenorhabditis elegans* [Chen et al., 2006, Kaiser and Hilgetag, 2006, Pérez-Escudero and de Polavieja, 2007, Pérez-Escudero

et al., 2009]. There are also studies on the relation of wiring economy and neuron placement in larger brains. For example, Rivera-Alba et al. [2011] try to explain the placement of neurons in a module of the *Drosophila melanogaster* brain; Chklovskii et al. [2002] associate wiring optimization with the optimal arrangement of elements of neuronal circuits in the mouse neocortex; Kaiser and Hilgetag [2006] further examine the concept of wiring economy analyzing 3D spatial positions of connected cortical areas in the macaque brain; Rivera-Alba et al. [2014] use the concept of wiring economy and the dimensions of neuronal components to predict the microarchitecture of the neuropile across brain areas and species. Karbowski [2015] combines different forms of wiring minimization with the maximization of dendritic spine proportion in the cerebral cortex across species.

Regarding the morphological properties of single neurons, Cuntz et al. [2007, 2008, 2010] and Schneider et al. [2014] use simulations of synthetic neuronal structures to show that optimal wiring explains dendritic branching patterns. Wen and Chklovskii [2008] and Wen et al. [2009] attempt to disclose the relationship between the dimensions and branching structure of dendritic arbors and synaptic distribution by minimizing wiring cost. Other studies formulate mathematically the relation between optimal wiring and different dendritic characteristics. For example, Cuntz et al. [2012] have shown that optimal wiring predicts a 2/3 power law between dendritic wiring length and the number of branching points and also a 2/3 power law between wiring and the number of synapses.

Part III

CONTRIBUTIONS TO POINT PROCESS STATISTICS

Chapter 5

Three-dimensional replicated point pattern-based analysis applied to cortical synapses

5.1 Introduction

The aim of this chapter is to perform a complete analysis of the 3D spatial distribution of several groups of replicates (groups of patterns considered as instances of the same process). For that, we first analyzed the intensity in each group and examined whether there were significant differences between groups. Second, we performed spatial modeling to find a suitable model for each replicate from different groups. Third, we used replicated spatial point patterns to analyze similarities and differences in the spatial distribution between groups of replicates. To illustrate each step of this spatial analysis, it was applied to the study of the 3D distribution of synapses in the cerebral cortex, particularly aiming to find out whether there is a general pattern of distribution of synapses for the six cortical layers, and identify any possible similarities and differences between layers.

One major issue in cortical circuitry is to ascertain how synapses are distributed and whether or not synaptic connections are specific [DeFelipe et al., 2002b]. To understand the anatomical design principles of cortical circuits, it is essential to analyze the ultrastructure of all components of the neuropil (i.e., the very dense network of neuronal and glial processes that occupy the space between the cell bodies of neurons, glia and blood vessels) and in particular the number and spatial distribution of synapses. Furthermore, synaptic size plays an important role in the functional properties of synapses [Lüscher et al., 2000, Schikorski and Stevens, 1997, Takumi et al., 1999, Tarusawa et al., 2009]. Thus, numerous researchers have been trying to find simple and accurate methods for estimating the distribution, size and number of synapses. To this end, two sampling procedures are currently available: one is based on serial reconstructions and the other on single sections. Clearly, serial reconstruction should be the method of choice for the challenging task of unraveling the extraordinary

complexity of the nervous system. Indeed, serial sectioning transmission electron microscopy is a well-established and mature technology for collecting 3D data from ultrathin sections of brain tissue [Bock et al., 2011, Harris et al., 2006, Hoffpauir et al., 2007, Mishchenko et al., 2010, Stevens et al., 1980]. It is based on imaging ribbons of consecutive sections with a conventional transmission electron microscope. However, the major limitation is that it is extremely time-consuming and difficult to obtain long series of ultrathin sections, often making it impossible to reconstruct large volumes of tissue. Hence, the recent development of automated electron microscopy techniques is a vital step forward in the study of neuronal circuits [Briggman and Denk, 2006, Knott et al., 2008, Merchán-Pérez et al., 2009]. Using combined focused ion beam (FIB) milling and scanning electron microscopy (SEM), we obtained 3D samples from the six layers of the rat somatosensory cortex and identified and reconstructed the synaptic junctions. A total volume of tissue of approximately 4500 μm^3 and around 4000 synapses from three different animals were analyzed. Different samples, layers and/or animals were aggregated and compared using replicated spatial point processes.

The research included in this chapter has been published in Anton-Sanchez et al. [2014].

Chapter outline

The chapter is organized as follows. Section 5.2 describes the intensity analysis in each group of replicates. Section 5.3 introduces the modeling of spatial point processes for each replicate. Section 5.4 details the proposed methodology for replicated point pattern-based analysis. Section 5.5 shows the results of the spatial analysis described in the previous sections applied to the study of the distribution of cortical synapses. Section 5.6 briefly describes the software developed for the spatial analysis of synapses. Finally, Section 5.7 ends with some discussion and conclusions.

5.2 Intensity

The first step in our analysis was to estimate the synaptic density of each layer and, more specifically, to study whether there were significant differences between synaptic densities in different layers of the somatosensory cortex. We used the simulation process described below along with a multiple mean comparison test.

We calculated a fixed-volume sampling box to extract subsamples from the original experimental samples. The x, y, z dimensions of this box were equal to the smallest x, y, z dimensions of the experimental samples, so the box could be applied to any of the samples without exceeding their boundaries. We then used this box to extract centroids from randomly selected samples of each layer at random locations. We repeated this process 50 times for each layer, thus obtaining 50 different synaptic densities per layer. See Fig. 5.1.

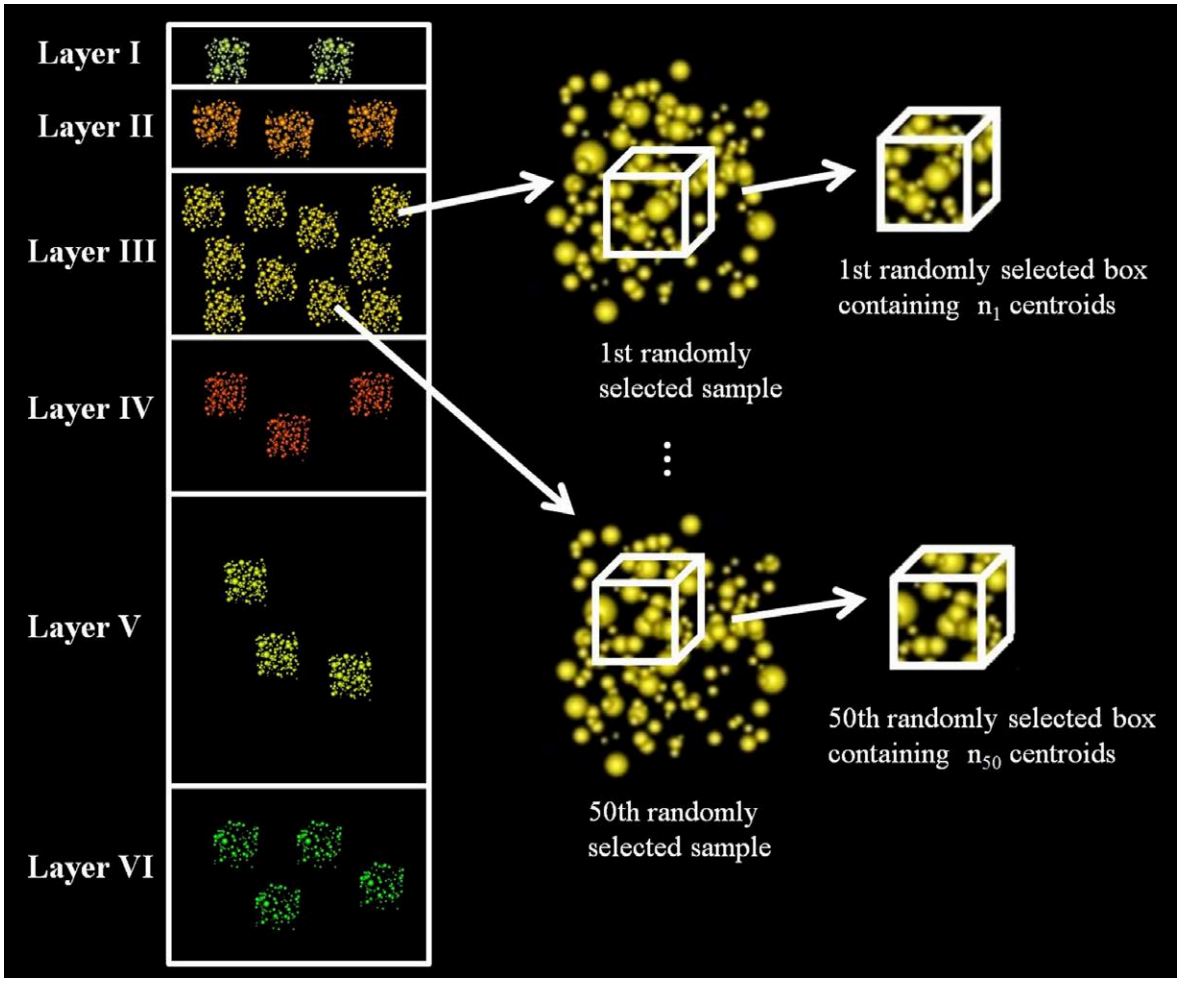


Figure 5.1: Diagram of data extraction to analyze whether the synaptic densities of cortical layers are significantly different. The figure shows how we randomly selected a sample from layer III, then we extracted, also randomly, a box inside this sample and counted the number of synaptic junctions in the box. We repeated this process 50 times for each layer. The dimensions of the box were the same for all layers, and it had the maximum volume that could be extracted from all the samples, i.e., it had the minimum length in each dimension (x, y, z) considering all samples

5.3 Modeling of spatial point processes

The second step in the analysis of the entire cerebral cortex was to find a suitable model for each of the samples from layer I to VI. Because Merchán-Pérez et al. [2014] recently showed that the random sequential adsorption (RSA) process [Evans, 1993] adequately describes the spatial distribution of synaptic junctions in layer III, we tested the RSA model for each sample of all layers.

An RSA process is a type of hard-core process, i.e., two points cannot be placed closer than a minimum distance, where locations are chosen randomly, subject only to the distance constraint. These minimum distances can be fixed or, as in our case, calculated according

to a probability density function (Section 2.2.3.3). Considering that the synaptic junctions cannot overlap, and therefore the minimum distances between synapses are limited by the size of the junctions at least, the RSA process is particularly well suited here. We have used Feret's diameter of each synaptic junction as an estimate of its size (the diameter of the smallest sphere circumscribing the synaptic junction). As in Merchán-Pérez et al. [2014] for layer III, we found that Feret's diameters in all layers were lognormally distributed.

To test the RSA models we used one of the summary characteristics most commonly used in the analysis of spatial point processes, namely Ripley's K function and, particularly, a common transformation of it, Besag's L function [Ripley, 1977] (see Section 2.2 for details). The Miles-Lantuéjoul-Stoyan-Hanisch translation edge-correction is often used to estimate $K(d)$ [Baddeley et al., 1993, Ohser, 1983]:

$$\hat{K}(d) = \frac{vol(B)^2}{N(B)^2} \sum_{x_k \in B} \sum_{x_l \neq x_k} \frac{\mathbf{1}\{||x_k - x_l|| \leq d\}}{\gamma_B(x_k - x_l)}, \tag{5.1}$$

where $N(B)$ is the number of points falling in a region $B \subset \mathbb{R}^3$, x_k, $k = 1, ..., N(B)$ are the observed points, $vol(B)$ is the volume of the region B and γ_B is the 'set covariance', $\gamma_B(x_k - x_l) = vol(\{x | x + x_k - x_l \in B\}) = vol(B \cap (B - (x_k - x_l)))$.

The 3D CSR process has the following expression for the K function (a clustered pattern curve will be shifted to the left, whereas a regular pattern curve will be shifted to the right):

$$K_{CSR}(d) = \frac{4}{3}\pi d^3. \tag{5.2}$$

The 3D expression of Besag's L function is:

$$L(d) = \sqrt[3]{\frac{3}{4\pi} K(d)}. \tag{5.3}$$

As explained in Section 2.2.3.1, this transformation converts the CSR K function to the straight line $L_{CSR}(d) = d$, making the plots much easier to assess visually. For the L function, a regular pattern curve will be below the diagonal (CSR) and a clustered pattern will be above.

The expression of Ripley's K function for the RSA process is analytically unknown, so we have to use RSA simulations. To simulate an RSA process we need to know its intensity and the probability density function of the minimum distances between points. In our case, we need the synaptic density λ and the μ and σ parameters of the lognormal distribution of Feret's diameters. An RSA process simulation starts with an empty window to which spheres, whose radii follow the lognormal distribution fitted using Feret's diameters, are added randomly one at a time. If the new simulated synapse intersects with any existing sphere, the new sphere is rejected, and another sphere is generated with another location and radius. The process continues until the target intensity is reached.

For example, Fig. 5.2 shows the K and L summary functions of experimental sample 1 from layer I (blue), the average of 99 RSA simulations performed for this sample (green) and the functions for a CSR process (red). Each RSA simulation had the same intensity as the

original sample, and the size of simulated synapses was calculated according to the lognormal distribution fitted using Feret's diameters of all the synapses of the sample. Generally, the K functions were very similar to each other across all distances for all the samples. Moreover, for short distances (200-300 nm), the L functions of the samples and RSA processes were well below the diagonal line (CSR) representing the empty space around centroids which should not contain any centroid (non-overlapping synapse constraint). From about 400 nm onwards, the L functions of both models and experimental samples were again very similar to each other.

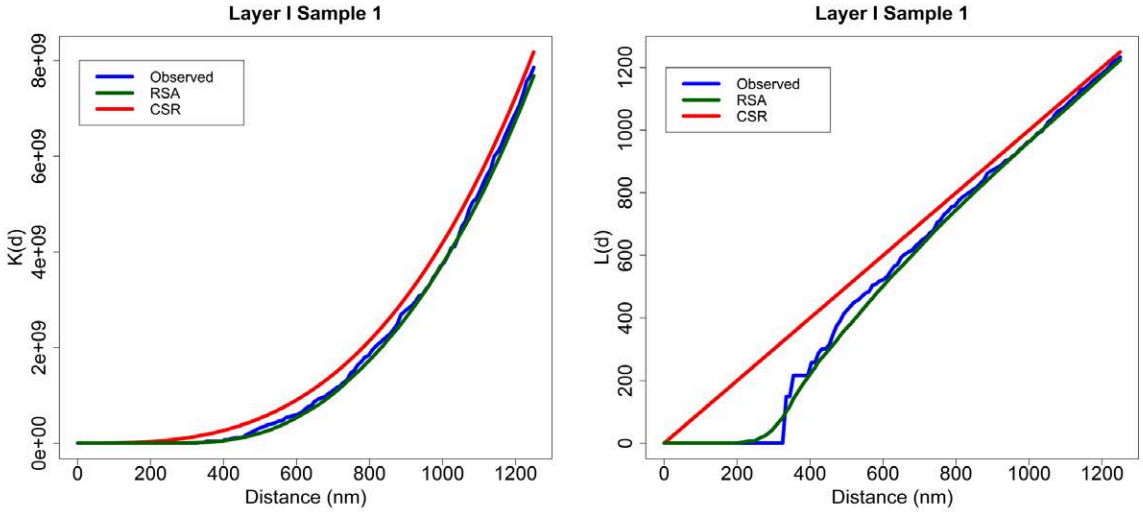


Figure 5.2: Layer I, Sample 1. An example of K and L functions for CSR and RSA processes. K (left) and L (right) functions of the experimentally observed data (blue) along with the theoretical CSR (red) and the average of 99 RSA process simulations fitted for sample 1 (green). The K functions of the sample, CSR and RSA processes are very similar. The L functions of the RSA and the experimentally observed sample are positioned well below the diagonal (CSR) for short distances and are fairly close to the diagonal for larger distances

To test differences between two summary functions we used simulation-based envelopes (Section 2.2.4). The statistical rationale of this common procedure is to be found in Monte Carlo testing. Taking the advice of Baddeley et al. [2014a], we transformed the K function into the L function and used global envelopes since we had no prior information about the range of spatial interaction. Note that Monte Carlo tests 'are strictly invalid, and probably conservative, if parameters have been estimated from the data' [Diggle, 2003]. To overcome this obstacle, we adjusted an RSA process for each sample j in each layer i (i = I, ..., VI) and estimated the parameters $\hat{\lambda}_{ij}$, $\hat{\mu}_{ij}$ and $\hat{\sigma}_{ij}$ using only the remaining samples of the same layer. The sizes of the simulated synapses were calculated according to the lognormal distribution fitted using Feret's diameters of these remaining (m_i-1) samples in layer i, where m_i is the number of samples in layer i. If vol_{it} denotes the volume of sample t in layer i, then

$$\hat{\lambda}_{ij} = \frac{\sum_{\substack{t=1 \\ t \neq j}}^{m_i} \lambda_{it} vol_{it}}{\sum_{\substack{t=1 \\ t \neq j}}^{m_i} vol_{it}}. \tag{5.4}$$

The RSA null hypothesis was tested as follows. For each sample, we performed 99 RSA simulations with the described parameters. We calculated the average L function of all these simulations and took this average, $\bar{L}$, to be an estimate of the theoretical mean value of the L summary statistic for the RSA model. The global envelope is a region of constant width $2w_{max}$, where w_{max} is determined as the furthest deviation between $\bar{L}$ and any of the L functions of a separate set of 99 RSA simulations with the same parameters at any distance d along the horizontal axis. We rejected the null hypothesis if the L function of the sample lay outside the envelope for any value of d (see Section 5.5.3 and Fig. 5.5).

Numerous R packages implement functions for spatial data analysis [Bivand et al., 2013], particularly for the analysis of spatial point patterns. Among the most commonly used are the **spatial** package [Venables and Ripley, 2002], the **splancs** package [Rowlingson and Diggle, 1993] and the **spatstat** package [Baddeley and Turner, 2005, Baddeley et al., 2015]. Another tool with a more user-friendly interface for analyzing three-dimensional spatial point patterns is the matlab-based software *Spatial Analysis 3D* (*SA3D*[1]) [Eglen et al., 2008]. In this chapter we analyzed spatial patterns using R software and the **spatstat** package. We obtained the translation edge-correction estimator of Ripley's K function in three dimensions for both the observed samples and the RSA simulations using the *K3est* function included in the **spatstat** package and we directly calculated the L functions from K functions using Eq. (5.3). To compute the simulation envelopes of the L functions we used the *envelope.pp3* function, also included in the **spatstat** package. We used this function with the 3D point pattern for each sample and 198 3D point patterns of RSA simulations performed for that sample.

5.4 Replicated spatial point patterns

We performed the Diggle test [Diggle et al., 1991, 2000] to study similarities and differences between groups of replicated data (see Section 2.4 for details). This test uses a bootstrap procedure to check whether there are significant differences between empirical K functions of independent replicates. Using 5000 bootstrap iterations, we studied whether there were differences between the study animals and between different cortical layers.

It is scientifically correct to construct an aggregated estimator of the K function without assuming a common intensity across all replicates because the K function is defined as independent of the intensity. This assumes that the hypothesis of a common K function and varying intensity is plausible, as would be the case if the replicates were different intensity

[1] http://www.nri.ucsb.edu/Labs/breese/SA3D.html

versions of a common underlying process [Diggle, 2013]. To test if this applied in our case, we adjusted a global spatial model for groups of replicates in which the Diggle test found no significant differences. Then we applied different random thinning procedures (i.e., randomly deleting points from the original model, Section 2.2.3.3) and introduced a cross-validation technique to honestly estimate the goodness-of-fit of the resulting models.

More explicitly, assume that A, B and C were the groups where the Diggle test found no significant differences, and let m_A, m_B and m_C be the number of samples in each group. We adjusted the global spatial model RSA_{global} with parameters μ_{global}, σ_{global} and λ_{global}. Parameters μ_{global} and σ_{global} were obtained by fitting the lognormal distribution of Feret's diameters considering all synapses of all samples from groups A, B and C and were used to estimate the size of the synapses in the global model. Let λ_{ij} be the synaptic density for the jth sample in the ith group, λ_{global} was chosen such that $\lambda_{global} > \lambda_{ij}$ for all i,j, i.e., we considered a global model that was *denser* than each of the samples separately (we chose to make λ_{global} 1% denser than the maximum density of each sample separately).

Our goal, then, was to check whether groups A, B and C, whose K functions were found not to be significantly different, were different thinned versions of a common underlying process. In other words, we wanted to find out whether the processes that described the spatial distribution of samples from groups A, B and C were different thinned versions of the global spatial model RSA_{global}.

To do this, we ran 198 *dense* RSA_{global} simulations with the estimated parameters μ_{global}, σ_{global} and λ_{global}. Then we thinned each of these *dense* simulations for each sample in each group. We used a cross-validation technique to check if these simulations had the same spatial distribution as the experimentally observed sample. Specifically, we applied the following cross-validation process for each sample j (test sample) in each group i:

1. First, we estimated $\hat{\lambda}_{ij}$ using the remaining (m_i-1) samples (training samples) in group i. The aggregated $\hat{\lambda}_{ij}$ was calculated by weighting the densities of the training samples by their volume as in Eq. (5.4).

2. Second, we randomly thinned the 198 *dense* RSA_{global} simulations until we obtained an intensity equal to the estimated density $\hat{\lambda}_{ij}$. Thus we obtained a set of 198 thinned RSA_{ij} simulations for sample j of group i. These simulations were like the original simulations but had a density equal to the intensity estimation for the test sample. This process is shown in Fig. 5.3.

3. Finally, we again used simulation-based envelopes to test for differences in the spatial distributions of the thinned simulations and the experimentally observed sample. We used 99 simulations to estimate the theoretical mean value of the L function for the RSA_{ij} model. We used the other 99 to calculate the maximum absolute difference from this theoretical mean value, which is necessary to build the envelope.

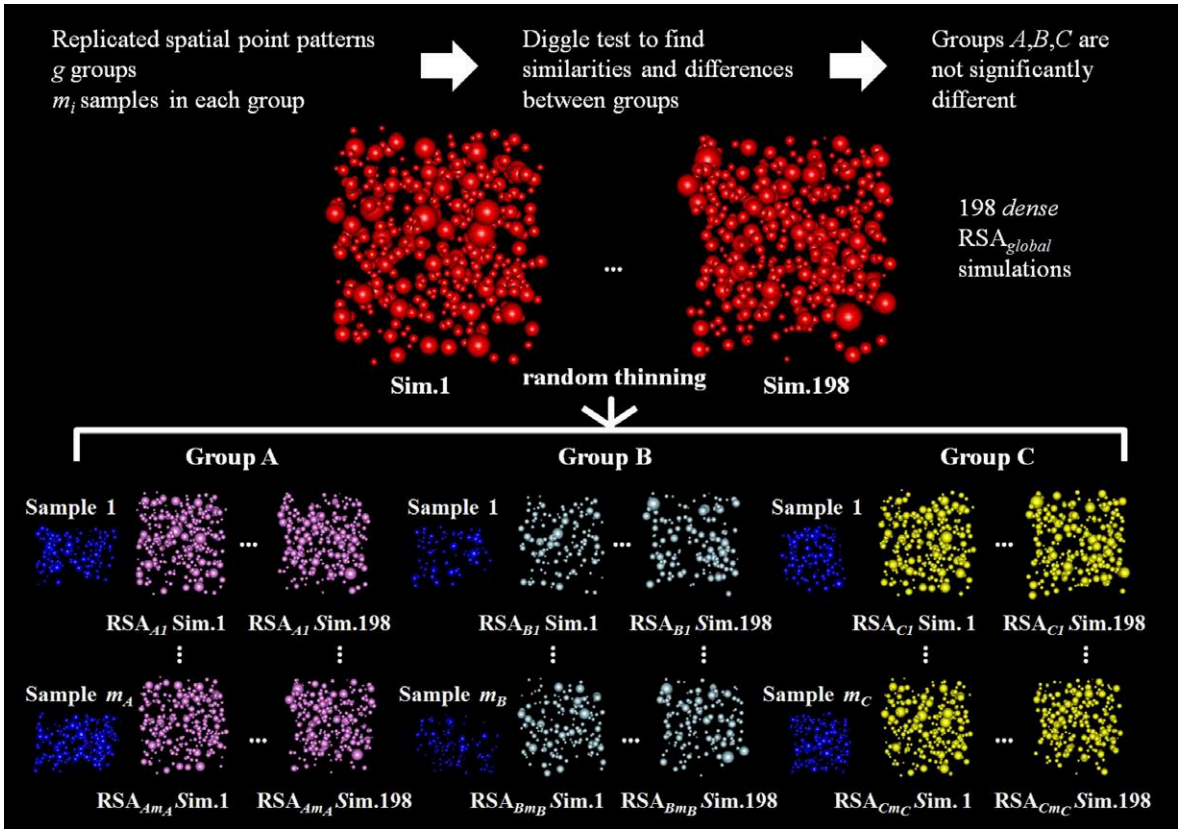


Figure 5.3: Diagram of the random thinning process for three groups of replicated point patterns, A, B and C, for which the Diggle test did not find significant differences. Our goal is to check if these groups are differentially thinned versions of a common underlying RSA process. Random thinning of *dense* simulations is performed for each experimentally observed sample j in each group i (test sample, shown in blue). Random thinning continues until we reach the intensity $\hat{\lambda}_{ij}$, estimated from all samples in group i excluding sample j. Then, for each experimentally observed sample j in each group i, we used simulation-based envelopes to test for differences in the spatial distributions of the thinned RSA simulations and the sample (we used 99 thinned simulations to estimate the L function for the RSA$_{ij}$ model and the other 99 to calculate the maximum deviation necessary to build the envelope)

5.5 Results

5.5.1 Data

We obtained 25 samples from the six layers of the somatosensory cortex of three 14-day-old rats by FIB/SEM. Although virtually all cortical synapses can be accurately identified as asymmetric and symmetric using FIB/SEM [Merchán-Pérez et al., 2009], we considered synaptic junctions as a whole. This was because it was not feasible to test RSA models for such a small number of symmetric synapses (they accounted for less than 10% of the total number of synapses found in any cortical layer). Thus, for simplicity's sake, we will use synaptic junctions to refer to both types of synapses. Synaptic junctions were visualized, automatically

segmented and reconstructed in three dimensions using Espina software [Morales et al., 2011]. We had a total reconstructed tissue volume of approximately 4500 μm^3 containing almost 4000 3D reconstructions of synapses. For each of these synapses, we had information on its 3D position (center of gravity or centroid) and an estimate of its size based on Feret's diameter. We obtained the density of each sample, that is, the number of synapses per unit volume, and the mean density for each layer (Table 5.1).

Table 5.1: Animal ID, volume, counts and density of synaptic junctions per sample in each layer of the somatosensory cortex. Total quantities and mean for each layer are shown

	Sample	Animal	Volume (μm^3)	No. of synapses	synapses/μm^3
Layer I	1	w33	210.61	180	0.855
	2	w35	177.20	128	0.722
Layer II	1	w33	224.35	230	1.025
	2	w35	139.51	127	0.910
	3	w35	149.03	206	1.382
Layer III	1	w31	149.13	147	0.986
	2	w31	157.15	109	0.694
	3	w33	186.45	173	0.928
	4	w33	176.44	178	1.009
	5	w33	176.28	167	0.947
	6	w33	175.55	165	0.940
	7	w33	191.28	189	0.988
	8	w35	247.58	198	0.800
	9	w35	178.40	201	1.127
	10	w35	165.06	168	1.018
Layer IV	1	w33	154.59	172	1.113
	2	w35	140.63	178	1.266
	3	w35	123.81	162	1.308
Layer V	1	w33	165.62	117	0.706
	2	w33	218.01	198	0.908
	3	w33	207.95	175	0.842
Layer VI	1	w33	185.32	92	0.496
	2	w35	183.55	85	0.463
	3	w31	179.97	102	0.567
	4	w31	280.09	107	0.382
		All Samples	4543.55	3954	0.870
MEAN		Layer I	193.91	154	0.794
		Layer II	170.96	188	1.098
		Layer III	180.33	170	0.940
		Layer IV	139.68	171	1.222
		Layer V	197.19	163	0.828
		Layer VI	207.23	97	0.466

5.5.2 Intensity

The density of the samples range from 0.382 synapses/μm^3 in a sample of layer VI to 1.382 synapses/μm^3 in a sample of layer II. The overall mean density is 0.870 synapses/μm^3 in all layers. See Table 5.1 for details. As shown in Fig. 5.4, the mean density of layer I is 0.794 synapses/μm^3, whereas layers II and III have mean densities of 1.098 and 0.940 synapses/μm^3 respectively, which increases up to the maximum mean density of 1.222 synapses/μm^3 in layer IV and then drops again in layer V (0.828 synapses/μm^3) down to the minimum mean density in layer VI, 0.466 synapses/μm^3.

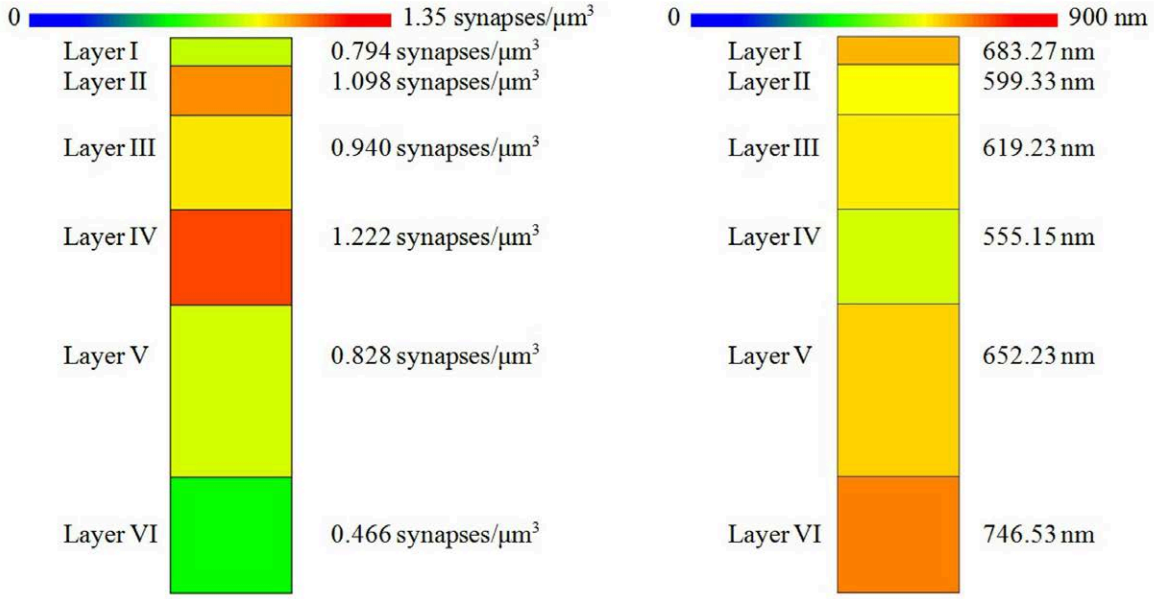


Figure 5.4: (Left) Mean synaptic density of the six layers of the somatosensory cortex. The synaptic density of the six layers is significantly different. However, we found no significant differences between the densities of layers I vs V or between the densities of layers II vs III. (Right) Mean distance to nearest synapse for each layer. Nearest synapse distances are significantly different in the six layers of the somatosensory cortex, but we found no significant differences between distances of layers I vs V, I vs VI, II vs III and III vs V

Following the simulation process described in Section 5.2, we looked for significant differences between the densities of the different layers of the somatosensory cortex. We performed a multiple mean comparison test on the 50 extracted densities for each of the six cortical layers. Because not all of the necessary assumptions for ANOVA were satisfied (data were normally distributed but homoscedasticity was not met, i.e., the variance of data in each layer was not the same), we used the Kruskal-Wallis test and then applied the Mann-Whitney test with the Bonferroni method to adjust the p-values for pair-wise comparisons. We found that there were differences between the density of layers (p-value $\leq 2.2 \times 10^{-16}$), which is consistent with a recent work [Crandall, 2013]. Pair-wise comparisons revealed that there was no significant difference between the densities of layers I vs V or between the densities of layers

II vs III.

In addition to the location and Feret's diameters of synapses of each sample, which were on average 404.73 nm, we measured the distance of each synapse to its nearest synapse. The mean distances to nearest neighbour measured between centroids of synaptic junctions ranged from 533.78 nm in a sample of layer II to 794.63 nm in a sample of layer VI, and the overall mean distance to the nearest synapse was 641.58 nm. This information is shown in Table 5.2. Using the Kruskal-Wallis test we found that there were significant differences between the distances to the nearest synapse between layers of the somatosensory cortex (p-value $\leq 2.2 \times 10^{-16}$). We applied the Mann-Whitney test and adjusted the p-values using the Bonferroni method for pair-wise comparisons. There were no significant differences for layers I vs V, I vs VI, II vs III and III vs V. Notice that we found no differences between the synaptic densities of layers I vs V and II vs III either (see Fig. 5.4).

5.5.3 Modeling of spatial point processes

A recent paper [Merchán-Pérez et al., 2014] analyzed the 3D spatial distribution of synapses in the somatosensory cortex. Merchán-Pérez and colleagues adjusted CSR and RSA models showing that RSA processes modeled the synaptic distribution more adequately. However, this study was limited to layer III of the somatosensory cortex. We extend this analysis to all layers of the cortex here.

To test the null hypothesis of RSA we used simulation-based envelopes. As an example, Fig. 5.5 shows the envelopes of the first sample of each layer of the somatosensory cortex. The averages of the L functions of 99 RSA simulations performed for each sample are represented in green. The shaded area is a region of constant width $2w_{max}$. The width w_{max} was calculated with a separate set of 99 RSA simulations as described in Section 5.3 using the **spatstat** package. The dashed red lines show the theoretical value for CSR for visual comparison only.

The null hypothesis is rejected if the L function of the experimentally observed sample (blue) lies outside the envelope for any value of distance d. The L functions of samples 2 and 7 from layer III and sample 2 from layer IV were very close to the upper boundary of the envelope at a distance of about d =300 nm but did not lie outside the envelope. The remaining samples were completely within the envelope for all values of d. So, we did not reject the RSA model for any of the 25 analyzed samples.

5.5.4 Replicated spatial point patterns

Taking advantage of the fact that we had several samples of each layer of the somatosensory cortex, we used replicated spatial point patterns in order to detect similarities and differences between groups. Because we had seen that synaptic densities between layers of the somatosensory cortex were different, we used the K function because it does not depend on intensity. We aggregated the K functions of each group using the number of synapses ($w_{ij} = n_{ij}$, Eq. 2.15) [Diggle, 2013, Diggle et al., 1991].

Table 5.2: Mean distances from a synapse to its nearest neighbour and mean Feret's diameters. Nearest neighbour distances are measured between centroids of synaptic junctions. Feret's diameters are an estimate of the size of synaptic junctions (diameter of the smallest sphere circumscribing each junction)

	Sample	Mean distance to nearest neighbour (nm) ± sd	Mean Feret's diameter of synaptic junctions (nm) ± sd
Layer I	1	682.09 ± 201.96	459.01 ± 196.20
	2	684.95 ± 242.28	442.01 ± 207.62
Layer II	1	613.06 ± 191.74	429.69 ± 183.35
	2	680.80 ± 204.30	453.67 ± 184.03
	3	533.78 ± 177.72	340.96 ± 143.25
Layer III	1	600.10 ± 193.62	377.19 ± 159.63
	2	680.33 ± 200.79	462.18 ± 177.52
	3	620.15 ± 206.34	437.62 ± 168.04
	4	615.28 ± 208.79	414.22 ± 169.04
	5	647.70 ± 228.39	466.03 ± 215.91
	6	605.46 ± 231.85	423.38 ± 169.83
	7	599.08 ± 244.67	397.29 ± 168.22
	8	643.36 ± 193.31	427.90 ± 168.15
	9	580.30 ± 203.76	378.35 ± 166.60
	10	625.62 ± 209.32	405.43 ± 175.62
Layer IV	1	562.38 ± 228.22	397.83 ± 155.06
	2	539.84 ± 208.77	354.90 ± 129.26
	3	564.29 ± 214.38	353.52 ± 134.01
Layer V	1	701.03 ± 235.69	414.84 ± 161.68
	2	632.66 ± 263.23	380.71 ± 173.12
	3	641.75 ± 216.35	404.49 ± 186.79
Layer VI	1	730.74 ± 272.02	425.60 ± 146.11
	2	766.04 ± 371.24	394.42 ± 176.28
	3	694.07 ± 301.23	325.66 ± 114.03
	4	794.63 ± 357.46	351.45 ± 153.30

As discussed, we performed the Diggle test to compare different groups of K functions [Diggle et al., 1991, 2000]. The first step was to check whether there were any differences between the three animals. We applied the Diggle test to $g = 3$ groups of sizes $m_1 = 12$, $m_2 = 9$ and $m_3 = 4$ and obtained a p-value $= 0.724$. Thus, we did not detect differences between animals in the study. Fig. 5.6 shows the aggregated K and L functions for each of the three animals. After ruling out differences between animals, we studied whether there were differences in the synaptic distribution between layers.

Considering each layer of the cortex as a group of replicates, we calculated the aggregated L function of each group transforming the aggregated K function of the group (Eq. (2.15)).

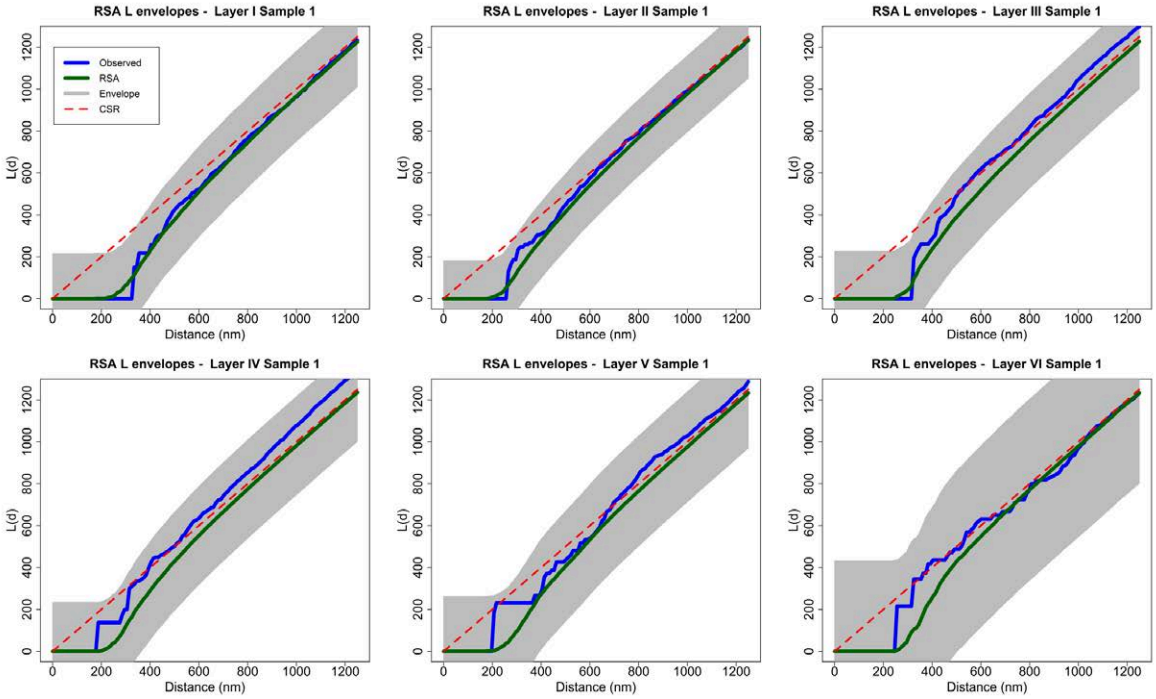


Figure 5.5: Analysis of spatial patterns using global envelopes (sample 1 for each layer of the somatosensory cortex). The L functions of the experimentally observed samples are shown in blue, and the averages of 99 RSA simulations are shown in green. The shaded area represents the envelopes of values calculated from a separate set of 99 RSA simulations. We do not reject the RSA null hypothesis for any sample because no observed L function lies outside the envelope for any value of distance d. The results for all samples in the study were the same. Dashed red lines show the theoretical value for CSR (for the purpose of visual comparison only)

Fig. 5.7 shows the L function of each observed sample in each layer as dashed blue lines, the aggregated L function of each layer in dark blue and the average of 99 RSA simulations fitting the RSA model for all the samples of the layer in green. We calculated the parameters $\hat{\lambda}_i$, $\hat{\mu}_i$ and $\hat{\sigma}_i$ of the RSA$_i$ model for each layer i, i = I, ..., VI, calculating the volume-weighted average of the parameters λ_{ij} of each sample j in layer i and fitting the lognormal distribution of Feret's diameters using all synapses in this layer. Fig. 5.7 also shows the envelope obtained using a separate set of 99 RSA simulations with the same parameters, as explained in Section 5.3. For visual comparison, we added the theoretical L function for a random pattern (dashed red diagonal). Because all the aggregated L functions were within the boundaries of the envelopes, we did not reject the RSA model for any layer of the somatosensory cortex.

Applying the Diggle test for $g = 6$ groups of sizes $m_1 = 2$, $m_2 = 3$, $m_3 = 10$, $m_4 = 3$, $m_5 = 3$ and $m_6 = 4$, we obtained a p-value of 0.002. Thus, we could conclude that there were differences between the six layers of the cortex. To better understand synaptic spatial distribution, we applied the Diggle test six times with $g = 2$ groups, each time forming a

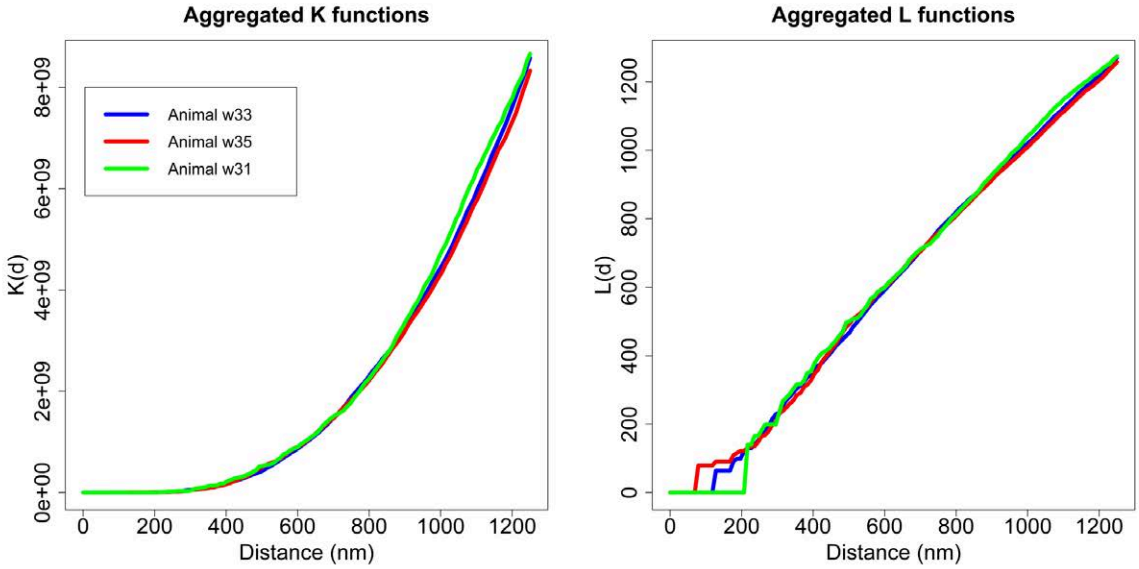


Figure 5.6: Aggregated K and L functions for each animal. The Diggle test found no significant differences between the three animals used in the study

group with the K functions of all samples of one layer and the other group with the K function of all samples of the remaining layers. In this analysis, the group of samples from layer I was the only one significantly different from the other group (samples from layers II to VI) with a p-value of 0.009. The Diggle test found no significant differences between groups of replicates formed by layers II to VI ($g = 5$, p-value $= 0.1176$). Moreover, the Diggle test found no significant differences between the distribution of samples from layers II to VI in pair-wise comparisons of these layers. Fig. 5.8 shows the aggregated K and L functions of all six layers (the two identified groups are shaded differently, i.e., layer I in green and layers II to VI in violet). Layer I functions are slightly shifted to the right compared to the other layers, so the repulsion in the spatial distribution of its synapses appears to be greater.

In Section 5.5.2 we saw that layers of the somatosensory cortex did not have a common synaptic density, so we wanted to find out whether we had different thinned versions of a common underlying process in layers from II to VI [Diggle, 2013]. We did this analysis introducing for the first time in this context a cross-validation technique to honestly estimate the goodness-of-fit of the resulting models.

With the simulation and thinning process described in Section 5.4, we performed 198 *dense* RSA_{global} simulations with a volume of 300 μm^3 and a density of 1.4 synapses/μm^3 ($\lambda_{global} = 1.4$, a density greater than the density of any of the samples), i.e., each RSA_{global} simulation had 420 synapses. For each sample j (test sample) in group i (we had a group consisting of layers II to VI), we calculated the synaptic density of its RSA_{ij} model using the remaining samples of the same layer (Eq. (5.4)). Table 5.3 shows the estimated intensity $\hat{\lambda}_{ij}$ for each experimental sample. For each sample, we randomly thinned each of the 198 *dense* RSA_{global} simulations until they had the estimated intensity $\hat{\lambda}_{ij}$. The sizes of the simulated

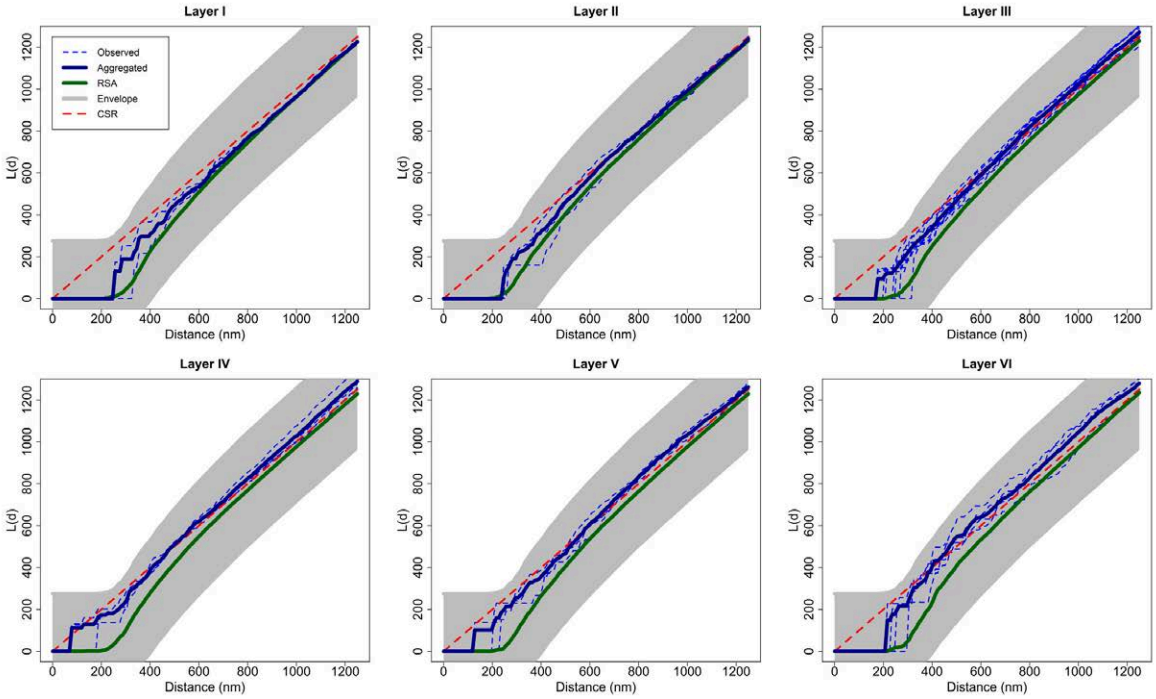


Figure 5.7: For each layer, aggregated L function (dark blue) of experimentally observed data (dashed blue) along with the average of 99 RSA simulations (green) fitting the model for all samples of the layer. This figure shows the envelope obtained using a separate set of 99 RSA simulations. We do not reject the RSA model for any layer of the somatosensory cortex because all the aggregated L functions were within the boundaries of the envelopes. We added the theoretical L function for a random pattern (dashed red diagonal) for the purpose of visual comparison

synapses were calculated using the lognormal distribution fitted using Feret's diameters of all samples of the group. Table 5.3 also shows these parameters. Note that μ_{global} and σ_{global} are equal because all these layers were modeled as a common RSA_{global} process. Fig. 5.9 shows one *dense* RSA_{global} simulation for the group of layers II to VI and two thinned RSA simulations for two different samples in the study.

We validated the RSA_{ij} model with the test sample i using simulation-based envelopes. To do this, we used the function *envelope.pp3* included in the **spatstat** package. The L functions of sample 7 from layer III and sample 2 from layer IV touched the upper boundary of the envelope slightly at distances around 200-300 nm but did not lie outside the envelope. However, sample 1 from layer IV did lie just outside the envelope at distances around 300-400 nm. The remaining samples were completely within the envelope. Thus, for all 23 samples in layers II to VI, except for only sample 1 in layer IV, we did not reject the null hypothesis of RSA, i.e., we validated the hypothesis that the synaptic distribution of layers II to VI of the somatosensory cortex are different thinned versions of a common underlying RSA process.

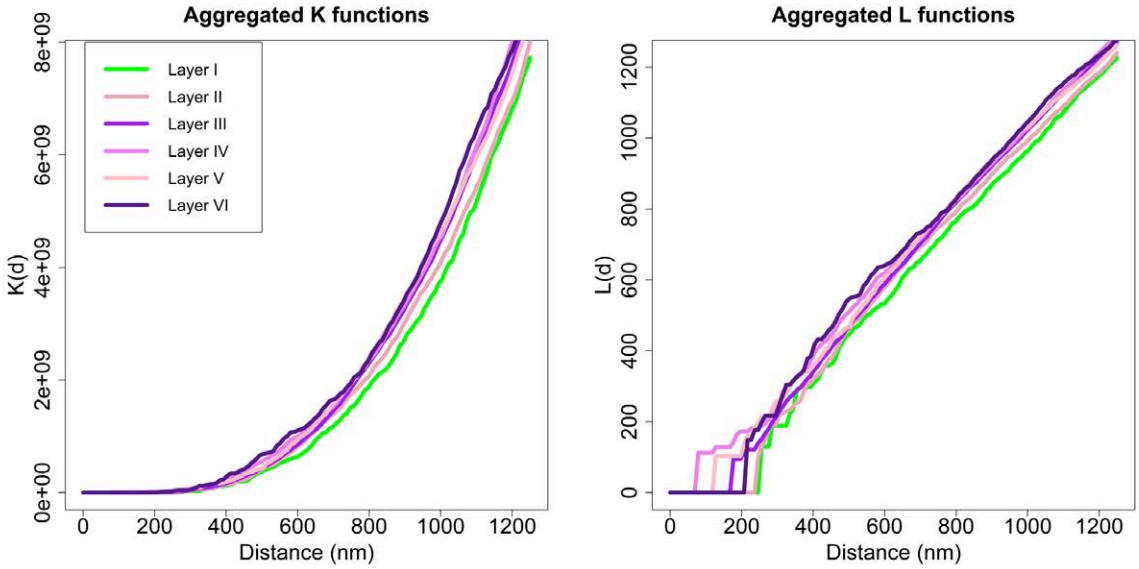


Figure 5.8: Aggregated K and L functions for each layer. The Diggle test found no significant differences between K functions of layers II, III, IV, V and VI (shown in different shades of violet). Layer I (green) is significantly different from other layers

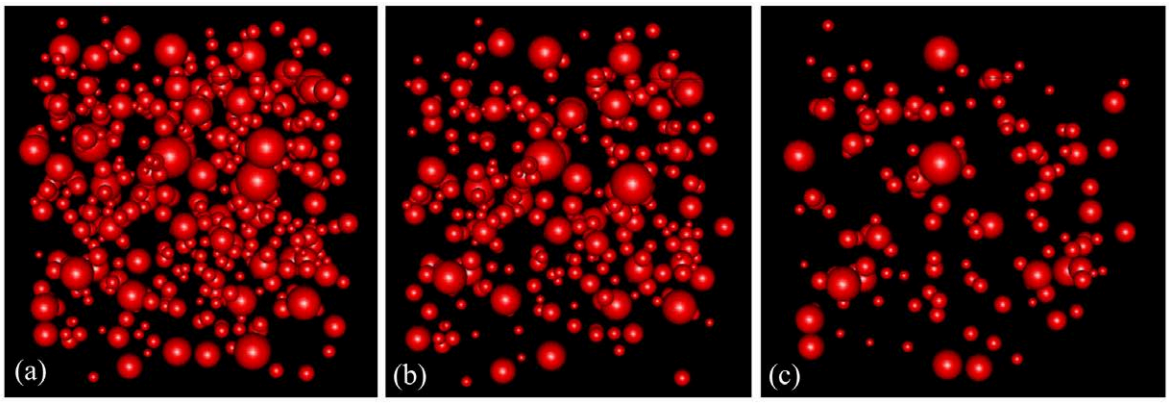


Figure 5.9: (a) RSA simulation with $\lambda = 1.4$ for the group of layers II, III, IV, V and VI. (b) Thinned RSA simulation, $\lambda = 0.932$, for sample 10 of layer III. λ estimated from the remaining nine samples of layer III. (c) Thinned RSA simulation, $\lambda = 0.457$, for sample 1 of layer VI. λ estimated from the remaining three samples of layer VI

5.6 Software

We developed a tool available at CeSViMa server[2] to process and analyze the 3D spatial distribution of synapses in the cerebral cortex (Fig. 5.10). The tool was developed using R software with a graphical user, based on *shiny*, an embedded web-interface, and it uses four of the summary functions most commonly applied in the analysis of spatial point processes, namely, the F, G, K and L functions (Section 2.2.2.2).

[2] http://vps136.cesvima.upm.es:3838/hbp/synapsesSA/

Table 5.3: Estimated intensity $\hat{\lambda}_{ij}$ for samples in layer II to VI using only the remaining samples of the same layer (Eq. (5.4)). μ_{global} and σ_{global} parameters are the same because layers II to VI form a group, and they were obtained using Feret's diameters of all samples of the group. We thinned RSA_{global} simulations modeled with $\lambda_{global} = 1.4$ and parameters μ_{global} and σ_{global} until we reached the estimated intensity $\hat{\lambda}_{ij}$ for each sample

			Density	Size (Feret's diameters)	
	Sample	Animal	$\hat{\lambda}_{ij}$	μ_{global}	σ_{global}
Layer II	1	w33	1.154	5.911	0.404
	2	w35	1.168		
	3	w35	0.981		
Layer III	1	w31	0.936	5.911	0.404
	2	w31	0.963		
	3	w33	0.941		
	4	w33	0.932		
	5	w33	0.939		
	6	w33	0.940		
	7	w33	0.934		
	8	w35	0.962		
	9	w35	0.919		
	10	w35	0.932		
Layer IV	1	w33	1.286	5.911	0.404
	2	w35	1.200		
	3	w35	1.186		
Layer V	1	w33	0.876	5.911	0.404
	2	w33	0.782		
	3	w33	0.821		
Layer VI	1	w33	0.457	5.911	0.404
	2	w35	0.466		
	3	w31	0.438		
	4	w31	0.508		

For each synapse, the 3D coordinates (x,y,z) of the centroid, its Feret's diameter and its layer must be provided. With this information the user can process and visualize the data from cortical synapses. The view supports zoom and rotation where each synapse is depicted as a sphere, using the Feret's diameter as the spherical diameter. The main tasks that can be performed with this software are: model the spatial distribution of the synapses to find out any possible distribution pattern; replicate, via simulations based on real data, samples of cortical synapses; and perform a layer comparison of synaptic density and distance to the nearest synapse.

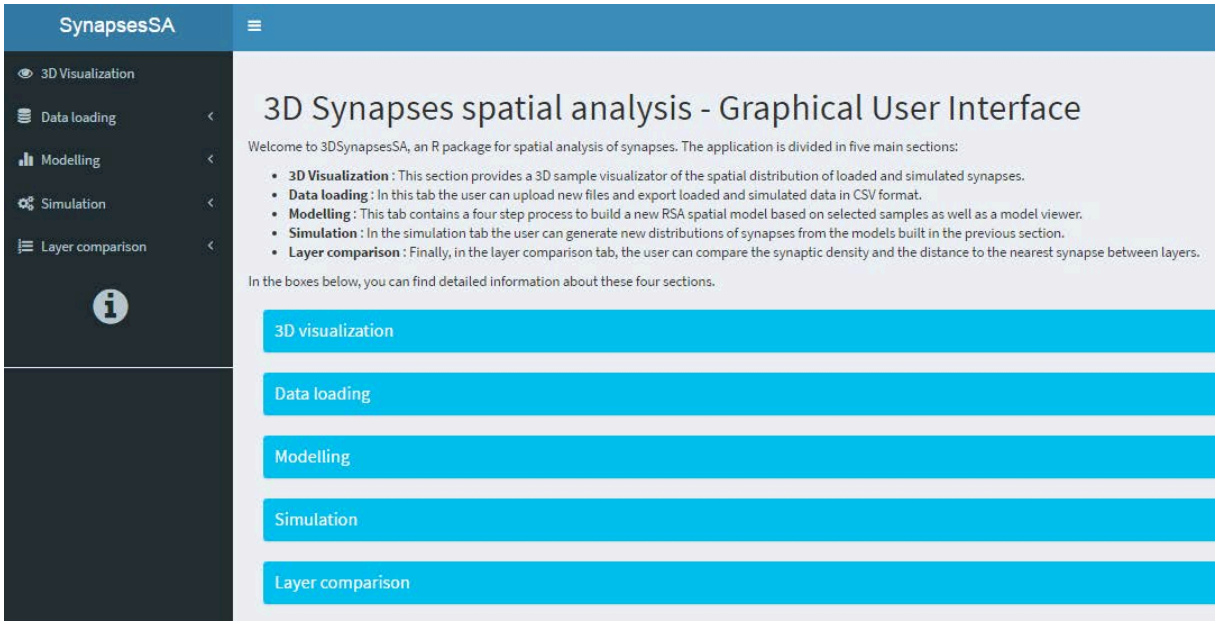


Figure 5.10: Home screen of the tool to analyze the 3D spatial distribution of synapses

5.7 Conclusions

The field of replicated point patterns is growing strongly due to technological advances, particularly in 3D sampling. In fact, much of the research on replicated point patterns is related to biological issues, including applications to neuroanatomical data [Baddeley et al., 1993, Burguet and Andrey, 2014, Burguet et al., 2011, Diggle et al., 1991, 2000, Jafari-Mamaghani et al., 2010, Myllymäki et al., 2012, Wager et al., 2004]. Indeed, neuroanatomical data in the form of spatial point patterns is fundamental for revealing the spatial architecture of the different brain regions at all levels of analysis, from light microscopy (e.g., spatial distribution of neurons) to electron microscopy (e.g., spatial distribution of synapses). In this chapter, we performed an analysis in the context of replicated point patterns by exploiting the fact that we have been able to obtain a relatively large number of samples containing the spatial distribution of synapses in the neuropil from several layers of the rat cerebral cortex. Using the Diggle test [Diggle et al., 1991, 2000] we detected groups of replicates (groups of patterns considered as instances of the same process) whose spatial distribution was found not to be significantly different. Then we modeled these groups using a global RSA replicated spatial point process. In order to collect and explain the variability in each group's synaptic density, we introduced a thinning procedure in the global model. We proposed a cross-validation technique for models within each group of replicates to honestly estimate the goodness-of-fit of the resulting models.

Our results confirmed the assumption that the spatial distribution of synaptic junctions in the neuropil is nearly random, with the only constraint that synapses cannot overlap in space —a scenario that can be modeled by an RSA process. This model had already been suggested for layer III synapses [Merchán-Pérez et al., 2014] and is now extended to

all neocortical layers. We found that the spatial distribution of synapses in all samples of each layer can be described by RSA processes. We also found that the spatial distribution of synapses in the neuropil of layers II to VI follows a common underlying RSA process with different synaptic densities. Interestingly, the results showed that the synaptic spatial distribution in layer I is slightly different than in other layers, suggesting that, although an RSA process suitably fits layer I synaptic distribution, the repulsion in the spatial distribution of synapses in this layer is slightly higher than in the other layers.

Since the synaptic density in the cerebral cortex changes with age, e.g., Bourgeois and Rakic [1993], DeFelipe et al. [1997], Rakic et al. [1986, 1994], and we used P-14 rats, the conclusion of this study regarding spatial distribution may not be applicable at other time points during development. Note, however, that the spatial distribution of synapses follows the same pattern in different cortical layers in spite of significant differences in their synaptic densities. Furthermore, preliminary results in the adult human cerebral cortex also suggest that the spatial distribution of synapses is nearly random [Blazquez-Llorca et al., 2013]. Therefore, random spatial distribution of synapses is probably a common general pattern of cortical synaptic organization. Nevertheless, further studies in other cortical areas, species and ages would be necessary to verify these conclusions.

The assumption that the distribution of synapses in the neuropil of layers I to VI follows an RSA model with different intensities (synaptic densities) per layer has several interesting implications. First, the position of a given synapse in the neuropil is practically independent of the position of neighbouring synapses, so they can be arbitrarily close to one another with the only physical constraint that they cannot overlap. Second, the density of synapses varies by layers and also locally. Importantly, early studies of the cerebral cortex proposed that the density of synapses was relatively constant throughout the cortical layers, as well as across different cortical areas and different species. This uniformity in synaptic density led O'Kusky and Colonnier [1982] to propose that it probably reflects the optimal number of synapses and that it may be due to some limiting metabolic or structural factor. However, most comparisons were only qualitative and not based on statistical analyses. It now appears that, using appropriate stereological counting methods (disector or size-frequency methods; see DeFelipe et al. [1999]), there are significant differences in the estimated number of synapses per volume between certain layers in several species (reviewed in DeFelipe et al. [2002a]). In this study, we also found using FIB/SEM that there may be significant differences between certain cortical layers. This method has the advantage that it provides the actual number of synapses per volume (instead of estimations) based on the analysis of single electron microscope images [Merchán-Pérez et al., 2009].

Our results showed no significant differences in the synaptic distribution between the different rats used in the study, and RSA processes properly described the spatial distribution of synapses in all cortical layers. This argues in favor of a common general principle of synaptic organization. However, the mean density of synapses across the six layers was significantly different, with the exception of layers I vs V and layers II vs III. This is an important observation in terms of connectivity, as these differences or similarities in density of synapses

between layers may provide us with some fundamental rules to generate virtual circuits in order to gain a better understanding of cortical organization. This also means that, due to physical constraints, the volume of the neuropil that the dendritic tree of a given neuron occupies may vary depending of the density of neurons in the layer where this neuron is located. In turn, its chances of establishing synapses would be greater the more neuropil volume it occupies. This idea was put forward by Von Economo [1926] in his interpretation of Nissl's observation in terms of the evolutionary significance of the differences between species in cortical neuronal density [Nissl, 1898]. Nissl observed that "in the mole and dog, cortical neurons were more crowded than in man". Von Economo proposed that the greater separation between neurons the richer the fiber plexus between them will be, increasing the chance for neuronal interactions. Thus, the larger separation of neurons in humans compared to other species could be construed as a sign of a greater complexity of the connections between neurons. Using this approach, several authors have identified an inverse relationship in the adult cerebral cortex between neuronal density and the number of synapses per neuron in different cortical areas/layers/species, but this principle does not appear to be generally applicable [DeFelipe et al., 2002a]. Since in this study we found no significant differences in the density of synapses in layer I vs V —the density of neurons in layer V is much greater than in layer I—, or between layer II vs III —the density of neurons in layer III is much less than in layer II —, this principle does not appear to be applicable to the 14-days-old rat somatosensory cortex either. In this regard it is important to keep in mind that the dendrites present in the neuropil of a given layer belong to both local neurons and neurons located below and above that layer, as dendrites, of pyramidal cells particularly, may cross several layers during their ascending course towards layer I, whereas their basal dendrites may invade the layer underneath, respectively. It follows that the number of synapses that a given neuron receives cannot be predicted solely on the basis of the synaptic density of the layer in which it is located.

Finally, the application of FIB/SEM to analyze the neuropil also revealed the existence of local variability in the synaptic density within each layer. This local variability would be the product of mere chance and can be explained (and modeled) by RSA processes. The between-layers variability, however, cannot be put down to chance, except possibly for the differences between layers I and V and between layers II and III. This would imply, as previously suggested [Merchán-Pérez et al., 2014], that spatial specificity in the neocortex is scale dependent. It is well known that at the macroscopic and mesoscopic scales the mammalian nervous system is a highly ordered and stereotyped structure, where connections are established in a highly specific and ordered way, like, for example, the connecting pathways of the visual system. Even at the microscopic level, it is clear that different areas and layers of the cortex receive specific inputs [Nieuwenhuys, 1994]. At the ultrastructural level, however, our results seem to indicate that the number and distribution of synapses follow a nearly random pattern. This could mean that, as the axon terminals reach their destination, the spatial resolution that they achieve is fine enough to find a specific cortical layer but not to make a synapse on a smaller target, such as a specific dendritic branch or dendritic spine

within that layer. For example, axon terminals from a certain thalamic nucleus reach specific areas and layers of the cerebral cortex but, once there, they would form synapses randomly among their possible targets to a greater or lesser extent depending on particular classes of the postsynaptic neurons.

Chapter 6

Three-dimensional network spatial analysis applied to spine modeling along dendritic networks

6.1 Introduction

Existing techniques for network spatial analysis (Section 2.3) assume that the network is two dimensional. In this chapter, we extend these techniques to the 3D space in order to model the spatial distribution of dendritic spines (for simplicity, spines) of pyramidal neurons.

Since spines are the main postsynaptic target of excitatory synapses in the cerebral cortex, many researchers are interested in ascertaining their spatial distribution within the two main dendritic domains of pyramidal cells: the apical and basal dendritic trees. The apical arbor stems from a main apical shaft whose origin is the upper pole of the pyramidal cell body. This apical dendrite is radially directed towards the pia mater and gives off a number of oblique branches. A system of large basal dendrites (generally, from three to six) emerges from the base of the pyramidal cell body and is directed laterally or downward. Generally speaking, proximal dendrites receive excitatory inputs from local sources (collaterals in the same area or from an adjacent area), whereas the distal apical tuft receives inputs from more distant cortical and thalamic locations [DeFelipe and Fariñas, 1992]. While the proximal portions of pyramidal cell dendrites are devoid of spines (approximately 10-15 μm from the cell body), there is a progressive increase in the density of spines. The highest densities are found at variable distances from the soma, depending on the cortical area and species. In the human temporal cortex, the highest density is found at a distance of 75-125 μm from the cell body. Thereafter, there is a progressive decrease towards the distal tips of dendrites, where the density is again low [Elson and DeFelipe, 2002].

Spines must necessarily lie on the dendritic shaft. Therefore, the application of network spatial analysis is appropriate. Some recent research [Baddeley et al., 2014b, Jammalamadaka et al., 2013] also used network spatial analysis to analyze spine distribution along dendrites.

However, using the justification that neurons in cell culture *in vitro* are almost flat, they ignored the third dimension and used a 2D projection. To the best of our knowledge, this is the first time that 3D network spatial analysis has been applied.

Taking advantage of the fact that we had several dendritic arborizations from each pyramidal cell, which can be treated as a group of instances of the same neuron, we also used replicated point patterns to detect differences and similarities between different pyramidal neurons and between apical and basal dendrites. As well as our research of Chapter 5, numerous works related to biological issues, particularly with applications to neuroanatomical data, use replicated spatial pattern techniques [Baddeley et al., 1993, Burguet et al., 2011, Diggle et al., 1991, 2000, Myllymäki et al., 2012, Wager et al., 2004]. These techniques are used here together with network spatial analysis for the first time.

The research included in this chapter has been submitted for publication [Anton-Sanchez et al., 2017c].

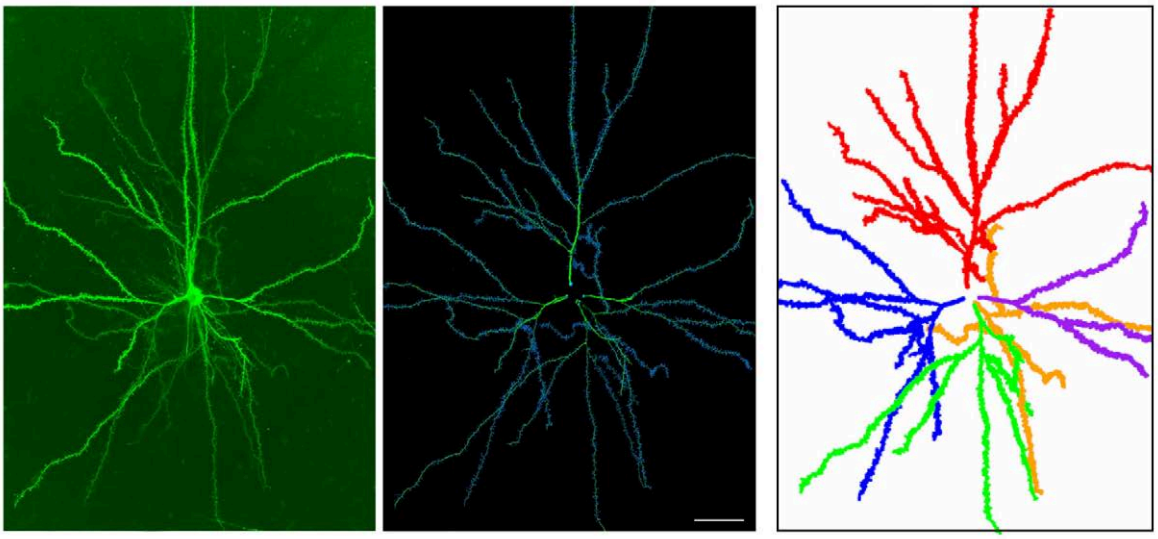
Chapter outline

This chapter is organized as follows. Section 6.2 describes the pyramidal cells involved in this study. Section 6.3 illustrates the statistical and computational methods used to carry out the 3D analysis of the spatial distribution of spines along the dendritic arborizations. Section 6.4 details the use of these methods with replicated point patterns. Section 6.5 describes the results for the analyzed pyramidal neurons. Finally, Section 6.6 includes a discussion and conclusions.

6.2 Data

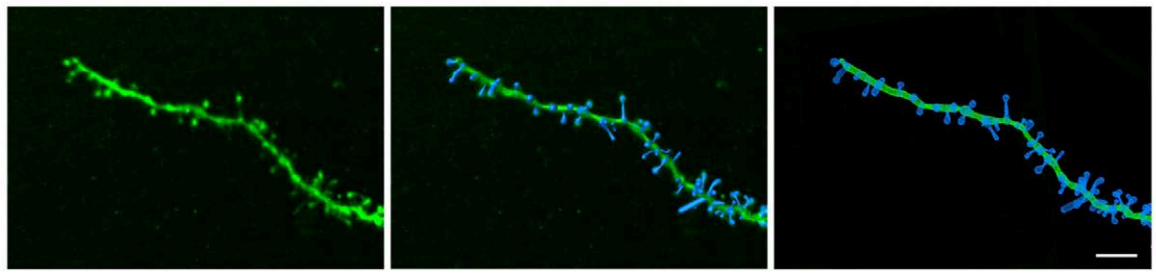
We analyzed five detailed and complete reconstructions of adult human pyramidal neurons that were intracellularly injected with Lucifer Yellow (LY) in layer III of the temporal cortex (area 20 of Brodmann) from two human males (aged 40 and 66) obtained at autopsy (2-3 h post-mortem) that died in traffic accidents. The brain samples were obtained following the guidelines and with the approval of the Institutional Ethical Committee. The tissue from these human brains has been used and described as histologically normal in previous studies [Blazquez-Llorca et al., 2010, Garcia-Marin et al., 2009]. Detailed information regarding tissue preparation, injection methodology and immunohistochemistry processing was described in Benavides-Piccione et al. [2013]. The injected cells were fully imaged at high magnification using the tile scan mode in a Leica TCS 4D confocal scanning laser attached to a Leitz DMIRB fluorescence microscope (Fig. 6.1). Consecutive stacks of images at high magnification (x63 glycerol) were acquired to capture dendrites along the apical and basal dendritic arbors. Using this method, some apical and basal dendrites near to the surface of the section are lost, but it has been estimated that at least two-thirds of the cell are preserved [Krimer et al., 1997]. In addition, the apical dendrites that run for more than 900 μm from the soma were not filled with dye, and therefore apical tuft was not included in the analysis. The dendritic arborization was reconstructed using Imaris 7.6.5, Filament Tracer module software

(Bitplane AG, Zurich, Switzerland). Therefore, we were able to place their spines, adjusting the position, length and volume of each spine individually (Fig. 6.2). We analyzed a total of more than 32,000 spines, 44% in apical dendrites and 56% in basal dendrites.



(a) (b) (c)

Figure 6.1: Example of one of the analyzed pyramidal neurons. (a) Confocal microscopy image of an intracellularly injected layer III pyramidal neuron of the human temporal cortex (Neuron 1 in Tables 6.1 and 6.2), visualized in 3D from high-resolution confocal stacks of images. (b) 3D reconstruction of the complete morphology of the cell shown in (a). (c) 3D reconstruction of the same neuron showing the apical dendrite in red and the four reconstructed basal dendrites in blue, green, orange and purple. We use the blue basal tree in (c) throughout the manuscript to illustrate the analysis performed. Scale bar (in (b)): 50 μm



(a) (b) (c)

Figure 6.2: Example of basal dendritic segment. (a) High magnification confocal microscopy image showing a basal dendritic segment from Neuron 1. (b, c) Reconstruction of the dendritic shaft and spines shown in (a) in a solid (b) and mesh (c) view

6.3 Network spatial analysis

Each dendritic arborization is a tree, i.e., a network with no cycles, in which all points are connected. Considering that spines can only lie on the dendritic shaft, we model spine distribution along the dendritic networks of pyramidal neurons in both basal and apical dendrites using network spatial analysis and, particularly, the geometrically corrected K function proposed by Ang et al. [2012] (see Section 2.3 for details).

To test whether the deviation between two summary functions, usually between the empirical summary function and the summary function of the model to be tested, is statistically significant, the standard approach is to use a Monte Carlo test based on envelopes of the summary function obtained from simulated point patterns (Section 2.2.4). We used global envelopes since we had no prior information about the range of spatial interaction. We calculated the envelopes by generating 19 simulations of the model to be tested, computing the summary functions of the simulated patterns. The global envelope is a region of constant width $2w_{max}$, where w_{max} is determined as the maximum absolute difference between the theoretical mean value of the summary function of the model to be tested and any of the summary functions of the simulated patterns. This corresponds to a Monte Carlo test with significance level $1/(1+19)=0.05$ [Diggle, 2003]. If the empirical summary function is completely contained in the envelope, the model is not rejected.

Existing computational techniques for spatial analysis along networks assume that the network is 2D [Baddeley et al., 2015, Okabe and Sugihara, 2012]. Although dendritic networks are 3D, recent research analyzing the distribution of spines along dendritic arborizations [Baddeley et al., 2014b, Jammalamadaka et al., 2013] ignored the third dimension, arguing that neurons in cell culture *in vitro* are more or less flat. They used a 2D projection to represent the spatial layout of the dendrites. In our case, cell reconstructions have a third dimension that should not be overlooked. For example, Fig. 6.3(a) shows the first basal dendrite of the first analyzed pyramidal neuron, clearly illustrating that the dendritic tree is not flat. We extended the functionality provided by the **spatstat** package [Baddeley, 2010, Baddeley and Turner, 2005] designed to manage 2D networks in order to handle 3D networks. Thus, we have performed the first spatial analysis along 3D networks. Eq. (2.12), (2.13) and (2.14) are applicable to 3D linear networks, although key values like $d_L(x_i, x_j)$ or $m(u, t)$ are much harder to compute taking into account the third dimension.

From the specifications of pyramidal neurons in .vrml format, we obtained the 3D axes of the dendritic arborizations and the spine attachment points (network and network events in the model, respectively, see Fig. 6.3(b)). After processing the .vrml files using R software, we used the **spatstat** package and the extensions that we implemented for the 3D analysis in order to represent the networks and analyze the distribution of the spines along dendritic networks. The networks (dendrites) and network events (spines) of the five human pyramidal neurons analyzed in this study can be found on Figshare[1].

[1] https://figshare.com/articles/3D_human_pyramidal_dendrites_with_spines/4892630

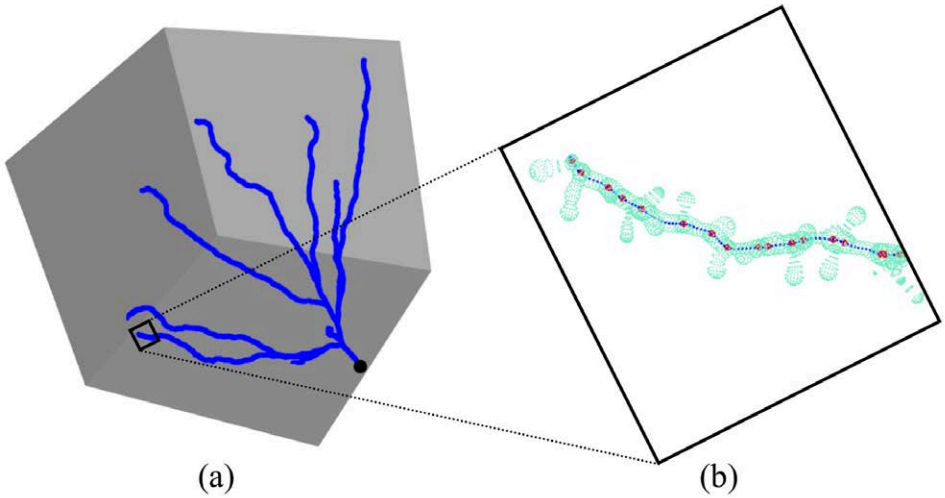


Figure 6.3: First basal arborization of Neuron 1 illustrating the analysis (some of its characteristics are shown in Table 6.2). (a) 3D representation of the basal network. The tree root is shown in black. (b) Zoom of a small part of the same dendrite (end of the dendritic segment shown in Fig. 6.2) to illustrate the computation of the dendrite axis (i.e., the network in dark blue) and the attachment points of the spines (network events in red) from the reconstruction provided in the .vrml file (light blue)

6.4 Replicated spatial point patterns

For replication in groups with which we are concerned, there are g different experimental groups. In group i ($i = 1, ..., g$), we observe m_i point patterns that can be regarded as independent replicates within this group. Replication provides for the analysis of the differences in spatial point patterns between and within groups for decision making on whether there are statistically significant differences between groups. We had several basal dendritic trees from each pyramidal neuron that can be regarded as replicates of the same observation (the neuron). By conducting an analysis in the context of replicated point patterns, we investigated whether there were significant differences between the basal arborizations of the same pyramidal neuron and between different pyramidal neurons, that is, we performed a study with g=5 groups, where each group was composed of the basal dendrites of each pyramidal neuron. We were also interested in analyzing whether there were significant differences in the distribution of spines along the apical and basal networks, that is, in performing a study with g=2 groups (one group with apical dendrites and the other with all basal dendrites of all neurons). We tested the null hypothesis of similarity between groups with the studentized permutation test proposed by Hahn [2012] (Section 2.4).

As mentioned in Section 2.3, the geometrically corrected K function compensates for the geometry of the network, whereby the corrected K functions obtained from different point patterns in different networks are directly comparable. Therefore, this is the first time that the geometrically corrected K function (Eq. (2.13) or (2.14)) has been applied in the context of replicated point patterns to compare different groups of 3D point patterns on

linear networks. We used the studentized permutation test provided in **spatstat**, which we expanded to be used with the K function on linear networks. Because the geometrically corrected K function stabilizes the variance we use the studentized permutation test given by Eq. (2.19). We used 1000 permutations for the test (default value).

6.5 Results

Table 6.1 shows some important characteristics of the apical dendrites of each of the five analyzed pyramidal neurons: number of spines, total length of the network, average number of points per unit length in the network, *circumradius* and number of branching points (complexity measure of the dendritic tree). Table 6.2 shows the same information for basal dendrites, grouped according to the pyramidal neuron to which they belong. Apical dendrites are clearly more complex than basal dendrites, as a comparison of the mean number of characteristics shown in Tables 6.1 and 6.2 patently shows. While the mean number of spines in apical dendrites is 2845, it is 1074 in basal dendrites. The mean length is also much greater in apical than in basal arborizations: 2497.25 μm and 951.85 μm, respectively. The same applies to the mean number of branching points (20 in apical networks vs 6 in basal networks).

Table 6.1: Description of the analyzed apical dendrites. The table shows the number of spines n, total length of the network $|L|$ (in μm) average number of points per unit length in the network $n/|L|$, *circumradius* R (in μm), and number of branching points in the dendrite #BP

Neuron	n	$\|L\|$	$n/\|L\|$	R	# BP
1	2750	2182.42	1.26	237.48	16
2	3019	3073.93	0.98	325.49	22
3	2195	1852.01	1.19	231.55	18
4	2599	2123.39	1.22	261.55	19
5	3660	3254.50	1.12	332.11	23
Mean	2845	2497.25	1.16	277.64	20

The first property to be analyzed is the intensity or average density of points along the network. Spatial inhomogeneity can be conflated with clustering between points. Therefore, it is important to analyze any evidence of spatial variation in point intensity. Indeed, the distribution of dendritic spines along the dendrites of pyramidal cells has been shown not to be uniform in different cortical areas and species (reviewed in Elson and DeFelipe [2002]). We defined the distance function to the tree root r by the shortest path in the dendritic network $d_L(u,r) = d_L(u)$, $u \in L$, and we analyzed the relationship $\lambda(u) = \rho(d_L(u))$, where ρ is an unknown function to be estimated. Kernel smoothing methods can be used to estimate the intensity function as discussed in Okabe et al. [2009] and in McSwiggan et al. [2016]. Fig. 6.4 shows the kernel-smoothing estimate of function ρ of the first basal dendrite (see Table 6.2), confirming that spine intensity depends on the distance to the cell body when it is analyzed

Table 6.2: Description of the analyzed basal dendrites grouped by neuron. The table shows the number of spines n, total length of the network $|L|$ (in μm), average number of points per unit length in the network $n/|L|$, *circumradius* R (in μm), and number of branching points in the dendrite #BP

Neuron	n	$\|L\|$	$n/\|L\|$	R	# BP
1	1889	1527.45	1.24	257.74	8
1	584	615.79	0.95	208.72	3
1	1214	957.66	1.27	225.92	5
1	1272	1156.29	1.10	235.34	7
2	287	391.74	0.73	137.46	4
2	791	928.87	0.85	220.23	11
2	270	327.29	0.82	139.79	2
2	715	914.87	0.78	184.14	9
3	852	664.26	1.28	237.90	3
3	2149	1467.43	1.46	209.18	8
3	662	594.30	1.11	177.59	4
4	778	556.49	1.40	169.34	4
4	1487	1282.15	1.16	213.84	8
4	1004	834.83	1.20	202.17	5
5	2244	2231.19	1.01	309.20	14
5	978	879.04	1.11	204.76	7
5	1088	851.73	1.28	211.03	6
Mean	1074	951.85	1.10	208.49	6

along the complete network. The results for all analyzed dendritic networks for both basal and apical dendrites were very similar.

We used the cumulative distribution function (CDF) test to study the hypothesis of independence of intensity on a spatial covariate (distance to the cell body, in our case). The CDF test was first described by Berman [1986] (in the context of spatial data, using Kolmogorov-Smirnov statistic). For a linear network, the test compares the observed distribution of the values of the covariate in the network events with the null distribution of the covariate at random points on the network. For all analyzed apical and basal dendritic networks, we found strong evidence of the dependence of spine intensity on distance to the cell body (a p-value $< 10^{-6}$ was obtained in the CDF test in 90% of the cases, the highest p-value=0.00591 being in one of the basal dendrites of Neuron 2).

In view of the above results, we fitted an inhomogeneous Poisson model for each dendritic network, in which the spine intensity $\lambda(u)$, $u \in L$ depends on the distance to the cell body. Considering the shape of function ρ (Fig. 6.4), we decided to adjust a log-quadratic intensity in d, that is, $\lambda(u) = \exp(\theta_0 + \theta_1 d(u) + \theta_2 d(u)^2)$, where θ_0,θ_1,θ_2 are the parameters for estimation. Fig. 6.5(a) shows the estimation of the geometrically corrected 3D inhomogeneous K_{LI} function (Eq. (2.14)) for the basal example and 5% critical envelopes based on 19 simulations of an inhomogeneous Poisson process with log-quadratic intensity in d. The chart shows that

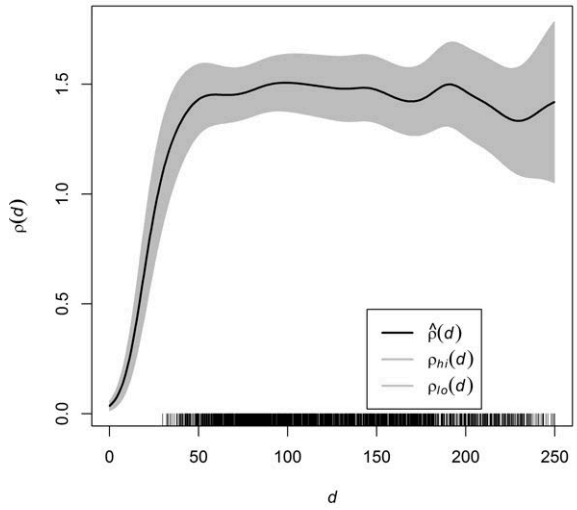


Figure 6.4: Estimate of the intensity of the first basal arborization of Neuron 1 as a function of the distance (in μm) to the tree root

the spatial distribution of the spines along the network is consistent with an inhomogeneous Poisson process. Fig. 6.5(b) is analogous to Fig. 6.5(a) using the 2D implementation of K_{LI} provided in the **spatstat** package instead. Although the fit is not bad, it is not as good as in 3D where the estimation of the K_{LI} function is almost completely superimposed on the Poisson function for all distances d. Fig. 6.5(c) shows the result of applying the 3D K function to the spines of the same basal arbor, using only spine spatial coordinates and ignoring the network. This figure incorrectly suggests that spines are strongly clustered. The error stems, however, from the choice of a mistaken null hypothesis because the envelopes are computed from 3D CSR simulations without considering the network.

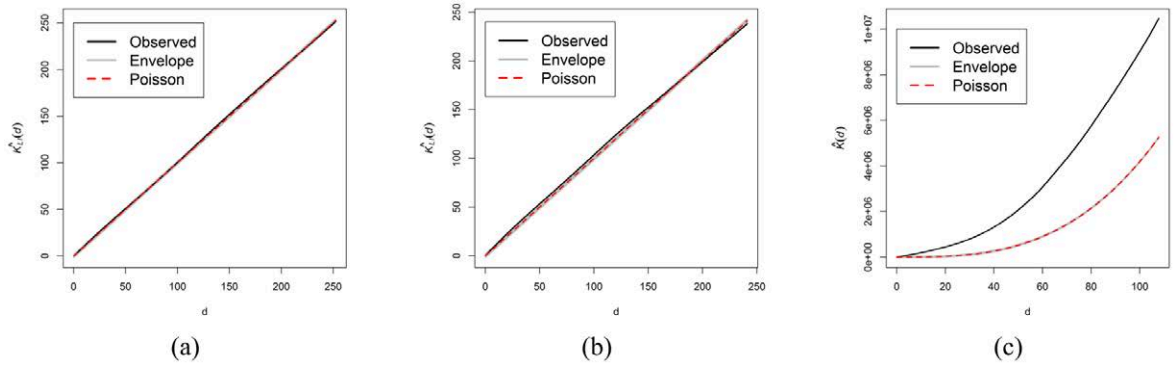


Figure 6.5: 5% critical envelopes of the first basal arborization of Neuron 1. (a) Estimation of 3D geometrically corrected inhomogeneous K_{LI} function. (b) Estimation of 2D geometrically corrected inhomogeneous K_{LI} function. (c) Estimation of 3D K function ignoring the network (the envelope is just below the red dotted line)

The results were similar for all analyzed dendritic trees, suggesting that there does not appear to be any evidence of clustering or regularity of dendritic spines after considering spatial inhomogeneity. For three of the analyzed basal networks, however, the estimation of the K_{LI} function lay slightly below the lower boundary of the envelope at long distances, indicating that there were fewer points within distance d of an arbitrary point than within an inhomogeneous Poisson process with log-quadratic intensity in d. Conversely, the K_{LI} function estimations of two of the apical networks remained outside the envelope at long distances but above the upper boundary of the envelope, suggesting that points tended to be closer than within an inhomogeneous Poisson process at long distances. This could mean that spine distribution differs slightly in basal and apical arborizations as the distance to the cell body increases. Then we used the studentized permutation test to analyze if there were differences between different pyramidal neurons and between basal and apical dendrites (Section 6.4).

First, we compared groups of basal arborizations of different neurons, that is, we applied the test with g=5 groups (neurons) using their previously estimated 3D K_{LI} functions in the range of distances [0,134.70]. We used a maximum distance that was 2% lower than the minimum *circumradius* R of all the networks used in the test. We obtained a p-value of 0.808. Thus, we concluded that there were no significant differences in spine distributions along the basal trees among the five neurons at the analyzed distances (Fig. 6.6(a)). Then, we applied the test again, forming a group with all basal arborizations of the five pyramidal neurons and another group with all apical arborizations. The resulting p-value was 0.109. Therefore, we concluded that there were no statistically significant differences between spine distributions of these two groups up to a distance of 134.70 μm (Fig. 6.6(b)).

We wanted to analyze if there were differences in spine distribution between basal and apical dendrites taking into account distances farthest from the cell body. In the studentized permutation test, each group should contain at least three patterns to achieve reasonably precise estimates for the within-group variance of the estimates. To do this, we decided to remove Neuron 2 from the analysis because two of its basal trees had a small *circumradius* (137.46 μm and 139.79 μm, respectively), and we repeated the analysis with all the other neurons up to a distance of 165.96 μm (distance that was 2% shorter than the minimum *circumradius* R of the remaining basal networks). First, we compared groups of basal arborizations. We obtained a p-value of 0.565 for g=4 groups (Neurons 1, 3, 4 and 5). Therefore, we concluded that there were no significant differences in spine distribution along basal trees in the range of distances [0, 165.96] either. We applied the test again, forming a group with the 13 basal dendrites analyzed in the previous step, and another group with all apical dendrites (all with a *circumradius* longer than 165.96 μm). We obtained a p-value of 0.045, and, with the usual 5% significance level, we concluded that, contrary to previous cases, there were significant differences in spine distribution along apical and basal dendritic networks considering distances up to 165.96 μm. Fig. 6.6(c) suggests that apical dendritic spines are more clustered than basal dendritic spines as the distance from the cell body increases.

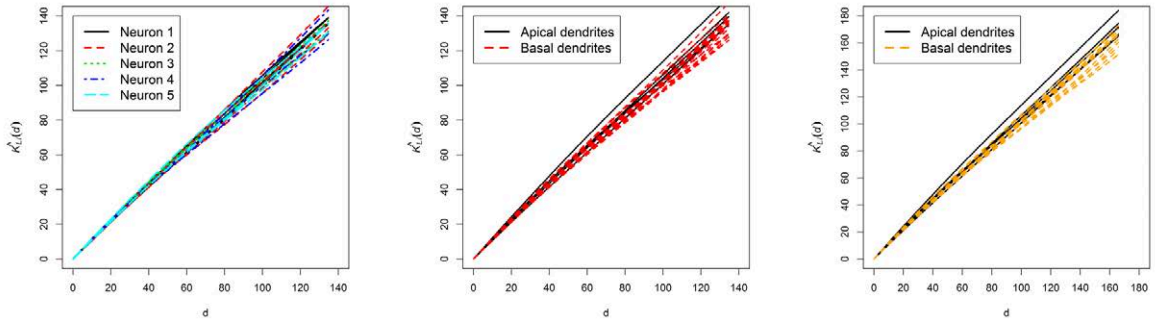


Figure 6.6: Estimated 3D K_{LI} functions used in the studentized permutation test. (a) Estimated 3D K_{LI} functions of all basal networks grouped by neuron in the distance range [0,134.70] (g=5 groups, p-value=0.808). (b) Estimated 3D K_{LI} functions of all apical networks forming a group and all basal networks forming another group in the distance range [0,134.70] (g=2, p-value=0.109) (c) Estimated 3D K_{LI} functions of all apical networks forming a group and the basal dendrites of Neurons 1, 3, 4 and 5 forming another group in the distance range [0,165.96] (g=2, p-value=0.045)

6.6 Conclusions

We analyzed the spatial distribution of spines along both basal and apical dendritic networks of human pyramidal neurons. To do this, we used network spatial analysis, implementing methods to analyze 3D linear networks for the first time. We studied whether there were differences in the spatial distribution of spines between different pyramidal neurons and between basal and apical dendrites, using replicated point patterns in conjunction with network spatial analysis. To do this, we took advantage of the geometrically corrected K function in order to compare the corrected K functions obtained from different point patterns in different networks [Ang et al., 2012].

A non-constant intensity of points can be easily confused with clustering between points. Therefore, we set out to thoroughly analyze spine intensity in dendritic networks. We found that there was spatial variation in spine intensity which depended on the distance to the cell body. Therefore, we fitted an inhomogeneous Poisson model. The model used appeared to adequately explain the spatial distribution of spines along dendritic networks in most cases. Additionally, we found that there were no significant differences in spine distribution between basal trees of the same and different neurons. This suggests that dendritic spine distribution in the basal dendritic arbors conforms to common rules. Neither did we find statistically significant differences between basal and apical trees up to distances of 134.70 μm away from the cell body. Excluding the smaller basal networks and analyzing distances farthest from the cell body (up to 165.96 μm), however, we did find significant differences in the distribution of spines along basal and apical networks. The spines of apical dendrites are more clustered than basal spines. Therefore, not only do apical and basal dendritic arbors have distinct morphologies, but the rules of spine distribution are also different. These observations further

emphasize that synaptic input information is processed differently within these two dendritic domains. Note, however, that, as stated in Baddeley et al. [2014b], the analysis performed may be very sensitive to the fitted intensity, especially in tree-like networks. Because of this, it might be interesting to examine other models that further characterize the spatial distribution of spines along the basal and apical networks, especially at distances further from the cell body.

Recent research analyzing the distribution of spines along dendritic networks yielded different results. Jammalamadaka et al. [2013] concluded that spine intensity is completely spatially random. Baddeley et al. [2014b], who studied only one example pattern of the cells analyzed by Jammalamadaka et al. [2013], found that different branches may have different patterns of spine distribution. The dendrites investigated in these studies belong to cell culture *in vitro* rat dissociated hippocampal neurons, while we analyzed adult human neocortical pyramidal cells obtained at autopsy. Thus, differences in spine distribution are not comparable because of possible differences between human and rat pyramidal cell structures, as well as between the experimental approaches used to obtain the tissue, neuron labeling and methods of analysis.

This is the first work to take into account the third dimension of spatial analysis on linear networks. This approach has been applied to the example of spines along dendritic networks but could be useful for the spatial analysis of other real-world 3D networks. The shortest path distances in the network are much harder to compute than Euclidean distances in traditional spatial statistics. Besides, the inclusion of the third dimension considerably increases the computational load especially with increased network complexity. As future work, we would like to improve the efficiency of the implementation developed for 3D networks. Also, it would be interesting to consider the network (dendrite) volume and the possibility of events (spines) occurring on the surface of the network with volume. In this case, 3D analysis could be even more useful, although the methodology would need to be expanded. The inclusion of marks in the analysis, such as some spine characteristics like length, volume or type [Arellano et al., 2007, Benavides-Piccione et al., 2013], may also be beneficial for elucidating important aspects of the spatial distribution of spines. Finally, alterations of spine distribution are common in the diseased brain (for a review see Fiala et al. [2002]). Thus, the analysis performed in this study may shed light on the possible alterations of neuronal circuits in brain diseases.

Chapter 7

Nearest neighbour distances to describe dendritic morphology organization

7.1 Introduction

In this chapter we use the average nearest neighbour ratio R, that measures the degree of clustering of points in a given volume, to study the relationship between spatial input distributions in dendrites and the respective dendritic morphology. The measure R has been used in a wide variety of scientific disciplines, such as physics, biology, geography and astronomy [Bishop, 2007a,b, Chandrasekhar, 1943, Clark and Evans, 1954]. In particular, it has been applied to graph theoretical problems such as minimum spanning trees (MSTs) [Dry et al., 2012], but to the best of our knowledge it has not yet been considered to characterize neuronal morphology.

The primary function of dendritic trees is to collect inputs from other neurons in the nervous tissue [Chklovskii, 2004, Stepanyants and Chklovskii, 2005]. Different cell types play distinct roles in wiring up the brain and are typically visually identifiable by the particular shape of their dendrites [Ramón y Cajal, 1899]. However, so far no branching statistic exists that reliably associates individual morphologies to their specific cell class [Ascoli et al., 2008, Torben-Nielsen and Cuntz, 2014], indicating that we have not yet identified the morphological features that are characteristic for the differences in how neurons connect to one another. Theoretical considerations have provided systematic qualitative insight into the question of how dendrite shape relates to specific connectivity. Dendrites are thought to collect their inputs using the shortest amount of cable and minimizing conduction times in the circuit [Cuntz et al., 2007, 2010, Ramón y Cajal, 1899, Wen and Chklovskii, 2008] and they have been proposed to maximize the possible connection repertoire [Wen et al., 2009]. Of the possible connections that a neuron could make by anatomical proximity only a small, relatively invariable number become functional synapses [Fares and Stepanyants, 2009]. But it has

generally been proposed that the connection probability between a dendrite and an axon can be determined by the amount of anatomical overlap between the two [Binzegger et al., 2004, Hill et al., 2012, Peters and Payne, 1993]. Furthermore, dendrite shape has been linked to the number of synapses based on the optimal wiring assumptions described above, linking total dendrite length and the number of synapses that determine the morphology [Cuntz et al., 2012]. This leads to the question whether specific axonal arrangements or synapse distribution patterns may lead to specific typical dendritic morphological characteristics [Cuntz, 2012].

A useful concept to relate dendritic trees with their underlying connectivity comes from extended MSTs that connect a set of target points while minimizing total cable length and path lengths in the tree toward the root where signals get integrated [Cuntz et al., 2007, 2010]. Such MSTs were shown to produce accurate dendritic morphologies when the corresponding target points were selected adequately [Beining et al., 2017, Cuntz et al., 2008, 2010, 2012]. This approach has previously linked both the distribution of target points to actual synapse locations, and as well the number of branching points and terminal points [Cuntz et al., 2012].

Here we analyze the measure R in branching and terminal points of real dendrites to estimate how regularly the dendrites spread out. Then, we use MST-based morphological models generated on different target point distributions to compare the spatial distributions of branching and terminal points with the underlying distribution of target points as a proxy for their corresponding synaptic input distributions.

The research included in this chapter has been submitted for publication [Anton-Sanchez et al., 2017b].

Chapter outline

The chapter is organized as follows. Section 7.2 describes the average nearest neighbour ratio R. Section 7.3 details how we compute the supporting volume of a point cloud in order to estimate R. Section 7.4 introduces the inconveniences resulting from edge effects in the estimation of R and our Monte Carlo approximation to avoid them. Section 7.5 describes our implementation of a point pattern generator with specific R values. Section 7.6 details how the analysis of nearest neighbour distances in dendritic morphology is performed, both in real dendrites and in MST-based morphological models. Then, Section 7.7 reports the results of the nearest neighbour analysis in real and synthetic branching structures. Finally, discussion and final comments are included in Section 7.8.

7.2 Average nearest neighbour ratio R

The average nearest neighbour (NN) ratio $R = \bar{r}_0/\bar{r}_E$ compares the observed average NN distance $\bar{r}_0$ between a set of N points with the expected average distance $\bar{r}_E$ between nearest neighbours under the assumption of a uniform random distribution (with the same number of points covering the same total area or volume). This approach was first described by Hertz [1909] and Clark and Evans [1954].

R provides a measure of the clustering of the points in a point cloud C. Concretely, the closer the points are to a random (Poisson) distribution, the closer to 1 the value of R becomes (as the values of $\bar{r}_0$ and $\bar{r}_E$ are more similar). Values of R less than 1 correspond to clustering ($\bar{r}_0 < \bar{r}_E$). When all points overlap (R=0) the most clustered condition is reached. For values of R greater than 1, points are further apart than it would be expected for a random distribution ($\bar{r}_0 > \bar{r}_E$). In 2D arrangements, the most dispersed situation is the one in which the points are spaced on a triangular lattice, yielding a value of R=2.1491 [Clark and Evans, 1954]. The measure R has the advantage of being easily interpretable. For example, R=0.5 indicates that nearest neighbours are, on average, half as distant as expected under random conditions.

Formally, for a finite point cloud C, i.e., a set of N points, the average NN distance is

$$\bar{r}_0 = \frac{1}{N} \sum_{\substack{i=1 \\ i \neq j}}^{N} \min\{d_{ij}\},$$

where d_{ij} denotes the Euclidean distance between the i-th and the j-th point in C. This is the numerator in the definition of $R = \bar{r}_0/\bar{r}_E$. The denominator in R is the expected NN distance $\bar{r}_E$ for a Poisson process that can be analytically computed as $\bar{r}_E = 1/2\sqrt{\lambda}$ in the 2D case and as $\bar{r}_E = \Gamma(4/3)/\sqrt[3]{4\pi\lambda/3}$ in the 3D case, where $\Gamma(\,\cdot\,)$ is the gamma function and λ is the point density, i.e., the mean number of points per unit area or volume V. For a uniform random distribution, an unbiased estimator of λ is $\hat{\lambda} = N/V$. Thus, to obtain the point density, an accurate estimate of the supporting volume V of the point cloud C is required.

7.3 Computing the supporting volume of a point cloud

In order to estimate R, a volume V supporting a given point cloud C needs to be estimated. The most common way to do this is to use the convex hull of C. Yet, with this choice the supporting volume is overestimated if it is non-convex, which results in incorrect values of R. Better estimates of R are obtained using α-shapes. α-shapes were devised to characterize the shapes of point clouds and can be seen as an extension to the notion of a convex hull [Edelsbrunner and Mucke, 1994, Edelsbrunner et al., 1983].

Formally, to any given finite point cloud C in 2D or 3D Euclidean space a one parameter family of curves or surfaces S_α called α-shapes can be constructed, with $\alpha \in [0, \infty]$. By construction, S_∞ corresponds to the convex hull and S_0 to the point cloud itself. For any finite C, S_α is a finite set and a smallest value α_0 exists (called critical value of α) such that S_{α_0} is connected and contains all points of C. Furthermore a smallest value $\alpha_k < \infty$ exists for which S_{α_k}=S_∞. The α-spectrum of C is defined as the monotonically increasing, finite sequence of values $(\alpha_i)_{0 \leq i \leq k}$, $0 \leq \alpha_i \leq \infty$, $\alpha_i \leq \alpha_{i+1}$ for which each S_{α_i} is a distinct α-shape and the shapes do not change between two consecutive values α_i, α_{i+1}. To compute what we call a 'tight hull' around a point cloud C we selected the center point $\alpha_{k/2}$ of the α-spectrum, for which we rounded the index $k/2$ to the next integer value. Especially for point clouds

with non-convex supporting volumes, this yielded much better estimates of the true volume and thus less biased values of R. Fig. 7.1 shows an example of the convex hull compared to the tight hull of point clouds with non-convex support, and the resulting R values.

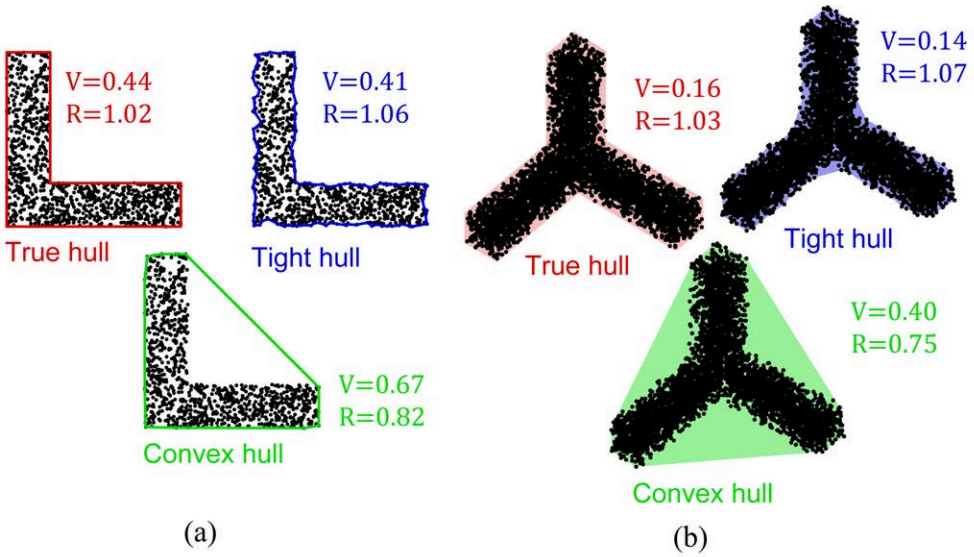


Figure 7.1: Examples of the implemented approximation to compute the R measure through a tight hull in 2D for an L-shape with 1,000 points (a), and in 3D for a double L-shape with 3,000 points (b). (Left) Expected R using the correct volume V (computed analytically). (Middle) V and R computed from the convex hull. (Right) V and R computed for a tight hull

7.4 Edge effects and Monte Carlo approximation of R

Assume that for a given point cloud C we estimate a supporting volume V using α-shapes as described in the previous section. Note that the expected NN distance of a uniform random distribution used for calculating R is usually obtained analytically, assuming the case of infinitely many points contained in an unbounded volume. Yet, in practice all our volumes V containing a given point cloud C are finite and bounded. The spatial analysis of any finite region implies that there is a boundary but most spatial statistic theories are based on the assumption of an infinite space, so the analysis of a bounded region gives rise to what are known as boundary or edge effects. Two well-known techniques correcting for such boundary induced biases are the toroidal edge correction and the border area edge correction (see Section 2.2.2.2). Analytical bias corrections were also derived for convex planar surface areas as supporting volumes [Ripley and Rasson, 1977].

Without correction for edge effects, nearest neighbour distances will be positively biased [Donnelly, 1978]. Since we usually work here with point counts with non-convex areas and volumes (see for example Fig. 7.2) and many of them do not have a high number of points, instead of computing $\bar{r}_E$ analytically from an estimate of the point density and using an edge

correction technique, we decided to use a Monte Carlo (MC) simulation approach to estimate $\bar{r}_E$. This approach does not attempt to eliminate edge effects but rather to repeatedly simulate the phenomenon of interest (Poisson point clouds in our case) for a given study region and estimate the distribution of a test statistics (average NN distance in our case) in the presence of edge effects.

For a point cloud C consisting of N points contained in a volume V, we first computed $\bar{r}_0$ as the observed mean NN distance in C. We then sampled M=100 uniform random point clouds within V, each containing N points. For each of those point clouds we computed its average NN distance and obtained an estimate of $\bar{r}_E$ as the mean of the M=100 values of the simulations. No edge corrections are necessary here because all the average NN distances are biased by the same edge effects. To check the correctness and convergence properties of this approach, we generated point clouds with known R values and compared them to the R values estimated from our MC based method (Fig. 7.3).

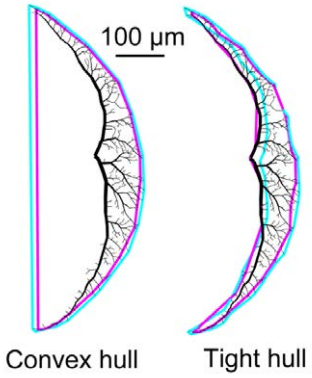


Figure 7.2: Example of non-convex dendrite. Magenta shows the hull around branching points and cyan shows hull around terminal points. (Left) Convex hull. (Right) Tight hull

7.5 Point pattern generator with target R

In order to study a wide range of different spatial organizations we implemented a procedure for obtaining point clouds with specified R values (Fig. 7.4). First, we generated a number N of random points within a square. We then iteratively estimated the R value using our MC method and moved each point in the direction of or away from its NN, depending on whether the target R was smaller or greater than the current R, respectively, (Fig. 7.4(b)) until the target R value was reached. The shift was proportional to the difference between the current R value and the target R, i.e., the closer the values of both, the smaller the movements. Fig. 7.4(c) shows the number of iterations required for our algorithm to reach different values of R, from highly clustered (R=0.2) to highly regular (R=1.8) given 1,000 initial points. We obtained very similar results for different numbers of points. For very small or very large values of the target R, it was increasingly expensive to find a corresponding point configuration.

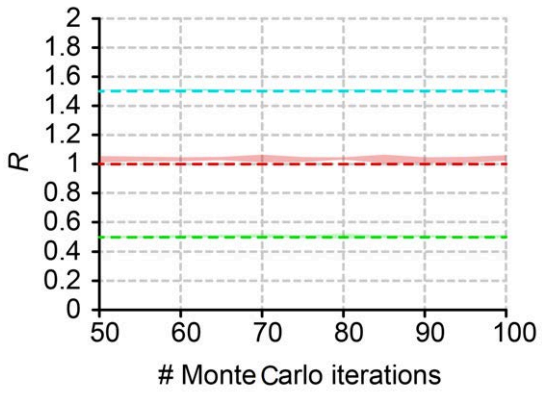


Figure 7.3: Estimated R values via MC for point clouds with known R in a square area with N=1,000 points. Dashed lines show the true R values. The mean and standard deviation of 10 estimated R values are shown in green (R=0.5), red (R=1) and cyan (R=1.5). Here we used from 50 to 100 Monte Carlo iterations to obtain each estimated R

7.6 Nearest neighbour distances in dendritic morphology

7.6.1 R values for dendrites from NeuroMorpho.Org

To evaluate the measure R on real cells, we obtained a number of reconstructions of dendritic trees from NeuroMorpho.org [Ascoli et al., 2007], version 7.0 (January 2016) using the TREES toolbox[1], an open-source software package for MATLAB (Mathworks, Natick, MA) [Cuntz et al., 2011]. Specifically, we chose reconstructions belonging to eight cell classes, namely cortical pyramidal cells, hippocampal pyramidal cells, dentate granule cells, motoneurons, retinal ganglion cells, cerebellar Purkinje cells, fly larva dendritic arborization (da) neurons and fly Lobula Plate tangential cells (TCs). The first four classes were 3D cells and the last four classes were 2D.

For selecting the reconstructions, we obtained all reconstructions from NeuroMorpho.org that were classified as either having 'moderate' or 'complete' reconstructions of their dendritic trees and belonged to the control group (to exclude mutant cells). We then grouped all reconstructions by archive and sorted out archives that contained poor reconstructions by manual visual inspection as well as archives containing one cell only. This left us with a number of reconstructions of each cell type, denoted in parentheses in the following list: cortical pyramidal cells (3786), hippocampal pyramidal cells (399), dentate granule cells (154), motoneurons (83), retinal ganglion cells (322), cerebellar Purkinje cells (15), fly da neurons (68), fly TCs (55).

After downloading the reconstructions in .swc format, these were read into and pre-processed using the TREES toolbox. For each reconstruction, this process involved deleting the soma and the axon if present and then re-joining the parts of the tree if the deletion

[1] www.treestoolbox.org

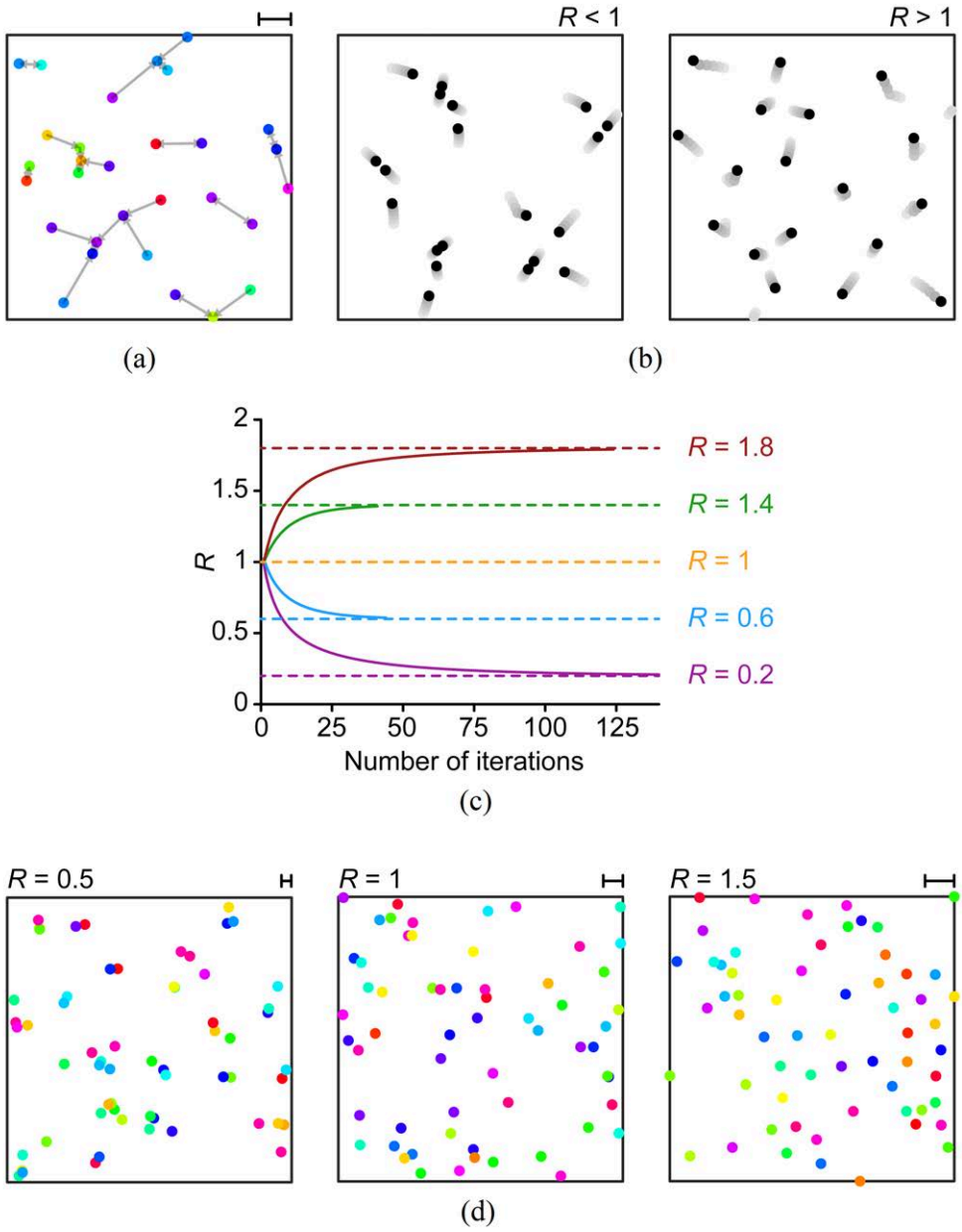


Figure 7.4: Point pattern generator for target average NN distance. (a) Illustration of NN distances for uniform random Poisson distribution of 20 colored points and arrows indicating the individual NN. Scale bar (upper right) shows average nearest neighbour distance $\bar{r}_0$. (b) Movements of the points in the first 5 iterations (from light grey to dark grey) of our point pattern generator towards a clustered (left) and a more regular (right) pattern. (c) Number of iterations required in our algorithm to obtain different values of R (0.2 - purple, 0.6 - blue, 1 - yellow, 1.4 - green and 1.8 - red) from an initial point cloud with 1,000 random points. Dashed lines show target R values. (d) Sample distributions of 50 points for R=0.5 (left), R=1 (middle) and R=1.5 (right). Scale bars show average nearest neighbour distance $\bar{r}_0$

operation yielded several roots, followed by a final removal of higher order multifurcations. This process was not necessary for fly TCs that were available with the TREES toolbox [Cuntz et al., 2008]. Da neurons were furthermore subdivided into da class I-IV cells and TCs into horizontal system northern (HSN) cells, horizontal system equatorial (HSE) cells, and vertical system (VS2, VS3, and VS4) cells. For each tree we then computed a number of statistics: total dendrite length, number of branching points, mean branch order of branching and terminal points, mean branch angle, mean asymmetry, mean path length, R for the set of branching points (R_{BP}), R for the set of terminal points (R_{TP}), volume of the convex hull, volume of the tight hull and cable density as total length per volume in the tight hull. The tight hulls as well as their volumes needed for estimating R were computed using α-shapes as described previously.

7.6.2 Morphological models connecting points with different R values

Using the point pattern generator described in Section 7.5 we generated a large number of point clouds in 2D or 3D spaces. Planar arrangements were fixed to 200 μm x 200 μm and 3D arrangements were set to 200 μm x 200 μm x 200 μm. A large variety of number of points (50-800 points) and R values (R_{Input}, 0.2-1.8) were computed. We subsequently computed morphological models based on optimal wiring principles that connected these point clouds.

In this context, we considered that the simulated target point clouds were the positions of putative synapses and we computed synthetic branching structures connecting those targets with minimal resources using the extended MST algorithm described in Cuntz et al. [2010] and the algorithms available in the TREES toolbox. Briefly, optimal wiring minimizes both total cable length and the path length from any point along the tree to the root, using a balancing factor bf to weigh the second cost (that is, total $cost = cable\ length\ cost + bf \cdot path\ length\ cost$). For bf=0 the algorithm only seeks to minimize the total cable length while for large bf it seeks also to minimize the length of the connections from the root to any point. Values of bf greater than 0 and less than 1 represent a mixture of the two objectives that are realistic for real dendrites. As a further constraint, we did not allow multifurcations (more than two daughter branches at each branching point) in the computed synthetic trees.

The minimization was achieved via a greedy minimum spanning tree algorithm [Prim, 1957]. We computed synthetic dendritic trees from all the point clouds, connecting the points to a root in the center and using bf values from 0.2 to 0.8. We obtained 100 trees for each individual condition (point density, R and bf value). For each synthetic dendritic tree, R_{Input} of its target points was known (since the inputs were obtained using the point pattern generator), and we estimated R values of its branching points (R_{BP}) and terminal points (R_{TP}) in order to study the relation between these measures. In addition, we analyzed the relation between R_{Input} and other branching statistics commonly used to describe dendritic morphology. Specifically, we studied the total length, the number of branching points, the mean path length from the root of branching and terminal points, the mean branching angle, the mean branching order and the mean asymmetry at the branching points of each synthetic dendritic tree. The asymmetry for each branching point was defined as the ratio of $v1/(v1+$

$v2$) for $v1 < v2$, where $v1$ and $v2$ are the counts of terminal points in each of the two daughter branches. All statistics were computed using the TREES toolbox.

7.7 Results

7.7.1 R values for dendrites from NeuroMorpho.Org

We first calculated R for the set of branching and terminal points in real dendrites using our MC based approach to estimate how regularly the dendrites spread in the circuit (Fig. 7.5).

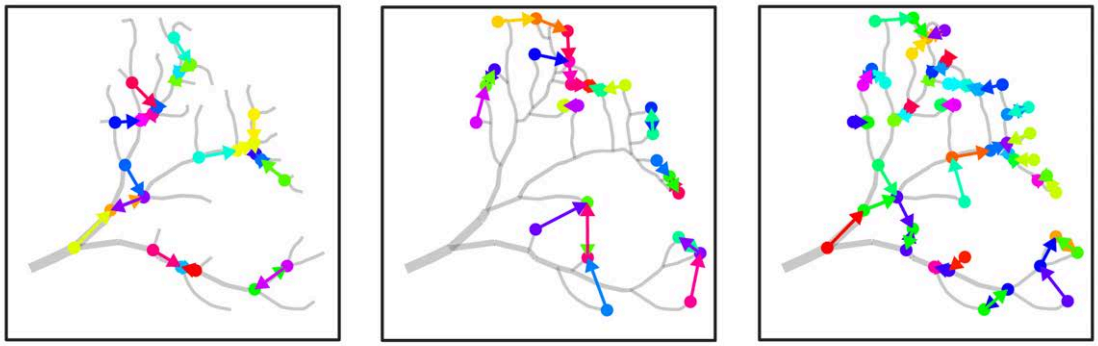


Figure 7.5: Sketch describing nearest neighbour distances in branching and terminal points. Sample dendrite and nearest neighbours (colored arrows) between branching points (left), terminal points (middle) and branching and terminal points (right)

The mean estimated values of R in four 3D and four 2D cell types varied widely (Fig. 7.6). For almost all cases we observed a tendency of R_{TP} being slightly larger than R_{BP}. Considering 3D dendrites, the spatial distribution of branching and terminal points was most regular in dentate granule cells, followed by cortical pyramidal cells, hippocampal pyramidal cells and finally motoneurons; the latter, on average, exhibited near random distributions with R close to 1 (Fig. 7.6(a)). In the case of the four planar cell types (Fig. 7.6(b)), dendritic arborization (da) neurons in the fly larva were well characterized by the clustering of their branching points. The spatial organization of the terminal points of Lobula Plate tangential cells (TCs) in the fly appeared to be similar to cerebellar Purkinje cells, but branching points were clustered more strongly in TCs than in Purkinje cells. Retinal ganglion cells, a large inhomogeneous group of cell types, exhibited comparably more regular distributions of branching and terminal points with larger R values than the other planar cell types that we studied.

It is important to note that all eight populations in Fig. 7.6 were composed of subgroups with strong differences in their functional role in the nervous system. Moreover, morphologies within the separate subgroups were partly obtained in different species, preparations and developmental ages. To illustrate the effect this can have on the analysis, we dissected fly

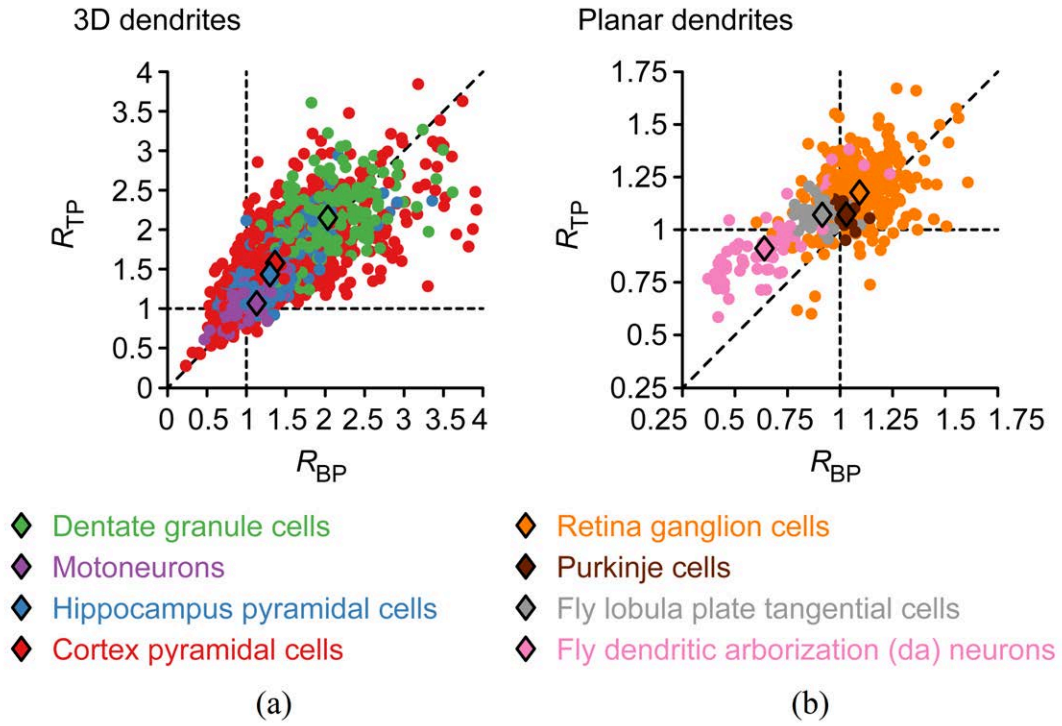


Figure 7.6: Nearest neighbour distances in 3D and 2D dendrites of real neurons. (a) R values of branching points (x-axis) and terminal points (y-axis) for four different populations of 3D dendrites. (b) Similar analysis as in (a) for four populations of planar dendrites. Data were obtained from NeuroMorpho.Org [Ascoli et al., 2007]. Colored dots correspond to individual reconstructions and diamonds represent mean values for each cell type

da neurons and TCs into their respective characteristic subgroups (Fig. 7.7). Da neurons are known to subdivide into morphologically distinct classes (I-IV) and, apart from the classes I and II, these can be separated into clusters (Fig. 7.7(a)) corresponding to their specific R values. In particular, class III da neurons with their large number of small terminal segments (STS) exhibited small R values consistent with the clustering of branching and terminal points due to these STS. On the other hand, sub-classes of TCs (two types of horizontal system cells - HSN and HSE, and three types of vertical system cells - VS2, VS3 and VS4) did not separate into different clusters according to their R values (Fig. 7.7(b)). Terminal points were more regularly distributed than branching points in all TC classes but all R values were close to 1, indicating random distributions. This was not surprising since TCs were previously characterized in detail using morphological models and shown to have similar inner branch rules even though their spanning areas are easy to distinguish [Cuntz et al., 2008].

The statistic R for branching and terminal points is therefore a useful measure to distinguish between cell classes and characterize the relationship between dendritic tree structure and input architecture. However, it remains to be shown that the use of this local statistic in dendritic morphology is not simply an altered version of another traditional branching statistic. In order to test this and to check whether the input architecture as measured by R

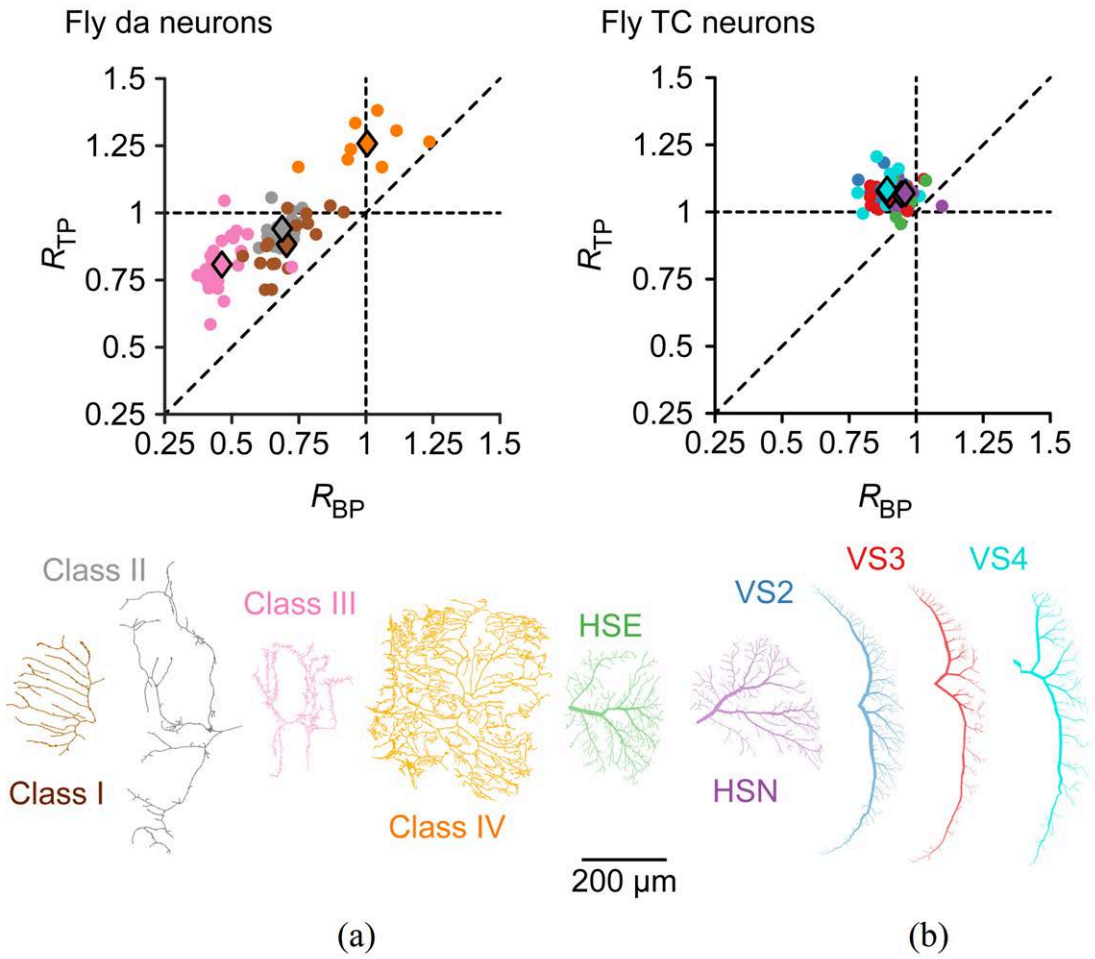


Figure 7.7: R values of fly da neurons and TCs subdivided into individual classes. (a) Similar plots as in Fig. 7.6 but subdividing da neurons into their four different classes (I-IV). (b) Similar subdivision as in (a) but for fly TCs. Here the five difference cell types exhibit similar R values. Note that the scale is different than in Fig. 7.6. Individual morphologies are shown to visualize the differences in how regularly the branches are distributed in the different classes

is reflected in other branching statistics, we computed the correlations between R and other commonly used statistics in 3D (Fig. 7.8(a)) and 2D (Fig. 7.8(b)) dendrites. We found that R_{BP} and R_{TP} do not have strong correlations with other typical branching statistics of dendritic trees in both cases. Since R_{BP} and R_{TP} were different in distinct cell types and were weakly correlated with other branching statistics, we postulate that these measures are a useful addition to the collection of branching statistics used to classify dendritic morphology.

7.7.2 Morphological models connecting points with different R values

In order to estimate how the clustering structure of input locations affects the clustering of branching and terminal points, we generated morphological models targeting different sets of

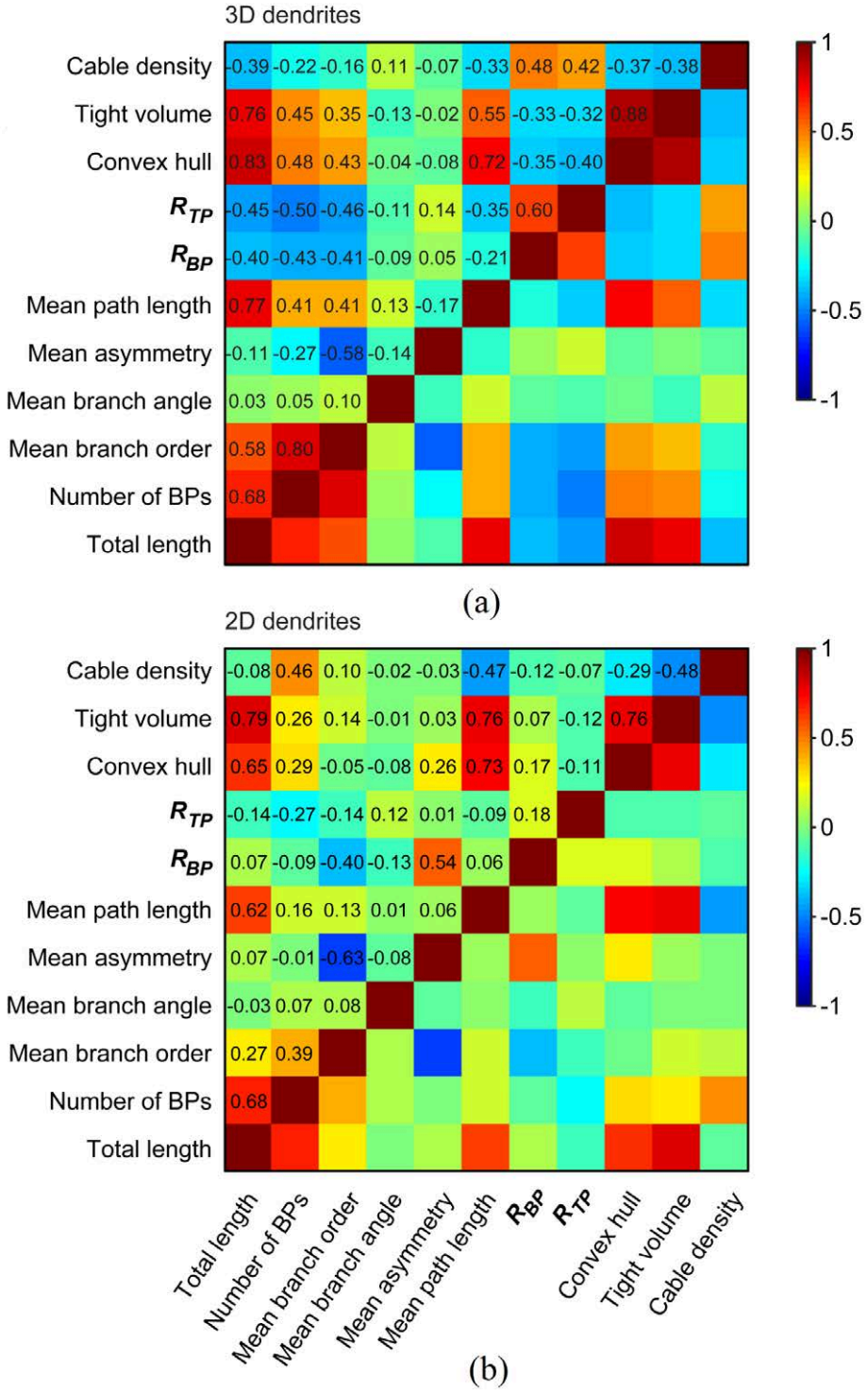


Figure 7.8: Correlation matrix between R values and other branching statistics. (a) Correlation matrix of R_{BP} and R_{TP} with other typical branching statistics in the lumped 3D cells from Fig. 7.6(a). (b) Similar correlation matrix but using the 2D cells of Fig. 7.6(b)

input points with specified values R_{Input} of the statistic R. For this, we use the point pattern generator described in Section 7.5. Dendrites were considered as tree structures connecting these target points [Cuntz et al., 2010, 2012]. Fig. 7.9(a) shows planar sample trees obtained from connecting 100 target points in a 200 μm x 200 μm square with different R_{Input} values, using the extended MST for different values of bf (see Section 7.6.2).

Generating morphological models on different sets of carrier points with specific values of R_{Input}, we found that with higher R_{Input} the trees became denser and the branches were more regularly distributed. Compared with R_{Input}, branching and terminal points describing the dendritic geometry were more regularly distributed in all cases (Fig. 7.9(b) for 3D cases similar to Fig. 7.9(a)) with much larger values in R and almost no values below 1. Furthermore, the spatial organization of branching and terminal points was clearly different: in line with reconstructions of real dendritic trees, R_{TP} values were consistently larger than R_{BP}. As might be expected, more regularly distributed inputs generally resulted in more regular branching structures. For higher point densities ($N > 100$) the results were similar, but R values of branching and terminal points were more similar to the R values of the input configuration. We obtained similar results in the 2D case with the exception that the distribution of branching and terminal points for high point densities was less regular than the input target point distribution, while in the 3D case both distributions tended to become similar for high densities. Overall, we believe that constructing synthetic versions of real dendritic trees with more detailed morphological models would therefore be useful for inferring the underlying spatial organization of the synaptic inputs.

Apart from its impact on the R values of branching and terminal points, it is interesting to study the impact of R_{Input} on branching statistics typically used to characterize dendritic trees (Fig. 7.10). As was the case for R_{BP} and R_{TP} in real dendritic tree reconstructions, R_{Input} was weakly correlated with other branching statistics, suggesting that input architecture is not well captured by traditional branching statistics whereas R_{BP} and R_{TP} would be useful measures for this and to classify dendritic morphology accordingly. However, both total length and number of branching points increased reliably with R_{Input}, requiring the minimum spanning tree to use more cable to connect the points that are more widely spread and more branches to reach out to all distributed inputs in space. This correlation clearly affected the scaling behavior that was previously observed between number of inputs and total length as well as between number of inputs and number of branching points [Cuntz et al., 2012]. Here, the previously reported 2/3 power between these measures was not affected by R_{Input}, but a clear increase in total length was observed as an offset in the relationship (Fig. 7.11). MST-based dendrites connecting target points with an increased R_{Input} required much more cable length. This is consistent with a correlation of around 0.4 observed in Fig. 7.8 between R values and the cable density in real 3D dendrites.

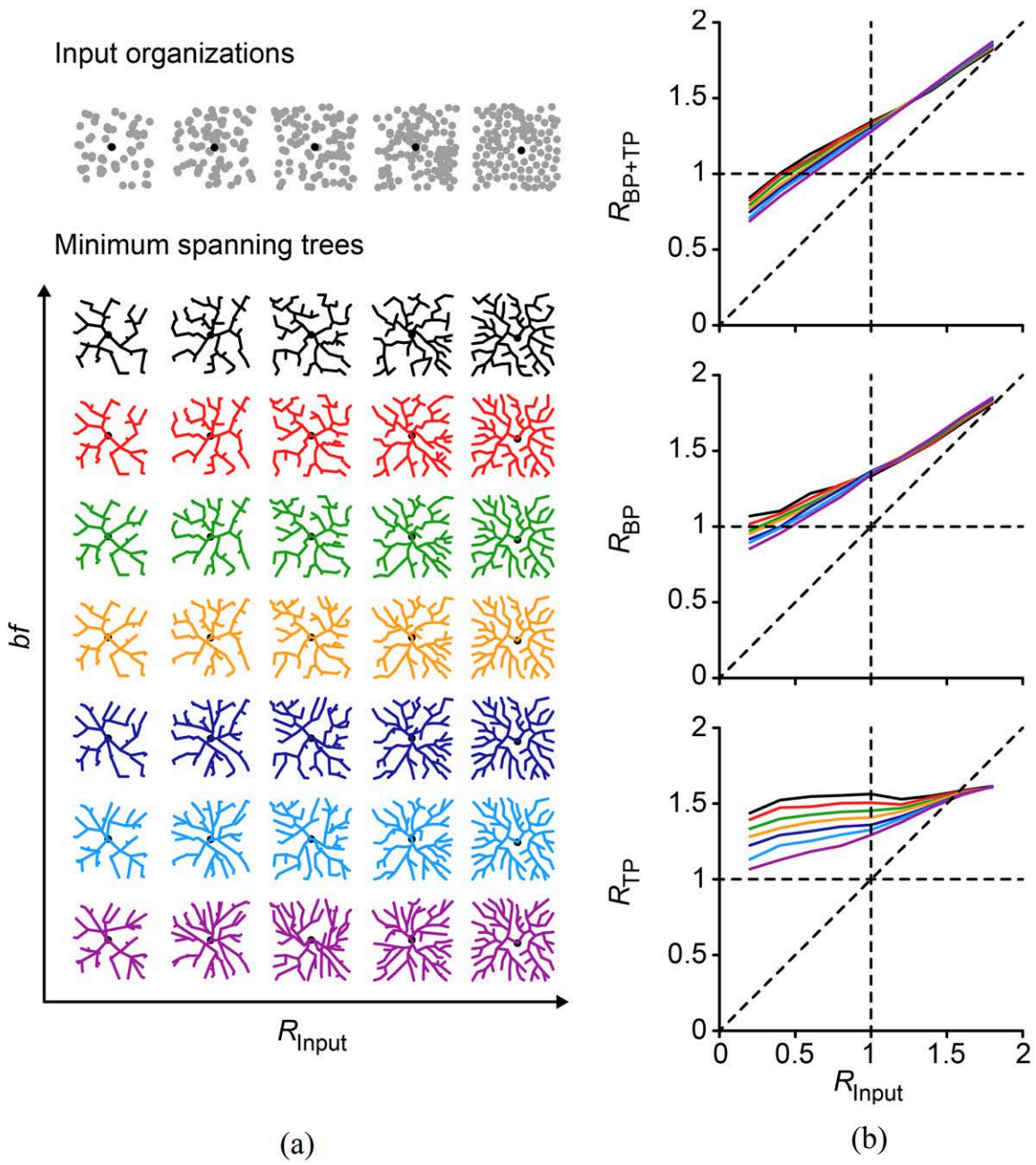


Figure 7.9: Relation between nearest neighbour distances in the input distribution and in the branching and terminal points. (a) Top row: 100 points (grey) distributed with different values of R (0.2, 0.6, 1, 1.4 and 1.8 from left to right) using our point pattern generator. Synthetic trees to optimize wiring for different bf values are shown below each point cloud (from 0.2 to 0.8 in 0.1 increments from top to bottom), in 2D for better visualization. (b) Relationship between R_{Input} and R_{BP} as well as R_{TP} for all morphological models in (a) (average of 100 trees for each individual condition) for the 3D morphological models

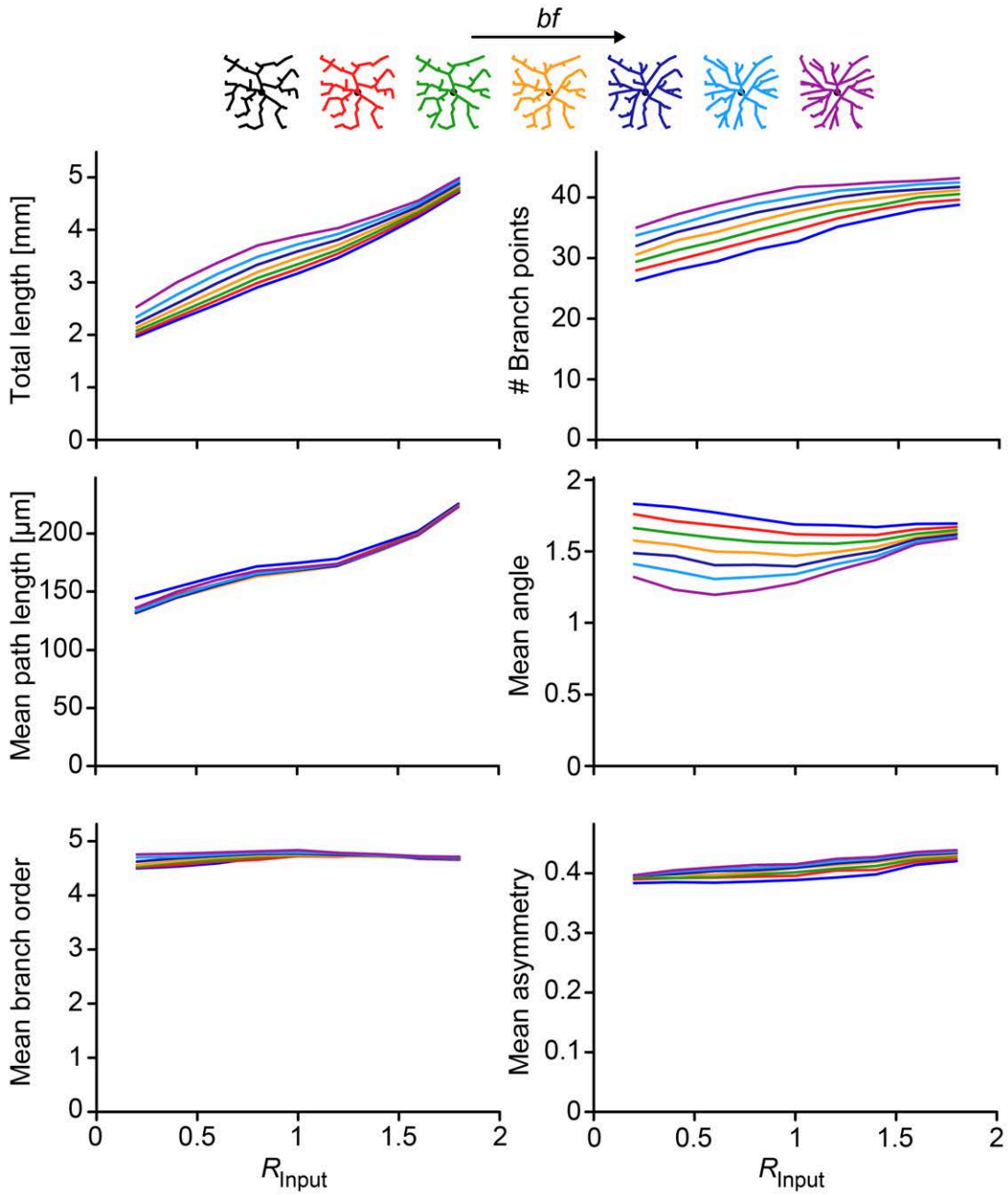


Figure 7.10: Relation between R_{Input} and dendritic branching statistics in the morphological model. Total dendrite length, number of branching points, mean path length from any point to the root, mean angle at branching points, mean branch order for any point on the dendrite and mean asymmetry at the branching points for all cases in Fig. 7.9. Colors indicate the same different values of bf in the morphological model with sample morphologies plotted from Fig. 7.9 at the top of the figure

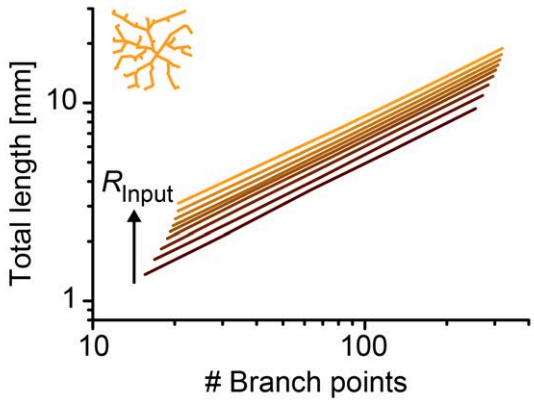


Figure 7.11: Scaling relation of total dendrite length. Total dendritic length vs. number of branching points in the 3D morphological model of Fig. 7.9 (bf=0.5). Increasing R is indicated with lighter color, same values as in Fig. 7.9. Results were similar with different bf values

7.8 Conclusions

We have presented a new branching statistic R, which is based on the average nearest neighbour (NN) distance between points of a given set, capturing their clustering structure. Specifically, R is defined as the ratio of the observed average NN distance to the one expected in a matching random point cloud. This makes R independent of the absolute scale of the dendritic arbor, but rather captures the clustering characteristic of the branching and terminal points and allows for comparison of cells of different sizes. We found that the measure allowed to distinguish dendritic trees from different cell classes for which the local statistics of the spatial input distribution differed. The values of R computed for the sets of branching points (R_{BP}) and terminal points (R_{TP}) of reconstruction of real dendritic trees correlated little with most other commonly considered branching statistics, indicating that these measures provide new descriptive power for dendritic trees that was not captured by existing measures. Using morphological models, we then found that overall R_{BP} and R_{TP} attained higher values compared to the value of the input distribution (R_{Input}). This indicates that dendritic branching structures are more regularly spread than the inputs that they collect. We also showed that in spatial distributions with higher values of R_{Input}, the total length of the dendrite increased dramatically. Overall, we expect the proposed measure R to be able to predict certain features of their input organization for given dendritic tree types better than the existing branching statistics, such as for example how regularly inputs are spread. As more realistic morphological models become available based on minimum spanning trees, as is the case for example for TCs [Cuntz et al., 2008] and dentate gyrus granule cells [Beining et al., 2017], this information can be further refined.

One issue when computing R that has been given little attention in the literature so far is the finding that a naive calculation yields a biased result due to edge effects. Our approach

to remedy these adverse effects was to use Monte Carlo (MC) simulations to predict the expected NN distances of uniform distributions numerically, using instances of Poisson point clouds. This MC based approach would most likely be useful in further studies beyond the scope of dendritic morphology since point processes in small volumes will necessarily exhibit similar important edge effects.

For most of the different types of real cells that we analyzed, we estimated values of R close to or greater than 1, indicating a more regular distribution than uniform random. Our results showed that also for the case of synthetic dendrites, the spatial distribution of dendritic branching structures is more regular than the distribution of the inputs that they collect, and that the higher the input density, the more similar both distributions become. Specifically, R values of branching and terminal points greater than 1 corresponded to slightly lower R_{Input} values, which indicated that inputs were potentially distributed almost randomly. These results are consistent with the spatial analysis carried out in Chapter 5 where we concluded that the 3D spatial distribution of synapses is close to uniformly random, with the only constraint that they cannot overlap. Furthermore analyses of dendritic spines, on which the majority of the excitatory synapses in the brain are established, showed that their spatial distribution is close to random [Morales et al., 2014].

There are several ways in which the measure R could be generalized that were not the focus of this study. First of all, for simplicity we assumed point clouds with uniform homogeneous densities when computing R. This can be extended to the non-homogenous case by including local estimates of point densities. This would lead to a localized version of measure R. Second, we only considered one nearest neighbour per point. This can easily be extended to neighbourhoods of higher order, containing the k-th nearest neighbours for each point, $k \geq 2$. Both of these extensions are subject of future studies.

Overall, we presented a new statistic R for dendrites that allows to relate their morphologies with the specific individual input organization that a neuron implements. It has low correlation with most commonly used statistics of dendritic branching and is extendable in several ways, providing a useful new statistic for the classification of dendritic trees. Furthermore, the MC based approach for small point clouds might be of interest to application in other areas of research.

Part IV

CONTRIBUTIONS TO NETWORK DESIGN OPTIMIZATION

Chapter 8

Network design with degree- and role-constrained minimum spanning trees

8.1 Introduction

We define a variation on the DCMST problem described in Section 3.2, which we call the degree- and role-constrained minimum spanning tree (DRCMST) problem. A DRCMST is a DCMST where we determine a priori the role of each node in the tree by choosing among the following: root node, intermediate node and leaf node. It may be useful to constrain the role of the nodes in network design. In computer networking, for example, the service has to reach the leaf nodes, and these nodes are clearly different from the central processor (which has a fixed number of ports). In such a network the cost could be associated with distances between nodes or with the material costs needed to connect nodes. A DRCMST could also be useful in a business network, for example, for the design of the project staff structure, where we would differentiate between the project manager (root), middle managers (intermediate nodes) and the rest of the team who are not in charge of any other staff (leaf nodes). The cost of this problem might be associated with team member preferences for project managers.

The DRCMST problem is NP-hard, because it contains the particular case where we determine one root node and one leaf node with the constraint that the degree of each node has to be less than or equal to two. This is equivalent to the shortest Hamiltonian path problem between two nodes. In addition, a forest rather than a single tree can be built, i.e., we do not limit the number of root nodes to one so we can solve more complex problems, e.g., we might design several computer networks and several business networks by simultaneously considering several central processors and several project managers in the above examples, respectively.

We introduce a new permutation-based representation for building forests of DRCMSTs. One permutation simultaneously encodes all the DRCMSTs in the forest. Due to problem

complexity, we address a wide variety of DRCMST problem instances using the evolutionary computation techniques detailed in Section 3.3. Individuals of the populations are encoded with the proposed representation.

The research included in this chapter has been published in Anton-Sanchez et al. [2017a].

Chapter outline

The organization of this chapter is as follows. Section 8.2 formally describes the DRCMST problem. Section 8.3 introduces the proposed permutation-based representation to encode forests of DRCMSTs. This representation is used to approximate a variety of DRCMST instances using the evolutionary computation algorithms described in Section 8.4. Section 8.5 details the characteristics of the 30 simulated test problem instances used to compare the performance of the different evolutionary computation techniques for the problem of finding forests of DRCMSTs. Section 8.6 analyzes the results and compares the algorithms. Section 8.7 illustrates the construction of forests with DRCMSTs in a real-world application, specifically trans-European transport network design. Finally, some discussion and conclusions are provided in Section 8.8.

8.2 Problem definition

A DRCMST is a DCMST where the role of the nodes in the tree is determined a priori (by the user/expert). In a DRCMST problem, we define three subsets of nodes R, I and L for root nodes, intermediate nodes and leaf nodes, respectively, where each node $v \in V$ must belong to one and only one subset, $\{R, I, L\}$ is a partition of the set of nodes and $R \neq \emptyset$. Note that the following conditions must be met: $d_v \geq 1\ \forall v \in R$, $d_v \geq 2\ \forall v \in I$ and $d_v = 1\ \forall v \in L$. If we choose only one root node ($|R| = 1$), we construct a single tree. With a higher number of roots, we build a forest of DRCMSTs.

Given an undirected complete graph $G = (V, E)$ with a set of vertices (nodes) V and a set of edges E, a forest of G is a subgraph $F = (V, E_F)$, $E_F \subset E$ that contains all vertices in V and consists of a spanning tree in each connected component of F. Given a definition of degree constraints and subsets R, I and L that satisfies the requirements specified in the previous paragraph, the DRCMST problem consists in finding a minimum forest $F^* = (V, E_{F^*}), E_{F^*} \subset E$ with $|R|$ connected components such that

$$F^* = \underset{F}{\operatorname{argmin}} \sum_{(u,v)\in E_F} c_{uv}, \tag{8.1}$$

subject to

$$\begin{gathered} \deg(v) \leq d_v \ \text{ for all } v \in V \\ \text{role}(v) = \mathit{root} \text{ for all } v \in R \\ \text{role}(v) = \mathit{intermediate} \text{ for all } v \in I \\ \text{role}(v) = \mathit{leaf} \text{ for all } v \in L, \end{gathered}$$

where role$(v)\in \{root, intermediate, leaf\}$ gives the role of node v in the forest.

Proposition 1. Given an undirected complete graph $G = (V, E)$, the subsets of nodes R, I and L such that $\{R, I, L\}$ is a partition of the set V, $R \neq \emptyset$, and a degree constraint for each $v \in V$ satisfying $d_v \geq 1\ \forall v \in R$, $d_v \geq 2\ \forall v \in I$ and $d_v = 1\ \forall v \in L$, the DRCMST problem is feasible if and only if

$$\sum_{v\in R} d_v + \sum_{v\in I} d_v - |I| \geq |I| + |L|. \tag{8.2}$$

Proof. Note that the proof has a double implication. First, we assume that the problem is feasible, and we see that Inequality (8.2) holds.

$\Rightarrow$

Suppose the problem is feasible, i.e., the subgraph of G, $F = (V, E_F)$, consists of $|R|$ trees where the constraints of problem (8.1) are satisfied. Let us prove that Inequality (8.2) holds when d_v is replaced by $\deg(v)$ for each $v \in V$, and then it will also hold for $d_v, v \in V$, because $d_v \geq \deg(v)$, $\forall v \in V$. Then, we prove

$$\sum_{v\in R}\deg(v) + \sum_{v\in I}\deg(v) - |I| \geq |I| + |L| \Longleftrightarrow \sum_{v\in R}\deg(v) + \sum_{v\in I}\deg(v) \geq 2|I| + |L|$$

We add $\sum_{v\in L}\deg(v)$ to both sides of Inequality (8.2):

$$\sum_{v\in R}\deg(v) + \sum_{v\in I}\deg(v) + \sum_{v\in L}\deg(v) \geq 2|I| + |L| + \sum_{v\in L}\deg(v)$$

It is known that $\sum_{v\in V_G}\deg(v) = 2|E_G|$ holds for any graph $G_G = (V_G, E_G)$. Further, because $d_v = \deg(v) = 1\ \forall v \in L$, $\sum_{v\in L}\deg(v) = |L|$ and we have

$$\sum_{v\in V}\deg(v) \geq 2|I| + |L| + |L| \Longleftrightarrow 2|E_F| \geq 2|I| + 2|L| \Longleftrightarrow |E_F| \geq |I| + |L|$$

Since $|E_T| = |V_T| - 1$ holds for every tree T and our initial assumption was that the forest F has $|R|$ trees $\Rightarrow |E_F| = |V| - |R|$. Then, we have that $|V| - |R| \geq |I| + |L|$ and this becomes an equality because $|V| = |R| + |I| + |L|$. Hence, because Inequality (8.2) is true for $\deg(v)$, $v \in V$, it is also true for $d_v, v \in V$, and we have proved the first part of the double implication.

Second, we assume that Inequality (8.2) holds and we prove that the problem is feasible.

$\Leftarrow$

If Inequality (8.2) is satisfied, we can build a forest satisfying the degree constraints with the following two steps. First, we incorporate into the forest a path starting at one of the roots and including all the intermediate nodes. Second, we include an edge linking each leaf to either a root node or an intermediate node. After the first step of the construction, the sum of the residual degrees is precisely $\sum_{v\in R} d_v + \sum_{v\in I} d_v - 2|I|$, which is greater or equal than $|L|$ because Inequality (8.2) holds, ensuring that the second step of the construction can be carried out. Hence, we have also proved this part of the implication.

□

In other words, the DRCMST problem is feasible if and only if the maximum allowed number of 'outputs' (left-hand side of (8.2)) is greater than or equal to the number of 'inputs' (right-hand side of (8.2)). See Fig. 8.1 for an infeasible example. Notice that we are working with undirected graphs so no input or output edges exist. By establishing root nodes, however, tree structure is implicitly directed since the roots (leaves) are considered to be the origins (ends) of a tree.

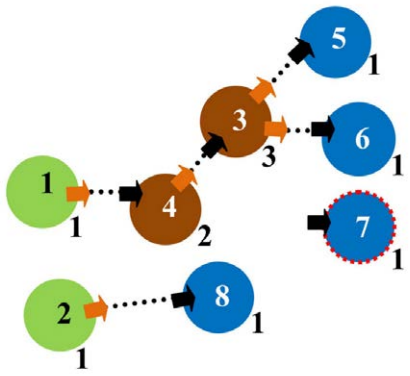


Figure 8.1: Example of an infeasible DRCMST instance. The maximum allowed degree d_v is shown on the right of each node. Root nodes are shown in green, intermediate nodes in brown and leaf nodes in blue. Since $|R| = 2$, $|I| = 2$ and $|L| = 4$, we need six 'inputs' (black arrows): two for intermediate nodes and four for leaf nodes. However, we only have five possible 'outputs' (orange arrows): two from the root nodes and three from intermediate nodes. In this example, node number 7 cannot be connected to either of the two trees in this forest. This example does not satisfy Inequality (8.2): $2 + 5 - 2 \not\geq 2 + 4$

8.3 Problem representation

We set out to encode the DRCMST problem using a permutation representation. A permutation is understood as a vector $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_n)$ of the indices $1, ..., n$ such that $\sigma_k \neq \sigma_s$ for all $k \neq s$. We say that index s is in position k in $\boldsymbol{\sigma}$ when $\sigma_k = s$.

In a forest encoded by the proposed representation, all nodes have a degree $\deg(v)$ equal to their maximum allowed degree d_v. To enforce this constraint, we add a new type of nodes called dummy nodes, see Fig. 8.2. We add as many dummy nodes as are necessary to make $\deg(v) = d_v, \forall v \in V$. Let D be the subset of dummy nodes. In a forest of DRCMSTs encoded by our representation, Inequality (8.2) becomes an equality:

$$\sum_{v \in R} d_v + \sum_{v \in I} d_v - |I| = |I| + |L| + |D|, \tag{8.3}$$

and hence the number of dummy nodes to be added is

$$|D| = \sum_{v \in R} d_v + \sum_{v \in I} d_v - 2|I| - |L|. \tag{8.4}$$

Dummy nodes are added for representation purposes only, and they are all leaf nodes so their degree is always equal to 1. The cost of every edge that reaches a dummy node is zero. Then, $m = |V| + |D| = |R| + |I| + |L| + |D|$ is the total number of nodes in the encoded forest.

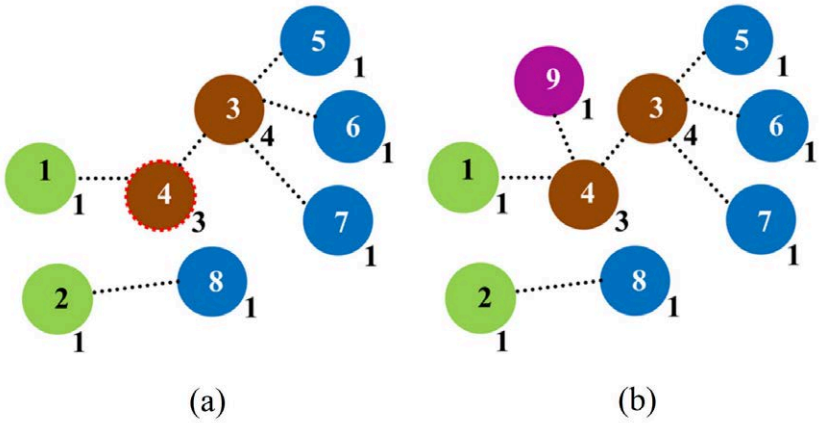


Figure 8.2: Example of a DRCMST forest with two trees. The maximum allowed degree d_v is shown on the right of each node. In (a) the degree $\deg(v)$ of all nodes is equal to their maximum allowed degree d_v, except node number 4 where $\deg(4)$=2 and d_4=3. To encode this forest with our permutation-based representation, we add one dummy node, node 9, connected to node 4. Forests (a) and (b) are equivalent because dummy nodes are added for representation purposes only and do not affect the calculation of tree costs

In our representation, each index of the permutation denotes a connection between two nodes, i.e., each index represents an edge in the forest. Since $|E_T| = |V_T| - 1$ holds for every tree T and we encode $|R|$ trees in one permutation, the permutation length n (total number of edges in the encoded forest) can be calculated as $n = m - |R|$.

The length of the permutation can also be obtained using Eq. (8.3):

$$n = \sum_{v \in R} d_v + \sum_{v \in I} d_v - |I| = |I| + |L| + |D|.$$

To find out which nodes are connected by the edges represented in each position of the permutation, we need two auxiliary arrays, *parent* and *child*, both of length n. These arrays remain unchanged for all permutations of the same problem. The *parent* auxiliary array represents the 'outputs' of the edges in the forest. Since each root node has d_v 'outputs' and each intermediate node has $(d_v - 1)$ 'outputs', each root node appears d_v times for all $v \in R$ and each intermediate node appears $(d_v - 1)$ times for all $v \in I$ in the *parent* array. The *child* auxiliary array represents the 'inputs' of the edges. All intermediate nodes, leaf nodes and dummy nodes have one 'input', therefore the *child* array includes each node $v \in I \cup L \cup D$ once. With these arrays, our permutation is such that $\sigma_k = s$ represents that node $parent_s$ in the forest (note that we use the subscript to indicate the element in position s of the auxiliary array) is the parent of node $child_k$, see Fig. 8.3. A simple version of this novel representation considering only binary trees was introduced in Anton-Sanchez et al. [2013].

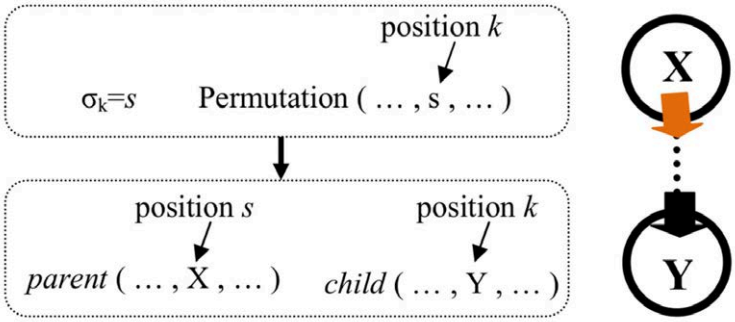


Figure 8.3: Decoding the proposed permutation-based representation. $\sigma_k = s$ represents that, in the forest, node $parent_s$ (node X) is the parent of node $child_k$ (node Y)

To illustrate this representation, consider the example in Fig. 8.4. Fig. 8.4(a) shows the $|V| = 10$ nodes of an example graph $G = (V, E)$. The maximum allowed degree d_v is shown on the right-hand side of each node $v \in V$. Nodes selected as root nodes are shown in green, intermediate nodes in brown and leaf nodes are shown in blue, thus, $|R| = 2$, $|I| = 3$ and $|L| = 5$. This problem is feasible because it satisfies Inequality (8.2). We check whether it is necessary to add any dummy nodes to solve the problem using Eq. (8.4):

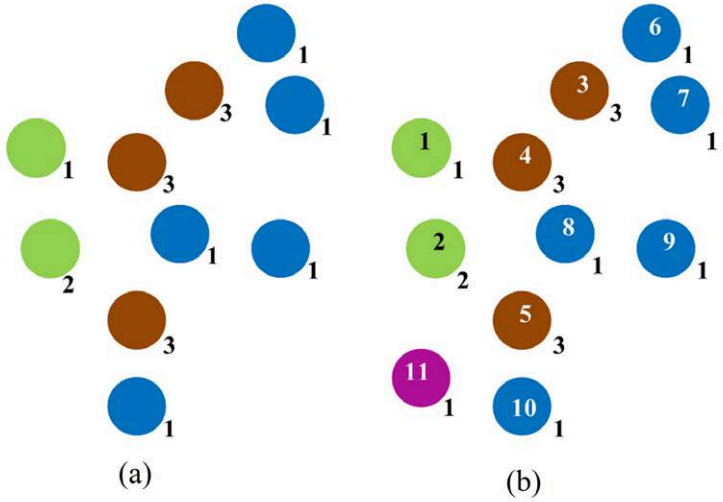
$$|D| = \sum_{v \in R} d_v + \sum_{v \in I} d_v - 2|I| - |L| = 3 + 9 - 2 \cdot 3 - 5 = 1,$$

i.e., we have to add one dummy node. Fig. 8.4(b) shows the numbered nodes and the added dummy node (node number 11 in pink). Then, a forest in the example will have two trees ($|R| = 2$) with $m = |V| + |D| = 10 + 1 = 11$ nodes and it will be represented by permutations of length $n = m - |R| = 11 - 2 = 9$.

We build the *parent* and *child* auxiliary arrays both needed to encode the permutations. As indicated, we add each root node d_v times ($v \in R$) and intermediate nodes ($d_v - 1$) times each ($v \in I$) to the *parent* array. A possible *parent* auxiliary array is shown in Fig. 8.4(c). A possible *child* auxiliary array, including each intermediate, leaf and dummy node once, is shown in Fig. 8.4(d). Note that *parent* and *child* auxiliary arrays must be established before starting to solve the problem because they determine which DRCMST problem solution each permutation represents. The order of the nodes in these arrays is in fact irrelevant.

Fig. 8.4(e) represents the permutation (6,1,2,4,5,8,7,9,3) which would be a correct individual (forest) using the defined *parent* and *child* auxiliary arrays, i.e., $parent_6$ (node 4) is the parent of $child_1$ (node 3), $parent_1$ (node 1) is the parent of $child_2$ (node 4) and so on until the last position of the permutation, which indicates that $parent_3$ (node 2) is the parent of $child_9$ (node 11).

Our permutation-based representation implicitly ensures that all constraints of problem (8.1) are satisfied in an encoded forest. However, permutations encoding any cycle represent invalid forests. For example, Fig. 8.4(f) represents an invalid individual (permutation (1,**8**,**6**,4,5,7,9,2,3)) because it contains a cycle (in red) between nodes 4 and 5. The second



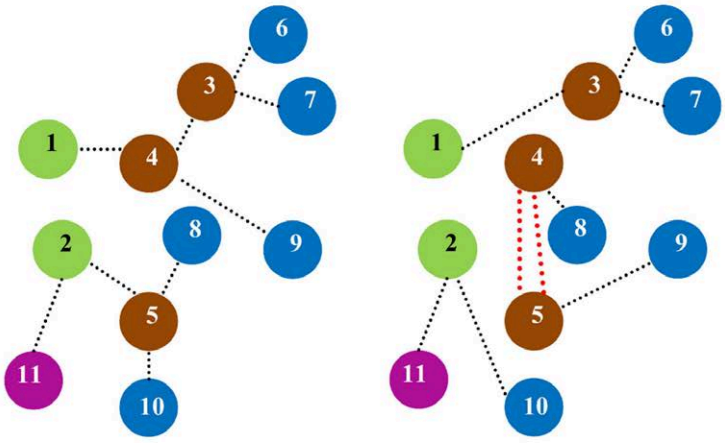
(a) (b)

position	1	2	3	4	5	6	7	8	9
parent	**1**	**2**	**2**	**3**	**3**	**4**	**4**	**5**	**5**

(c)

position	1	2	3	4	5	6	7	8	9
child	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**

(d)



(e) (f)

Figure 8.4: An example of permutation representation. (a) Nodes of an example graph. The maximum allowed degree d_v is specified to the right of each node. The role of each node is indicated in different colors: root nodes in green, intermediate nodes in brown and leaf nodes in blue. (b) Numbered nodes. According to Eq. (8.4) ($|D| = 3 + 9 - 2 \cdot 3 - 5 = 1$), a dummy node is needed to solve the problem. This is added as node 11 in pink. (c)-(d) *parent* and *child* auxiliary arrays required to determine which forest each permutation represents. (e) Example of valid individual, permutation (6,1,2,4,5,8,7,9,3). (f) Example of invalid individual because it contains a cycle, permutation (1,8,6,4,5,7,9,2,3)

position of the permutation indicates that $parent_8$ (node 5) is the parent of $child_2$ (node 4) and the next position indicates that $parent_6$ (node 4) is the parent of $child_3$ (node 5).

We can ensure that permutations corresponding to acyclic graphs are correct forests, i.e., they represent the required number of trees $|R| = m-n$ (it is trivial that if a graph $G = (V, E)$ has m nodes, n edges and no cycles, then it is a forest composed of $(m - n)$ trees).

Note that this representation yields several permutations encoding the same individual. For example, permutation (6,1,**3**,4,5,8,7,9,**2**) is the same individual as permutation (6,1,**2**,4,5,8,7,9,**3**) (Fig. 8.4(e)) because both $parent_3$ and $parent_2$ represent node number 2. We call these positions redundant positions, and we remove redundancy. To do this, we always choose the individual whose numbers of redundant positions are ordered from lowest to highest, i.e., (6,1,2,4,5,8,7,9,3) in the example.

Furthermore, note that cycles of length one, i.e., cycles that indicate that a node is its own parent, are very easy to detect with our representation. For example, as regards the problem illustrated in Fig. 8.4, we know that numbers 4 or 5 cannot occupy the first position of the permutation because this would indicate that $parent_4$ or $parent_5$, i.e., node 3, is the parent of $child_1$, also node 3. For longer cycles, we must traverse the permutation and build the trees that it encodes to identify any cycles. We read the permutation, in which each position indicates a connection between two nodes, sequentially. If the two nodes in the connection already belong to the same connected component, we will have detected a cycle. Otherwise, we join the nodes. To do this, we use the weighted quick-union algorithm [Sedgewick and Wayne, 2011]. The worst-case order of growth of all operations of this algorithm is log$n$, where n is the length of the permutation.

8.4 Problem-solving approach

We used genetic algorithms (GAs) [Holland, 1975] and estimation of distribution algorithms (EDAs) [Larrañaga and Lozano, 2002] to solve and compare a variety of synthetic simulated DRCMST instances. GAs have been widely studied for solving permutation-based optimization problems [Larrañaga et al., 1999], and they are known to perform satisfactorily [Bielza et al., 2010, Reeves, 1995, Ruiz and Maroto, 2005]. However, although several papers using probabilistic models on rankings with EDAs have recently been published [Aledo et al., 2013, Ceberio et al., 2011, 2014, 2015], EDAs have not been so extensively developed for permutation-based optimization problems [Ceberio et al., 2012]. The probabilistic model learned in an EDA is expected to reflect the structure of the problem, and therefore this approach should provide an effective exploitation of promising solutions.

As described in Chapter 3, we used the gGA [Cobb and Grefenstette, 1993], the ssGA [Syswerda, 1991], the NHBSA [Tsutsui, 2006] and the MKEDA [Ceberio et al., 2015], and the jMetal framework [Durillo and Nebro, 2011, Durillo et al., 2010] in order to compare the performance of these algorithms. jMetal already contained gGA and ssGA algorithms, and we plugged our NHBSA and MKEDA implementations into jMetal. We made improvements to all the algorithms due to the specific characteristics of our representation. On the one hand,

we ruled out the generation of individuals containing cycles of length one (which are easy to detect as described in Section 8.3) and, on the other hand, we removed the redundancy of our representation by selecting a representative individual from the redundant individuals as already explained. In order to detect cycles of length longer than one, it was necessary, as already mentioned, to traverse each permutation sequentially, building the trees that it encoded. If a cycle was identified, the individual was immediately ruled out.

For GAs, we used the default operators provided for permutations in jMetal: partially matched crossover (PMX) and swap mutation. PMX builds a child by choosing a subsequence of one parent (permutation) using two random cut points, and it preserves the order and position of as many indices as possible from the other parent. Swap mutation selects two indices at random and swaps their positions. We set a crossover probability ($CrossProb$ in Alg. 3.1) equal to 0.9 and a mutation probability ($MutProb$ in Alg. 3.1) equal to $1/n$, where n is the length of the permutation. For each problem, we established a population size equal to $10|V|$ for all GAs and EDAs. For each execution of each problem, the initial population was randomly generated including the improvements discussed above (length-one cycles and redundancy). We decided to stop any algorithm if there was no more than a 0.1% improvement of the best fitness over the last 500 generations.

We applied the non-parametric Friedman test to detect statistically significant differences considering the whole set of algorithms [Friedman, 1937]. The null hypothesis for the Friedman test states equality between all the algorithms. If the null hypothesis is rejected, a post-hoc test can be applied to find out which pairwise comparisons cause the differences. We opted for the Bergmann-Hommel procedure [Bergmann and Hommel, 1988]. Although computationally expensive, this is the best-performing procedure for comparing all the algorithms with one another [Derrac et al., 2011]. We set a significance level of $\alpha = 0.05$. We used the implementation of the Friedman test for multiple comparison and the Bergmann-Hommel procedure provided in the MULTIPLETEST package available at the SCI2S public website[1].

8.5 Test problem generation

We simulated five problem instances for each of the following sizes $|V|$= 20, 40, 60, 80, 100, 200 nodes. The number of roots and intermediate nodes were randomly generated, although some constraints were included. The number of problem root nodes was less than or equal to 20% of all problem nodes, i.e., $|R| \leq 0.2|V|$. The number of intermediate nodes was less than or equal to 75% of all problem nodes ($|I| \leq 0.75|V|$). The number of leaf nodes was derived as $|L| = |V| - |R| - |I|$. The maximum allowed degree of root and intermediate nodes (leaf nodes always have a degree equal to 1) was also simulated randomly for each node as follows. The maximum allowed degree d_v was between 1 and 4 for root nodes and between 2 and 5 for intermediate nodes. We simulated the coordinates (x, y, z) of each point between $x_{min} = y_{min} = z_{min} = 1$ and $x_{max} = y_{max} = z_{max} = 100$. Then, we computed the cost matrix with real Euclidean distances between pairs of points. A small fitness was preferred

[1] http://sci2s.ugr.es/sicidm/

Table 8.1: Description of the simulated DRCMST instances. The table shows the number of root, intermediate, leaf and dummy nodes and the length of the permutations that encode the problem solutions

Problem	$\|R\|$ roots	$\|I\|$ intermediate	$\|L\|$ leaf	$\|D\|$ dummy	n permutation length
Problem 1-20Nodes	3	9	8	10	27
Problem 2-20Nodes	3	6	11	0	17
Problem 3-20Nodes	1	9	10	3	22
Problem 4-20Nodes	3	11	6	10	27
Problem 5-20Nodes	2	12	6	11	29
Problem 1-40Nodes	5	18	17	15	50
Problem 2-40Nodes	7	12	21	6	39
Problem 3-40Nodes	4	22	14	16	52
Problem 4-40Nodes	4	18	18	8	44
Problem 5-40Nodes	7	16	17	13	46
Problem 1-60Nodes	5	24	31	1	56
Problem 2-60Nodes	1	36	23	13	72
Problem 3-60Nodes	11	17	32	8	57
Problem 4-60Nodes	9	26	25	18	69
Problem 5-60Nodes	11	23	26	22	71
Problem 1-80Nodes	13	42	25	50	117
Problem 2-80Nodes	11	25	44	1	70
Problem 3-80Nodes	13	36	31	36	103
Problem 4-80Nodes	12	33	35	5	73
Problem 5-80Nodes	15	29	36	21	86
Problem 1-100Nodes	15	45	40	29	114
Problem 2-100Nodes	7	56	37	39	132
Problem 3-100Nodes	6	52	42	14	108
Problem 4-100Nodes	18	42	40	39	121
Problem 5-100Nodes	15	56	29	49	134
Problem 1-200Nodes	23	74	103	24	201
Problem 2-200Nodes	30	100	70	91	261
Problem 3-200Nodes	11	102	87	43	232
Problem 4-200Nodes	10	103	87	16	206
Problem 5-200Nodes	20	99	81	59	239

when we evaluated individuals (permutations) of our population.

Table 8.1 shows the characteristics of each of the simulated instances. For each instance, it lists the number of nodes of each role and the length of the permutation representing the forest. Note that the length of the permutation for each problem instance depends on the number of nodes of each role and their maximum allowed degree.

We obtained a wide variety of problem instances. The number of trees in the forest ranged from a single tree ($|R| = 1$, two times out of 30) to 30 trees (problem number 2 with 200 nodes). The number of intermediate nodes ranged from 28% to 60% of all network nodes. The number of leaf nodes ranged from 29% to 55% of $|V|$ depending on the problem. No dummy node had to be added in one of the problems (problem number 2 with 20 nodes), whereas problem number 1 with 80 nodes had 50 dummy nodes (62.5% of the problem size). On average, the permutation length of the problems was 15.6% greater than the number of nodes in the problem.

8.6 Results

We used the algorithms described in Section 3.3 to solve the 30 simulated DRCMST instances described in Section 8.5. Each problem was run 20 times with each algorithm (with a new randomly generated initial population in each run). All the results were obtained using the Magerit supercomputer. Magerit is offered by the high performance computing area at the CeSViMa. Magerit is a general-purpose cluster with dual architecture (Intel and POWER). We used POWER7 nodes with 3.3 GHz (422.4 GFlops) 32 GB of RAM and 300 GB of local hard disk.

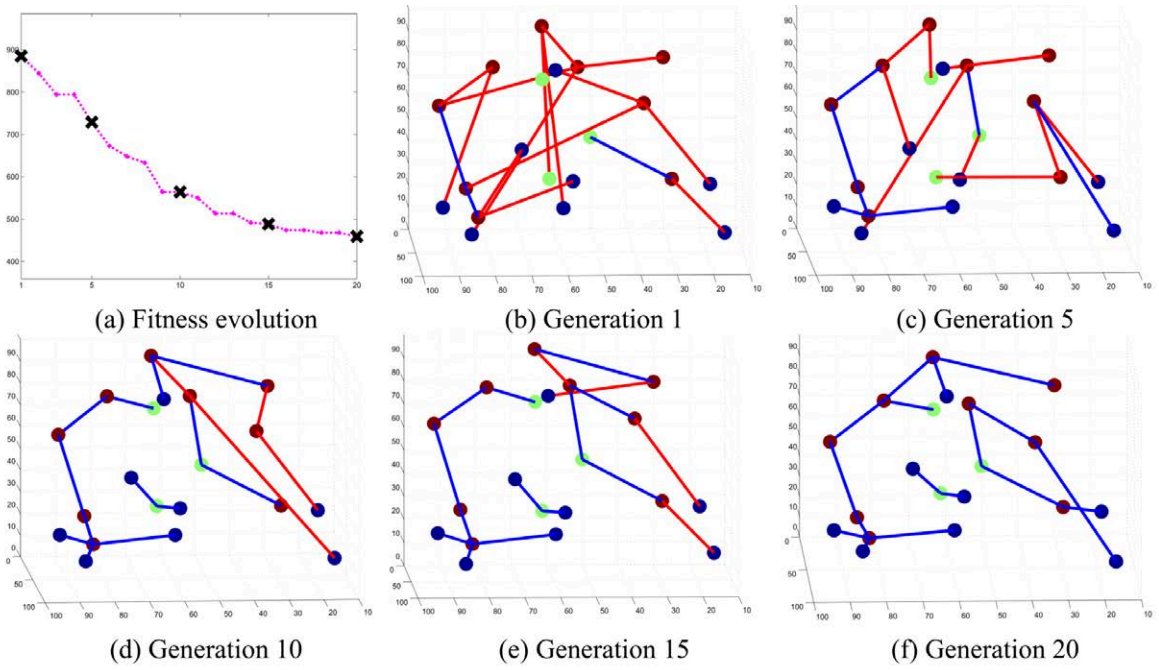


Figure 8.5: Evolution of the best fitness found in 20 generations by the NHBSA for problem number 1 with 20 nodes. (a) Fitness evolution over 20 generations (the crosses indicate the fitness of individuals shown in (b)-(f)). (b)-(f) Forest encoded by the best solutions found in generations 1, 5, 10, 15 and 20. Root nodes are shown in green, intermediate nodes in brown and leaf nodes in blue. Edges that differ from the best forest found by the algorithm are shown in red. The algorithm did not improve after generation 20

Fig. 8.5(a) shows the best fitness values found by the NHBSA in the first 20 generations of a run for problem number 1 with 20 nodes. For this problem, we were interested in building a forest of three trees. Fig. 8.5(b)-(f) shows the trees that represent the best individuals found in generations 1, 5, 10, 15 and 20. Again, the roots are represented in green, intermediate nodes are shown in brown and leaf nodes in blue. In each forest, the edges that differ from the best forest found by the algorithm are shown in red. We observe that the number of red edges gradually diminishes as the number of generations increases because the algorithm is approaching the best solution found. The forest output in generation 20 does not have any

red edges because the algorithm did not improve after this generation, i.e., it provides the best fitness value found for this problem.

As described in Section 8.4, we applied the Friedman test to detect statistically significant differences among the algorithms [Friedman, 1937]. We applied the test two times: once on the mean best fitness found by the algorithms for the 30 problem instances and again on the mean execution time. The Friedman test rejected the null hypothesis of equality for both the fitness and execution time (p-value $\leq 10^{-11}$ in both cases). Once the null hypothesis of equality between all pairs of algorithms was rejected, we applied the Bergmann-Hommel procedure [Bergmann and Hommel, 1988] to perform all the pairwise comparisons.

Fig. 8.6 illustrates the results of both the Friedman test and the Bergmann-Hommel procedure. These diagrams were introduced in Demšar [2006] and neatly illustrate statistically significant differences between algorithms. The Friedman test ranks the algorithms such that the best-performing algorithm should have rank 1, the second best rank 2, etc. In the diagrams the lowest (best) ranks are to the right so the algorithms on the right-hand side can be viewed as better. Groups of algorithms that are not significantly different (p-value > 0.05 in the Bergmann-Hommel procedure pairwise comparisons) are connected. Analyzing pairwise comparisons, the results showed that there were no significant differences in the best fitness for the NHBSA, the gGA and the ssGA (Fig. 8.6(a)). Looking at the execution times, however, we found significant differences between all the algorithms (Fig. 8.6(b)). Both EDAs had a longer execution time than GAs. The gGA and the ssGA had similar execution times but the hypothesis of equal mean times was rejected. We could, therefore, conclude that the ssGA was preferable because it had a better execution time.

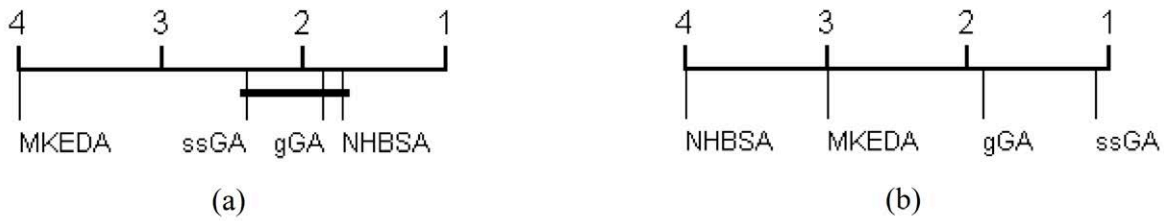


Figure 8.6: Comparison of the four algorithms using the Friedman test and the Bergmann-Hommel procedure. Groups of algorithms that are not significantly different (p-value > 0.05) are connected. The lowest (best) ranks are to the right so the algorithms on the right-hand side can be viewed as better. (a) Fitness diagram. (b) Execution time diagram

We also wanted to compare the four heuristic algorithms with an exact method for further evaluation. The evaluation of all permutations of length n for a DRCMST problem requires an execution time of order $n!$. This is unworkable even for small values of n. Therefore, we implemented the following branch-and-bound method to solve small instances of DRCMST problems exactly.

We know that, in a DRCMST problem, each permutation position adds the connection cost between two points to the objective function. Moreover, we build the forest represented

by a permutation, traversing the permutation and accumulating the cost of each position sequentially. Suppose that we want to solve exactly a DRCMST problem represented by permutations of length n. To do this, we start by generating and evaluating the permutations of length n in lexicographical order. Let x be the best solution found by our heuristic methods. If, in a specific permutation, we are at position j, such that the cost accumulated so far is greater than the x cost, then we can rule out the following $(n-j)!$ permutations in lexicographical order, i.e., all permutations that have the same indices up to position j.

Although, using this branch-and-bound method, a lot of the permutations do not have to be evaluated, we were unable to solve exactly the instances of 20 nodes (Table 8.1), because it would have taken several months. Therefore, we simulated some smaller problem instances. Specifically, we simulated five problems with 10 nodes (with a permutation length equal to 10, 11 and 12) and five problems with 15 nodes (all with a permutation length equal to 13) as detailed in Section 8.5. We solved these 10 problem instances with the implemented branch-and-bound method, evaluating, on average, 23.63% of all permutations. The instances with 10 nodes were fairly easy to solve, whereas the execution time on a desktop computer for each instance with 15 nodes was of the order of several days.

We also solved these 10 problems 20 times using each of the four heuristic algorithms. This was done sequentially (10 problems x 20 times x 4 algorithms = 800 runs) on the same computer as for the exact method. The execution time was just over two minutes. The gGA, ssGA and NHBSA found the global optimum in all cases, and the MKEDA found the global optimum for six out of the 10 problems (the solutions for the other four problems were 1.75% worse, on average, than the global optimum).

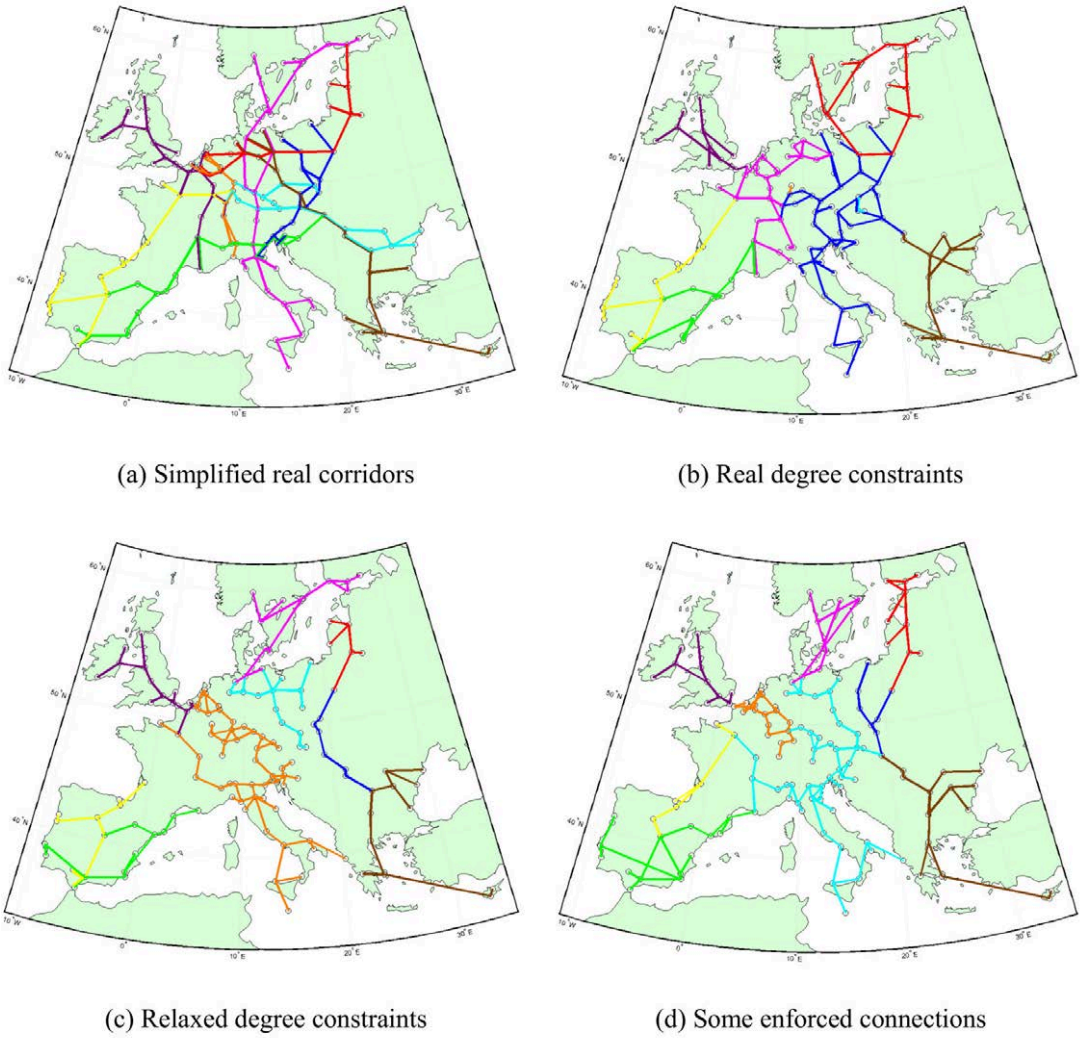
8.7 Trans-European transport network

We use the trans-European transport network in order to illustrate the applicability of forests with DRCMSTs in a real-world network design. This transport network, available at the European Commission website[2], is composed of nine corridors and comprises 138 cities. Its main purpose is to facilitate the transport of passengers and goods throughout the European Union providing for faster international long-distance travel. To illustrate the interest of our approach, we tried to design a similar network facilitating transport between the above European cities by formulating the network design as a DRCMST problem. As in the real network we built nine corridors and were interested in minimizing the total length of the transport network.

Not all the corridors of the trans-European transport network are trees in the sense that some contain cycles. To fit the design of a forest of DRCMSTs, we simplified the real network by deleting 18 connections between cities to remove cycles. Fig. 8.7(a) shows the real transport corridors after removing these connections.

We used straight lines to represent connections between pairs of cities. In our simplified

[2] http://ec.europa.eu/transport/themes/infrastructure/ten-t-guidelines/corridors/doc/ten-t-corridor-map-2013.pdf



(a) Simplified real corridors

(b) Real degree constraints

(c) Relaxed degree constraints

(d) Some enforced connections

Figure 8.7: Application of forests with DRCSMTs to the nine trans-European transport network corridors. We consider the geodesic distance between two cities as their connection cost. (a) Simplified trans-European transport network where we removed 18 connections to cut out cycles in the real corridors. Total length= 33,138 km. (b)-(d) Solutions provided by the ssGA. (b) We chose one city from each corridor as the root node. In the example, the roots are Bilbao, Craiova, Frankfurt, Hamburg, Katowice, London, Perpignan, Vienna and Warsaw. The cities where a corridor ends in the real network are considered leaf nodes; the remaining nodes are intermediate nodes. We assigned a city a maximum degree equal to the number of connections it has in the simplified real corridors. Total length= 32,177 km. (c) Role constraints as in (b). In this case, we relaxed degree constraints and allowed a maximum degree equal to 3 for all intermediate nodes. Total length=28,759 km. (d) Degree and role constraints as in (c). We established some compulsory connections between cities to enforce corridor interconnections, in particular, we enforced the following connections: Bordeaux-Paris, Marseille-Lyon, Vienna-Wels/Linz and Craiova-Timişoara. Total length= 31,166 km

example, however, we considered the geodesic distance between two cities as their connection cost. The total geodesic length of the real corridors (with cycles removed) is 33,138 km (Fig. 8.7(a)).

There are 138 cities in the trans-European transport network. Given that some cities in the real network belong to several corridors, we replicated these cities as many times as the number of corridors they belong to. This strategy makes the network design more flexible. After replicating these cities, the result was a total of 204 nodes (cities) for our problem.

As regards node roles, we chose one city from each of the real corridors (located more or less in the middle of the corridors) to play the root role in order to build nine transport corridors. Specifically, the roots were Bilbao, Craiova, Frankurt, Hamburg, Katowice, London, Perpignan, Vienna and Warsaw in the examples shown in Fig. 8.7(b)-(d). The cities where a corridor ended in the real network were considered leaf nodes, and the other cities were intermediate nodes.

Regarding the degree constraints of each city, we ran two different tests. First, we optimized the design of the nine corridors by defining the maximum allowed degree in each city as the real number of connections it had in the (simplified) real corridors. In this case, no dummy nodes had to be added and, since we built nine trees, the length of the permutations was 204+0-9=195. We used the ssGA to solve the problems since this technique performed significantly better for DRCMST problems in Section 8.6. The best solution found for the real degree of each city is shown in Fig. 8.7(b) with a total length of 32,177 km. In this solution most of the European territory is covered by seven rather than nine transport corridors, since the orange and cyan corridors were composed of only two cities (Frankfurt and Mannheim) and three cities (Vienna, Brno and Bratislava), respectively.

Then we relaxed the degree constraints to make the network design more flexible. Specifically, we assigned a maximum number of connections equal to three to every intermediate node. In this case, we had to add 82 dummy nodes, and the permutation length needed to represent the nine corridors was 204+82-9=277. Fig. 8.7(c) shows one of the solutions using these degree constraints with a total length of 28,759 km. Fig. 8.7(c) reveals that the corridors are divided into four groups: 1) yellow and green corridors, 2) purple and orange corridors, 3) magenta and cyan corridors and, finally, 4) the remaining three corridors. The corridors in each group are connected with each other but not with other groups.

The ssGA looks for a forest, in our case with a minimum total length, where the degree and role constraints hold. We can easily add additional constraints, for example, require the solution to have specific connections between cities, using the proposed representation. Since each permutation position represents a connection between two nodes of the forest, if we want two nodes always to be connected, the permutation should be required to have a specified number in a specified position. In an attempt to find a good solution with some interconnected corridors, we optimized the corridor design as in Fig. 8.7(c) but enforced the following four connections: Bordeaux-Paris, Marseille-Lyon, Vienna-Wels/Linz and Craiova-Timişoara. One of the solutions provided by the ssGA is shown in Fig. 8.7(d). This solution had a total length of 31,166 km.

Besides adding constraints depending on the specific characteristics of the network design problem, the minimization criterion for building the trees could take into account more complex criteria, like the varying cost of building the connection between different cities. Furthermore, alternative distances more in line with reality than the geodesic distance could be used.

8.8 Conclusions

This chapter has presented a novel permutation-based representation to solve a new variant of the DCMST problem, which we have called DRCMST problem. A DRCMST is a DCMST with supplementary constraints that determine the role of the nodes in the tree (root, intermediate or leaf nodes). Establishing the roles of the nodes may be useful in some problems such as network design. Most research about computing DCMST outputs a single tree. We increase the flexibility of the problem by not limiting the number of root nodes to one so, generally, we compute forests of DRCMSTs. We used metaheuristic techniques to approximate the problem solution because the DRCMST problem is NP-hard. Specifically, we opted to use two genetic algorithms (gGA and ssGA) and two estimation of distribution algorithms (NHBSA and MKEDA). Using the proposed representation, we solved a wide range of synthetic simulated DRCMST instances. The results showed that the NHBSA, the gGA and the ssGA found the best solutions, but the ssGA ran in significantly less time. Finally, we formulated the nine corridors of the trans-European transport network as a DRCMST problem and optimized it using the ssGA to illustrate the applicability and flexibility of our approach.

The main advantage of our permutation-based representation is that it can encode more than one tree simultaneously. Moreover, the degree constraint can be different for each node. Another strength is that it is simple to add constraints related to a specific problem. For example, if two nodes must (cannot) be connected in the problem statement, then a specific number will be enforced (forbidden) at a specific position of the permutation.

Probably the weakest point of the proposed representation is that it encodes invalid individuals (cycles). Cycles of length equal to one are easy to detect and thus avoid (this is the cause of the highest percentage of invalid individuals). However, the permutation must be decoded to detect the existence of cycles of length longer than one. We intend to work on improving cycle detection, which could speed up the algorithms. Furthermore, different permutations may encode the same forest. We remove this redundancy by selecting a representative individual within the set of redundant individuals.

Other aspects could be taken into account such as considering a more complete fitness evaluation function. For example, if the network is designed for signal transmission from server nodes to leaf nodes, then, distances from root nodes to leaf nodes should be as short as possible, since distances are closely related to transmission time. In this case, besides minimizing the total cost (distance) of the resulting forest, it might also be beneficial to minimize the distances between roots and leaves. If there are several optimization criteria to be considered, we might also think about the convenience of optimizing either a single-

objective problem (for example, weighting the different objectives) or moving towards a multi-objective problem. Another aspect to be considered is problem solving with an extremely large number of nodes. In this case, it might be handy to decompose the original problem into subproblems of smaller size and parallelize problem solving.

Chapter 9

Neuronal wiring economy

9.1 Introduction

Santiago Ramón y Cajal formulated the fundamental anatomical principles of the organization of nerve cells more than a century ago. He stated that the structure of axons and dendrites is designed in such a way as to save space, time and matter [Ramón y Cajal, 1899]. In this chapter we aim to show that dendritic and axonal trees of different types of neurons optimize brain connectivity in terms of neuronal wiring cost. Although the concept of wiring cost is not clearly defined, it is basically based on the assumption that the further away two elements are, the more expensive the connection between them is. Therefore, wiring cost can be expressed as a function of the distance between elements, this being the criterion to be minimized.

The significance of the neuronal wiring cost hypothesis, regarded as underlying principles of brain morphology and organization, has been widely studied (reviewed in Chklovskii [2004]). Some researchers have suggested that the organization of certain regions of the brain is related to the need to reduce wiring costs [Chklovskii et al., 2002, Wen and Chklovskii, 2008, Wen et al., 2009]. Other studies have constructed synthetic neuronal structures to show that optimal wiring explains dendritic branching patterns [Cuntz et al., 2007, 2008, 2010]. In this chapter we analyze wiring economy in single neurons, taking a different approach from previous research considering a specific criterion of wiring cost assessment, namely, wiring length. We start from the branching and terminal point cloud of real neuronal trees, which we search for the shortest arborization. We force the computed wiring to pass through the branching points to reach the terminal points, and we limit the number of times that the points branch out, since multifurcations rarely occur in real neurons. We hypothesize that by imposing constraints that provide realistic synthetic arborizations, we can for the most part explain the wiring economy of single neurons considering only wiring length.

We use graph theory to test our wiring optimization hypothesis using the DRCMST presented in Chapter 8. Graph theory is suitable for representing the point clouds and their connections and has been successfully applied in previous works studying dendritic structures [Cuntz et al., 2007, 2008] and neocortical axons [Budd et al., 2010]. With the imposed

constraints, our wiring design problem is NP-hard, so we had to use heuristic methods for problem solving. We opted for evolutionary computation techniques (Section 3.3). Relatively few heuristics have been used to analyze wiring design. For example, Cuntz et al. [2007, 2008] used a greedy algorithm which locally minimizes the total amount of wiring in their synthetic neuronal structures, whereas Pérez-Escudero et al. [2009] and Rivera-Alba et al. [2011] used simulated annealing [Kirkpatrick et al., 1983] to find low-cost neuronal element configurations.

The research included in this chapter has been published in Anton-Sanchez et al. [2016a,b].

Chapter outline

Section 9.2 presents the analysis of both the dendritic and axonal wiring of a set of single interneurons with complex and different morphologies through the methodology proposed in Chapter 8. Section 9.3 tests the hypothesis of optimal neuronal wiring in single pyramidal cells, a much more homogeneous population of neurons, and examines if there are differences in wiring optimality across all cortical layers.

9.2 Wiring economy of GABAergic interneurons

9.2.1 Data

In this section we analyze the dendritic and axonal wiring of six morphological types of neocortical interneurons, including Martinotti (MA), large basket (LB), common type (CT), horse tail (HT), chandelier (CH) and common basket (CB) cells [DeFelipe et al., 2013]. These interneurons are characterized by different dendritic and axonal morphologies and synaptic connections (see e.g., Ascoli et al. [2008]). We used a set of 12 3D reconstructed interneurons (two neurons of each type, Fig. 9.1) classified into different types according to their morphology by 42 leading neuroscientists [DeFelipe et al., 2013]. These neurons were originally extracted from NeuroMorpho.Org [Ascoli et al., 2007]. Table 9.1 shows the cell type and unique identifier of these neurons in NeuroMorpho.Org.

9.2.2 Wiring analysis

Fig. 9.2(a) shows neuron CT2 in Fig. 9.1 with superimposed point clouds formed by the roots, branching and terminal points of the dendrites (red) and the axon (blue). We searched for the optimal (the shortest) dendritic wiring from the red point cloud and for the optimal axonal wiring from the blue point cloud.

All branching points in the analyzed neurons were bifurcations. Therefore, we forced these nodes to divide into two branches too. Hence, in our neuronal wiring analysis, we are looking for minimal cost trees, with constraints on the number of bifurcations. Additionally, to assure that the extent of the dendritic and axonal arborizations is fixed, the roots (i.e., points of origin of the dendrites and axons from the cell body) and terminal points of real neuronal trees should also be unchanged in the searched structures. We can deal with this by building the DRCMSTs presented in Chapter 8. As shown in Section 8.6, genetic algorithms, and,

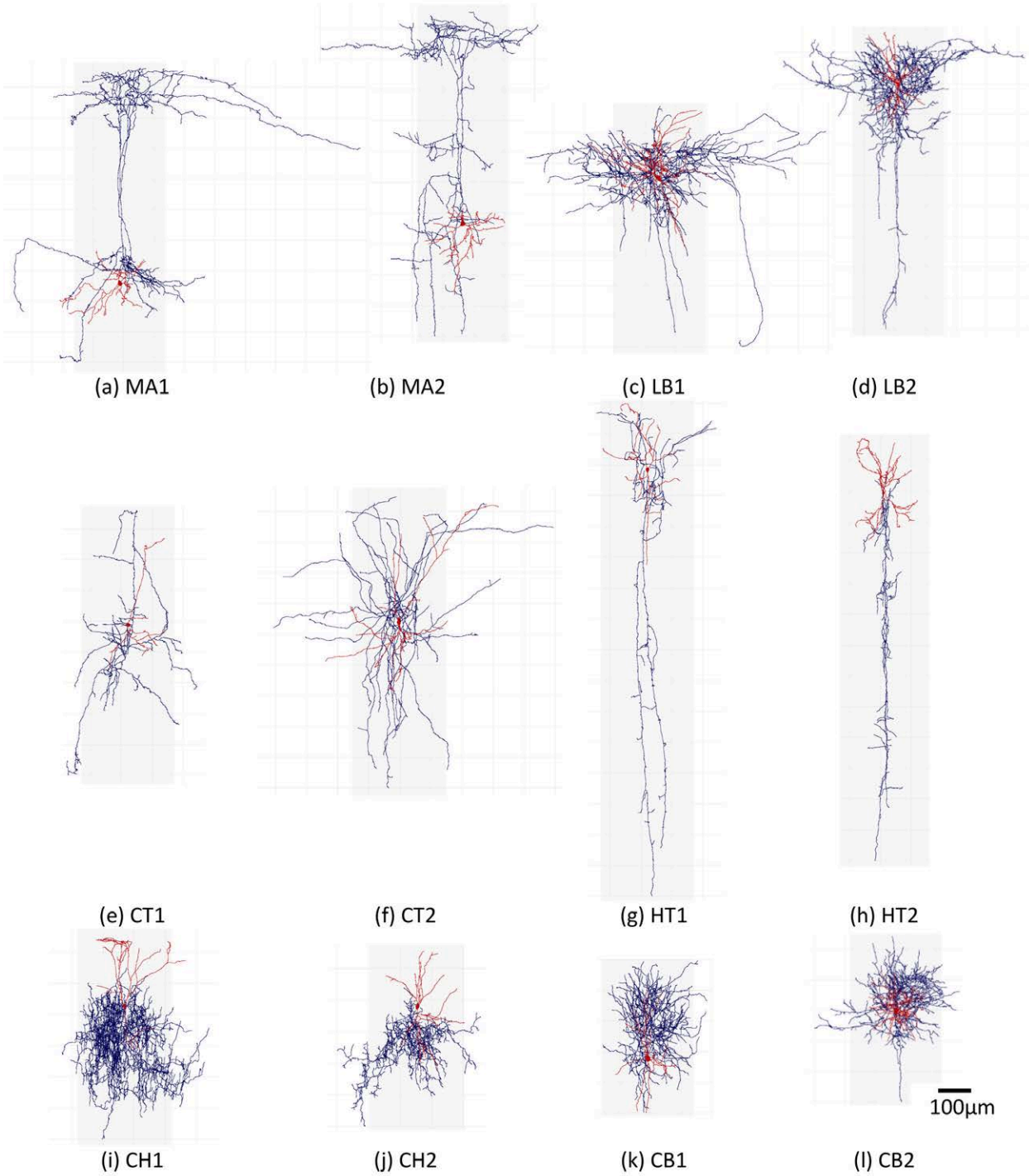


Figure 9.1: The twelve analyzed interneurons. Dendrites are shown in red and axons in blue. We consider six different types of interneurons depending on their morphology: (a,b) Martinotti (MA), (c,d) large basket (LB), (e,f) common type (CT), (g,h) horse tail (HT), (i,j) chandelier (CH) and (k,l) common basket (CB), as defined in a previous work for the classification on GABAergic interneurons [DeFelipe et al., 2013]

Table 9.1: NeuroMorpho.Org identifier and cell type of the 12 analyzed interneurons. We analyzed the morphology files of the repository version 6.1 (May 2015)

Neuron	NeuroMorpho.Org ID	Type
MA1	NMO_02204	Martinotti
MA2	NMO_00334	Martinotti
LB1	NMO_04572	Large basket
LB2	NMO_04582	Large basket
CT1	NMO_02732	Common type
CT2	NMO_04558	Common type
HT1	NMO_04577	Horse tail
HT2	NMO_00337	Horse tail
CH1	NMO_04548	Chandelier
CH2	NMO_00291	Chandelier
CB1	NMO_01858	Common basket
CB2	NMO_04574	Common basket

in particular, the ssGA [Syswerda, 1991], performed significantly better for the DRCMST problem. Therefore, we solved our neuronal wiring design problems using this technique. One of the main issues that need to be addressed when using genetic algorithms is the definition and encoding of individuals. In our case, an individual of the population is a feasible neuronal arborization, and each individual is encoded by the permutation-based representation explained in Section 8.3. The smaller the total wiring length is, the fitter an individual is considered to be.

Axonal arborizations consist of a single tree but dendritic arborizations are, generally, formed by a group of trees. The methodology proposed in Chapter 8 can simultaneously optimize one or more trees. Therefore it is applicable to our wiring design problems for both axons and dendrites. Thus, by restricting the number of branches (degree) and the role played by each point in the trees, we search for a single tree with optimal wiring in axonal point clouds and we search for a group of trees with optimal wiring in dendritic point clouds. Then, we compare the resulting structures and the real arborizations. Fig. 9.2(b) shows the axonal point cloud of neuron CT2 in three different colors, differentiating the three roles with which we work. Fig. 9.2(c) shows the colored dendritic point cloud. Note that, in this case, we have five roots because the neuron has five dendritic trees (none of the roots are readily appreciable because it is a 3D point cloud).

To search for the optimal arborization that meets the discussed constraints, we formulate and optimize DRCMST problems where an arborization is represented by a permutation of length $n - t$, where n is the total number of points and t is the number of trees to be built. Each position of the permutation represents a connection between two points. The use of the auxiliary arrays *parent* and *child* to decode the permutation-based representation guarantees degree and role constraints in the trees.

Fig. 9.3 shows an example with two of the dendritic trees of neuron CT2. Fig. 9.3(a) shows the point cloud of these two trees and uses different colors to identify the roles. Fig. 9.3(b)

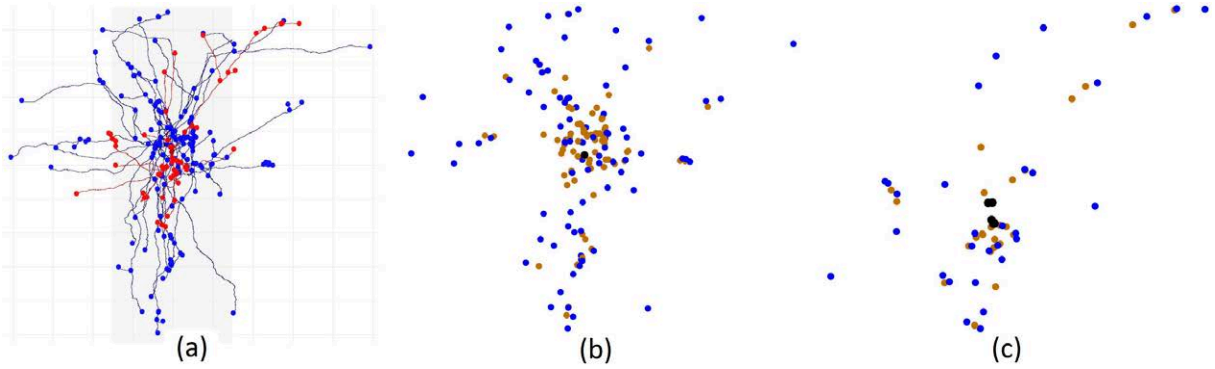


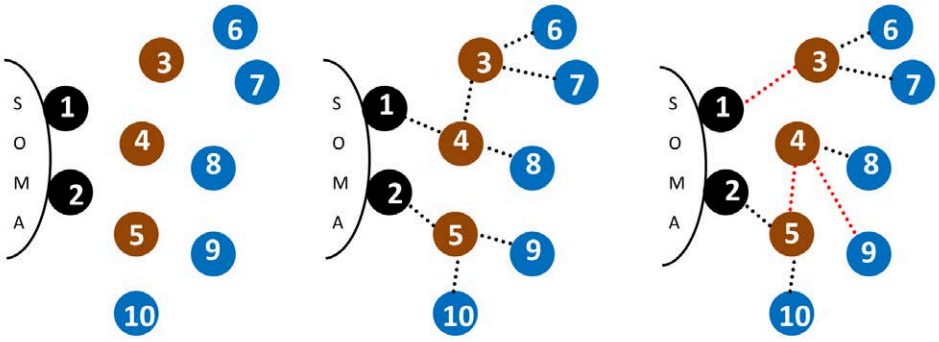
Figure 9.2: Example of point clouds. (a) Neuron CT2 with superimposed point clouds formed by the roots, branching points and terminal points of the dendrites (red) and the axon (blue). (b,c) Axonal (b) and dendritic (c) point clouds: the root points are shown in black, the branching points in brown and the terminal points in blue

shows a solution which matches the real neuronal trees. Fig. 9.3(c) shows another possible valid set of trees. This small example has $n = 10$ points and $t = 2$ trees. Therefore, the length of the permutations that represent this arborization is $n - t = 8$. Fig. 9.3(d) shows the auxiliary arrays *parent* and *child* needed for permutation decoding. Fig. 9.3(e) shows the permutations that represent the arborizations in (b) and (c).

9.2.3 Axon partition

As reported in Section 8.5, DRCMST problems up to 200 nodes can be readily solved. This is the case of dendritic wiring design problems. The computational cost of solving axonal design problems in the same way would be huge because they are much more complex, and it would be very time consuming. Therefore, we introduce parallel computing to address complex problems, that is, we partition the overall axonal point cloud into smaller clouds, and we solve these smaller clouds separately. We can simultaneously solve each of the parts (which takes a few seconds or minutes depending on their size) and then combine the best (shortest) solutions found in each part to ouput the solution that provides the complete axonal tree (negligible time compared to the rest of the process).

The axon is represented by a permutation of length $n - 1$, where n is the total number of points in the axonal point cloud. The creation of sub-regions in the overall point cloud is equivalent to partitioning this permutation into as many parts as sub-regions we need to solve. First, we optimize each of the parts into which we divide the permutation, searching for the shortest tree structures in different regions of the point cloud (different colors in Fig. 9.4). Each sub-region is solved according to the procedure reported in Chapter 8 as described above. Second, we put together the shortest solutions found in each sub-region (sub-part of the global permutation) to output a permutation that represents the entire axonal tree. Third, we try to improve the global solution found. To do this, we iteratively swap permutation positions that are close to the junctions of the parts making up the whole permutation (Fig. 9.5).



(a) (b) (c)

position	1	2	3	4	5	6	7	8
array1	**1**	**2**	**3**	**3**	**4**	**4**	**5**	**5**

position	1	2	3	4	5	6	7	8
array2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**

(d)

position	1	2	3	4	5	6	7	8
Trees in (b)	**5**	**1**	**2**	**3**	**4**	**6**	**7**	**8**

position	1	2	3	4	5	6	7	8
Trees in (c)	**1**	**7**	**2**	**3**	**4**	**6**	**5**	**8**

(e)

Figure 9.3: Two examples of dendritic trees of neuron CT2 shown in Fig. 9.1 and their codification with the proposed permutation-based representation. (a) Numbered point cloud of the two trees. The roots are shown in black, terminal points in blue and branching points (bifurcations) in brown. (b) Equivalent structure to real trees. (c) Another valid solution. Differences to (b) are shown in red. Note that as the roots are unchanged, the number of constructed trees is always equal to the number of trees in the neuron. However, branching and terminal points from different dendritic trees can be mixed. (d) Auxiliary arrays *parent* and *child* needed to decode the permutations. (e) Permutations that represent the arborizations in (b) and (c). Decoding is as follows. A number s at position k of the permutation means that the node at position s of auxiliary array *parent* is connected to the node at position k of auxiliary array *child*. For example, in the permutation shown in (e), top, representing arborization (b), we find $s = 5$ at position $k = 1$. This means that the node at position 5 in auxiliary array *parent* (node 4) is connected to the node which is at position 1 of auxiliary array *child* (node 3). The number at position $k = 2$ is $s = 1$, which means that the node at position 1 of auxiliary array *parent* (node 1) is connected to the node at position 2 of auxiliary array *child* (node 4), and so on (see Section 8.3 for further details on decoding)

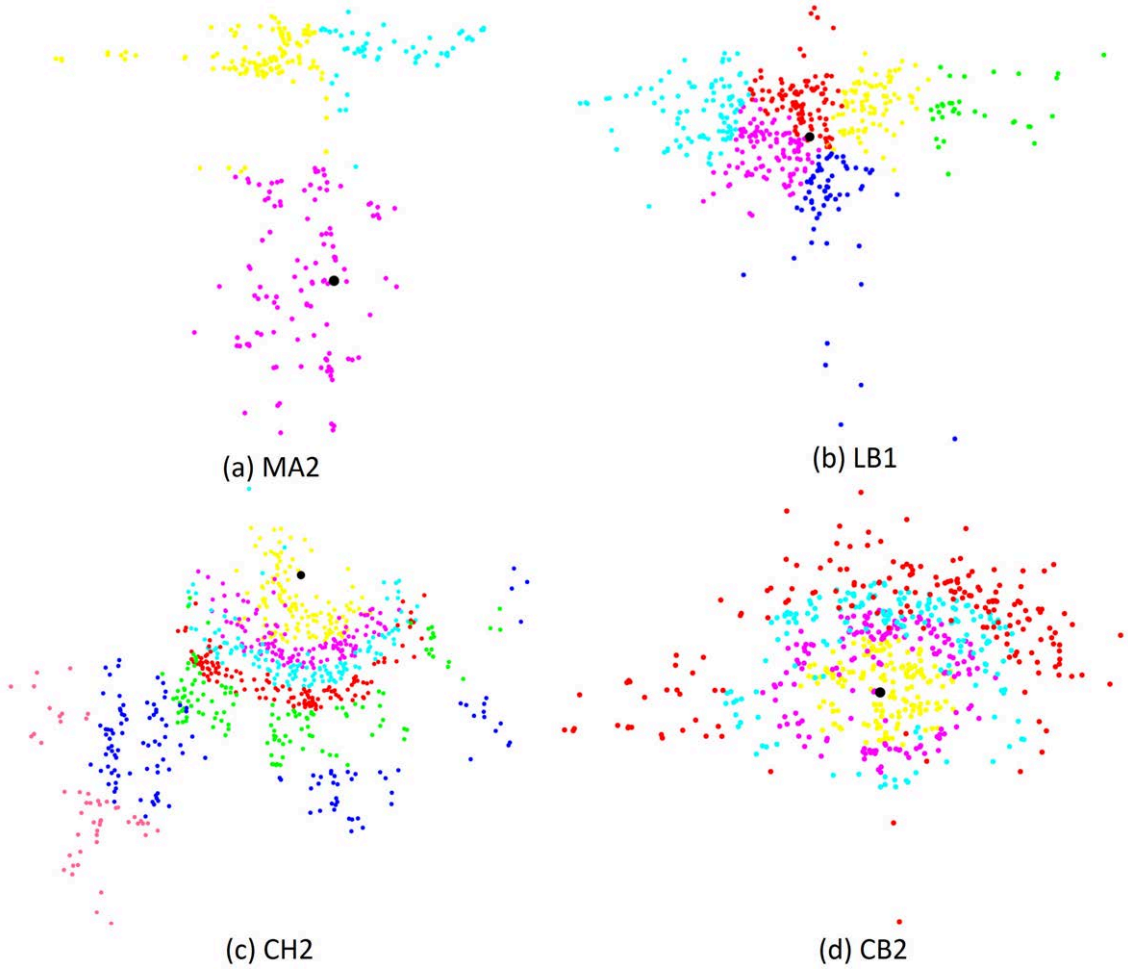


Figure 9.4: Axonal point clouds of some of the analyzed interneurons divided into smaller clouds to reduce complexity. Sub-regions are shown in different colors and the root of the tree is shown in black. (a) Neuron MA2. 274 points. Groups created using the k-means algorithm with k=3 groups. (b) Neuron LB1. 500 points. k=6 groups. (c) Neuron CH2. 800 points. Groups created by distances from nodes to the soma with group size of 125. (d) Neuron CB2. 674 points. Group size of 165

Due to the diversity of axon shapes (spherical, elongated, etc.), we try out two different methods to create the smaller point clouds within the overall set of points. For all the analyzed neurons, we optimize the axonal wiring using the two methods described below. For each neuron, we choose the result provided by the method that performs best, that is, the method that provides the shortest total axonal wiring, and we compare its length with the real axonal wiring.

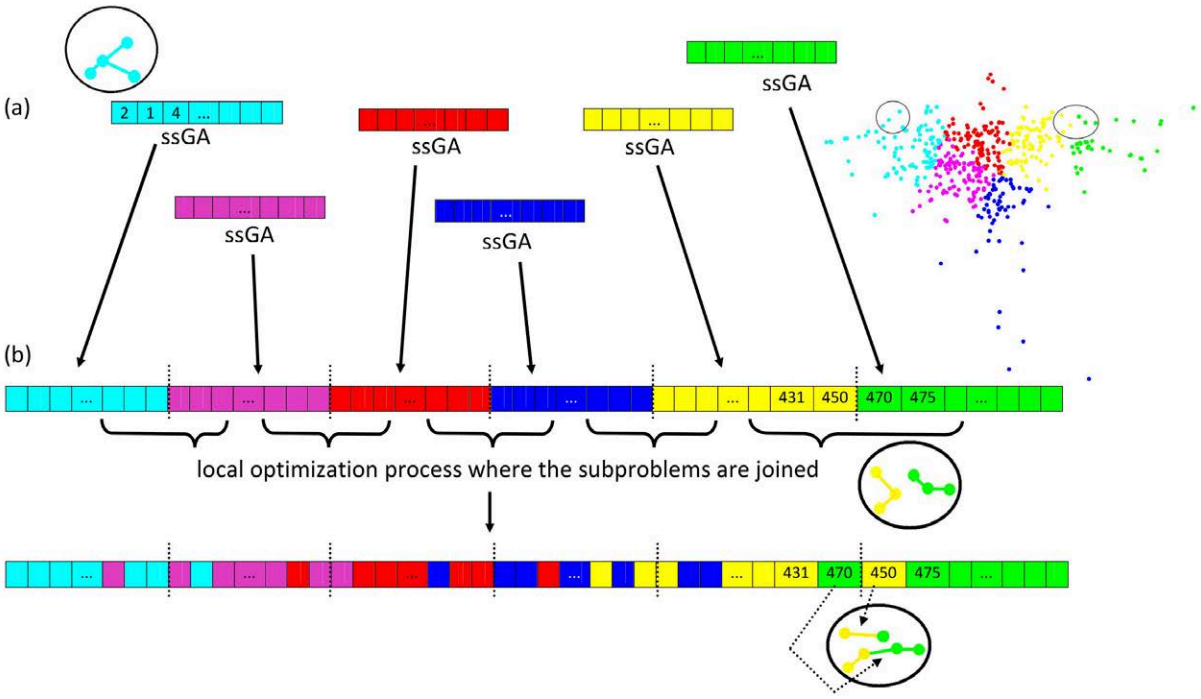


Figure 9.5: Description of the partitioning process for complex problems with a high number of nodes. Example with the axon of neuron LB1. (a) The axonal point cloud (500 nodes) is divided into six smaller point clouds (using the k-means algorithm in this case): cyan, magenta, red, blue, yellow and green. The genetic algorithm (ssGA) is applied to each sub-region separately searching for the shortest arborization in each of the smaller point clouds. A sub-region is part of the global permutation depicting the complete axon. Each position of the permutation represents a connection between two nodes (e.g., the first three positions of the cyan permutation correspond to the three connections of the magnified region in this color). (b) We put together the best solutions found in each part to output the global permutation for the complete axonal tree. We apply a local optimization process in the neighbourhoods where the sub-region solutions meet: we iteratively switch positions near the junctions of the parts that form the global permutation trying to find better solutions. Switching positions at those permutation locations means changing connections between nearby nodes of two different sub-regions (an example is shown in the magnified region of the yellow and green zones). After the local optimization processes we choose the permutation depicting the best (shortest) complete axonal tree. We repeat the procedure in (a) and (b) 20 times for each neuron (maintaining the same sub-regions). Then we choose and compare with the real axonal tree the best arborization found

k-means algorithm

One method for creating subsets of nodes is the k-means unsupervised clustering algorithm [MacQueen, 1967] to group nodes according to the distance between them (Fig. 9.4(a) and 9.4(b)). We choose a value of k nearest to how many hundreds of nodes there are in the point cloud. For example, we choose $k = 3$ for neuron MA2 whose axonal point cloud has 274 nodes.

Soma distance

The other method is to form groups of nodes based on their distance to the soma. By setting a group size, e.g. 100, we form the first group with the first 100 nodes of the point cloud that are closest to the soma, the second group with the next 100 nodes closest to the soma, after excluding the nodes used in previous groups, and so on. We test several group sizes for each of the neurons in order to achieve good results for comparison with the real neuronal trees (Fig. 9.4(c) and 9.4(d)).

9.2.4 Software

We developed software enabling the user to analyze the wiring optimality of a three-dimensional neuron from its specification in .asc format. The software and a user manual are available for download at the Computational Intelligence Group's webpage[1] (Software section). It is capable of processing wiring design problems with point clouds up to size 200. Larger problems are costly for a personal computer and are better addressed using parallel computing. Both dendritic and axonal wiring can be analyzed. We implemented the necessary preprocessing for the .asc files in Java and we used the single-objective ssGA implementation provided in jMetal framework [Durillo and Nebro, 2011, Durillo et al., 2010].

9.2.5 Results

Table 9.2 summarizes the characteristic features of the 12 neurons analyzed in this study: number of dendritic trees, total number of points (roots, branching and terminal points) of the dendritic point cloud and total number of points of the axonal point cloud (always a single tree). Furthermore, it shows the ratio between the total length of the shortest trees found and the total length of real neuronal trees (see below). The wiring length between two connected points is measured, in both the real and found tree structures, using the Euclidean distance between them. Therefore, we use an approximate real wiring length because we ignore the path tortuosity.

Dendritic wiring optimization

The number of dendritic trees in the analyzed neurons varies from 2 to 11 (Table 9.2); the total number of nodes in these cases is between 32 and 132. For dendritic wiring optimization, we did not apply the partitioning methods described in Section 9.2.3 because they were not complex problems. The results for dentritic trees are similar across all types of neurons. In all cases, the ssGA algorithm slightly improves upon the real neuronal arborization, i.e., it finds a slightly lower total wiring. The ratio between the length of the best dendritic structure found and the length of the real dendritic trees (fourth column of Table 9.2) shows that

[1] http://cig.fi.upm.es/

Table 9.2: Characteristics of the 12 interneurons shown in Fig. 9.1. Number of dendritic trees. Total number of points in the dendritic point cloud (considering all trees). Total number of points in the axonal point cloud (always a single tree). Ratio between the total length of the best (shortest) structure found for each neuron and the total length of the real neuronal trees. Below 100% (boldface), the length of the best found structure is shorter than the real wiring

	Dendrites			Axon	
Neuron	Trees	Points	Best/Real	Points	Best/Real
MA1	4	54	**95.50%**	476	102.59%
MA2	4	66	**97.33%**	274	**92.98%**
LB1	6	100	**93.34%**	500	**97.98%**
LB2	7	58	**98.00%**	822	111.04%
CT1	3	32	**97.51%**	236	**96.00%**
CT2	5	60	**99.91%**	168	**88.21%**
HT1	3	44	**98.92%**	228	**98.12%**
HT2	2	90	**97.97%**	156	**86.46%**
CH1	3	46	**95.24%**	780	101.36%
CH2	3	48	**98.59%**	800	113.01%
CB1	7	46	**95.29%**	560	**94.20%**
CB2	11	132	**91.98%**	674	109.36%

the greatest improvement is achieved for neuron CB2, where the genetic algorithm finds a solution whose total length is 8% shorter than the real neuronal wiring.

To check the range of variation of the wiring function, we performed the optimization process by reversing the direction, that is, we searched for structures that maximized the wiring length while meeting the constraints. As shown in Fig. 9.6, the maximum wiring of the dendritic arborizations was much longer than the real dendritic wiring (the results ranged from 270.51% in neuron LB2 to 605.42% in neuron CT1).

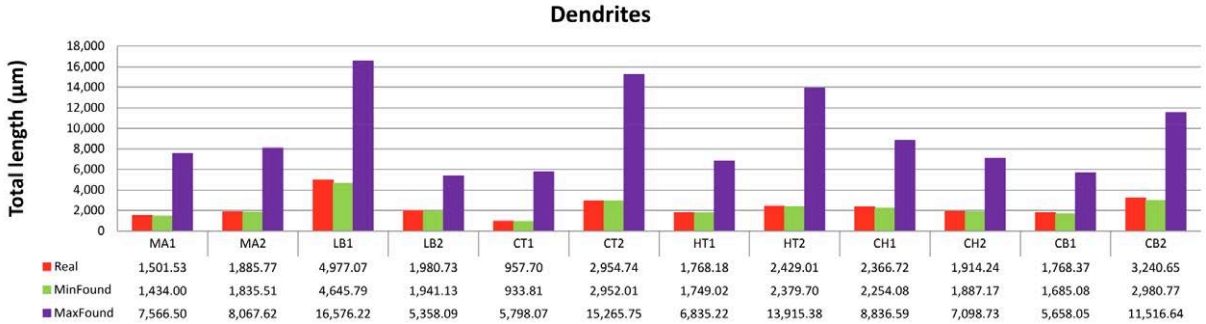


Figure 9.6: Total dendritic length (μm) of the 12 analyzed interneurons (red) versus total length of the minimum and maximum arborizations found (green and purple, respectively). In all cases, the optimization algorithm finds a better (shorter) solution than the real wiring. The maximum wiring found is much longer than the real wiring (four times on average)

Going back to our running example with the dendrites of neuron CT2, Fig. 9.7 illustrates the difference between the real dendritic wiring and some structures found during the optimization process for the entire dendritic arborization of this neuron. Fig. 9.7(a) shows the dendritic point cloud with the connections between nodes that exist in the real trees of this neuron. The five dendritic trees of this neuron are shown in five different colors. Fig. 9.7(b) shows the dendritic connections in the shortest structure found. It is a slight improvement upon the real dendritic wiring. Fig. 9.7(d) shows the connections of the structure that maximizes the wiring of this neuron. It is more than five times longer than the real wiring (Fig. 9.6). Fig. 9.7(c) shows a wiring which is in-between the minimum and maximum found by the optimization algorithm. It is about three times longer than the real wiring. Note that all connections are drawn as straight lines as we measure the (straight) length between points.

Axonal wiring optimization

The axonal point clouds of the 12 analyzed neurons have from 156 to 822 nodes (Table 9.2) with an average number of nodes greater than 470. As mentioned in Section 9.2.3, we use two different techniques to create sub-regions in the overall point cloud of each axon to reduce complexity. For each method, we combine the shortest solutions found in each sub-region so that our approach outputs the global minimum arborization (Fig. 9.5). We choose the result of the technique that returns the shortest total wiring for each neuron.

For Martinotti, large basket and common type neurons (Fig. 9.1(a)-(f)), the best solutions found were clearly better with the k-means algorithm (in the case of neuron LB1, there was a 15% difference in the best solutions found by both methods). For chandelier and common basket neurons (Fig. 9.1(i)-(l)), the best solutions found were vastly better creating groups of nodes depending on their distances to the soma (up to 33% better than k-means algorithm in the case of neuron CB2). For horse tail neurons (Fig. 9.1(g)-(h)), we also achieved better results by grouping the nodes by their distance to the soma. However, the best solutions found for this type of neurons were very similar using both methods.

The results of the two methods used to split the axonal point clouds clearly differentiated which method it is better to apply for each type of neuron. This was predictable considering the shape of the axons. In spherical-shaped axons, like chandelier and common basket neurons, it is better to group the nodes around the root tree. In axons with much less homogeneous shapes, like Martinotti and large basket neurons, it is better to group the nodes taking into account the distance between them regardless of a reference point.

Unlike dendrites, the tree structures output by the optimization algorithm do not improve upon the real axonal wiring in all cases. In the last column of Table 9.2, a figure below 100% shows that the best solution found by the ssGA has a total wiring length shorter (better) than the real axonal tree. A number greater than 100% indicates that the algorithm cannot find a solution that improves the real wiring. For neuron HT2, for example, we obtain a tree whose total length is almost 14% less than the real axonal tree. However, for neuron CH2 (one of the most complex axons analyzed with 800 nodes), the best solution found was 13%

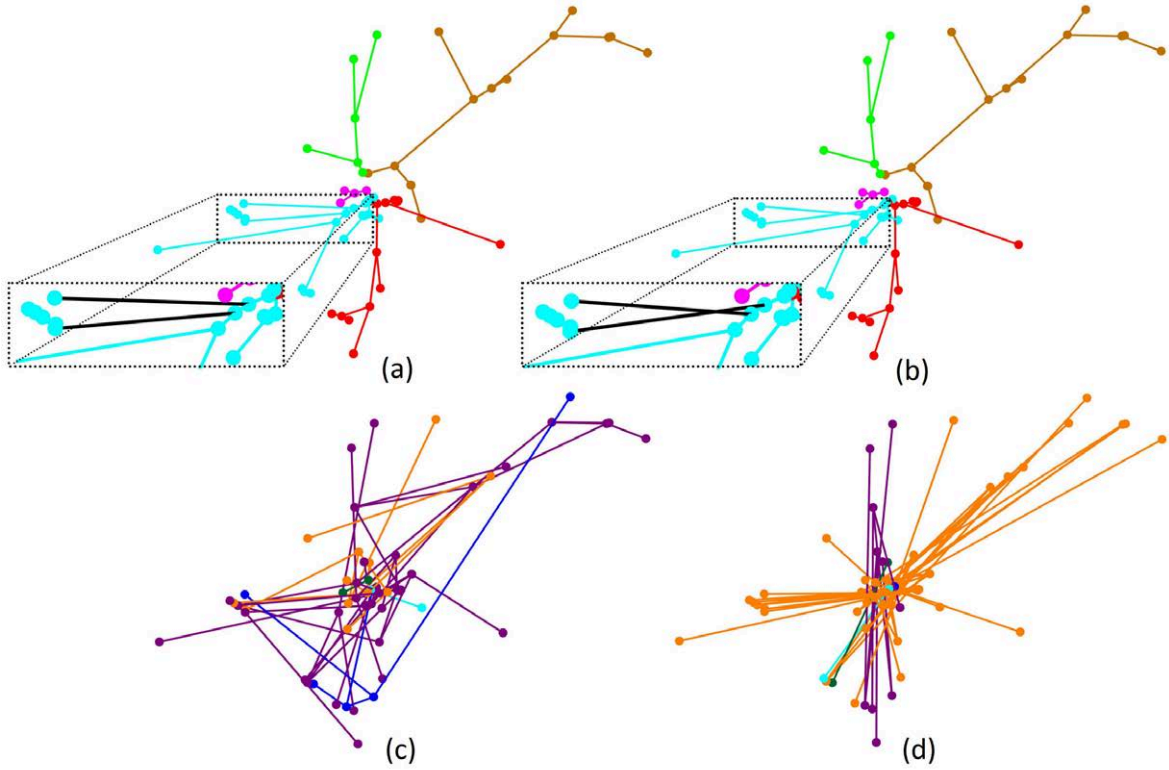
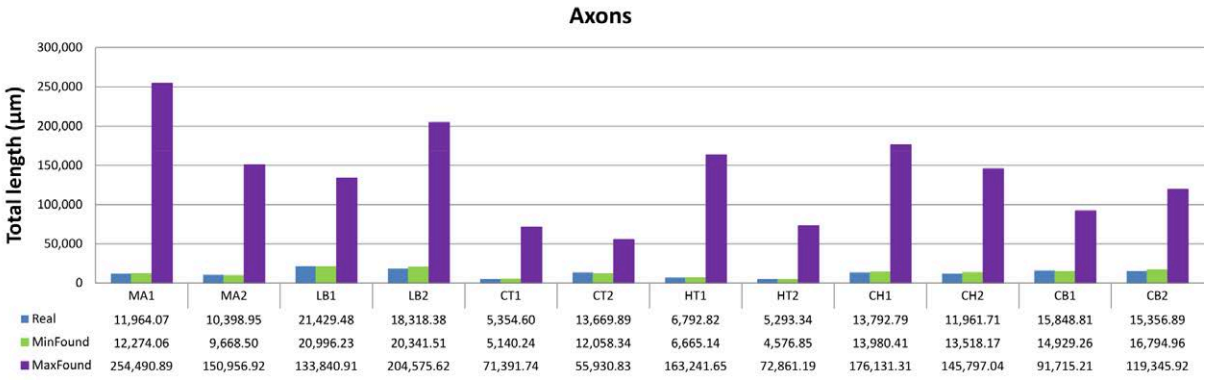


Figure 9.7: Example of neuron CT2 and differences between real and optimized dendritic wiring. (a) Dendritic point cloud with real connections between points. Five dendritic trees are shown in different colors: brown, green, magenta, cyan and red. (b) Dendritic point cloud with the connections in the shortest structure found by the algorithm. The optimization algorithm finds a structure that improves the real neuronal wiring by only two microns. With the exception of only two edges, the tree structure provided by the algorithm is identical to the real dendritic wiring (black connections in the magnified regions in (a) and (b)). (c) Structure whose wiring is three times longer than the real wiring. (d) Dendritic point cloud with the connections in the largest structure found, which is five times longer than the real wiring. The trees in (c) and (d) are very different from the real dendritic trees, and their colors were chosen arbitrarily

worse (longer total length) than the real axonal wiring. We also searched for the trees that maximized the axonal wiring of each neuron. The results varied from 409.15% in neuron CT2 to 2403.15% in neuron HT1, i.e., the maximum wiring found was between four and 24 times longer than the real wiring. Fig. 9.8 shows the total real lengths of the 12 axons and the total length of the minimum and maximum solutions found for each neuron.

In some of the axonal wiring design problems, the algorithm was unable to find the real configuration, which was known to exist. Therefore, we performed the following test to check algorithm performance. We generated random point clouds with n points and built their MSTs using Prim's algorithm [Prim, 1957]. From these MSTs, we constrained the degree and role of each point to match the degree and role in the MSTs. Then, from the original point clouds and with the imposed constraints, we searched for the DRCMSTs. We did this



	MA1	MA2	LB1	LB2	CT1	CT2	HT1	HT2	CH1	CH2	CB1	CB2
Real	11,964.07	10,398.95	21,429.48	18,318.38	5,354.60	13,669.89	6,792.82	5,293.34	13,792.79	11,961.71	15,848.81	15,356.89
MinFound	12,274.06	9,668.50	20,996.23	20,341.51	5,140.24	12,058.34	6,665.14	4,576.85	13,980.41	13,518.17	14,929.26	16,794.96
MaxFound	254,490.89	150,956.92	133,840.91	204,575.62	71,391.74	55,930.83	163,241.65	72,861.19	176,131.31	145,797.04	91,715.21	119,345.92

Figure 9.8: Total axonal length (μm) of the 12 analyzed interneurons (blue) versus total length of the minimum and maximum trees found (green and purple, respectively). For most axons, the optimization algorithm finds a solution that is shorter than or very close to real wiring. The maximum axonal wiring found is much longer than the real wiring (12 times on average)

for $n = 50, 100$ without problem partitioning. For $n = 200, 400, 800$, we divided the point clouds into smaller sub-regions using both of the partitioning methods described in Section 9.2.3.

For small problems, the optimization algorithm was very close to the MST length (2% larger for $n = 50$ and 6% for $n = 100$). For larger problems, we applied the partitioning methods to create sub-regions. By optimizing the sub-regions of the point cloud separately, we may not come as near to the global optimum. This is the price we pay for making these problems computationally tractable. For random problems with $n = 200$, the optimization algorithm yielded solutions 14% larger than the MST length. For $n = 400, 800$, the solutions were 23% and 26% larger than their MSTs, respectively. For $n = 200, 400$, we found the best results, i.e., shortest wirings, creating the sub-regions according to the soma distance. For $n = 800$, the ssGA found the best solutions using the k-means algorithm.

The mean number of points in the 12 dendritic wiring design problems was 65, and the mean best-to-real ratio for the shortest solutions found was 96.63%. Therefore, we concluded that, because the algorithm performed quite well for similar values of n, dendritic wiring was very nearly optimal in terms of wiring length. Comparing axons and dendrites, axonal wiring was not as optimal in terms of wiring length for neurons whose axonal point clouds had the lowest number n of points (although n was greater than the largest dendritic point clouds). Specifically, the best-to-real ratios in neuron HT2 ($n = 156$) and neuron CT2 ($n = 168$) were 86.46% and 88.21%, respectively (Table 9.2). Neuronal trees appear to expand more optimally in less complex branching structures. Consequently, dendritic wiring, generally simpler than axonal wiring, should come closer to the optimum in terms of the wiring length discussed in this study. In future research, we intend to refine the resolution of large problems in order to explore what happens in the axons for which our algorithm failed to improve upon the real wiring length.

The test that we conducted gives an idea of how well the genetic algorithm performs for problems of different sizes using both partitioning methods, but we must take into account that the comparison of the MST and DRCMST solutions is unfair. The MST for a big point cloud is easily obtainable in polynomial time. However, if the problem has degree and/or role constraints, the problem becomes NP-hard, and large problems are extremely difficult to solve. On this ground, it is necessary to use heuristic methods.

9.2.6 Analysis of other examples

In addition, we extended the study to analyze both the optimality of dendritic and axonal wiring of another 16 neurons to substantiate that the results were similar with groups consisting of more than two neurons. In addition to the cells illustrated in Fig. 9.1, we optimized the wiring of the eight Martinotti cells and the eight large basket cells shown in Table 9.3. These cell types were selected because they are common place in the literature and they have recognizable morphological characteristics [DeFelipe et al., 2013]. We applied the k-means algorithm to create the sub-regions in the axonal point clouds of the new neurons because this technique worked best for Martinotti and large basket cells. As shown in Table 9.4, the results for all ten neurons were very similar to what we found for individual neurons (Table 9.2).

Table 9.3: NeuroMorpho.Org identifier of the eight analyzed Martinotti and large basket neurons. Repository version 6.1 (May 2015)

Martinotti	Large basket
NMO_01848	NMO_00366
NMO_02629	NMO_00293
NMO_01839	NMO_01851
NMO_02203	NMO_04560
NMO_02579	NMO_04581
NMO_02648	NMO_00272
NMO_00306	NMO_00382
NMO_00427	NMO_04576

Table 9.4: Mean number of points ($\bar{n}$) and mean and standard deviation ($\bar{x}_{\pm s}$) of the ratios between the total length of the shortest dendritic and axonal wiring solutions found for each neuron, and the total length of the real trees for ten Martinotti and ten large basket neurons. The results include eight cells of each type (Table 9.3) on top of the two neurons already analyzed in Fig. 9.1

	Dendrites		Axon	
Type	$\bar{n}$	$\bar{x}_{\pm s}$	$\bar{n}$	$\bar{x}_{\pm s}$
Martinotti	81.0	$96.66\%_{\pm 2.2\%}$	527.2	$101.58\%_{\pm 10.7\%}$
Large basket	53.4	$97.81\%_{\pm 1.8\%}$	488.4	$104.85\%_{\pm 14.1\%}$

9.2.7 Conclusions

In this section we have presented a new approach to test the hypothesis of optimal neuronal wiring in single neurons using graph theory and evolutionary computation. We analyzed both the dendritic wiring and the much more complex axonal wiring. We found that the tree structure of different types of neocortical interneurons, which included Martinotti, large basket, common type, horse tail, chandelier and common basket cells, is near-optimal in terms of wiring length, although dendritic wiring was generally nearer to the optimum than axonal wiring. This is a remarkable finding since a characteristic of these neurons is that the postsynaptic targets and spatial characteristics of their dendritic and axonal arborizations are rather different (see below). Our analysis stresses the importance of the wiring cost to which some morphological and organizational principles in the brain have been attributed [Chklovskii, 2004].

Dendritic wiring optimization was solved properly using the method described in Chapter 8. To address axonal wiring design problems, however, we had to reduce their size. The method proposed here is to divide the axonal point cloud into different sub-regions and find the shortest tree structures in each of these sub-regions. The results show that this method performs well in many cases, providing a more efficient method in terms of time and computational cost savings. However, for some of the more complex axons, the optimization algorithm output a tree structure whose total length was close to but larger than real wiring (i.e., the algorithm could not find an equal or better solution than the real situation). Future research needs to improve the way in which the sub-regions are created and how the best solutions found in these sub-regions are combined to output the overall solution. Thus, it would be possible to deal with larger problems.

For all dendrites and many axons, the genetic algorithm (ssGA) used output tree structures with a total length slightly shorter than the real trees. This indicates that dendrite and axon spanning uses the least amount of wiring needed to achieve their functions but that there are also other important factors that influence neuron growth. For example, we might consider a more complete wiring cost function minimizing the distance of each non-root point to the root of the tree. This is closely related to minimizing the time that it takes for a signal to reach a synaptic contact from the soma (see e.g., Budd et al. [2010], Cuntz et al. [2007], Wen and Chklovskii [2008]).

For dendritic trees of the same neuron, we could check if the optimal arborization found has the same number of trees as the real neuron by not fixing the number of trees in advance.

Note also that there are 'obstacles', like blood vessels and cell somata, that the dendrite and axon trajectory has to circumvent. The more such obstacles there are, the greater the wiring cost would be. Moreover, the larger the arbor is, the more the trajectory modifications are. Thus, the wiring may not be perfectly optimal, particularly in axons. However, we did not take tortuosity into consideration (although it would have been more realistic) on the grounds of the complexity of the problem. Moreover, tortuosity is, at least in part, due to the presence of obstacles, and we did not have access to this information. In addition, different types of interneurons connect with different postsynaptic targets, and this is related to the

spatial characteristics of their axons. For example, the pattern of postsynaptic contacts may be 'distributed' or evenly spaced, whereas others may show a 'gradient' pattern where the distribution of contacts changes in a specific direction. 'Clustered' terminal branches are characteristic of chandelier cells that innervate pyramidal-cell axon initial segments (see, e.g., Ascoli et al. [2008], Blazquez-Llorca et al. [2015]). Further studies using more complete data on the synaptic characteristics of the cells under study and the local spatial distribution and density of the blood vessels and somata where the neuron is localized will make the wiring rules of single neurons easier to interpret.

9.3 Wiring economy of pyramidal neurons

In Section 9.2, we found that wiring was near optimal in most of the tested dendritic and axonal trees of the different types of interneurons that we examined, including Martinotti, large basket, common type, horse tail, chandelier and common basket cells. These GABAergic interneurons account for no more than a minority of all neurons in the cerebral cortex and are genetically, molecularly, anatomical and physiologically distinct from pyramidal cells, the most abundant type of neuron in the cerebral cortex [Anderson et al., 1999, DeFelipe, 1993, DeFelipe and Fariñas, 1992, Fishell and Rudy, 2011, Spruston, 2008].

In this section we analyze the neuronal wiring of individual pyramidal cells to check if this type of neuron also optimizes brain connectivity in terms of neuronal wiring cost. Since it is well established that pyramidal cell structure varies between different cortical areas and species (see Elston [2007] and DeFelipe [2011] for reviews), our study focuses on the hindlimb somatosensory cortex of Wistar rats at postnatal day 14. Furthermore, we used this experimental animal and at this age since we intended to integrate these data with other anatomical, molecular, and physiological data that have already been collected from the same cortical region of the postnatal day 14 Wistar rats. The final goal is to create a detailed, biologically accurate model of circuitry across all layers of the primary somatosensory cortex within the framework of the Blue Brain Project [Markram et al., 2015].

Unfortunately, current methodological limitations restrict the analysis to either the complete basal arbors (horizontal sections) or truncated apical and basal arbors (coronal sections) of pyramidal cells. For the sake of consistency with our previous studies, we opted to study the basal dendrites first. Therefore, we investigated the dendritic architecture of complete basal arbors of pyramidal neurons in all cortical layers (layers II, III, IV, Va, Vb and VI) as it has been shown that dendritic morphologies are statistically different in each cortical layer [Rojo et al., 2016]. Thus, we were also interested in examining whether, within a cortical area, there are possible differences in wiring optimality across all cortical layers.

9.3.1 Data

We analyzed the neuronal wiring of 288 3D-reconstructed complete basal arborizations of pyramidal cells across cortical layers II, III, IV, Va, Vb and VI of the somatosensory neocortex of the P14 rats (48 cells per layer). These basal dendritic arbors are made up of several main

trunks, which are in turn composed of several dendrites. For the sake of simplicity and unless otherwise stated, we refer to these single trunks of basal dendritic arbors as *dendritic trees* (Fig 9.9). Data on the reconstruction and dendritic structure of these cells has been already published [Rojo et al., 2016] and can be found on Figshare[2].

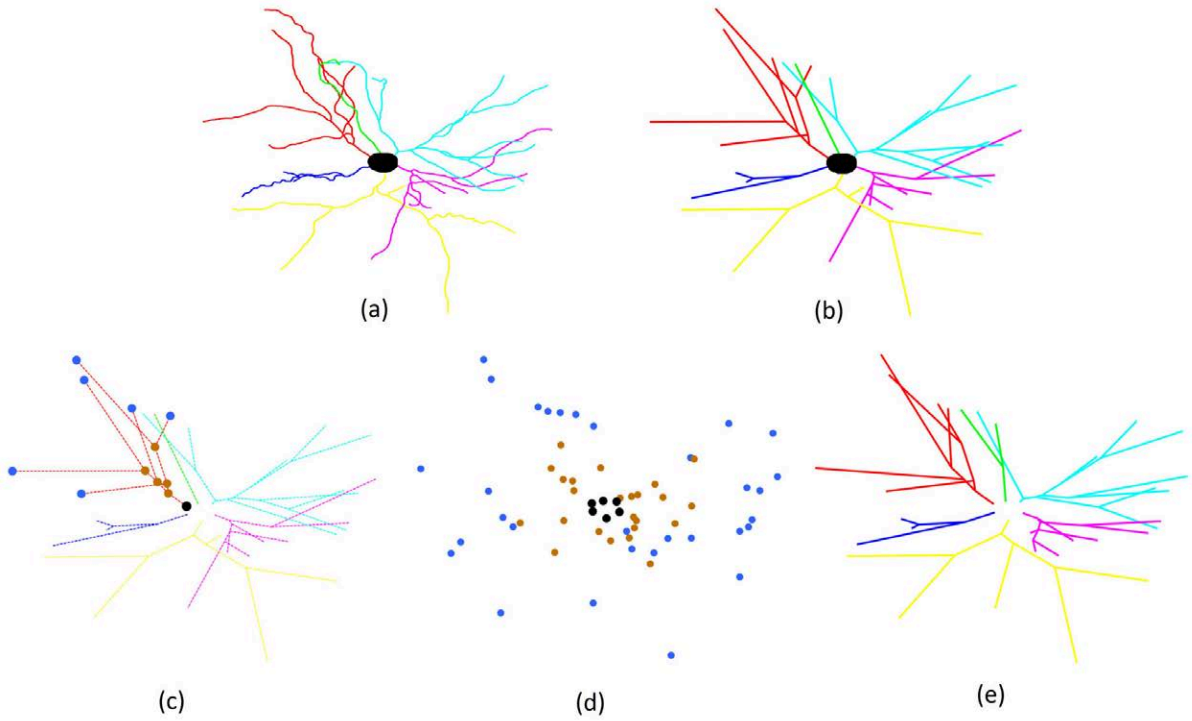


Figure 9.9: Example of one basal dendritic arbor of a pyramidal cell in layer II. (a) Real 3D Neurolucida reconstruction where each dendritic tree is shown in a different color. (b) Simplified real dendritic arbor where all connections are drawn as straight lines as we measure the (straight) length between points. (c) Example of the identification of the root (black), branching points (brown) and terminal points (blue) of one dendritic tree. (d) Point cloud formed by all the roots, branching and terminal points of the six basal dendritic trees. (e) Shortest arborization found for the point cloud shown in (d)

9.3.2 Wiring analysis

We analyzed the neuronal basal wiring of single pyramidal neurons following the procedure described in Section 9.2 for dendritic wiring. Briefly, for each neuron, we started from a point cloud formed by the roots, branching points and terminal points of the real neuron and searched for the optimal dendritic arborization. This is defined as the arborization that has the shortest total length, subject to the output structure retaining the roots and terminal points of real neuronal trees. In addition, the output structure also contained the same number of branches of each real branching point. Then we compared the output minimal wiring length with the real wiring length. We used an approximate wiring length in both the

[2] https://figshare.com/articles/pyramidalCells_ascFiles_zip/4193457/1

real and output tree structures, because we measured the Euclidean distance between two connected points, that is, we ignored the path tortuosity.

Fig 9.9 illustrates this procedure with a pyramidal cell from layer II (all the neurons analyzed in this study are shown in Rojo et al. [2016]). The basal dendritic arbor of the neuron in Fig 9.9 has six main dendritic trees (shown in different colors in Fig 9.9(a)). Fig 9.9(b) shows the simplified dendritic arbor where all connections are straight lines with a total wiring length of 1954.97 μm. For each dendritic tree, we identify the root of the tree, the branching points and the terminal points. Fig 9.9(c) shows the three types of points in the red tree (the root is shown in black, the branching points in brown and the terminal points in blue). We form the point cloud (Fig 9.9(d)) with the roots, branching and terminal points of all dendritic trees. We search for the minimum length arborization going through the above points. Fig 9.9(e) shows the best (shortest) structure found for this neuron, with a total length of 1912.79 μm (2.16% shorter than the real wiring in Fig 9.9(b)). Note that since the roots are unchanged, the number of constructed trees always matches the number of trees in the real neuron. The branching and terminal points of different dendritic trees can be combined to arrive at an arborization with minimum length wiring.

As in Section 9.2, to get the shortest arborization, we formulated the search for the optimal wiring of each neuron as a combinatorial optimization problem. Specifically, we used the steady-state genetic algorithm [Syswerda, 1991] through graph theory with the permutation-based representation presented in Chapter 8. Due to the stochasticity of genetic algorithms, we repeated the search for the optimal dendritic wiring 20 times for each cell and then chose the best (shortest) structure found for comparison with the real dendritic arborization.

9.3.3 Results

The dendritic structure of the 288 analyzed cells are described in detail in Rojo et al. [2016]. The mean length of basal dendritic wiring, in microns, grouping the cells by layer is shown in red in Fig 9.10. This figure shows that, on average, the wiring of the neurons in layers Va, Vb and VI is longer than the neurons belonging to the three more superficial layers.

The mean number of trees in the basal arborizations of the 48 cells analyzed in each cortical layer ranges from 4.96 (layer IV) to 7.48 (layer VI), while the mean number of points in these arborizations is between 42.67 (layer IV) and 66.67 (layer Va) (Table 9.5). To check whether there were significant differences between layers, we performed a multiple mean comparison test on the number of trees and the number of points. First, we checked if the necessary assumptions to apply ANOVA were satisfied, i.e., if data were normally distributed (Kolmogorov-Smirnov test) and if homoscedasticity was met (Levene's test). For the number of trees, none of the above assumptions were met, on which ground we used the Kruskal-Wallis test. The resulting p-value was 3.605e-12, i.e., there were differences in the number of trees between layers. Then, we applied the Mann-Whitney test with the Bonferroni method to adjust the p-values for pairwise comparisons. We found that there were differences between the number of trees in layers II vs. Vb, II vs. VI, III vs. IV, III vs. VI, IV vs. Va, IV

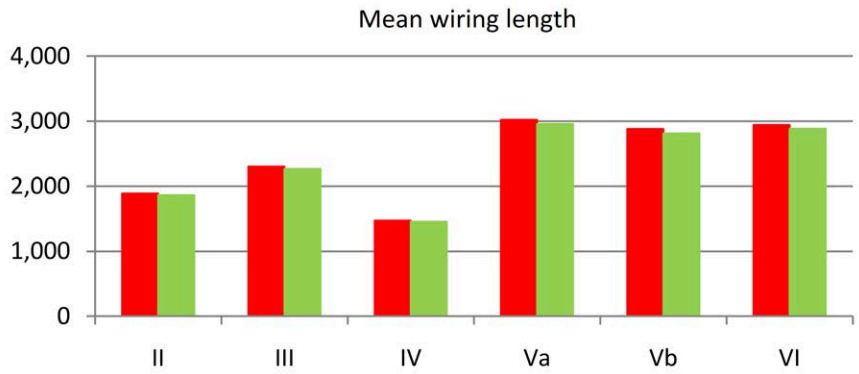


Figure 9.10: Mean wiring length (μm) of the 48 analyzed cells in each cortical layer (red) versus mean wiring length of the shortest arborizations found by our optimization algorithm for each layer (green). The optimization algorithm found an equal or slightly better (shorter) wiring for all the neurons in all the layers. We found the biggest difference with respect to the real wiring in layer Va, where the synthetic wiring was, on average, 2.06% shorter than the real wiring. The smallest difference occured in layer IV, where the optimized wiring was, on average, 1.01% shorter than the real wiring

vs. Vb and IV vs. VI. For the number of points in the arborizations of each layer, data were normally distributed but homocedasticity was not met. Therefore, we again applied the Kruskal-Wallis test and found significant differences between layers (p-value = 3.05e-11). In pairwise comparison, we found differences between layer IV and all the remaining layers.

Table 9.5: Mean and standard deviation ($\bar{x}_{\pm s}$) of the number of dendritic trees and the number of points of the dendritic point clouds (roots, branching points and terminal points) of the 48 cells of each cortical layer

Layer	Trees	Points
II	$5.50_{\pm 1.01}$	$61.98_{\pm 9.19}$
III	$5.94_{\pm 1.29}$	$64.04_{\pm 10.56}$
IV	$4.96_{\pm 1.20}$	$42.67_{\pm 15.67}$
Va	$6.52_{\pm 2.04}$	$66.67_{\pm 22.12}$
Vb	$6.75_{\pm 1.72}$	$60.63_{\pm 17.70}$
VI	$7.48_{\pm 2.03}$	$62.75_{\pm 18.82}$

We computed the optimal wiring length of the 288 cells. In order to compare the optimized wiring with the real wiring of each neuron, we calculated the percentage resulting from dividing the length of the shortest solution found by the real neuronal length. A figure of 100% shows that the length of the best solution found by our algorithm is equal to the total real wiring length of the neuron. A figure below 100% denotes that the solution found is better (shorter) than the real length, while a figure above 100% shows that the optimization algorithm is not able to improve the real dendritic tree.

Fig 9.11 shows the box plot of the results. We found a shorter wiring than the real wiring for all neurons, except for one neuron in layer IV, for which the best solution found matched

the real situation, that is, the percentage optimality for this neuron was 100%. A neuron in layer Va was found to have a wiring length that was nearly 5% shorter than the real one (95.06%, best result found). The percentage optimality for the remaining neurons ranged from 95.06% to 100%. The mean wiring length of the best solutions found for the 48 cells of each layer is shown in green in Fig 9.10.

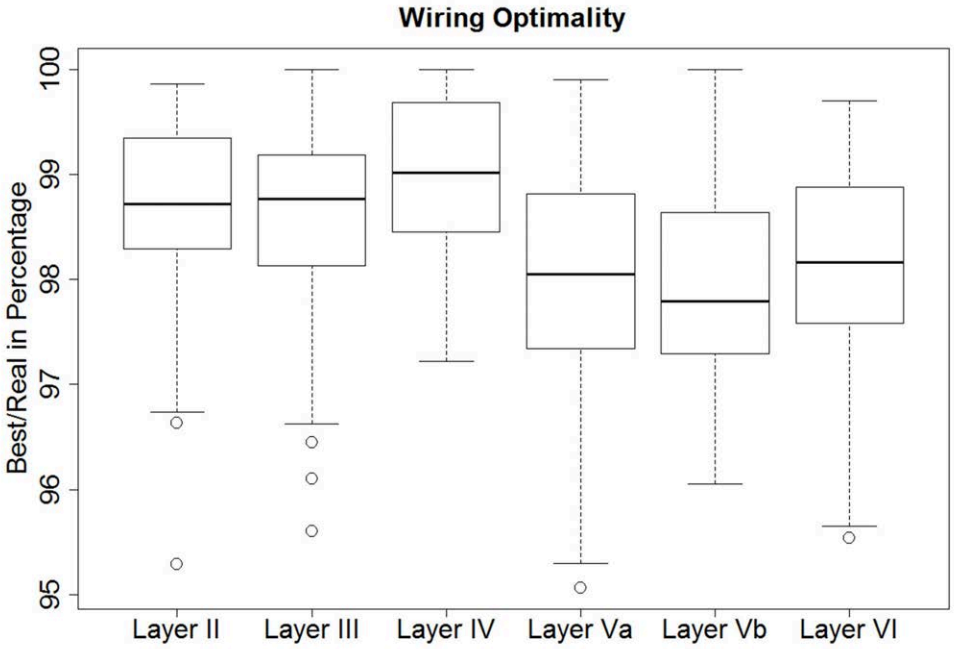


Figure 9.11: Box plot of the wiring analysis results for all layers. All the solutions are equal to or less than 100%, signifying that the solutions found by the optimization algorithm had a length equal to or shorter than the real wiring of the cells. For layers II, III and IV, the real neuronal wiring was closer to the shortest solutions found. Deeper layers had a higher degree of dispersion (steeper spacing between the parts of the box)

Fig 9.12 shows the mean percentage optimality for the neurons of each layer. Bluish colors denote that the wiring length of the best solutions found was further removed from the real wiring length, i.e., represented structures that offer a bigger improvement on the real neuronal structures. Reddish colors show that the total length of the resulting solutions was closer to the real neuronal length. Note, however, that the results ranged from 97.94% in layer Va to 98.99% in layer IV. Accordingly, all the results were very close to 100%.

We analyzed whether there were any significant differences in the wiring optimality, grouping neurons by layer. Since the data were normally distributed and homoscedasticity was met, we applied ANOVA. The resulting p-value was 1.54e-08, i.e., we rejected the null hypothesis of equal optimality in all six cortical layers. To find the differences between groups, we ran Tukey's HSD (honest significant difference) test to evaluate all pairwise comparisons. The results showed that there were significant differences between the following pairs of layers: Va vs. II, Va vs. III, Va vs. IV, Vb vs. II, Vb vs. III, Vb vs. IV and VI vs. IV. Analyzing the results, we concluded that the behavior of our algorithm by layers could be divided into two

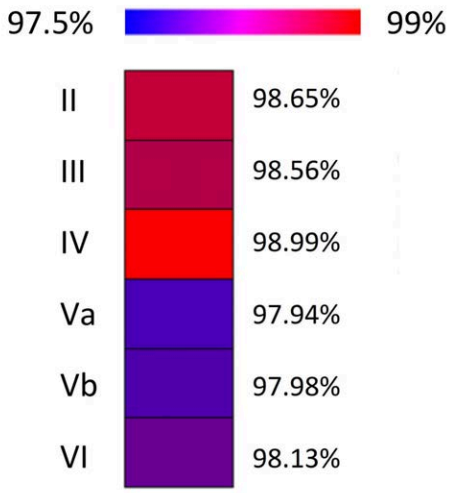


Figure 9.12: Mean optimality percentages of each cortical layer. Figures closer to 100% denote that the real neuronal wiring was closer to the shortest solutions found

groups: (i) layers II, III and IV (reddish in Fig 9.12) and (ii) layers Va, Vb and VI (bluish in Fig 9.12). We grouped cells accordingly and found that the percentage optimality of the second group (layers Va, Vb and VI) was significantly lower.

9.3.4 Conclusions

In this section we analyzed the neuronal basal wiring of single pyramidal neurons across cortical layers following the procedure described in Section 9.2 for dendritic and axonal wiring optimization of GABAergic neurons. The interneurons discussed in Section 9.2 had rather complex morphologies and they showed many different anatomical characteristics, whereas pyramidal cells represent a much more homogeneous population of neurons. Thus, we tested the hypothesis of optimal neuronal wiring in single pyramidal cells with this method which represents a different approach from previous research on neuronal wiring. Specifically, the method imposes constraints that provide realistic synthetic arborizations, that is, forces the synthetic wiring of a specific cell to pass through the branching points to reach the terminal points of this neuron. It also limits the number of times that the points branch out. With this procedure, we proved that we can explain the wiring economy of single pyramidal cells considering only one specific criterion, i.e., wiring length.

The morphological characteristics of the same 288 pyramidal cells were analyzed in Rojo et al. [2016], concluding that there is a systematic layer-specific variation of the basal dendritic pattern in pyramidal cells. More specifically, the branching structure of pyramidal cells became progressively larger and more complex from superficial to deeper layers, save for layer IV, which contained the simplest cells. Although the morphological characteristics are statistically different in each cortical layer, our study has found that basal wiring arborizations were near optimal in terms of wiring length in all cases (the biggest difference between the shortest solution found for a neuron and the length of its real basal wiring was less than 5%).

However, there appears to be a relationship between dendrite complexity and wiring economy since the solutions for the most superficial layers found by our algorithm were closer to the real wiring in our study, that is, the real cells manage to grow more optimally if they have a simpler branching structure. More specifically for cells in layer IV, the simplest according to Rojo et al. [2016], the real and optimal neuronal wirings were closer than in other layers. Nevertheless, it is noteworthy that neuronal connectivity depends not only on the dendritic wiring; but also on the density of dendritic spines, at least in the case of pyramidal cells. This is because most synapses on pyramidal cells are on their dendritic spines, and there are variations in the density of spines [DeFelipe and Fariñas, 1992]. However, we do not know whether or not the density of spines and wiring optimization are related. Thus, further studies should be performed to determine whether neurons with high densities of dendritic spines have more or less optimal wiring attributes compared to neurons with low densities of dendritic spines.

9.4 Conclusions

On the whole, the studies of this chapter show that the wiring economy of cortical neurons is not related to the type of neurons or their morphological complexities but to general principles of wiring economy. The study on the wiring of GABAergic interneurons in Section 9.2 concluded that dendritic wiring was near optimal in the tested neurons in spite of the clear morphological differences between Martinotti, large basket, common type, horse tail, chandelier and common basket cells. Nevertheless, this rule seems to apply to dendrites in particular since the wiring length of axonal trees of interneurons was, albeit near optimal, less so than for dendrites. In addition, in Section 9.3 we found that, although the differences in the wiring optimality between the basal dendritic arbors of pyramidal cells in different layers were small, they were statistically significant. As a result, the real wiring of the analyzed cells was nearer optimal in layers II, III and IV, whose branching structures are less complex according to Rojo et al. [2016], than in the deeper layers. Therefore, although wiring economy seems to be the general rule of optimization for cortical neurons irrespective of their anatomical and functional features, other factors may have an influence on the growth of the neuronal arborizations. More specifically, as previously discussed in Section 9.2, the trajectory of cellular processes could have 'obstacles', like blood vessels and cell somata, that the dendrite and axon trajectories have to circumvent. The more obstacles there are, the greater the wiring cost would be. Therefore, less optimal wiring might be expected in regions with a higher density of blood vessels and neurons.

Further studies in other cortical areas, layers and species are necessary to examine whether: (i) wiring economy is applicable to the dendritic and axonal arborizations of other types of neurons, including the apical dendrites and axons of pyramidal cells, and (ii) what is the biological significance, if any, of the small differences in the wiring of the basal dendritic arbors between pyramidal cells in different layers identified in Section 9.3 or between the dendritic or axonal arborization of certain types of interneurons (Section 9.2).

Part V

CONCLUSIONS AND FUTURE WORK

Chapter 10

Conclusions and future work

Chapter outline

This concluding chapter is organized as follows. Section 10.1 summarizes the main contributions and conclusions of this dissertation. Section 10.2 includes the list of publications and submissions derived from this research. Finally, Section 10.3 discusses the main lines of future work and some open issues.

10.1 Summary of contributions

The contributions have been organized into two parts:

- Part III includes our work on point process statistics. In Chapter 5 we perform a replicated point pattern-based analysis applied to the 3D spatial distribution of synapses in the cerebral cortex. The chapter describes a simulation process, along with a multiple mean comparison test, to investigate whether there were differences in the intensity (synaptic density) between groups (layers). We find that RSA processes described the spatial distribution of synapses in all samples of each layer which argues in favor of a common general principle of synaptic organization. We also find that the synaptic distribution in layers II to VI conforms to a common underlying RSA process with different synaptic densities per layer. Interestingly, the results show that synapses in layer I have a slightly different spatial distribution from the other layers. In order to collect and explain the variability in each group's intensity, we introduce for the first time in this context a simulation and thinning procedure in conjunction with a cross-validation technique to honestly estimate the goodness-of-fit of the resulting models within each group of replicates. This chapter also presents the software developed to process and analyze the 3D spatial distribution of synapses in the cerebral cortex.

  Chapter 6 expands the existing 2D computational techniques for spatial analysis along networks to perform a 3D network spatial analysis. In this chapter we apply this 3D analysis to the modeling of spine distribution along dendritic networks of human

pyramidal neurons in both basal and apical dendrites. Considering the dendritic arborizations of each pyramidal cell as a group of instances of the same observation (the neuron), we use replicated point patterns together with network spatial analysis for the first time to search for significant differences in the spine distribution of basal dendrites between different cells and between all basal and apical dendrites. To do this, we use a recent variant of Ripley's K function defined to work along networks. Our results suggest that dendritic spine distribution in basal dendritic arbors adheres to common rules and highlight that synaptic input information processing is different between apical and basal dendritic arborizations.

In Chapter 7 we characterize the spatial distribution of branching and terminal points of dendritic trees using nearest neighbour distances, particularly, a measure R defined as the ratio between the observed mean nearest neighbour distance of any set of points in a given volume and the expected mean nearest neighbour distance of the same number of points distributed uniformly in the same volume. We find that the distribution of branching and terminal points depend strongly on the cell types. Moreover, we find that R is only weakly correlated with other commonly used branching statistics, suggesting that it might reflect features of dendritic morphology that are not captured by commonly studied branching statistics. Besides studying the spatial distribution of these points in different types of real cells, in this chapter we also use morphological models based on optimal wiring principles to study the relation between different initial point distributions and resulting dendritic branching structures. Using our models, we find that branching and terminal points in dendrites are generally spread out more regularly than the target points from which the dendrite structures are determined. The most common way to obtain the unknown enclosing volume of a point cloud is to use the convex hull. However, with this choice the supporting volume is often overestimated. In this chapter, we use an extension to the notion of convex hull called α-shapes to obtain an accurate tight hull of the volume surrounding a given point cloud. Further, a naive calculation of R yields a biased result due to edge effects. We address this issue by developing a Monte Carlo based approach to estimate R values.

- Part IV includes our work on network design optimization. Finding the DCMST of a graph is a widely studied NP-hard optimization problem whose one of its most important applications is network design. Chapter 8 proposes a new variant of the DCMST problem: the DRCMST problem. A DRCMST is a DCMST where the role of each node in the tree is determined a priori by choosing among root node, intermediate node and leaf node. In addition, the number of root nodes is not limited to one, i.e., a forest rather than a single tree can be built. In this chapter we propose a novel permutation-based representation to encode forests of DRCMSTs. In this new representation, one permutation simultaneously encodes all the trees to be built. We compare the performance of GAs and EDAs to solve a variety of synthetic simulated DRCMST instances using the jMetal framework. jMetal already contained the implementation of GAs and

we include the implementation of EDAs. The modeling of network design problems can benefit from the possibility of generating more than one tree and determining the role of the nodes in the network. To illustrate the applicability of our approach, we formulate the trans-European transport network as a DRCMST problem. In this network design, we simultaneously optimize nine transport corridors and show that it is straightforward using the proposed representation to add constraints depending on the specific characteristics of the network.

In Chapter 9 we test the hypothesis that individual cortical neurons optimize brain connectivity in terms of wiring length, formulating it as a network design optimization problem, particularly, a DRCMST problem. The optimal arborization is defined as that with the shortest total wiring length provided that all neuron bifurcations are respected and the extent of the neuronal arborizations remains unchanged. We analyze both the axonal and dendritic trees of a set of different types of cortical GABAergic interneurons and the basal dendritic arborizations of a homogeneous population of pyramidal cells, examining in the latter group if there are differences in wiring optimality across all cortical layers. Our results show that wiring economy of cortical neurons is related to the way in which neuronal arborizations grow irrespective of the type of neurons or their morphological complexities. We develop and make available to the user software enabling to analyze the wiring optimality of a 3D neuron from its specification in .asc format. Both dendritic and axonal wiring can be analyzed, in both pyramidal neurons and interneurons. This chapter also introduces a parallelization strategy in order to solve large DRCMST problems (like those of axonal arbors). Complex problems are divided into sub-problems that are solved using parallel computing; then their solutions are put together and recombined.

10.2 List of publications

The publications and submissions derived from this research are listed below.

A. Peer-reviewed JCR journals

- L. Anton-Sanchez, C. Bielza, A. Merchán-Pérez, J. R. Rodríguez, J. DeFelipe, and P. Larrañaga. Three-dimensional distribution of cortical synapses: A replicated point pattern-based analysis. *Frontiers in Neuroanatomy*, 8:Article 85, 2014. Also available in the ebook: http://www.frontiersin.org/books/Quantitative_Analysis_of_Neuroanatomy/829. Impact factor (JCR 2014): 3.544. Ranking: 3/21 (Quartile 1). Category: Anatomy & morphology.

- L. Anton-Sanchez, C. Bielza, R. Benavides-Piccione, J. Felipe, and P. Larrañaga. Dendritic and axonal wiring optimization of cortical GABAergic interneurons. *Neuroinformatics*, 14(4):453-464, 2016. Impact factor (JCR 2015): 2.864. Ranking: 15/104 (Quartile 1). Category: Computer science, interdisciplinary applications.

- L. Anton-Sanchez, C. Bielza, P. Larrañaga, and J. DeFelipe. Wiring economy of pyramidal cells in the juvenile rat somatosensory cortex. *PLoS One*, 11(11):1-10, 2016. Impact factor (JCR 2015): 3.057. Ranking: 11/63 (Quartile 1). Category: Multidisciplinary sciences.

- L. Anton-Sanchez, C. Bielza, and P. Larrañaga. Network design through forests with degree- and role- constrained minimum spanning trees. *Journal of Heuristics*, 23(1):31-51, 2017. Impact factor (JCR 2015): 1.344. Ranking: 36/105 (Quartile 2). Category: Computer science, theory & methods.

B. Submissions

- L. Anton-Sanchez, P. Larrañaga, R. Benavides-Piccione, I. Fernaud-Espinosa, J. DeFelipe, and C. Bielza. Three-dimensional spatial modeling of spines along dendritic networks in human cortical pyramidal neurons. *Submitted*, 2017.

- L. Anton-Sanchez, F. Effenberger, C. Bielza, P. Larrañaga and H. Cuntz. Local statistics of input space shape dendritic morphology. *Submitted*, 2017.

C. Communications

- L. Anton-Sanchez, C. Bielza, and P. Larrañaga. Towards optimal neuronal wiring through estimation of distribution algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2013 Companion*, pages 1647-1650, 2013. 1st award at the Student Workshop.

D. Collaborations

- J. Morales, R. Benavides-Piccione, M. Dar, I. Fernaud, A. Rodríguez, L. Anton-Sanchez, C. Bielza, P. Larrañaga, J. DeFelipe, and R. Yuste. Random positions of dendritic spines in human cerebral cortex. *Journal of Neuroscience*, 34(30):10078-10084, 2014. Impact factor (JCR 2014): 5.924. Ranking: 26/256 (Quartile 1). Category: Neurosciences.

10.3 Future work

The open issues and future lines of this dissertation have been already discussed in the specific conclusions section of each chapter. This section summarizes the most relevant.

In Chapter 5 we perform a 3D spatial analysis in the context of replicated point patterns. Replicated point patterns and point pattern analysis in more than two dimensions, are two of the spatial point process areas that need more development, as well as marked point patterns. We intend to continue research in these areas and provide improvements in existing methodology in order to realistically model complex spatial datasets.

In this dissertation the third dimension has been taken into account for the first time when working with spatial analysis on linear networks (Chapter 6). We aim to improve the computational efficiency of our approach in order to be really useful for the spatial analysis of 3D real-world networks (the inclusion of the third dimension considerably increases the computational load especially with increased network complexity). In addition, we believe that considering the network volume and the possibility of events occurring on the surface of the network with volume, 3D network spatial analysis could still be more useful. We intend to develop additional methodology for this purpose.

The statistic R, used in Chapter 7 to measure the degree of clustering of a set of points in a given volume based on average nearest neighbour distances, can be extended to deal with more general cases. On the one hand, we would like to extend R to the inhomogeneous case. On the other hand, we are interested in analyzing the behavior of nearest neighbour distances considering the k-th nearest neighbours of each point with $k \geq 2$.

In relation to our proposal for network design optimization, i.e., the DRCMST presented in Chapter 8, we intend to work on improving cycle detection, which could speed up the algorithms. Depending on the problem, it might be interesting to consider a more complete fitness evaluation function and, if there were several optimization criteria to be considered, we might also think about the convenience of moving towards an approach based on multi-objective. Our proposal to solve DRCMST problems with a large number of nodes is to decompose the original problem into subproblems of smaller size and parallelize problem solving (Chapter 9). In future research, we aim to improve the way in which the subproblems are created and how the best solutions found in these subproblems are combined to output the overall solution.

Regarding neuroscience applications developed in this dissertation, in Chapter 5 we model the 3D spatial distribution of cortical synapses in P-14 rats. We found that random spatial distribution of synapses is probably a common general pattern of cortical synaptic organization. Nevertheless, since the synaptic density in the cerebral cortex changes, the conclusion of this study regarding spatial distribution may not be applicable at other time points during development. We intend to analyze other cortical areas, species and ages to verify our conclusions. In Chapter 6 we model the 3D spatial distribution of dendritic spines along dendritic arborization. We found that apical and basal dendritic arbors not only show distinct morphologies but also different rules of spine distribution. As future work, we intend to examine other models that further characterize the spatial distribution of spines along the basal and apical networks, especially at distances further from the cell body where these two types of arborizations show more differences.

Both in the study developed using spatial point process techniques with Euclidean distances (Chapter 5) and in the case of the study using network spatial analysis (Chapter 6), we could incorporate marks to the analysis, such as the type of synapse (symmetric or asymmetric) in the first case, or some characteristics of the spines as their length, volume or type, in the second case. The use of marked point patterns may be beneficial to elucidate important aspects of the spatial distribution of synapses and dendritic spines, respectively.

Regarding the study of wiring neuronal economy carried out in Chapter 9, we intend to analyze if wiring economy principle also holds in dendritic and axonal arborizations of other types of neurons, as well as in the apical dendrites and axons of pyramidal cells. In addition, our approach could easily be extended to consider other possible important aspects in wiring neuronal economy. For example, it would be easy to extend the wiring cost function to consider the minimization of both wiring length and the distance of each non-root point to the root of the tree (closely related to minimize the time that it takes for a signal to reach a synaptic contact from the cell body). Further, by not fixing the number of trees in advance, we would like to check if the optimal arborization found for a single neuron has the same number of trees than the real cell. We are also interested in analyzing whether neurons with high densities of dendritic spines have more or less optimal wiring compared to neurons with low densities of dendritic spines.

Bibliography

J. A. Aledo, J. A. Gámez, and D. Molina. Tackling the rank aggregation problem with evolutionary algorithms. *Applied Mathematics and Computation*, 222:632–644, 2013.

A. P. Alivisatos, M. Chun, G. M. Church, R. J. Greenspan, M. L. Roukes, and R. Yuste. The Brain Activity Map project and the challenge of functional connectomics. *Neuron*, 74(6): 970–974, 2012.

A. P. Alivisatos, M. Chun, G. M. Church, K. Deisseroth, J. P. Donoghue, R. J. Greenspan, P. L. McEuen, M. L. Roukes, T. J. Sejnowski, P. S. Weiss, and R. Yuste. The Brain Activity Map. *Science*, 339:1284–1285, 2013.

S. Anderson, M. Mione, K. Yun, and J. L. Rubenstein. Differential origins of neocortical projection and local circuit neurons: Role of Dlx genes in neocortical interneuronogenesis. *Cerebral Cortex*, 9:646–654, 1999.

Q. W. Ang, A. Baddeley, and G. Nair. Geometrically corrected second order analysis of events on a linear network, with applications to ecology and criminology. *Scandinavian Journal of Statistics*, 39(4):591–617, 2012.

L. Anton-Sanchez, C. Bielza, and P. Larrañaga. Towards optimal neuronal wiring through estimation of distribution algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2013 Companion*, pages 1647–1650, 2013.

L. Anton-Sanchez, C. Bielza, A. Merchán-Pérez, J. Rodríguez, J. DeFelipe, and P. Larrañaga. Three-dimensional distribution of cortical synapses: A replicated point pattern-based analysis. *Frontiers in Neuroanatomy*, 8:Article 85, 2014.

L. Anton-Sanchez, C. Bielza, R. Benavides-Piccione, J. DeFelipe, and P. Larrañaga. Dendritic and axonal wiring optimization of cortical GABAergic interneurons. *Neuroinformatics*, 14 (4):453–464, 2016a.

L. Anton-Sanchez, C. Bielza, P. Larrañaga, and J. DeFelipe. Wiring economy of pyramidal cells in the juvenile rat somatosensory cortex. *PLoS One*, 11(11):1–10, 2016b.

L. Anton-Sanchez, C. Bielza, and P. Larrañaga. Network design through forests with degree- and role-constrained minimum spanning trees. *Journal of Heuristics*, 23(1):31–51, 2017a.

L. Anton-Sanchez, F. Effenberger, C. Bielza, P. Larrañaga, and H. Cuntz. Local statistics of input space shape dendritic morphology. *Submitted*, 2017b.

L. Anton-Sanchez, P. Larrañaga, R. Benavides-Piccione, I. Fernaud-Espinosa, J. DeFelipe, and C. Bielza. Three-dimensional spatial modeling of spines along dendritic networks in human cortical pyramidal neurons. *Submitted*, 2017c.

J. I. Arellano, R. Benavides-Piccione, J. DeFelipe, and R. Yuste. Ultrastructure of dendritic spines: Correlation between synaptic and spine morphologies. *Frontiers in Neuroscience*, 1:Issue 1, 2007.

R. Armañanzas and G. Ascoli. Towards the automatic classification of neurons. *Trends in Neurosciences*, 38(5):307–318, 2015.

G. Ascoli, D. Donohue, and M. Halavi. Neuromorpho.org: A central resource for neuronal morphologies. *The Journal of Neuroscience*, 27(35):9247–9251, 2007.

G. Ascoli, L. Alonso-Nanclares, S. A. Anderson, G. Barrionuevo, R. Benavides-Piccione, A. Burkhalter, G. Buzsáki, B. Cauli, J. Defelipe, A. Fairén, D. Feldmeyer, G. Fishell, Y. Fregnac, T. F. Freund, D. Gardner, E. P. Gardner, J. H. Goldberg, M. Helmstaedter, S. Hestrin, F. Karube, Z. F. Kisvárday, B. Lambolez, D. A. Lewis, O. Marin, H. Markram, A. Muñoz, A. Packer, C. C. Petersen, K. S. Rockland, J. Rossier, B. Rudy, P. Somogyi, J. F. Staiger, G. Tamas, A. M. Thomson, M. Toledo-Rodriguez, Y. Wang, D. C. West, and R. Yuste. Petilla terminology: Nomenclature of features of GABAergic interneurons of the cerebral cortex. *Nature Reviews Neuroscience*, 9(7):557–568, 2008.

A. Baddeley. *Analysing Spatial Point Patterns in R. Workshop Notes.* Published online by CSIRO, 2010.

A. Baddeley and R. Turner. spatstat: An R package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6):1–42, 2005.

A. Baddeley, A. Boyde, C. Howard, and R. Moyeed. Analysis of a three-dimensional point pattern with replication. *Applied Statistics*, 42(4):641–668, 1993.

A. Baddeley, J. Møller, and R. Waagepetersen. Non- and semi-parametric estimation of interaction in inhomogeneous point patterns. *Statistica Neerlandica*, 54(3):329–350, 2000.

A. Baddeley, P. Gregori, J. Mateu, R. Stoica, and D. Stoyan. *Case Studies in Spatial Point Process Modeling.* Lecture Notes in Statistics, 185. Springer, 2006.

A. Baddeley, P. J. Diggle, A. Hardegen, T. Lawrence, R. K. Milne, and G. Nair. On tests of spatial pattern based on simulation envelopes. *Ecological Monographs*, 84(3):477–489, 2014a.

A. Baddeley, A. Jammalamadaka, and G. Nair. Multitype point process analysis of spines on the dendrite network of a neuron. *Journal of the Royal Statistical Society. Series C*, 63 (5):673–694, 2014b.

A. Baddeley, E. Rubak, and R. Turner. *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC Press, 2015.

S. Banerjee, B. P. Carlin, and A. E. Gelfand. *Hierarchical Modeling and Analysis for Spatial Data*. Monographs on Statistics and Applied Probability. Chapman and Hall/CRC, 2004.

G. Barnard. Discussion of: The spectral analysis of point processes (Barlett, MS). *Journal of the Royal Statistical Society. Series B*, 25:294, 1963.

M. Beining, T. Jungenitz, T. Radic, T. Deller, H. Cuntz, P. Jedlicka, and S. Schwarzacher. Adult-born dentate granule cells show a critical period of dendritic reorganization and are distinct from developmentally born cells. *Brain Structure and Function*, 222(3):1427–1446, 2017.

R. Benavides-Piccione, I. Fernaud-Espinosa, V. Robles, R. Yuste, and J. DeFelipe. Age-based comparison of human dendritic spine structure using complete three-dimensional reconstructions. *Cerebral Cortex*, 23(8):1798–1810, 2013.

B. Bergmann and G. Hommel. Improvements of general multiple test procedures for redundant systems of hypotheses. In *Multiple Hypotheses Testing*, volume 70 of *Medizinische Informatik und Statistik*, pages 100–115. Springer, 1988.

M. Berman. Testing for spatial association between a point process and another stochastic process. *Applied Statistics*, 35:54–62, 1986.

J. Berzhanskaya and G. Ascoli. Computational neuroanatomy. *Scholarpedia*, 3(3):1313, 2008.

J. Besag. Contribution to the discussion of Dr. Ripley's paper. *Journal of the Royal Statistical Society. Series B*, 39:193–195, 1977.

C. Bielza, J. A. Fernández del Pozo, P. Larrañaga, and E. Bengoetxea. Multidimensional statistical analysis of the parameterization of a genetic algorithm for the optimal ordering of tables. *Expert Systems with Applications*, 37(1):804–815, 2010.

T. Binzegger, R. J. Douglas, and K. A. C. Martin. A quantitative map of the circuit of cat primary visual cortex. *Journal of Neuroscience*, 24(39):8441–8453, 2004.

M. A. Bishop. Point pattern analysis of eruption points for the Mount Gambier volcanic sub-province: A quantitative geographical approach to the understanding of volcano distribution. *Area*, 39:230–241, 2007a.

M. A. Bishop. Point pattern analysis of north polar crescentic dunes, Mars: A geography of dune self-organization. *Icarus*, 191(1):151–157, 2007b.

R. S. Bivand, E. J. Pebesma, and V. Gómez-Rubio. *Applied spatial data analysis with R.* Springer, 2nd edition, 2013.

L. Blazquez-Llorca, V. Garcia-Marin, and J. DeFelipe. Pericellular innervation of neurons expressing abnormally hyperphosphorylated tau in the hippocampal formation of Alzheimer's disease patients. *Frontiers in Neuroanatomy*, 4:20, 2010.

L. Blazquez-Llorca, A. Merchán-Pérez, J. R. Rodríguez, J. Gascón, and J. DeFelipe. FIB/SEM technology and Alzheimer's disease: Three-dimensional analysis of human cortical synapses. *Journal of Alzheimer's Disease*, 34(4):995–1013, 2013.

L. Blazquez-Llorca, A. Woodruff, M. Inan, S. A. Anderson, R. Yuste, J. DeFelipe, and A. Merchán-Pérez. Spatial distribution of neurons innervated by chandelier cells. *Brain Structure and Function*, 220(5):2817–2834, 2015.

D. D. Bock, W.-C. A. Lee, A. M. Kerlin, M. L. Andermann, G. Hood, A. W. Wetzel, S. Yurgenson, E. R. Soucy, H. S. Kim, and R. C. Reid. Network anatomy and in vivo physiology of visual cortical neurons. *Nature*, 471(7337):177–182, 2011.

M. Bota and L. W. Swanson. The neuron classification problem. *Brain Research Reviews*, 56(1):79–88, 2007.

J. P. Bourgeois and P. Rakic. Changes of synaptic density in the primary visual cortex of the macaque monkey from fetal to adult stage. *The Journal of Neuroscience*, 13(7):2801–2820, 1993.

K. L. Briggman and W. Denk. Towards neural circuit reconstruction with volume electron microscopy techniques. *Current Opinion in Neurobiology*, 16:562–570, 2006.

J. M. L. Budd, K. Kovács, A. S. Ferecskó, P. Buzás, U. T. Eysel, and Z. F. Kisvárday. Neocortical axon arbors trade-off material and conduction delay conservation. *PLoS Computational Biology*, 6(3):1–25, 2010.

J. Burguet and P. Andrey. Statistical comparison of spatial point patterns in biological imaging. *PLoS One*, 9(2):1–12, 2014.

J. Burguet, Y. Maurin, and P. Andrey. A method for modeling and visualizing the three-dimensional organization of neuron populations from replicated data: Properties, implementation and illustration. *Pattern Recognition Letters*, 32(14):1894–1901, 2011.

J. Ceberio, A. Mendiburu, and J. A. Lozano. Introducing the Mallows model on estimation of distribution algorithms. In *Neural Information Processing*, volume 7063 of *Lecture Notes in Computer Science*, pages 461–470. Springer, 2011.

J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano. A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems. *Progress in Artificial Intelligence*, 1(1):103–117, 2012.

J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano. A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem. *IEEE Transactions on Evolutionary Computation*, 18(2):286–300, 2014.

J. Ceberio, A. Mendiburu, and J. A. Lozano. Kernels of Mallows models for solving permutation-based problems. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015*, pages 505–512, 2015.

S. Chandrasekhar. The law of distribution of the nearest neighbor in a random distribution of particles. *Reviews of Modern Physics. Stochastic Problems in Physics and Astronomy*, 15:86–87, 1943.

B. L. Chen, D. H. Hall, and D. B. Chklovskii. Wiring optimization can relate neuronal structure and function. *Proceedings of the National Academy of Sciences*, 103(12):4723–4728, 2006.

D. B. Chklovskii. Synaptic connectivity and neuronal morphology: Two sides of the same coin. *Neuron*, 43(5):609–617, 2004.

D. B. Chklovskii, T. Schikorski, and C. F. Stevens. Wiring optimization in cortical circuits. *Neuron*, 34(3):341–347, 2002.

P. S. Churchland, C. Koch, and T. J. Sejnowski. What is computational neuroscience? In *Computational Neuroscience*, pages 46–55. MIT Press, 1993.

P. J. Clark and F. C. Evans. Distance to nearest neighbor as a measure of spatial relationships in populations. *Ecology*, 35(4):445–453, 1954.

H. Cobb and J. Grefenstette. Genetic algorithms for tracking changing environments. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 523–530. Morgan Kaufmann, 1993.

R. Crandall. On the fractal distribution of brain synapses. In *Computational and Analytical Mathematics*, volume 50 of *Springer Proceedings in Mathematics & Statistics*, pages 325–348. Springer, 2013.

H. Cuntz. The dendritic density field of a cortical pyramidal cell. *Frontiers in Neuroanatomy*, 6:Article 2, 2012.

H. Cuntz, A. Borst, and I. Segev. Optimization principles of dendritic structure. *Theoretical Biology and Medical Modelling*, 4:21, 2007.

H. Cuntz, F. Forstner, J. Haag, and A. Borst. The morphological identity of insect dendrites. *PLoS Computational Biology*, 4(12):1–7, 2008.

H. Cuntz, F. Forstner, A. Borst, and M. Häusser. One rule to grow them all: A general theory of neuronal branching and its practical application. *PLoS Computational Biology*, 6(8):1–14, 2010.

H. Cuntz, F. Forstner, A. Borst, and M. Häusser. The TREES toolbox - probing the basis of axonal and dendritic branching. *Neuroinformatics*, 9(1):91–96, 2011.

H. Cuntz, A. Mathy, and M. Häusser. A scaling law derived from optimal dendritic wiring. *Proceedings of the National Academy of Sciences*, 109(27):11014–11018, 2012.

M. M. Czajko and J. Wojciechowski. Tree-based access network design under requirements for an aggregation network. *Elektronika - Konstrukcje, Technologie, Zastosowania*, 4:23–27, 2009.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes Volume I: Elementary Theory and Methods*. Springer, 2nd edition, 2003.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes. Vol. II*. Springer, 2nd edition, 2008.

C. Darwin. *On the Origin of Species by Means of Natural Selection*. Murray, 1859.

J. DeFelipe. Neocortical neuronal diversity: Chemical heterogeneity revealed by colocalization studies of classic neurotransmitters, neuropeptides, calcium-binding proteins, and cell surface molecules. *Cerebral Cortex*, 3:273–289, 1993.

J. DeFelipe. The evolution of the brain, the human nature of cortical circuits and intellectual creativity. *Frontiers in Neuroanatomy*, 5:Article 29, 2011.

J. DeFelipe and I. Fariñas. The pyramidal neuron of the cerebral cortex: Morphological and chemical characteristics of the synaptic inputs. *Progress in Neurobiology*, 39(6):563–607, 1992.

J. DeFelipe, P. Marco, A. Fairén, and E. Jones. Inhibitory synaptogenesis in mouse somatosensory cortex. *Cerebral Cortex*, 7:619–634, 1997.

J. DeFelipe, P. Marco, I. Busturia, and A. Merchán-Pérez. Estimation of the number of synapses in the cerebral cortex: Methodological considerations. *Cerebral Cortex*, 9:722–732, 1999.

J. DeFelipe, L. Alonso-Nanclares, and J. I. Arellano. Microstructure of the neocortex: Comparative aspects. *Journal of Neurocytology*, 31:299–316, 2002a.

J. DeFelipe, G. N. Elston, I. Fujita, J. Fuster, K. H. Harrison, P. R. Hof, Y. Kawaguchi, K. A. C. Martin, K. S. Rockland, A. M. Thomson, S. S. H. Wang, E. L. White, and R. Yuste. Neocortical circuits: Evolutionary aspects and specificity versus non-specificity of synaptic connections. Remarks, main conclusions and general comments and discussion. *Journal of Neurocytology*, 31(3-5):387–416, 2002b.

J. DeFelipe, H. Markram, and K. S. Rockland. The neocortical column. *Frontiers in Neuroanatomy*, 6(22):1–2, 2012.

J. DeFelipe, P. L. López-Cruz, R. Benavides-Piccione, C. Bielza, P. Larrañaga, S. Anderson, A. Burkhalter, B. Cauli, A. Fairén, D. Feldmeyer, G. Fishell, D. Fitzpatrick, T. F. Freund, G. González-Burgos, S. Hestrin, S. Hill, P. R. Hof, J. Huang, E. G. Jones, Y. Kawaguchi, Z. Kisvárday, Y. Kubota, D. A. Lewis, O. Marín, H. Markram, C. J. McBain, H. S. Meyer, H. Monyer, S. B. Nelson, K. Rockland, J. Rossier, J. L. R. Rubenstein, B. Rudy, M. Scanziani, G. M. Shepherd, C. C. Sherwood, J. F. Staiger, G. Tamás, A. Thomson, Y. Wang, R. Yuste, and G. A. Ascoli. New insights into the classification and nomenclature of cortical GABAergic interneurons. *Nature Reviews Neuroscience*, 14(3):202–216, 2013.

A. C. B. Delbem, A. de Carvalho, C. Policastro, A. Pinto, K. Honda, and A. C. García. Node-depth encoding for evolutionary algorithms applied to network design. In *Genetic and Evolutionary Computation*, volume 3102 of *Lecture Notes in Computer Science*, pages 678–687. Springer, 2004.

A. C. B. Delbem, T. W. de Lima, and G. P. Telles. Efficient forest data structure for evolutionary algorithms applied to network design. *IEEE Transactions of Evolutionary Computation*, 16(6):829–846, 2012.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

J. Derrac, S. García, D. Molina, and F. Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.

P. J. Diggle. *Statistical Analysis of Spatial Point Patterns*. Edward Arnold. 2003.

P. J. Diggle. *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. Chapman & Hall/CRC. 3rd edition, 2013.

P. J. Diggle, N. Lange, and F. M. Benes. Analysis of variance for replicated spatial point patterns in clinical neuroanatomy. *Journal of the American Statistical Association*, 86 (415):618–625, 1991.

P. J. Diggle, J. Mateu, and H. E. Clough. A comparison between parametric and non-parametric approaches to the analysis of replicated spatial point patterns. *Advances in Applied Probability*, 32(2):331–343, 06 2000.

K. Donnelly. Simulations to determine the variance and edge-effect of total nearest neighbour distance. In *Simulation Studies in Archaeology*, pages 91–95. Cambridge University Press, 1978.

M. Dry, K. Preiss, and J. Wagemans. Clustering, randomness, and regularity: Spatial distributions and human performance on the traveling salesperson problem and minimum spanning tree problem. *The Journal of Problem Solving*, 4(1):Article 2, 2012.

J. J. Durillo and A. J. Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

J. J. Durillo, A. J. Nebro, and E. Alba. The jmetal framework for multi-objective optimization: Design and architecture. In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2010.

M. Dwass. Modified randomization tests for nonparametric hypotheses. *Annals of Mathematical Statistics*, 28(1):181–187, 1957.

H. Edelsbrunner and E. P. Mucke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13:43–72, 1994.

H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983.

S. Eglen, D. Lofgreen, M. Raven, and B. Reese. Analysis of spatial relationships in three dimensions: tools for the study of nerve cell patterning. *BMC Neuroscience*, 9:68, 2008.

G. Elson and J. DeFelipe. Spine distribution in cortical pyramidal cells: A common organizational principle across species. *Progress in Brain Research*, 136:109–133, 2002.

G. Elston. Specialization of the neocortical pyramidal cell during primate evolution. In *Evolution of Nervous Systems: A Comprehensive Reference*, volume 4, pages 191–242. Academic Press, 2007.

J. W. Evans. Random and cooperative sequential adsorption. *Reviews of Modern Physics*, 65:1281–1329, 1993.

T. Fares and A. Stepanyants. Cooperative synapse formation in the neocortex. *Proceedings of the National Academy of Sciences*, 106(38):16463–16468, 2009.

J. C. Fiala, J. Spacek, and K. M. Harris. Dendritic spine pathology: Cause or consequence of neurological disorders? *Brain Research Reviews*, 39(1):29–54, 2002.

G. Fishell and B. Rudy. Mechanisms of inhibition within the telencephalon: Where the wild things are. *Annual Review of Neuroscience*, 34(1):535–567, 2011.

R. A. Fisher. *Design of Experiments*. Oliver and Boyd, Edinburgh, 1935.

M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

V. Garcia-Marin, L. Blazquez-Llorca, J.-R. Rodriguez, S. Boluda, G. Muntane, I. Ferrer, and J. DeFelipe. Diminished perisomatic GABAergic terminals on cortical neurons adjacent to amyloid plaques. *Frontiers in Neuroanatomy*, 3:Article 28, 2009.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

U. Hahn. A studentized permutation test for the comparison of spatial point patterns. *Journal of the American Statistical Association*, 107(498):754–764, 2012.

K. M. Harris, E. Perry, J. Bourne, M. Feinberg, L. Ostroff, and J. Hurlburt. Uniform serial sectioning for transmission electron microscopy. *The Journal of Neuroscience*, 26(47): 12101–12103, 2006.

P. Hertz. Über den geigenseitigen durchschnittlichen Abstand von Punkten, die mit bekannter mittlerer Dichte im Raume angeordnet sind. In *Mathematische Annalen*, volume 67, pages 387–398, 1909.

S. L. Hill, Y. Wang, I. Riachi, F. Schürmann, and H. Markram. Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits. *Proceedings of the National Academy of Sciences*, 109(42):E2885–E2894, 2012.

B. K. Hoffpauir, B. A. Pope, and G. A. Spirou. Serial sectioning and electron microscopy of large tissue volumes for 3D analysis and reconstruction: A case study of the calyx of Held. *Nature Protocols*, 2(1):9–22, 2007.

J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.

J. Illian, A. Penttinen, H. Stoyan, and D. Stoyan. *Statistical Analysis and Modelling of Spatial Point Patterns*. Wiley-Interscience, 2008.

E. Irurozki, B. Calvo, and J. A. Lozano. Sampling and learning the Mallows and Generalized Mallows models under the Cayley distance. *Methodology and Computing in Applied Probability*, pages 1–35, 2016.

M. Jafari-Mamaghani, M. Andersson, and P. Krieger. Spatial point pattern analysis of neurons using Ripley's K-function in 3D. *Frontiers in Neuroinformatics*, 4:Article 9, 2010.

A. Jammalamadaka, S. Banerjee, B. S. Manjunath, and K. S. Kosik. Statistical analysis of dendritic spine distributions in rat hippocampal cultures. *BMC Bioinformatics*, 14:287, 2013.

M. Kaiser and C. C. Hilgetag. Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems. *PLoS Computational Biology*, 2(7):e95, 2006.

J. Karbowski. Cortical composition hierarchy driven by spine proportion economical maximization or wire volume minimization. *PLoS Computational Biology*, 11(10):e1004532, 2015.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

G. Knott, H. Marchman, D. Wall, and B. Lich. Serial section scanning electron microscopy of adult brain tissue using focused ion beam milling. *The Journal of Neuroscience*, 28(12): 2959–2964, 2008.

J. Knowles, D. Corne, and M. Oates. A new evolutionary approach to the degree constrained minimum spanning tree problem. *IEEE Transactions on Evolutionary Computation*, 4: 125–134, 2000.

L. S. Krimer, R. L. Jakab, and P. S. Goldman-Rakic. Quantitative three-dimensional analysis of the catecholaminergic innervation of identified neurons in the macaque prefrontal cortex. *Journal of Neuroscience*, 17(19):7450–7461, 1997.

M. Krishnamoorthy, A. Ernst, and Y. Sharaiha. Comparison of algorithms for the degree constrained minimum spanning tree. *Journal of Heuristics*, 7(6):587–611, 2001.

J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic. Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2):129–170, 1999.

P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer, 2002.

C. Lüscher, R. A. Nicoll, R. C. Malenka, and D. Muller. Synaptic plasticity and dynamic modulation of the postsynaptic membrane. *Nature Neuroscience*, 3:545–550, 2000.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297. University of California Press, 1967.

C. L. Mallows. Non-null ranking models. *Biometrika*, 44(1-2):114–130, 1957.

H. Markram. The blue brain project. *Nature Reviews Neuroscience*, 7(2):153–160, 2006.

H. Markram. The human brain project. *Scientific American*, 306(6):50–55, 2012.

H. Markram, E. Muller, S. Ramaswamy, M. Reimann, M. Abdellah, C. A. Sanchez, A. Ailamaki, L. Alonso-Nanclares, N. Antille, S. Arsever, G. A. Kahou, T. K. Berger, A. Bilgili, N. Buncic, A. Chalimourda, G. Chindemi, J.-D. Courcol, F. Delalondre, V. Delattre,

S. Druckmann, R. Dumusc, J. Dynes, S. Eilemann, E. Gal, M. E. Gevaert, J.-P. Ghobril, A. Gidon, J. W. Graham, A. Gupta, V. Haenel, E. Hay, T. Heinis, J. B. Hernando, M. Hines, L. Kanari, D. Keller, J. Kenyon, G. Khazen, Y. Kim, J. G. King, Z. Kisvarday, P. Kumbhar, S. Lasserre, J.-V. Le Bé, B. R. C. Magalhães, A. Merchán-Pérez, J. Meystre, B. R. Morrice, J. Muller, A. Muñoz Céspedes, S. Muralidhar, K. Muthurasa, D. Nachbaur, T. H. Newton, M. Nolte, A. Ovcharenko, J. Palacios, L. Pastor, R. Perin, R. Ranjan, I. Riachi, J.-R. Rodríguez, J. L. Riquelme, C. Rössert, K. Sfyrakis, Y. Shi, J. C. Shillcock, G. Silberberg, R. Silva, F. Tauheed, M. Telefont, M. Toledo-Rodriguez, T. Tränkler, W. Van Geit, J. V. Díaz, R. Walker, Y. Wang, S. M. Zaninetta, J. DeFelipe, S. L. Hill, I. Segev, and F. Schürmann. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456–492, 2015.

G. McSwiggan, A. Baddeley, and G. Nair. Kernel density estimation on a linear network. *Scandinavian Journal of Statistics*, pages 1–22, 2016.

A. Merchán-Pérez, J.-R. Rodriguez, L. Alonso-Nanclares, A. Schertel, and J. DeFelipe. Counting synapses using FIB/SEM microscopy: A true revolution for ultrastructural volume reconstruction. *Frontiers in Neuroanatomy*, 3:Article 18, 2009.

A. Merchán-Pérez, R. Rodríguez, S. González, V. Robles, J. DeFelipe, P. Larrañaga, and C. Bielza. Three-dimensional spatial distribution of synapses in the neocortex: A dual-beam electron microscopy study. *Cerebral Cortex*, 24(6):1579–1588, 2014.

Y. Mishchenko, T. Hu, J. Spacek, J. Mendenhall, K. M. Harris, and D. B. Chklovskii. Ultrastructural analysis of hippocampal neuropil from the connectomics perspective. *Neuron*, 67(6):1009–1020, 2010.

J. Møller and R. P. Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*. Chapman & Hall, 2004.

J. Morales, L. Alonso-Nanclares, J. R. Rodríguez, J. DeFelipe, Á. Rodríguez, and A. Merchán-Pérez. Espina: A tool for the automated segmentation and counting of synapses in large stacks of electron microscopy images. *Frontiers in Neuroanatomy*, 5:Article 18, 2011.

J. Morales, R. Benavides-Piccione, M. Dar, I. Fernaud, A. Rodríguez, L. Anton-Sanchez, C. Bielza, P. Larrañaga, J. DeFelipe, and R. Yuste. Random positions of dendritic spines in human cerebral cortex. *Journal of Neuroscience*, 34(30):10078–10084, 2014.

M. Myllymäki, I. G. Panoutsopoulou, and A. Särkkä. Analysis of spatial structure of epidermal nerve entry point patterns based on replicated data. *Journal of Microscopy*, 247(3): 228–39, 2012.

R. Nieuwenhuys. The neocortex. An overview of its evolutionary development, structural organization and synaptology. *Anatomy and Embryology*, 190(4):307–337, 1994.

F. Nissl. Nervenzellen und graue Substanz. *Munchener Medizinische Wochenschrift*, 45: 988–992,1023–1029,1060–1062, 1898.

J. Ohser. On estimators for the reduced second moment measure of point processes. *Mathematische Operationsforschung und Statistik. Series Statistics*, 14:63–71, 1983.

A. Okabe and K. Sugihara. *Spatial Analysis along Networks*. John Wiley & Sons, 2012.

A. Okabe and I. Yamada. The K-function method on a network and its computational implementation. *Geographical Analysis*, 33(3):271–290, 2001.

A. Okabe, T. Satoh, and K. Sugihara. A kernel density estimation method for networks, its computational method and a GIS-based tool. *International Journal of Geographical Information Science*, 23(1):7–32, 2009.

J. O'Kusky and M. Colonnier. A laminar analysis of the number of neurons, glia, and synapses in the adult cortex (area 17) of adult macaque monkeys. *The Journal of Comparative Neurology*, 210(3):278–290, 1982.

Z. Pawlas. Estimation of summary characteristics from replicated spatial point processes. *Kybernetika*, 47(6):880–892, 2011.

A. Pérez-Escudero and G. G. de Polavieja. Optimally wired subnetwork determines neuroanatomy of Caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 104(43):17180–17185, 2007.

A. Pérez-Escudero, M. Rivera-Alba, and G. G. de Polavieja. Structure of deviations from optimality in biological systems. *Proceedings of the National Academy of Sciences*, 106(5): 20544–20549, 2009.

A. Peters and B. R. Payne. Numerical relationships between geniculocortical afferents and pyramidal cell modules in cat primary visual cortex. *Cerebral Cortex*, 3(1):69–78, 1993.

E. Pitman. Significance tests which may be applied to samples from any population. *Supplement to the Journal of the Royal Statistical Society*, 4(1):119–130, 1937.

R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technology Journal*, 36:1389–1401, 1957.

G. R. Raidl and B. A. Julstrom. Edge sets: An effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*, 7(3):225–239, 2003.

P. Rakic, J. P. Bourgeois, M. F. Eckenhoff, N. Zecevic, and P. Goldman-Rakic. Concurrent overproduction of synapses in diverse regions of the primate cerebral cortex. *Science*, 231: 232–235, 1986.

P. Rakic, J. P. Bourgeois, and P. S. Goldman-Rakic. Synaptic development of the cerebral cortex: Implications for learning, memory, and mental illness. *Progress in Brain Research*, 102:227–243, 1994.

S. Ramaswamy, J.-D. Courcol, M. Abdellah, S. R. Adaszewski, N. Antille, S. Arsever, G. Atenekeng, A. Bilgili, Y. Brukau, A. Chalimourda, G. Chindemi, F. Delalondre, R. Dumusc, S. Eilemann, M. E. Gevaert, P. Gleeson, J. W. Graham, J. B. Hernando, L. Kanari, Y. Katkov, D. Keller, J. G. King, R. Ranjan, M. W. Reimann, C. Rössert, Y. Shi, J. C. Shillcock, M. Telefont, W. Van Geit, J. Villafranca Diaz, R. Walker, Y. Wang, S. M. Zaninetta, J. DeFelipe, S. L. Hill, J. Muller, I. Segev, F. Schürmann, E. B. Muller, and H. Markram. The neocortical microcircuit collaboration portal: A resource for rat somatosensory cortex. *Frontiers in Neural Circuits*, 9:Article 44, 2015.

S. Ramón y Cajal. Estructura de los centros nerviosos de las aves. *Revista Trimestral de Histología Normal y Patológica*, 1:1–10, 1888.

S. Ramón y Cajal. *Textura del Sistema Nervioso del Hombre y de los Vertebrados*. Nicolás Moya, Madrid, 1899.

C. R. Reeves. A genetic algorithm for flowshop sequencing. *Computers and Operations Research*, 22(1):5–13, 1995.

M. W. Reimann, J. G. King, E. B. Muller, S. Ramaswamy, and H. Markram. An algorithm to predict the connectome of neural microcircuits. *Frontiers in Computational Neuroscience*, 9:Article 120, 2015.

B. D. Ripley. Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society. Series B*, 39:172–212, 1977.

B. D. Ripley. *Spatial Statistics*. John Wiley & Sons, 1981.

B. D. Ripley and J. P. Rasson. Finding the edge of a Poisson forest. *Journal of Applied Probability*, 14:483–491, 1977.

M. Rivera-Alba, S. N. Vitaladevuni, Y. Mishchenko, Z. Lu, S.-Y. Takemura, L. Scheffer, I. A. Meinertzhagen, D. B. Chklovskii, and G. G. de Polavieja. Wiring economy and volume exclusion determine neuronal placement in the drosophila brain. *Current Biology*, 21(23): 2000–2005, 2011.

M. Rivera-Alba, H. Peng, G. G. de Polavieja, and D. B. Chklovskii. Wiring economy can account for cell body placement across species and brain areas. *Current Biology*, 24(3): R109–R110, 2014.

C. Rojo, I. Leguey, A. Kastanauskaite, C. Bielza, P. Larrañaga, J. DeFelipe, and R. Benavides-Piccione. Laminar differences in dendritic structure of pyramidal neurons in the juvenile rat somatosensory cortex. *Cerebral Cortex*, 26:2811–2822, 2016.

B. S. Rowlingson and P. J. Diggle. Splancs: Spatial point pattern analysis code in S-PLUS. *Computers and Geosciences*, 19:627–655, 1993.

R. Ruiz and C. Maroto. A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, 165(2):479–494, 2005.

T. Schikorski and C. F. Stevens. Quantitative ultrastructural analysis of hippocampal excitatory synapses. *Journal of Neuroscience*, 17(15):5858–5867, 1997.

C. J. Schneider, H. Cuntz, and I. Soltesz. Linking macroscopic with microscopic neuroanatomy using synthetic neuronal populations. *PLoS Computational Biology*, 10(10): e1003921, 2014.

R. Sedgewick and K. Wayne. *Algorithms*. Addison-Wesley, 4th edition, 2011.

S.-M. Soak, D. Corne, and B.-H. Ahn. A new encoding for the degree constrained minimum spanning tree problem. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 3213 of *Lecture Notes in Computer Science*, pages 952–958. Springer, 2004.

N. Spruston. Pyramidal neurons: Dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, 9:206–221, 2008.

A. Stepanyants and D. Chklovskii. Neurogeometry and potential synaptic connectivity. *Trends in Neurosciences*, 28(7):387–394, 2005.

J. K. Stevens, T. L. Davis, N. Friedman, and P. Sterling. A systematic approach to reconstructing microcircuitry by electron microscopy of serial sections. *Brain Research*, 2(3): 265–93, 1980.

D. Stoyan, W. Kendall, and J. Mecke. *Stochastic Geometry and its Applications*. John Wiley & Sons, 2nd edition, 1995.

G. Syswerda. A study of reproduction in generational and steady-state genetic algorithms. *Foundation of Genetic Algorithms*, 1:94–101, 1991.

Y. Takumi, V. Ramírez-León, P. Laake, E. Rinvik, and O. P. Ottersen. Different modes of expression of AMPA and NMDA receptors in hippocampal synapses. *Nature Neuroscience*, 2(7):618–24, 1999.

E. Tarusawa, K. Matsui, T. Budisantoso, E. Molnár, M. Watanabe, M. Matsui, Y. Fukazawa, and R. Shigemoto. Input-specific intrasynaptic arrangements of ionotropic glutamate receptors and their impact on postsynaptic responses. *The Journal of Neuroscience*, 29(41): 12896–12908, 2009.

B. Torben-Nielsen and H. Cuntz. Introduction to dendritic morphology. In *The Computing Dendrite: From Structure to Function*, pages 3–22. Springer, 2014.

S. Tsutsui. Node histogram vs. edge histogram: A comparison of probabilistic model-building genetic algorithms in permutation domains. In *Proceedings of IEEE Congress on Evolutionary Computation, 2006*, pages 1939–1946, 2006.

M. N. M. van Lieshout. A J-function for inhomogeneous point processes. *Statistica Neerlandica*, 65(2):183–201, 2011.

M. N. M. van Lieshout and A. Baddeley. A nonparametric measure of spatial interaction in point patterns. *Statistica Neerlandica*, 50(3):344–361, 1996.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S-Plus*. Springer, 4th edition, 2002.

C. Von Economo. Ein Koeffizient für die Organisationshöhe der Grosshirnrinde. *Klinische Wochenschrift*, 5:593–595, 1926.

C. G. Wager, B. A. Coull, and N. Lange. Modelling spatial intensity for replicated inhomogeneous point patterns in brain imaging. *Journal of the Royal Statistical Society. Series B*, 66(2):429–446, 2004.

Q. Wen and D. B. Chklovskii. A cost–benefit analysis of neuronal morphology. *Journal of Neurophysiology*, 99(5):2320–2328, 2008.

Q. Wen, A. Stepanyants, G. N. Elston, A. Y. Grosberg, and D. B. Chklovskii. Maximization of the connectivity repertoire as a statistical principle governing the shapes of dendritic arbors. *Proceedings of the National Academy of Sciences*, 106(30):12536–12541, 2009.

I. Yamada. Edge effects. In *International Encyclopedia of Human Geography*, volume 3, pages 381–388. Elsevier Science, 2009.