Aspects of Statistical Analysis of Spatial Point Patterns

Vihrs, Ninna

DOI (link to publication from Publisher):
10.54337/aau495039894

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

ASPECTS OF STATISTICAL ANALYSIS OF SPATIAL POINT PATTERNS

BY
NINNA VIHRS

DISSERTATION SUBMITTED 2022

AALBORG UNIVERSITY
DENMARK

Aspects of Statistical Analysis of Spatial Point Patterns

Ph.D. Dissertation

Ninna Vihrs

Dissertation submitted June 23, 2022

Dissertation submitted: June 23, 2022

PhD supervisor: Prof. Jesper Møller
Aalborg University

Assistant PhD supervisors: Prof. Rasmus P. Waagepetersen
Aalborg University

Assoc. Prof. Christophe A. N. Biscio
Aalborg University

PhD committee: Associate Professor Ege Rubak (chair)
Aalborg University, Denmark

Associate Professor Ute Hahn
Aarhus University, Denmark

Researcher Rémi Bardenet
Université de Lille, France

PhD Series: Faculty of Engineering and Science, Aalborg University

Department: Department of Mathematical Sciences

ISSN (online): 2446-1636
ISBN (online): 978-87-7573-877-9

Published by:
Aalborg University Press
Kroghstræde 3
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Stibo Complete, 2022

Abstract

This thesis is about statistical analysis of spatial point pattern data sets. A point pattern is a finite subset of some space S; in this thesis, $S \subseteq \mathbb{R}^d$ for some dimension d. In most application examples $d = 2$ or $d = 3$. Examples of point pattern data could be positions of trees in a forest or positions of cells in a human brain. It is often of interest to investigate whether a point pattern exhibits significant deviations from the case where all points are positioned completely at random. Such deviations can occur when the point pattern exhibit clustered and/or regular behaviour. This is usually investigated by means of functional summary statistics. A point pattern is considered to be a realization of a point process, which is a stochastic countable subset of S, and a statistical analysis of a point pattern may include fitting an appropriate point process model to the data. The task of estimating unknown parameters in a point process model is often difficult because the likelihood function is only tractable for very simple models, so likelihood based methods are often impractical. Therefore, there are many different estimation procedures for point process models including maximum pseudo-likelihood estimation and minimum contrast estimation. Which estimation method it is preferred to use depends on the point process model and the available theoretical knowledge for that particular model. This thesis contributes to the field of statistical analysis of spatial point patterns by introducing two new point process models, introducing a new procedure for estimating parameters in point process models, providing a discussion about whether to condition on the observed number of points when analysing point patterns, and providing an interesting application example of point pattern analysis involving the spatial organisation of pyramidal cells in the human cerebral cortex.

The thesis consists of two parts: Part I is an introduction, and Part II consists of six research papers. Part I provides some background information regarding statistical analysis of point patterns which is relevant in order to better understand the papers in Part II and their scientific contribution.

The most common and popular spatial point process models are models for either clustering or regularity. In Papers A and F, we introduce point process models exhibiting both clustering and regularity. In Paper A, we pro-

pose a doubly stochastic spatial point process model called an LGCP-Strauss process which combines a log-Gaussian Cox process with a Strauss process in order to obtain a model with repulsion at small scales and aggregation at a larger scale. Estimating parameters in this model is challenging because of a lack of theoretical knowledge, so we illustrate how to make Bayesian inference by means of the method of approximate Bayesian computations which is a method based entirely on the ability to simulate under the model for given parameters. In Paper F, we propose a cluster point process model with regularity between clusters thus yielding a model with clustering at a small scale and regularity at a larger scale. We call this class of models determinantal shot noise Cox processes since they are the special case of generalized shot noise Cox processes where a determinantal point process is used for the point process of cluster centres. We are able to derive various moment results, which are particularly tractable when either a Gaussian determinantal point process or a scaled Ginibre point process is used for the center process. These results can be used to easily estimate unknown parameters by means of minimum contrast estimation.

Inspired by the challenges we had with estimating parameters in LGCP-Strauss processes in Paper A, I suggest a method for estimating parameters in spatial point process models by means of neural networks in Paper E. The method is generally applicable to all point process models since the only requirement is the ability to simulate under the model. Through a simulation study, I compare the neural network approach to the most commonly used existing methods for some popular examples of point process models and conclude that the suggested neural network approach recovers parameters well.

In most applications of point pattern analysis, there is only one observed point pattern to analyse, that is, only one realization of an assumed generating spatial point process. It is thus impossible to validate any claims about the distribution of the number of points in the point pattern, but the common practice is still to directly or indirectly assume a model for the number of points. In Paper D, we discuss this practice and investigate whether we should instead condition on the number of points when modelling spatial point patterns. Regarding parameter estimation, we conclude that it is inconvenient to condition on the number of points. Concerning model validation with global envelopes, we conclude that when using some functional summary statistics, it will be an advantage to condition on the number of points; however, for the most popular choices of summary statistics for model validation it makes no real difference to condition on the number of points.

In Papers B–C, we investigate the spatial organisation of pyramidal cells in the human cerebral cortex by analysing three dimensional point patterns of cell positions. By means of the functional summary statistic called the cylindrical K-function, we were able to find evidence of a columnar structure

in the data. In Paper C, we also compared point patterns of cells from control subjects, subjects with a depression, subjects with schizophrenia, and subjects who had committed suicide, but the analysis was unable to detect any significant differences in the spatial organisation of pyramidal cells between groups.

Resumé

Denne afhandling omhandler statistisk analyse af rumlige punktmønstre. Et punktmønster er en endelig delmængde af et rum S. I denne afhandling er $S \subseteq \mathbb{R}^d$ for en dimension d. I de mest almindelige eksempler på anvendelser er $d = 2$ eller $d = 3$. Som eksempler på punktmønstre kan nævnes træers placering i en skov eller cellers placering in menneskehjernen. Det er ofte relevant at undersøge, om et punktmønster udviser signifikante afvigelser fra den situation, hvor alle punkter er placeret fuldstændigt tilfældigt. Sådanne afvigelser kan opstå, hvis punktermønstret ideholder tegn på klustering og/eller regularitet. Dette undersøges tit ved at benytte relevante summary statistics. Et punktmønster betragtes som en realisation af en punktprocess, hvilket er en stokastisk tællelig delmængde af S, og en statistisk analyse af et punktmønster kan inkludere, at en passende punktprocessmodel tilpasses data. Opgaven med at estimere ukendte parametre i en punktprocessmodel er ofte vanskelig, fordi likelihoodfunktionen kun er håndterlig for meget simple modeller, hvilket bivirker, at det ofte er upraktisk at benytte likelihoodbaserede estimationsmetoder. Derfor findes der mange forskellige estimationsmetoder for punktprocessmodeller, inklusiv maksimum pseudolikelihood estimation og minimum kontrast estimation. Hvilken estimationsmetode, det er bedst at bruge, afhænger af punktprocessmodellen og den tilgængelige teoretiske viden for denne specifikke model. Denne afhandling bidrager til området inden for statistisk analyse af rumlige punktmønstre ved at introducere to nye punktprocessmodeller; introducere en ny estimationsmetode for punktprocessmodeller; give en diskussion af, hvorvidt man bør betinge med det observerede antal punkter, når man analyserer punktmønstre; og give en interessant anvendelsesmulighed for punktmønsteranalyse, som omhandler organiseringen af pyramideceller i menneskets cerebrale cortex.

Afhandlingen består af to dele: Del I er en introduktion, og Del II består af seks artikler. Del I indeholder en del baggrundsinformation om statistisk analyse af punktmønstre, som er relevant for bedre at kunne forstå artiklerne i Del II og deres videnskabelige bidrag.

De mest almindelige og populære rumlige punkt process modeller er mo-

deller for enten klustering eller regularitet. I Artikel A og F introducerer vi punktprocessmodeller, som både indeholder klustering og regularitet. I Artikel A foreslår vi en dobbeltstokastisk rumlig punktprocessmodel kaldet en LGCP-Straussprocess, som kombinerer en log-Gaussisk Cox process med en Strauss process for at opnå en model med repulsion på lille skala og aggregering på større skala. At estimere parametre i denne model er udfordrenden på grund af manglende teoretisk viden, så vi illustrerer, hvordan man kan lave Bayesiansk inferens ved hjælp af metoden approximate Bayesian computations, som er baseret udelukkende på muligheden for at simulere under modellen for givne parametre. I Artikel F foreslår vi en klusterpunktprocessmodel med regularitet mellem klustrene, hvilket giver en model med klustering på lille skala og regularitet på større skala. Vi kalder denne modelklasse determinante shot noise Cox processer, fordi de er det specialtilfælde af generaliserede shot noise Cox processer, hvor en determinant punkt process benyttes for klustercentrene. Vi er i stand til at udlede forskellige momentegenskaber, som er særligt håndterlige, når der enten benyttes en Gaussisk determinant punktprocess eller en skaleret Ginibre punktprocess som centerprocess. Disse resultater kan bruges til nemt at estimere ukendte parametre med minimum kontrast estimation.

Inspireret af de udfordringer, som vi havde med at estimere parametre i LGCP-Strauss processer i Artikel A, foreslår jeg i Artikel E en metode til at estimere parametre i rumlige punktprocessmodeller, som benytter neurale netværk. Metoden kan generelt bruges for alle punktprocessmodeller, eftersom det eneste krav er, at det er muligt at simulere under modellen. Igennem et simulationsstudie sammenligner jeg den nye metode med de mest almindeligt brugte eksisterende estimationsmetoder for nogle populære eksempler på punktprocessmodeller og konkluderer, at den foreslåede fremgangsmåde med neural netværk giver gode parameterestimater.

I de fleste anvendelser af punktmønsteranalyse er der kun et observeret punktmønster at analysere. Det vil sige, at der kun er en realisation af en antaget frembringende rumlig punktprocess. Det er derfor umuligt at validere påstande omhandlende fordelingen af antal punkter i punktmønstret, men det mest almindelige er alligevel at enten direkte eller indirekte antage en model for antal punkter. I Artikel D diskuterer vi denne praksis og undersøger, om vi i stedet burde betinge med antallet af punkter, når vi modellerer rumlige punktmønstre. Hvad angår parameterestimation konkluderer vi, at det er upraktisk at betinge med antal punkter. Med hensyn til modelvalidering ved brug af globale envelopes konkluderer vi, at når man bruger nogle summary statistics, er det en fordel at betinge med antal punkter, men når man bruger de summary statistics, som er mest populære, når det kommer til modelvalidering, gør det ingen nævneværdig forskel at betinge med antal punkter.

I Artikel B–C undersøger vi den rumlige organisering af pyramideceller

i menneskets cerebrale cortex ved at analysere tre-dimensionelle punktmønstre bestående af cellernes placering. Ved brug af den summary statistic, som kaldes den cylindriske K-funktion, var vi i stand til at finde evidens for en søjlestruktur i data. I Artikel C sammenlignede vi også punktmønstre for celler fra emner med en depression, emner med skizofreni, emner som begik selvmord, og emner fra en kontrolgruppe, men analysen var ikke i stand til at finde nogen signifikante forskelle i den rumlige organisering af pyramideceller mellem grupperne.

Contents

Preface

This thesis contains the scientific research conducted by me and my collaborators during my time as a PhD student at the Department of Mathematical Sciences, Aalborg University. The work is partially funded by The Danish Council for Independent Research | Natural Sciences, grant DFF – 7014-00074 'Statistics for point processes in space and beyond'. The research concerns different aspects in relation to statistical analysis of spatial point patterns, including new spatial point process models, estimation procedures, practical and philosophical considerations, and examples of applications. Part I provides some background on spatial statistics related to the analysis of point patterns which helps to understand the scientific contributions of the thesis and to put it into perspective. Part II is a collection of the six research papers:

A: Vihrs, N, Møller, J, and Gelfand, A. E. *Approximate Bayesian inference for a spatial point process model exhibiting regularity and random aggregation.* Scandinavian Journal of Statistics (2022), 49, 185–210.

B: Larsen, N. Y., Li, X., Tan, X., Ji, G., Lin, J., Rajkowska, G., Møller, J. , Vihrs, N., Sporring, J., Sun, F., and Nyengaard, J. R. *Cellular 3D-reconstruction and analysis in the human cerebral cortex using automatic serial sections.* Communications Biology (2021), 4, article number 1030.

C: Larsen, N. Y., Vihrs, N., Møller, J., Sporring, J, Xueke, T., Xixia, L., Ji, G.; Rajkowska, G., Sun, F., and Nyengaard, J. R. *Layer III pyramidal cells in the prefrontal cortex reveal morphological changes in subjects with depression, schizophrenia, and suicide.* Submitted to Translational Psychiatry.

D: Møller, J. and Vihrs, N. *Should we condition on the number of points when modelling spatial point patterns?* Accepted for publication in International Statistical Review. ArXiv preprint available at `https://arxiv.org/abs/2108.10051`

E: Vihrs, N. *Using neural networks to estimate parameters in spatial point process models.* Spatial Statistics (2022), 51, 100668.

F: Møller, J. and Vihrs, N. *Determinantal shot noise Cox processes.* Submitted to Stat. ArXiv preprint available at `https://arxiv.org/abs/2112.04204`

The papers are included in this thesis in a version very similar to the publiced/submitted version, and therefore there may be some inconsistencies in notations and repetitions of contents across the papers. Each paper is fully self-contained. Papers B–D were submitted to the journals with supplementary material containing additional details; in this thesis, I have included such material in the appendices of the papers.

I wish to thank my supervisor Jesper Møller for his invaluable guidance and support. Thank you for always believing that I would be able to finish this PhD. I furthermore wish to thank Nick Yin Larsen for our many interesting talks about human brain cells during our collaboration on Papers B–C.

Ninna Vihrs
Aalborg University, June 23, 2022

Part I

Background

Introduction

This introduction provides the relevant background for understanding the contribution of this thesis. It consists of a brief overview of some of the basic aspects of statistical analysis of spatial point patterns and explains how Papers A-F contribute to this field. Note however that the papers in Part II are fully self-contained, and that there may be some inconsistencies in notation between this introduction and Papers A–F.

1 Point processes and point patterns

In brief, a (simple) spatial point process X on a space S can be thought of as a stochastic countable subset of S, often $S \subseteq \mathbb{R}^d$. When thinking of a point process in this way, measure theoretical details are avoided. For a more thorough introduction to point processes see e.g. Daley and Vere-Jones (2003), Møller and Waagepetersen (2004), and Diggle (2013). Let $n(x)$ denote the cardinality of a subset $x \subset S$. In this thesis, attention is restricted to the case where X takes values in the set of locally finite point configurations $\mathcal{N}_{\text{lf}} = \{x \subseteq S : n(x \cap B) < \infty \text{ when } B \subseteq S \text{ is bounded}\}$.

A realization x of X is called a point pattern, and it is usually only observed on an observation window $W \subseteq S$, meaning that only a realization of $X \cap W$ is available. The elements of x are referred to as points or events. In most real life applications, only a single realization x is available, which means that it is practically impossible to make any conclusions about for instance the distribution of $n(X)$ even though the models we use to model point patterns (see Section 2) either directly or indirectly assumes such a distribution. In Paper D we discuss the consequences of this practice and conclude that it is however convenient to make such an assumption for mathematical and computational reasons.

It is quite often assumed or showed that a point process is stationary or isotropic. A point process on $\mathbb{R}^d$ is stationary if its distribution is invariant under translations; it is isotropic if its distribution is invariant under rotations.

A common question of interest is whether a point process or point pattern is clustered/agrregated or repulsive/regular, or maybe both. It is quite intuitive what is meant by these terms, but it is difficult to make a precise definition of clustering and repulsiveness, and in some point patterns it is also difficult to spot by a mere visual inspection. The most common way to make statements about clustering and regularity is to consider the theoretical summary functions and functional summary statistics in Section 3. Another way to learn more about a point pattern is to find a point process model which fits the point pattern well. Section 2 describes some popular classes of point process models, Section 4 concerns the estimation of parameters in point process models, and Section 5 describes a method to validate a fitted model.

2 Point process models

2.1 Poisson processes

A Poisson process is maybe the simplest example of a point process model. It is the case where there is no interaction between points, and it is specified in terms of a function $\rho : S \to [0, \infty)$ which satisfies that $\int_B \rho(u)\,\mathrm{d}u < \infty$ when $B \subseteq S$ is bounded, i.e. that ρ is locally integrable. A process X is a Poisson process with intensity ρ if for any $B \subseteq S$ satisfying $\int_B \rho(u)\,\mathrm{d}u < \infty$ the following is satisfied:

1. $n(X \cap B)$ follows a Poisson distribution with rate $\int_B \rho(u)\,\mathrm{d}u$, or $n(X \cap B) = 0$ if $\int_B \rho(u)\,\mathrm{d}u = 0$,
2. conditioned on $n(X \cap B) = n$, the n points in $X \cap B$ are independent and identically distributed with unnormalized density function ρ.

The function ρ is called the intensity function, and note that

$$\mathrm{E}[n(X \cap B)] = \int_B \rho(u)\,\mathrm{d}u \tag{1}$$

In general for any point process X, a locally integrable non-negative function ρ satisfying (1) is called the intensity function (or sometimes just intensity) of X. Point processes with constant intensity are said to be homogeneous; otherwise, they are inhomogeneous. If a Poisson process on $\mathbb{R}^d$ is homogeneous, it is both stationary and isotropic. The Poisson process is also known as the case of complete spatial randomness, and statements about the behaviour of point processes and point patterns are often based on comparisons with a homogeneous Poisson process, for example as described in Section 3. A Poisson process with $\rho = 1$ is called the unit rate or standard Poisson process.

2.2 Cox processes

A Cox process (Cox, 1955) is driven by a stochastic intensity $Z = \{Z(u)\}_{u \in S}$ which is a non-negative random field satisfying $\mathrm{P}\left(\int_B Z(u)\,\mathrm{d}u < \infty\right) = 1$ for all bounded $B \subseteq S$. A process X is a Cox process driven by Z if X conditioned on Z is a Poisson process with intensity Z. Popular examples of Cox processes are log-Gaussian Cox processes where $Z = \exp(Y)$ for a Gaussian random field Y (Møller et al., 1998) and shot noise Cox processes on $\mathbb{R}^d$ where $Z(u) = \sum_{(c,\gamma)\in\Phi} \gamma k(c,u)$ for a kernel k and a Poisson process Φ on $\mathbb{R}^d \times (0,\infty)$ (see e.g. Møller, 2003). Shot noise Cox processes are examples of cluster point processes, and in this case X conditioned on Φ is the superposition $\cup_{(c,\gamma)\in\Phi} X_{(c,\gamma)}$ of independent Poisson point processes $X_{(c,\gamma)}$ with intensity function $\gamma k(c,\cdot)$. The process $X_{(c,\gamma)}$ is called the cluster with center c and intensity γ, and the set $\mathcal{C} = \{c : (c,\gamma) \in \Phi\}$ is called the centre process. Cox processes are usually considered as models for clustering, see Section 3.

2.3 Markov point processes

Markov point processes were introduced in spatial statistics by Ripley and Kelly (1977). I here give a brief overview of Markov point processes (also known as Gibbs point processes) and refer to Møller and Waagepetersen (2004) and the references therein for more details.

Markov point processes on a bounded set S, i.e. $|S| < \infty$, are defined in terms of their density with respect to a unit rate Poisson process. When I henceforth write density in connection with a spatial point process, it is always the density with respect to a unit rate Poisson process on S. Note that since S is bounded, X takes values in the set of finite point configurations $\mathcal{N}_{\mathrm{f}} = \{x \subseteq S \mid n(x) < \infty\}$. The density function f of a point process X is the function which satisfies that

$$P(X \in F) = \mathrm{E}[1[Y \in F] f(Y)], \quad F \subseteq \mathcal{N}_{\mathrm{f}},$$

for a unit rate Poisson process Y. Often, it is only possible to get a closed form expression for a density up to proportionality in which case we may consider an unnormalised density h such that $f(x) = h(x)/c$, $x \in \mathcal{N}_{\mathrm{f}}$, for a normalising constant c which does not depend on x.

An important characteristic which is based on the density function f is the Papangelou conditional intensity

$$\lambda(x,u) = \frac{f(x \cup \{u\})}{f(x)}, \quad x \in \mathcal{N}_{\mathrm{f}},\ u \in S \setminus x.$$

Since any normalising constant will cancel in this expression, the Papangelou conditional intensity can be derived if only the density is known up to proportionality. This can be a great advantage since many densities of point

processes involve an intractable normalising constant. This can for instance be used in connection with parameter estimation as will be described in Section 4.2. A heuristic interpretation of the Papangelou conditional intensity is that $\lambda(x,u)\mathrm{d}u$ is the probability that X has a point in an infinitesimal region B_u around u of size $\mathrm{d}u$ conditional on $X \cap (S \setminus B_u) = x$. The Papangelou conditional intensity can be used to make statements about clustering and repulsiveness in a point process as follows: Let $x, y \in \mathcal{N}_\mathrm{f}$ and $x \subset y$. If $\lambda(x,u) \leq \lambda(y,u)$, X is considered clustered; if $\lambda(x,u) \geq \lambda(y,u)$, X is considered repulsive.

The definition of a Markov point process requires a neighbour relation $\sim$ on S. This is a reflexive and symmetric relation, i.e. for all $u, v \in S$, $u \sim u$ and $u \sim v \Rightarrow v \sim u$. An example of an often used neighbour relation is $u \sim v \Leftrightarrow \|u - v\| \leq R$ for some $R > 0$. This will be referred to as the R-close neighbour relation. If $u \sim v$, the points $u, v \in S$ are said to be neighbours and the neighbourhood of a point $u \in S$ is the set $N_u = \{v \in S : u \sim v\}$ consisting of all neighbours to u. A Markov point process is a point process whose density function f is a Markov function with respect to some neighbour relation $\sim$ on S which means that

- f is hereditary, i.e. $f(x) > 0 \Rightarrow f(y) > 0$ for $y \subset x$, and
- for all $x \in \mathcal{N}_\mathrm{f}$ where $f(x) > 0$ and all $u \in S \setminus x$, $\lambda(x,u)$ only depends on x through $x \cap N_u$.

When f is hereditary, there is a one-to-one correspondence between f and λ.

An example of a Markov point process is a pairwise interaction point process where

$$f(x) \propto \prod_{u \in x} \varphi(u) \prod_{\{u,v\} \subseteq x} \varphi(\{u,v\}) \tag{2}$$

for a nonnegative function φ which makes (2) integrable with respect to a standard Poisson process. It is a Markov point process with respect to the R-close neighbour relation where

$$R = \inf\{r > 0 : \|u - v\| > r \Rightarrow \varphi(\{u,v\}) = 1 \text{ for all } u, v \in S\}. \tag{3}$$

The expression in (3) is known as the range of interaction or the interaction radius. Usually, $\varphi(\{u,v\}) \leq 1$, in which case it can be seen from the Papangelou conditional intensity

$$\lambda(x,u) = \varphi(u) \prod_{v \in x} \varphi(\{u,v\})$$

that pairwise interaction point processes are most often models for repulsion.

The special case of a pairwise interaction point process where $\varphi(u) = \beta$ and $\varphi(\{u,v\}) = \gamma^{1[\|u-v\| \leq R]}$ for parameters $\beta, R > 0$ and $\gamma \in [0,1]$ is known

as a Strauss process (Strauss, 1975, Kelly and Ripley, 1976). The density of a Strauss process is thus

$$f(x) \propto \beta^{n(x)} \gamma^{s_R(x)} \tag{4}$$

where $s_R(x) = \sum_{\{u,v\} \subseteq x} 1[\|u - v\| \leq R]$ is the number of R-close pairs in x. The Strauss process is a well known and much used example of a repulsive point process.

Note that since I have only defined Markov point processes on a bounded set S, they cannot be stationary. It is possible to extend the definition of a Markov point process to $\mathbb{R}^d$, but since this is rather involved and technical, I will not go into details in this introduction and refer instead to Ruelle (1969), Preston (1976), the review in Møller and Waagepetersen (2004) and the references therein for details about infinite Gibbs point processes.

2.4 Determinantal point processes

The general notion of determinantal point processes was introduced in Macchi (1975), and they have been studied from a statistical point of view in Lavancier et al. (2015). Determinantal point processes are defined in terms of their moment properties, specifically by means of the n'th order intensity functions, $n = 1, 2 \ldots$. The n'th order intensity function of a point process X on a Borel set $S \subseteq \mathbb{R}^d$ is a locally integrable function $\rho^{(n)} : S^n \to [0, \infty)$ which satisfies that for any Borel function $h : S^n \to [0, \infty)$,

$$\mathrm{E}\left[\sum_{x_1, \ldots, x_n \in X}^{\neq} h(x_1, \ldots, x_n) \right] = \int_S \cdots \int_S \rho^{(n)}(x_1, \ldots, x_n) h(x_1, \ldots, x_n) \, \mathrm{d}x_1 \cdots \mathrm{d}x_n$$

where the sum is over pairwise distinct $x_1, \ldots, x_n$ (as indicated by the sign $\neq$ over the sum).

Consider now a kernel $K : S \times S \to \mathbb{C}$. Let $[K](x_1, \ldots, x_n)$ denote the matrix with $K(x_i, x_j)$ as entry (i, j). Then X is a determinantal point process with kernel K if $\rho^{(n)}$, $n = 1, 2, \ldots$, is given by

$$\rho^{(n)}(x_1, \ldots, x_n) = \det[K](x_1, \ldots, x_n), \qquad (x_1, \ldots, x_n) \in S^n. \tag{5}$$

For a Borel set $B \subseteq S$ the restriction $X \cap B$ is also a determinantal point process whose kernel is the restriction of K to $B \times B$. The kernel K has to be a non-negative definite function. If it is furthermore continuous and Hermitian, it is known that K restricted to a compact set $B \subset \mathbb{R}^d$ has a spectral representation

$$K(x, y) = \sum_{k=1}^{\infty} \lambda_k^B \phi_k^B(x) \overline{\phi_k^B(y)}, \qquad (x, y) \in B \times B \tag{6}$$

with real eigenvalues λ_k^B and eigenfunctions ϕ_k^B which form an orthonormal basis for the set of square integrable functions. Then the existence of a determinantal point process with kernel K on $\mathbb{R}^d$ is equivalent to $\lambda_k^B \leq 1$ for all compact $B \subset \mathbb{R}^d$ and all $k = 1, 2, \ldots$. If one only needs the determinantal point process on a specific compact set B, it is enough to check the above criterion for existence for this set B.

The spectral representation (6) can be used to simulate X on B and for specifying the density of X defined on B, see Lavancier et al. (2015) for details. The eigenvalues can furthermore be used to make inference about $n(X \cap B)$ since

$$n(X \cap B) \sim \sum_{k=1}^{\infty} B_k \tag{7}$$

where B_k are independent Bernoulli variables with mean λ_k^B.

Lavancier et al. (2015) suggest several determinantal point process models where the kernel K is a real function, including the Gaussian determinantal point process where $K(x, y) = \rho \exp(-\|(x - y)/\alpha\|^2)$ for parameters $\alpha > 0$ and $0 \leq \rho \leq (\sqrt{\pi}\alpha)^{-d}$. For all the suggested models, the spectral representation (6) is unknown, and it is thus necessary to approximate the kernel in order to simulate and evaluate the density, see Lavancier et al. (2015).

A particularly tractable example of a determinantal point process with a complex Kernel is an α-Ginibre point process which has kernel

$$K(z, w) = \rho \exp\left(-\frac{\rho\pi}{\alpha}\left(\frac{|z|^2 + |w|^2}{2} - z\bar{w}\right)\right), \qquad z, w \in \mathbb{C}, \tag{8}$$

where $\rho > 0$, $\alpha \in (0, 1]$, and $|\cdot|$ denotes the modulus of a complex number. This can be used as a point process on the plane since $\mathbb{C} \simeq \mathbb{R}^2$. When $\alpha = 1$ and $\rho = \pi^{-1}$, X is a standard Ginibre process (Ginibre, 1965). Note that $\rho^{(1)}(z) = \rho(z) = K(z, z) = \rho$ so the parameter ρ is the intensity of X. It can be shown that X is both stationary and isotropic, see e.g. Appendix B in Paper F. Eigenfunctions satisfying (6) for an α-Ginibre point process restricted to the disc $b(0, r) \subset \mathbb{C}$ around zero with radius $r \in (0, \infty)$ are

$$\phi_k^{b(0,r)}(z) = \sqrt{\frac{\rho}{\alpha\gamma(k, \rho\pi r^2/\alpha)}} \exp(-\rho\pi|z|^2/(2\alpha)) \left(\sqrt{\frac{\rho\pi}{\alpha}} z\right)^{k-1}$$

where $\gamma(k, x) = \int_0^x t^{k-1} \exp(-t)\, \mathrm{d}t$ is the incomplete lower Gamma function, and the corresponding eigenvalues are

$$\lambda_k^{b(0,r)} = \frac{\alpha\gamma(k, \rho\pi r^2/\alpha)}{(k-1)!}. \tag{9}$$

The spectral representation on $b(0, r)$ is thus known, which is an advantage over the models in Lavancier et al. (2015) since it is thus unnecessary to approximate the kernel when simulating the process or evaluating the likelihood function when the process is considered on a disc.

2.5 Point process models exhibiting both regularity and clustering

The above classes of point process models are typically used to model either repulsion or clustering, but sometimes a model exhibiting both regularity and clustering is needed. In this regard, Lavancier and Møller (2016) suggested a model obtained by a dependent thinning of a repulsive point process, thereby obtaining repulsiveness at small scales and clustering at a larger scale. Their work was partly based on Stoyan (1979). It is also possible to obtain this behaviour in certain Gibbs point processes, see e.g. Baddeley et al. (2013) and Goldstein et al. (2015). However, models with this feature are not as well studied or as plenty as models with only clustering or repulsion. Papers A and F contribute with two new classes of point process models exhibiting both regularity and clustering. In Paper A, we introduce the LGCP-Strauss process model which is a combination of a Strauss process and a log-Gaussian Cox process. This model exhibits regularity at small scales and random aggregation at a larger scale. In Paper F, we introduce determinantal shot noise Cox processes, which are like shot noise Cox processes except that a determinantal point process is used for the centre process instead of a Poisson process. Determinantal shot noise Cox processes are thus cluster point processes with repulsion between clusters. Thereby we obtain clustering at small scales and some regularity at a larger scale, which yields realisations with more clearly separated clusters than in shot noise Cox processes.

3 Summary functions and functional summary statistics

Summary functions are often used to make statements about the features of a point process model, and their empirical estimates, referred to as functional summary statistics, are widely used when analysing point patterns. Summary statistics are for example used to make statements about clustering/regularity, to suggest appropriate models, and to estimate and validate models. Summary functions and/or functional summary statistics play an important role in all the papers in Part II. In this section, I define the summary functions which we use, discuss how they can be used to make statements about clustering/regularity, and explain how to estimate these empirically and non-parametrically. For a more thorough introduction to summary functions and functional summary statistics see e.g. Møller and Waagepetersen (2004), Diggle (2013), and Baddeley et al. (2015).

3.1 Theoretical summary functions

Since I have already defined the n'th order intensity function in Section 2.4, which includes the intensity $\rho = \rho^{(1)}$, it is straight forward to define the pair correlation function

$$g(u, v) = \frac{\rho^{(2)}(u, v)}{\rho(u)\rho(v)}$$

when ρ and $\rho^{(2)}$ exist and using the convention that $g(u, v) = 0$ if $\rho(u)\rho(v) = 0$. The intuitive interpretation of $\rho^{(2)}$ is that $\rho^{(2)}(u, v)\mathrm{d}u\mathrm{d}v$ is the probability of simultaneously observing a point from the process X in both of the balls centred at u and v of sizes $\mathrm{d}u$ and $\mathrm{d}v$, respectively. It is known that for a Poisson process $\rho^{(2)}(u, v) = \rho(u)\rho(v)$, so the pair correlation function is in this case $g(u, v) = 1$. Many pair correlation functions of interest only depend on u, v through their euclidean distance $\|u - v\|$, in which case the notation $g(u, v) = g(\|u - v\|) = g(r)$ is used. Then $g(r) > 1$ indicates that it is more likely to simultaneously observe a point near both u and v, for $\|u - v\| = r$, than it is in a Poisson process, which is therefore interpreted as an indication of clustering at interpoint distance r. If $g(r) < 1$, it is similarly interpreted as an indication of regularity at interpoint distance r.

If the pair correlation function exists and is invariant under translations, in which case the notation $g(u, v) = g(u - v) = g(u)$ is used, the second order reduced moment measure is

$$\mathcal{K}(B) = \int_B g(u)\,\mathrm{d}u \tag{10}$$

for a bounded Borel set $B \subset \mathbb{R}^d$. When letting B be a ball, we obtain Ripley's K-function $K(r) = \mathcal{K}(b(0, r))$. If the process X is stationary with intensity ρ, $\rho K(r)$ is the expected number of further points from X falling in $b(0, r)$ given that X has a point at the origin. Because of the stationarity, 0 can be thought of as a typical point of the process. In the case of a Poisson process, it is known that $K(r) = \omega_d r^d$ where ω_d is the volume of the d-dimensional unit ball, so the transformation $L(r) = (K(r)/\omega_d)^{1/d}$ is often considered. This transformation is a well-known summary function in itself referred to as the L-function. If $L(r) > r$, it suggests that we expect to see more points in $b(0, r)$ when it is given that $0 \in X$ than in a Poisson process, and it thus indicates clustering in the process at interpoint distance r. Similarly, $L(r) < r$ indicates regularity in the process at interpoint distance r. If one instead uses a cylinder $c_u(r, t)$ with midpoint 0, direction u, radius r, and height $2t$ as B in (10), the cylindrical K-function $K_u(r, t) = \mathcal{K}(c_u(r, t))$ is obtained (Møller et al., 2016). Its interpretation follows that of Ripley's K-function just with a cylinder instead of a ball. The cylindrical K-function is the most important functional summary statistic which we use to analyse the spatial structure of pyramidal brain cells in the human cerebral cortex in Papers B and C since

the cylindrical K-function has proven to be particularly useful for detecting a columnar structure in point pattern data and such brain cells are widely believed to be organized in columns.

Other summary functions are based on interpoint distances. There are three such summary functions which are common in use. The first one is the empty space function, also known as the F-function, and it is defined by

$$F(r) = P(X \cap b(0,r) \neq \emptyset), \quad r > 0.$$

For a stationary Poisson process with intensity ρ it is known that $F(r) = 1 - \exp(-\rho|b(0,r)|)$. If $F(r) < 1 - \exp(-\rho|b(0,r)|)$, it indicates more empty space than in a Poisson process, which usually suggest clustered behaviour. Similarly if $F(r) > 1 - \exp(-\rho|b(0,r)|)$, it indicates less empty space than in a Poisson process, which usually suggest regularity.

The second one is the nearest neighbour function, also known as the G-function, and it is given by

$$G(r) = \frac{1}{\rho|A|}\mathrm{E}\left[\sum_{u \in X \cap A} 1[(X \setminus \{u\}) \cap b(u,r) \neq \emptyset]\right], \quad r > 0, \tag{11}$$

which does not depend on the choice of $A \subset \mathbb{R}^d$ with $0 < |A| < \infty$. The G-function can also be expressed in terms of the so-called reduced Palm distribution at 0 denoted by $P_0^!$. To give a precise meaning to $P_0^!$ is rather technical, and I refer the reader to Appendix C in Møller and Waagepetersen (2004). However, $P_0^!$ can be interpreted as the conditional distribution of $X \setminus \{0\}$ given that $0 \in X$. It is known that

$$G(r) = P_0^!(X \setminus \{0\} \cap b(0,r) \neq \emptyset) = P(X \setminus \{0\} \cap b(0,r) \neq \emptyset \mid 0 \in X).$$

Because of stationarity, 0 can be thought of as a typical point of the process, and $G(r)$ is thus interpreted as the probability that there will be another point within distance r of a typical point. For a stationary Poisson process $G(r) = F(r) = 1 - \exp(-\rho|b(0,r)|)$. If $G(r) > 1 - \exp(-\rho|b(0,r)|)$, it means that it is more likely to see another point within distance r of a typical point than in a Poisson process, and it thus suggest clustering. Similarly, if $G(r) < 1 - \exp(-\rho|b(0,r)|)$, it indicates regularity.

The last summary function I will mention is the J-function

$$J(r) = \frac{1 - G(r)}{1 - F(r)},$$

which is only defined when $F(r) < 1$. For a stationary Poisson process, $J(r) = 1$. If $J(r) > 1$, it means that it is more likely to find no further points in $b(0,r)$ given that $0 \in X$ than it is to find no points in $b(0,r)$ which suggests

repulsive behaviour between points. Similarly, $J(r) < 1$ indicates aggregation of points.

As explained, it is possible to derive the theoretical summary functions mentioned in this section for a Poisson process. It is also possible to derive closed expressions of some of these theoretical summary functions for some other point processes. Consider for example an α-Ginibre point process X, which was defined in Section 2.4. It follows from (5) and (8) that the second order intensity function is

$$\rho^{(2)}(z,w) = \rho^2 \left(1 - \exp(-\rho\pi|z-w|^2/\alpha)\right).$$

Thus the pair correlation function is

$$g(z,w) = 1 - \exp(-\pi\rho|z-w|^2/\alpha). \tag{12}$$

Note that g only depends on z, w through the distance $r = |z-w|$ so it will be denoted $g_0(r)$. Since the pair-correlation function is isotropic, Ripley's K-function is

$$K(r) = 2\pi \int_0^r t g_0(t)\,\mathrm{d}t = \pi r^2 - (\alpha/\rho)(1 - \exp(-\pi\rho r^2/\alpha)).$$

To derive F, it can be used that $X \cap b(0,r)$ is a determinantal point process whose kernel has a known spectral representation. It thus follows from (7) and (9) that

$$F(r) = 1 - P(n(X \cap b(0,r)) = 0) = 1 - \prod_{k=1}^{\infty}(1 - \lambda_k^{b(0,r)}). \tag{13}$$

To derive G, it ca be used that the reduced Palm distribution of a determinantal point process X at the origin is known to be a determinantal point process with kernel

$$K^0(x,y) = \frac{K(x,y)K(0,0) - K(x,0)K(0,y)}{K(0,0)}$$

which in the case of an α-Ginibre point process gives

$$K^0(z,w) = (\exp(\rho\pi z\bar{w}/\alpha) - 1)\rho \exp(-\pi\rho(|z|^2 + |w|^2)/(2\alpha)).$$

In this case, the kernel K^0 has the same eigenfunctions as K on the disk $b(0,r)$ (I show this in Appendix A), and the corresponding eigenvalues

$$\lambda_k^{0,b(0,r)} = \begin{cases} 0, & \text{for } k = 1 \\ \lambda_k^{b(0,r)}, & \text{for } k > 1 \end{cases}, \tag{14}$$

so it follows from (7) that the G-function of X is

$$\begin{aligned} G(r) &= 1 - P_0^!(n((X \setminus \{0\}) \cap b(0,r)) = 0) \\ &= 1 - \prod_{k=1}^{\infty}(1 - \lambda_k^{0,b(0,r)}) \\ &= 1 - \prod_{k=2}^{\infty}(1 - \lambda_k^{b(0,r)}). \end{aligned} \tag{15}$$

By combining (13) and (15), the J-function is

$$J(r) = \left(1 - \lambda_1^{b(0,r)}\right)^{-1} = (1 - \alpha(1 - \exp(-\rho\pi r^2/\alpha)))^{-1}.$$

It is usually also possible to derive expressions of the pair correlation function (and thereby Ripley's K-function in the stationary case) for Cox processes by utilizing that $\rho(u) = \mathrm{E}[Z(u)]$, and $\rho^{(2)}(u,v) = \mathrm{E}[Z(u)Z(v)]$ for $u, v \in S$. For instance for a shot noise Cox process,

$$g(u,v) = 1 + \frac{\int_0^{\infty}\int_{\mathbb{R}^d} \gamma^2 k(c,u)k(c,v)\rho_{\Phi}(c,\gamma)\,\mathrm{d}c\mathrm{d}\gamma}{\rho(u)\rho(v)} \tag{16}$$

where ρ_{Φ} is the intensity function of the Poisson process Φ. For a log-Gaussian Cox process where the covariance function of the Gaussian random field is c,

$$g(u,v) = \exp(c(u,v)). \tag{17}$$

3.2 Non-parametric estimates

Non-parametric estimates of the summary functions in Section 3.1 are often used to make statements about clustering and regularity from a point pattern by following the interpretations of the theoretical summary functions. These non-parametric estimates are referred to as functional summary statistics. In this section, I give some of these non-parametric estimates in the case were the intensity ρ is assumed to be constant and a point pattern $x = \{x_1, \ldots, x_n\}$ on an observation window W is given. They all involve some kind of edge correction to account for the unobserved points in $S \setminus W$, but since Baddeley et al. (2015) noted that it is not so important which edge correction method to use as long as some correction is used, I will not go into to much detail about this.

A non-parametric estimate of the pair-correlation function is

$$\hat{g}(r) = \frac{|W|}{\sigma_d r^{d-1} n(n-1)} \sum_{i=1}^{n}\sum_{j\neq i, j=1}^{n} k_b(r - \|x_i - x_j\|)e_{ij}(r)$$

where σ_d is the surface area of the d-dimensional unit ball, k_b is a smoothing kernel with bandwidth b, and $e_{ij}(r)$ is an edge correction weight. This estimate is sensitive to the choice of bandwidth, and it is thus sometimes preferred to use other functional summary statistics when analysing point patterns.

A non-parametric estimate of Ripley's K-function is

$$\widehat{K}(r) = \frac{|W|}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} 1[\|x_i - x_j\| \leq r] e_{ij}(r)$$

where $e_{ij}(r)$ is again an edge correction weight. After estimating $K(r)$ it is straight forward to estimate $L(r)$ by following its definition.

One way to do edge correction when estimating the F- and G-functions is to use border correction by considering the eroded window $W_{\ominus r} = \{u \in W : b(u,r) \subseteq W\}$. The border corrected estimate of F based on some sample locations $\{u_1, \ldots, u_k\} \subset W_{\ominus r}$, usually these are points on a fine grid, is

$$\widehat{F}(r) = \frac{1}{k} \sum_{i=1}^{k} 1[x \cap b(u_i, r) \neq \emptyset].$$

If the points from x falling in $W_{\ominus r}$ are $\{x_1, \ldots, x_m\}$, then the border corrected estimate of G is

$$\widehat{G}(r) = \frac{1}{m} \sum_{i=1}^{m} 1[(x \setminus \{x_i\}) \cap b(x_i, r) \neq \emptyset].$$

It is straight forward to use these estimates to obtain an estimate of J based on its definition.

3.3 Discussion of summary functions in relation to clustering and regularity

As mentioned in Section 1 there is no clear definition of what is meant by the terms clustering and regularity in connection with spatial point processes and point patterns. If an expression for the pair correlation function can be found for a point process model, it is a very popular way to make claims about clustering and regularity in the process. For instance we see from (16) that the pair correlation function of a shot noise Cox process is greater than or equal to 1, which shows that when following this convention shot noise Cox processes are examples of point process models for clustering. In a log-Gaussian Cox process the covariance function c is often a non-negative function, and it then follows from (17) that $g(u,v) \geq 1$, so we again obtain a model for clustering. In the case of a determinantal point process, it follows from (5) that $g(u,v) = 1 - \frac{K(u,v)K(v,u)}{K(u,u)K(v,v)}$ when $K(u,u)K(v,v) > 0$. If K is Hermitian, i.e. $K(u,v) =$

$\overline{K(v,u)}$, which is often the case, $g(u,v) \leq 1$, so the process is considered to be regular.

However, as explained in Section 3.1, other summary functions than the pair correlation function are commonly used to make statements about clustering and regularity, and such statements may not agree entirely. For instance Figure 1 shows the pair correlation function and L-function for an example of a determinantal shot noise Cox process (the class of models introduced in Paper F). The pair correlation function suggests clustering at small

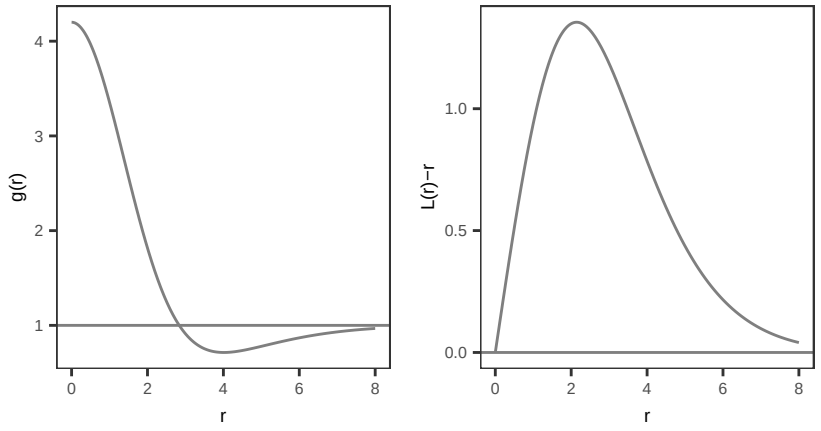

Fig. 1: Plots of pair correlation function (left) and $L(r) - r$ (right) for an example of a determinantal shot noise Cox process.

scales, but it is also below 1 for a range of larger r-values, which suggests regularity at a larger scale; however, $L(r) - r$ never gets below 0 for the considered range of r-values, so the conclusion made from the L-function would thus be that the process is only clustered since $L(r) - r > 0$.

The use of other functional summary statistics than the estimated pair correlation function is especially common when making statements about clustering and regularity in point patterns because the estimate of the pair correlation function is sensitive to the choice of bandwidth. Of course one is not limited to consider only one summary function and functional summary statistic when analysing point processes and point patterns, and different functional summary statistics are often considered in order to summarise different aspects.

It is also worth mentioning that statements about clustering made from theoretical summary functions may be unintuitive as the following example with a pair correlation function shows. In the case of an α-Ginibre point process X, it follows from (12) that $g_0(r) < 1$, which is interpreted as that X is

regular, which is in agreement with intuition when investigating realisations of this process. Impose now a distribution on the parameters (α, ρ) so that X conditioned on (α, ρ) is still an α-Ginibre point process. Specifically, assume that α and $\beta = \rho/\alpha$ are independent, with $\beta \sim \Gamma(\beta_1, \beta_2)$ following a gamma distribution with shape parameter $\beta_1 > 0$ and rate (inverse scale) parameter $\beta_2 > 0$, and α following some distribution on $[0, 1]$. The intensity of X becomes $\rho_X = \mathrm{E}[\rho] = \mathrm{E}[\alpha]\mathrm{E}[\beta] = \mathrm{E}[\alpha](\beta_1/\beta_2)$ and X has an isotropic pcf given by

$$g_0(r) = \mathrm{E}\left[\alpha^2\right] \mathrm{E}\left[\beta^2 \left(1 - \exp\left(-\pi\beta r^2\right)\right)\right] / \rho_X^2, \qquad r \geq 0,$$

where

$$\mathrm{E}\left[\beta^2 \left(1 - \exp\left(-\pi\beta r^2\right)\right)\right] = \frac{\beta_1(1+\beta_1)}{\beta_2^2} - \frac{\beta_1(1+\beta_1)\beta_2^{\beta_1}}{(\pi r^2 + \beta_2)^{\beta_1+2}}.$$

If

$$c = \frac{\mathrm{E}[\alpha^2]}{\mathrm{E}[\alpha]^2} \frac{1+\beta_1}{\beta_1},$$

then

$$g_0(r) = c\left[1 - \left(\frac{\beta_2}{\pi r^2 + \beta_2}\right)^{\beta_1+2}\right].$$

(The above doubly stochastic model and derivation of its theoretical summary functions are based on discussions with Jesper Møller.) Since g_0 is a smooth strictly increasing function with range $[0, c)$ where $c > 1$, there is regularity at a small scale and clustering at a larger scale if one follows the usual interpretation of the pair correlation function. Notice however that any single realization of the suggested process will be indistinguishable from a realization of an α-Ginibre point process. The α-Ginibre point process is a regular process; therefore, any realisation will look regular, and the apparent clustering at larger scale will not be visible. This means that it is questionable whether it is meaningful to deduce such a behaviour from the pair correlation function in this case. However, this interpretation is based on comparing the pair correlation function of X with the pair correlation function of a Poisson process. Considering how X is constructed, it may be more appropriate to compare X with a process Y defined in the following way: Let $\rho = \alpha\beta$, where α and β are distributed as above, and Y conditioned on ρ be a Poisson process with intensity ρ. Then the intensity, second order intensity, and pair

correlation function of Y are

$$\rho_Y = \mathrm{E}[\alpha\beta] = \mathrm{E}[\alpha]\frac{\beta_1}{\beta_2},$$
$$\rho_Y^{(2)} = \mathrm{E}[\alpha^2\beta^2] = \mathrm{E}[\alpha^2](\mathrm{Var}(\beta) + \mathrm{E}[\beta]^2) = \mathrm{E}[\alpha^2]\frac{\beta_1(1+\beta_1)}{\beta_2^2},$$
$$g_Y(r) = \frac{\mathrm{E}[\alpha^2]}{\mathrm{E}[\alpha]^2}\frac{1+\beta_1}{\beta_1} = c,$$

respectively. This shows that $g_0(r) < g_Y(r)$. Based on this comparison, it may be more correct to say that X is regular. Note that following the usual interpretation one would actually claim that Y, which is an example of what is known as a mixed Poisson process, is clustered since $g_Y(r) > 1$ even though any realization is a realization of a stationary Poisson process where points are completely random. These examples show that clustered and regular behaviour deduced from summary functions of a process may not be visible in a single realization from the process.

The above remarks are just made to point out that some care should be taken when interpreting summary functions in relation to clustering and regularity, and that one should not forget to consider the intuition behind such statements. When making statements about clustering and regularity in point processes and point patterns, a sensible approach must be chosen in the specific situation. For instance in Paper F the pair correlation function is expressible on closed form for determinantal shot noise Cox processes and can thus be used to describe the behaviour of such processes, whereas in Paper A we cannot derive any expressions for any summary functions for LGCP-Strauss processes and thus chose to explore the behaviour of the process by calculating functional summary statistics for some simulations instead. Since the estimate of the pair correlation function is sensitive to the choice of bandwidth, we decided to make this exploration of LGCP-Strauss processes based on the L-function instead of the pair correlation function.

4 Estimation

4.1 Maximum likelihood estimation

Recall from Section 2.3 that when I write density of a point process in this introduction, it always means the density with respect to the unit rate Poisson process. If the density of a point process is known, it can be used to estimate unknown parameters by means of maximum likelihood estimation. Unfortunately, densities of point processes are often intractable. For example are the densities of Markov point processes usually only known up to proportionality since they involve an intractable normalising constant which

makes likelihood based inference difficult. Maximum likelihood estimation can be done by using Markov chain Monte Carlo methods, but easier alternatives are often preferred. For more information about maximum likelihood estimation for spatial point process models see e.g. Geyer (1999) and the references therein.

One case for which likelihood based inference is easy is in the case of a Poisson process since a Poisson process on a bounded set S with intensity function ρ has density

$$f(x) = \exp\left(|S| - \int_S \rho(u)\,\mathrm{d}u\right) \prod_{u \in x} \rho(u).$$

If $\rho = \rho_\theta$ depends on an unknown parameter vector θ and a point pattern x is observed, the log-likelihood function is thus

$$l(\theta) = |S| - \int_S \rho_\theta(u)\,\mathrm{d}u + \sum_{u \in x} \log(\rho_\theta(u)), \tag{18}$$

and it can be maximised in order to find the maximum likelihood estimate of θ. It may be necessary to approximate the integral in (18) by numerical methods, but if ρ is a constant, it is easily seen that the maximum likelihood estimate is $\hat{\rho} = n(x)/|W|$ for a point pattern x observed in a window W. This estimate of the intensity is clearly unbiased, and it is also often used in situations where it cannot be shown to be the maximum likelihood estimate. I will sometimes refer to this estimate as the natural estimate of the intensity.

Another example where the likelihood function is known is a determinantal point process X on a compact set S with known spectral representation (6). Let

$$\tilde{K}(x,y) = \sum_{i=1}^{\infty} \frac{\lambda_i^S}{1-\lambda_i^S} \phi_i^S(x)\overline{\phi_i^S(y)}, \qquad x, y \in S \times S. \tag{19}$$

Then the density of X is

$$f(x) = \exp(|S|)\det[\tilde{K}](\{x_1, \ldots, x_n\}) \prod_{i=1}^{\infty}(1-\lambda_i^S). \tag{20}$$

When evaluating this likelihood in practice, it may be necessary to truncate the sum in (19) and the infinite product in (20). Furthermore, the determinant in the likelihood means that it can be very time consuming to evaluate this expression, and one might thus prefer to use the faster alternative of minimum contrast estimation, as will be described in Section 4.3. If a spectral representation for the kernel is not known, the kernel has to be approximated resulting in an approximate likelihood function, see Lavancier et al. (2015).

4.2 Maximum pseudo-likelihood estimation

If the density is intractable but it is possible to derive a closed form expression for the Papangelou conditional intensity λ, maximum pseudo-likelihood estimation (Besag, 1975, Ripley, 1988, Jensen and Møller, 1991) is a fast and reliable alternative to maximum likelihood estimation. The pseudo-likelihood function for an observed point pattern x is

$$PL_A(\theta) = \exp\left(-\int_A \lambda_\theta(x,u)\,\mathrm{d}u\right) \prod_{u\in x\cap A} \lambda_\theta(x\setminus\{u\},u)$$

for some set $A \subset S$ chosen in order to account for edge effects. The pseuod-likelihood function can then be maximised in order to find the maximum pseuod-likelihood estimate of θ. Usually this involves some numerical methods. The method of maximum pseudo-likelihood estimation is the most common way to make inference for Markov point processes since their density is usually known up to proportionality but involves an intractable normalising constant which cancels in the Papangelou conditional intensity.

4.3 Minimum contrast estimation

Minimum contrast estimation (Diggle, 1983, Diggle and Gratton, 1984) do not require any knowledge of the density. Instead it requires that it is possible to derive an expression for a theoretical summary function which depends on the parameters θ which are to be estimated. Also, a non-parametric procedure for estimating the chosen summary function is needed. If T_θ is such a summary function, which could for instance be any of the functions in Section 3.1, and $\widehat{T}$ a non-parametric estimate, the minimum contrast estimate of θ is

$$\hat{\theta} = \arg\min_\theta \left\{\int_{a_1}^{a_2} |T_\theta(r)^q - \widehat{T}(r)^q|^p\,\mathrm{d}r\right\}$$

for some user specified $0 \le a_1 < a_2$ and exponents p and q. Any summary function can be used, but the choice will usually affect the behaviour of the estimate as illustrated in the example below. Thus, there might be reasons to prefer a specific summary function in some cases.

As explained in Section 3.1, it is often possible to derive parametric expressions of summary functions in the cases of Cox and determinantal point processes. Therefore, minimum contrast estimation is the most common estimation procedure in these cases because it is much faster and easier than to do maximum likelihood estimation. For example in the case of an α-Ginibre process we have closed form expressions for both the pair correlation, K-, and J- functions, as shown in Section 3.1. These expressions depend on both α and ρ and can thus all be used to estimate both unknown parameters with minimum contrast estimation. The following simulation study compares the

estimates obtained with these different summary functions and also shows a comparison to maximum likelihood estimation, which is feasible for an α-Ginibre point process on a disc but time consuming. The summary statistics were calculated and the minimum contrast estimates found using the methods implemented in `spatstat`(Baddeley et al., 2015) with the default settings (except in estimating the pair correlation function where the argument `divisor = "d"` was used since it seemed to improve the results).

I considered simulations on the unit disc $b(0,1)$ (in order to do maximum likelihood estimation it is necessary that the observation window is a disc) of α-Ginibre point processes with parameters $\alpha = 0.1, 0.5, 0.9$ and $\rho = 100/\pi, 200/\pi, 300/\pi$. I made 500 simulations for each combination of parameters. Figure 2 shows a boxplot of the errors in the estimates of ρ obtained with the different estimation methods including the natural estimate $\hat{\rho} = n(x)/|b(0,1)|$. Overall, the natural estimate and the maximum likeli-

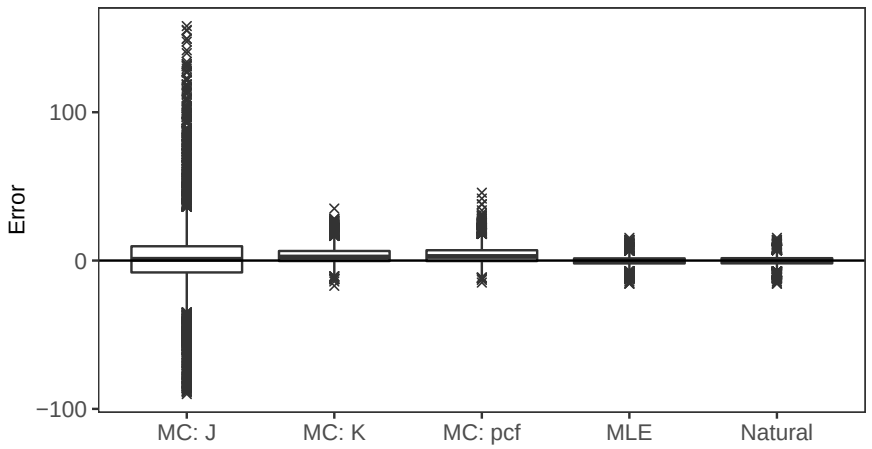

Fig. 2: Boxplot of the error (estimate minus true value) in ρ using the estimation methods maximum likelihood estimation (MLE), the natural estimate, and minimum contrast estimation based on K (MC:K), J (MC:J), and the pair correlation function (MC:pcf).

hood estimate shows a similar behaviour, and they are to be preferred above the estimates obtained with minimum contrast estimation since these exhibit more variation in the error. The maximum likelihood estimate of ρ cannot be found analytically for an α-Ginibre process, and an interesting question is, whether the maximum likelihood estimate is or is close to the natural estimate in this case. Figure 3 shows a comparison of the natural estimates of ρ and the maximum likelihood estimates. Overall, these estimates are similar and they approaches each other as α gets smaller which is in agreement with the fact that the α-Ginibre process approaches a Poisson process. Since the

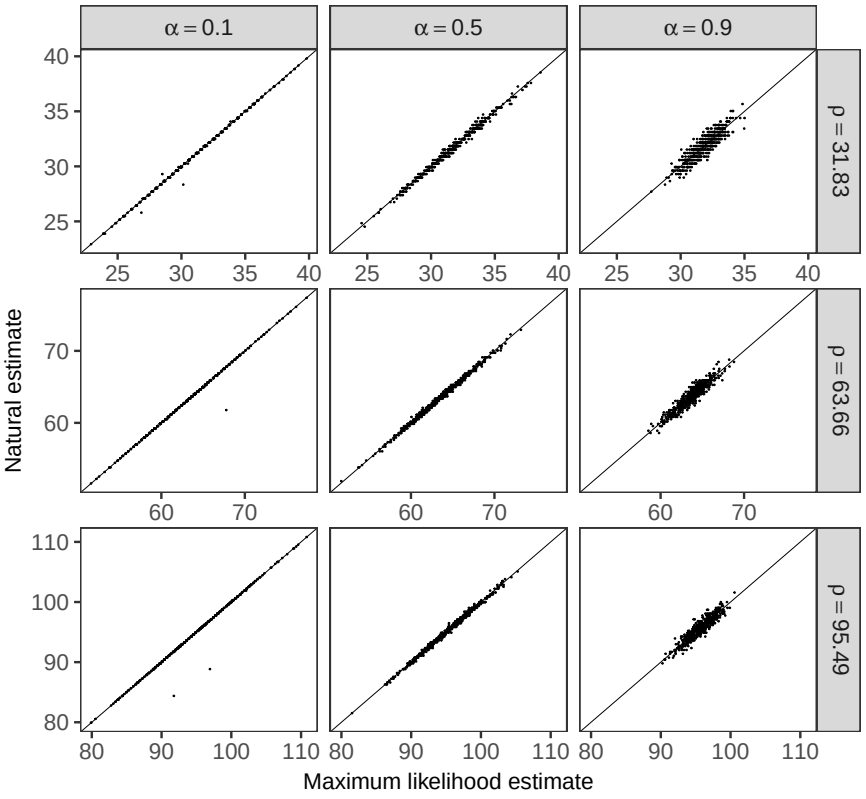

Fig. 3: The natural estimate of ρ plotted against the maximum likelihood estimate where the true values of α and ρ are stated at the top of each plot. The solid line is the identity line.

natural estimate is easy and fast to calculate and shows good performance, I prefer to use this. Then, estimating α with maximimum likelihood or minimum contrast estimation means solving an optimization problem with respect to just one variable, which is also easier than solving the optimization problems for both α and ρ. Figure 4 shows boxplots of the obtained estimates of α using different techniques where I have found the optimal value of $\alpha \in (0, 1]$ (for numerical reasons I more precisely considered $\alpha \in [0.01, 1]$ and $\alpha \in [0.01, 0.999]$ in the optimization problems for minimum contrast and maximum likelihood estimation, respectively). Maximum likelihood estimation provides the most accurate estimates, but the method is slow and only works when the observation window is a disc. Using the K-function for minimum contrast estimation shows a tendency for overestimating α whereas using the pair correlation function gives more accurate estimates in some of the cases. However, the best overall strategy for the α-Ginibre point process is

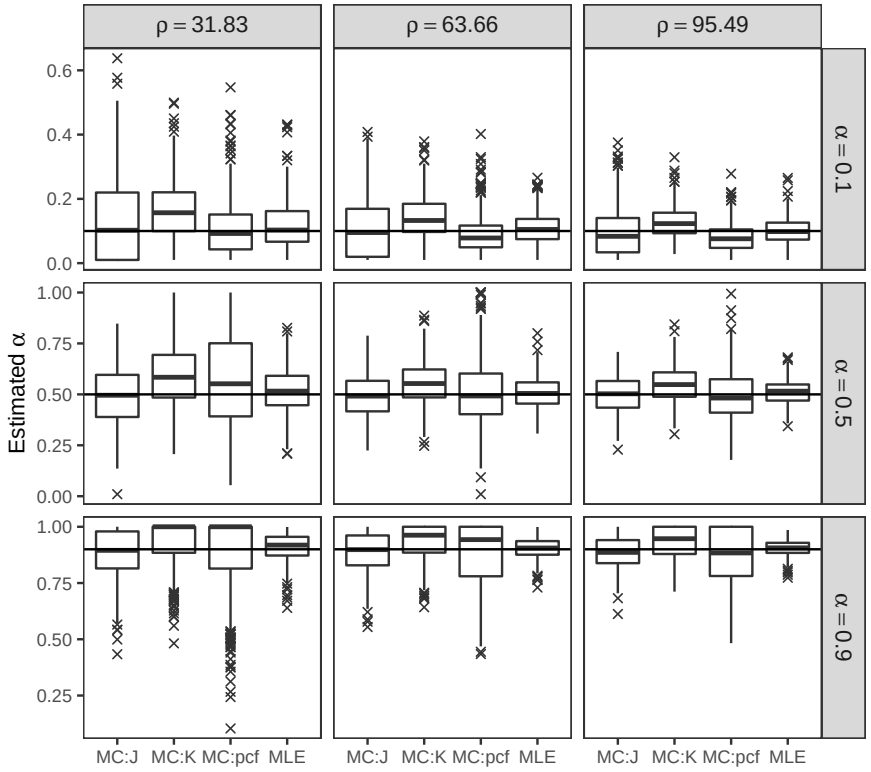

Fig. 4: Boxplots of the estimated values of α using the estimation methods maximum likelihood estimation (MLE) and minimum contrast estimation based on K (MC:K), J (MC:J), and the pair correlation function (MC:pcf). The true value of ρ is stated at the top of each column, and the true value of α is stated to the right of each row. The horizontal lines indicate the true value of α.

to use the J-function since the medians of the obtained estimates are in good agreement with the true values in all cases.

4.4 Purely simulation based methods

If the density, the Papangelou conditional intensity, and theoretical summary functions are all intractable, there is still the possibility to use estimation methods whose only requirement is the ability to simulate under the model. Since it is usually possible to simulate under point process models, such estimation methods are widely applicable.

In a Bayesian setup, there is the method of approximate Bayesian computations (ABC), which can be used for any parametric model (not just point

process models) which it is possible to simulate under. The simplest ABC method is the following rejection sampling procedure for a parametric model with parameter θ and given observed data x_{obs}:

1. Specify some summary statistic S (not necessarily related to the idea of functional summary statistics in Section 3; here, S is just some chosen way to summarize aspects of data) and a distance function χ.

2. Repeat sampling θ' from its prior distribution and simulating x' under the specified model with parameter θ', until $\chi(S(x'), S(x_{\text{obs}})) < \varepsilon$ where ε is a chosen tolerance.

3. Return θ' as an approximate sample from the posterior distribution.

The above procedure can be repeated a number of times in order to obtain a desired number of approximate samples from the posterior distribution. Such a sample can then be used to make statements about the posterior distribution. If the inequality in item 2. is replaced by the statement $S(x') = S(x_{\text{obs}})$ and S is either the identity function or a sufficient statistic, the algorithm returns an exact sample from the posterior distribution, but usually this is not a feasible approach, and the result will instead be a sample from an approximate posterior distribution, referred to as the ABC posterior. How well the ABC posterior resembles the true posterior will depend on the choice of ε and S. There exist many different ABC techniques, see e.g. Beaumont (2010) for an overview of some, but they all have in common that the only requirement for the model is the ability to simulate under it. In Paper A, we used ABC to make inference for our new class of point process models called LGCP-Strauss processes because it is an example of a process for which the other estimation procedures which I have described are infeasible.

In Paper E, I contribute with another method for parameter estimation based entirely on the ability to simulate. I consider the task of estimation as a prediction problem where the true parameter must be predicted from a realization of the process. This prediction problem can then be solved with machine learning techniques where I illustrate how to do it for point process models by using neural networks. In order to train the neural network, training data is needed, and it is for this task that the ability to simulate is used since this makes it possible to construct a large number of realizations where the true parameters are known.

5 Global envelopes

Presently, the most popular way to validate spatial point process models is to use global envelopes and corresponding tests. Global envelopes are calculated from a number of simulations under the model which is to be tested. It

is usually based on a functional summary statistic which is calculated from the observed data and then compared to the same functional summary statistic calculated for the simulations. Briefly, a $(100 - \alpha)\%$ global envelope is a region in which the observed curve falls completely if and only if the global envelope test cannot be rejected at level $\alpha\%$. For details about how to calculate a global envelope I refer to Myllymäki et al. (2017), Myllymäki and Mrkvička (2019), and Mrkvička et al. (2020). I show an example of a global envelope in Figure 5 where the first plot shows a point pattern consisting of the positions of 448 white oak trees which is part of the `lansing` data set available in `spatstat`, and the second plot shows a 95% global envelope and the p-value of the corresponding test based on the extreme rank length, the L-function, and 2499 simulations under a homogeneous Poisson process. It is clear from the p-value, that the hypothesis of a homogeneous Poisson

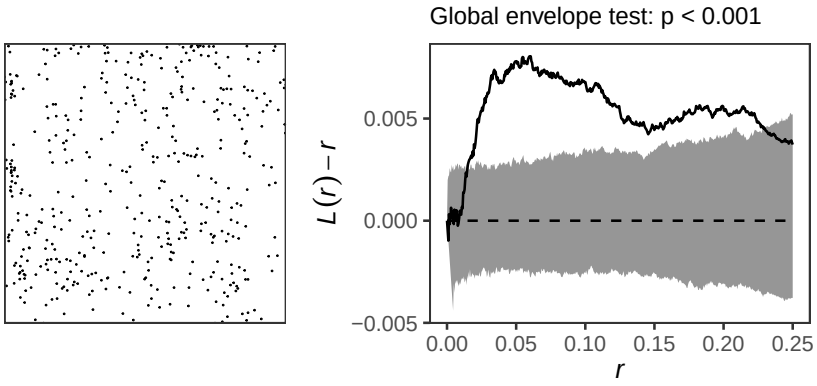

Fig. 5: Left: The positions of 448 white oak trees in a square region (scaled to a unit square) of Lansing Woods, Clinton County, Michigan USA. Right: 95% global envelope for testing the hypothesis of a homogeneous Poisson process. The envelope is the gray area, the solid curve is $\hat{L}(r) - r$ for the observed data, the dashed curve is the mean of $\hat{L}(r) - r$ calculated from 2499 simulations, and the p-value of the global envelope test is indicated at the top.

process is rejected. The real advantage of the envelope is that it provides an explanation for, why the test is rejected. It is seen that the observed values of $\hat{L}(r) - r$ are significantly higher for a large range of r-values than is to be expected under a homogeneous Poisson process. Following the interpretation in Section 3, it thus follows that the point pattern of white oak trees show significant signs of clustering. Such a statement can be used to suggest an appropriate model for this data, and in Paper F we find several cluster point process models which fit this data well, including the examples of determinantal shot noise Cox processes which are presented in this paper.

As exemplified above, global envelopes comparing a point pattern to a

homogeneous Poisson process are very valuable for making statements about clustering and regularity, and we use this procedure to make such statements about the spatial organization of pyramidal brain cells in the human cerebral cortex in both Papers B and C.

Global envelopes are not restricted to the task of validating models, but can be used to perform any test where one wishes to base the test on a summary statistic which is a curve. In Paper C where we studied the spatial organization of pyramidal cells in the human cerebral cortex from subjects with and without certain mental illnesses, we for instance used global envelopes to perform a permutation test for whether there were significant differences in functional summary statistics between the different groups of subjects.

A Details for deriving the G-function for an α-Ginibre point process

Here I show that K^0 for an α-Ginibre point process has the same eigenfunctions $\phi_k^{b(0,r)}$ as K. This is seen by the following calculations: Let $C_{\phi^k} = \sqrt{\frac{\rho^k \pi^{k-1}}{\alpha^k \gamma(k, \rho\pi r^2/\alpha)}}$. Then

$$
\begin{aligned}
&\int_{b(0,r)} K^0(z,s)\phi_k^{b(0,r)}(s)\,\mathrm{d}s \\
&= \int_{b(0,r)} K(z,s)\phi_k^{b(0,r)}(s)\,\mathrm{d}s - \int_{b(0,r)} \rho \exp(-\pi\rho(|z|^2+|s|^2)/(2\alpha))\phi_k^{b(0,r)}(s)\,\mathrm{d}s \\
&= \lambda_k^{b(0,r)}\phi_k^{b(0,r)}(z) - \\
&\quad C_{\phi^k}\rho\exp(-\pi\rho|z|^2/(2\alpha))\int_0^r \exp(-\pi\rho(t^2)/(\alpha))t^k \int_0^{2\pi}\exp((k-1)\theta\mathbf{i})\,\mathrm{d}\theta\mathrm{d}t
\end{aligned}
$$

where $\mathbf{i}$ is the imaginary unit. When $k > 1$, the last integral equals zero, so

$$
\int_{b(0,r)} K^0(z,s)\phi_k^{b(0,r)}(s)\,\mathrm{d}s = \lambda_k^{b(0,r)}\phi_k^{b(0,r)}(z), \quad k > 1.
$$

When $k = 1$,

$$
\begin{aligned}
&\int_{b(0,r)} K^0(z,s)\phi_1^{b(0,r)}(s)\,\mathrm{d}s \\
&= \lambda_1^{b(0,r)}\phi_1^{b(0,r)}(z) - C_{\phi^1} 2\pi\rho\exp(-\pi\rho|z|^2/(2\alpha))\int_0^r \exp(-\pi\rho(t^2)/(\alpha))t\mathrm{d}t \\
&= \lambda_1^{b(0,r)}\phi_1^{b(0,r)}(z) - C_{\phi^1}\exp(-\pi\rho|z|^2/(2\alpha))\alpha\int_0^{\pi\rho r^2/\alpha}\exp(-q)\mathrm{d}q \\
&= \phi_1^{b(0,r)}(z)\left(\lambda_1^{b(0,r)} - \alpha\gamma(1,\rho\pi r^2/\alpha)\right) \\
&= 0.
\end{aligned}
$$

This also shows that the eigenvalues are as in (14).

References

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

Baddeley, A., Turner, R., Mateu, J., and Bevan, A. (2013). Hybrids of Gibbs point process models and their implementation. *Journal of Statistical Software*, 55:1–43.

Beaumont, M. A. (2010). Approximate Bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics*, 41:379–406.

Besag, J. E. (1975). Statistical analysis of non-lattice data. *The Statistician*, 24:179–195.

Cox, D. R. (1955). Some statistical methods connected with series of events. *Journal of the Royal Statistical Society: Series B (Methodological)*, 17:129–157.

Daley, D. J. and Vere-Jones, D. (2003). *An Introduction to the Theory of Point Processes. Volume I:Elementary Theory and Methods*. Springer-Verlag, New York, second edition.

Diggle, P. J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

Diggle, P. J. (2013). *Statistical analysis of spatial and spatio-temporal point patterns*. CRC press, Boca Raton, third edition.

Diggle, P. J. and Gratton, R. J. (1984). Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society, series B*, 46:193–2012.

Geyer, C. (1999). Likelihood inference for spatial point processes. In *Stochastic geometry: Likelihood and Computation*, pages 79–140. Chapman and Hall/CRC, Boca Raton.

Ginibre, J. (1965). Statistical ensembles of complex, quaternion, and real matrices. *Journal of Mathematical Physics*, 6:440–449.

Goldstein, J., Haran, M., Simeonov, I., Fricks, J., and Chiaromonte, F. (2015). An attraction–repulsion point process model for respiratory syncytial virus infections. *Biometrics*, 71:376–385.

Jensen, J. L. and Møller, J. (1991). Pseudolikelihood for exponential family models of spatial point processes. *Annals of Applied Probability*, 3:445–461.

Kelly, F. P. and Ripley, B. D. (1976). A note on Strauss's model for clustering. *Biometrika*, 63:357–360.

Lavancier, F. and Møller, J. (2016). Modelling aggregation on the large scale and regularity on the small scale in spatial point pattern datasets. *Scandinavian Journal of Statistics*, 43:587–609.

Lavancier, F., Møller, J., and Rubak, E. (2015). Determinantal point process models and statistical inference. *Journal of Royal Statistical Society: Series B (Statistical Methodology)*, 77:853–877.

Macchi, O. (1975). The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122.

Møller, J. (2003). Shot noise Cox processes. *Advances in Applied Probability*, 35:614–640.

Møller, J., Safavimanesh, F., and Rasmussen, J. G. (2016). The cylindrical *K*-function and poisson line cluster point processes. *Biometrika*, 103:937–954.

Mrkvička, T., Myllymäki, M., Jilik, M., and Hahn, U. (2020). A one-way ANOVA test for functional data with graphical interpretation. *Kybernetika*, 56:432–458.

Myllymäki, M. and Mrkvička, T. (2019). GET: Global envelopes in R. arXiv preprint arXiv:1911.06583.

Myllymäki, M., Mrkvička, T., Grabarnik, P., Seijo, H., and Hahn, U. (2017). Global envelope tests for spatial processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79:381–404.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25:451–482.

Møller, J. and Waagepetersen, R. P. (2004). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman and Hall/CRC, Boca Raton.

Preston, C. (1976). *Random Fields*. Springer-Verlag, Berlin.

Ripley, B. D. (1988). *Statistical Inference for Spatial Processes*. Cambridge University Press, Cambridge.

Ripley, B. D. and Kelly, F. P. (1977). Markov point processes. *Journal of the London Mathematical Society*, 2:188–192.

Ruelle, D. (1969). *Statistical Mechanics: Rigorous Results*. W. A. Benjamin Reading, Massachusetts.

Stoyan, D. (1979). Interrupted point processes. *Biometrical Journal*, 21:607–610.

Strauss, D. J. (1975). A model for clustering. *Biometrika*, 62:467–475.

Part II

Papers

Paper A

Approximate Bayesian inference for a spatial point process model exhibiting regularity and random aggregation

Ninna Vihrs, Jesper Møller, and Alan E. Gelfand

The paper has been published in
Scandinavian Journal of Statistics (2022) **49**, 185-211

The layout has been revised.

Abstract

In this paper, we propose a doubly stochastic spatial point process model with both aggregation and repulsion. This model combines the ideas behind Strauss processes and log Gaussian Cox processes. The likelihood for this model is not expressible in closed form but it is easy to simulate realisations under the model. We therefore explain how to use approximate Bayesian computation (ABC) to carry out statistical inference for this model. We suggest a method for model validation based on posterior predictions and global envelopes. We illustrate the ABC procedure and model validation approach using both simulated point patterns and a real data example.

1 Introduction

Spatial point patterns are usually divided into three cases: regularity/repulsiveness, complete spatial randomness, and aggregation/clustering. There is a wide selection of point process models suitable for these situations, see e.g. the overview in Lavancier and Møller (2016, Section 1) and the references therein. However, some point patterns show repulsiveness between the points at small scale and aggregation at a larger scale, see Lavancier and Møller (2016) for a detailed discussion. In this regard, Lavancier and Møller (2016) suggested a model for this situation obtained by a dependent thinning of a repulsive point process. It is also possible to construct certain Gibbs point processes with this behaviour, see e.g. Baddeley et al. (2013) and Goldstein et al. (2015).

1.1 The log Gaussian Cox Strauss process

In this paper, we present a model for regularity at small scale and aggregation at larger scale which is a combination of a pairwise interaction point process and a log Gaussian Cox process. It is constructed by the following two steps.

First, we consider a pairwise interaction point process defined as follows. Let $\mathbf{X}$ be a spatial point process viewed as a finite random subset of a given bounded region $W \subset \mathbb{R}^2$ (we think of W as an observation window). Then $\mathbf{X}$ is a pairwise interaction point process if $\mathbf{X}$ follows a density (with respect to the unit rate Poisson process on W) of the form

$$f(\mathbf{x} \mid \psi, \varphi) = \frac{1}{C_{\psi,\varphi}} \prod_{i=1}^{n} \psi(x_i) \prod_{i<j} \varphi(\|x_i - x_j\|) \tag{1}$$

for all point patterns $\mathbf{x} = \{x_1, \ldots, x_n\} \subset W$ with $0 \leq n < \infty$ (if $n = 0$ then $\mathbf{x} = \emptyset$ is the empty point pattern), where the notation means the following: $\psi : W \to [0, \infty)$ is a so-called first order interaction function; $\varphi : [0, \infty) \to [0, \infty)$ is a so-called second order interaction function; $\|\cdot\|$ denotes usual

Euclidean distance; and $C_{\psi,\varphi} = 1/f(\emptyset \mid \psi, \varphi)$ is the normalising constant which is required to be positive and finite. Usually, $\varphi(\cdot) \leq 1$, in which case the density is well defined and results in a model for repulsion between the points. The first order interaction function may be used to model systematic aggregation of points.

Second, we consider a doubly stochastic construction, by replacing ψ with a random function Ψ in order to introduce random aggregation to the model. This is an extension of a Cox process (the case $\varphi = 1$, cf. Cox, 1955), and such a model was considered in Berthelsen and Møller (2008) when Ψ is the stochastic intensity function of a shot noise Cox process. Instead, we use the random intensity function of a log Gaussian Cox process (LGCP, see Møller et al., 1998), which is a popular model for random aggregation. Specifically, we let

$$\Psi(u) = \exp(Z(u)), \qquad u \in W, \tag{2}$$

where $\mathbf{Z} := \{Z(u)\}_{u \in W}$ is a Gaussian random field (GRF) with constant mean $\mu \in \mathbb{R}$ and exponential covariance function

$$c(u, v) = \sigma^2 \exp\left(-\|u - v\| / s\right), \qquad u, v \in W.$$

Here, $\sigma^2 \geq 0$ is the variance and $s > 0$ is a scale parameter. For $\sigma^2 > 0$, the flexible stochastic process $\Psi(u)$ may account for aggregation caused by unobserved covariates. Note that $\Psi(u) = \exp(\mu)$ if $\sigma^2 = 0$.

For the second order interaction function in (1), Berthelsen and Møller (2008) used a piecewise linear function, whereas we will use the much simpler second order interaction function of a Strauss process (Strauss, 1975, Kelly and Ripley, 1976). This gives us a density for $\mathbf{X}$ (with respect to the unit rate Poisson process on W) of the form

$$f(\mathbf{x} \mid \theta) = \mathrm{E}\left[\frac{1}{C_\theta(\mathbf{Z})} \prod_{i=1}^{n} \exp\left(Z(x_i)\right) \prod_{i<j} \gamma^{1\left[\|x_i - x_j\| \leq R\right]}\right], \tag{3}$$

where $\theta = \left(\mu, \sigma^2, s, \gamma, R\right)$ is the parameter vector. Here, the expectation is with respect to the GRF; $C_\theta(\mathbf{Z})$ is the normalising constant obtained by conditioning on $\mathbf{Z}$; $1[\cdot]$ denotes the indicator function; and we use the convention $0^0 = 1$. The parameter $R > 0$ is called the interaction radius and the parameter $\gamma \in [0, 1]$ controls the repulsion between points. This model for $\mathbf{X}$ will be referred to as an LGCP-Strauss process.

The model includes some well-known special cases:

(a) Conditioned on $\mathbf{Z}$, $\mathbf{X}$ is an inhomogeneous Strauss process.

(b) If $\sigma^2 = 0$, $\mathbf{X}$ is a usual Strauss process. If in addition $\gamma = 0$, $\mathbf{X}$ is a hard core Gibbs process with hard core parameter R; or if in addition $\gamma = 1$, $\mathbf{X}$ is a homogeneous Poisson process on W with intensity $\exp(\mu)$.

(c) If $\gamma = 1$, $\mathbf{X}$ is an LGCP.

The following coupling result becomes useful when interpreting the meaning of γ and when we later discuss simulation of the LGCP-Strauss process. To stress the dependence on γ, we write $\mathbf{X} = \mathbf{X}_\gamma$. Then, using a dependent thinning technique (Kendall and Møller, 2000) it follows that there exists a coupling of the LGCP-Strauss processes $\mathbf{X}_\gamma$ for all $\gamma \in [0, 1]$ such that $\mathbf{X}_\gamma \subseteq \mathbf{X}_{\gamma'}$ whenever $0 \leq \gamma < \gamma' \leq 1$. In particular, the special case of the LGCP $\mathbf{X}_1$ (item (c) above) dominates any of the LGCP-Strauss processes $\mathbf{X}_\gamma$. The intensity of $\mathbf{X}_1$ is $\exp(\mu + \sigma^2/2)$ (Møller et al., 1998), so $\exp(\mu + \sigma^2/2)|W|$ provides an upper bound on the expected number of points in $\mathbf{X}_\gamma$. Here, $|W|$ denotes the area of W.

Note that if we are not in any of the above special cases (a)–(c), both the intensity and other moment characteristics of $\mathbf{X}$, the density (3), and the Papangelou conditional intensity (see e.g. Møller and Waagepetersen, 2004) are not expressible in closed form. Therefore, in general, usual approaches for estimation based on likelihood, pseudo-likelihood, composite likelihood, and minimum contrasts (see the review in Møller and Waagepetersen, 2017) are not feasible for the LGCP-Strauss process. This makes statistical inference challenging. Finally, note that for a Poisson process 'everything is known', whilst for a Strauss process the Papangelou conditional intensity but not the moment characteristics are expressible in closed form, and for an LGCP the moment characteristics but not the Papangelou conditional intensity are expressible in closed form, cf. the above-mentioned references.

1.2 Objective and outline

In this paper, we show how to use approximate Bayesian computation (ABC) to make statistical inference for spatial point process models such as the LGCP-Strauss process model. In brief, ABC is a flexible method for approximate inference in a Bayesian framework, which does not require the likelihood to be expressible in closed form. Instead, it is based on the ability to make simulations under the assumed model, which are then compared to the observed data by using summary statistics.

In previous work on ABC in the setting of spatial point process models, Shirota and Gelfand (2017) explained how ABC can be used for Strauss process models and determinantal point process models. For the Strauss process model they estimated the interaction radius using maximum profile pseudo likelihood and then kept the interaction radius fixed at this estimate during the ABC procedure. Further, Soubeyrand et al. (2013) presented an ABC method using functional summary statistics such as the pair correlation function, which they exemplified for a Thomas process model and a marked point process model. Finally, Stoica et al. (2017) presented an ABC method for spatial point process models dealing with an intractable normalising constant in

the likelihood. This method will not help for the LGCP-Strauss process since it is not only a normalising constant but also the expectation in (3) which makes the density intractable.

In contrast to Shirota and Gelfand (2017), the method we use for statistical inference is based entirely on ABC, and unlike Shirota and Gelfand (2017) and Soubeyrand et al. (2013) we do not fix any of the unknown parameters during the ABC procedure. Furthermore, we provide a discussion of the choice of summary statistics for ABC when making statistical inference for the LGCP-Strauss process. We also suggest a method for model validation and comparison based on posterior predictions and global envelopes. We use this in a simulation study to assess the quality of ABC results for LGCP-Strauss processes and to investigate whether realisations of the LGCP-Strauss process can be distinguished from LGCPs and Strauss processes.

The remainder of this paper is organized as follows. Section 2 presents simulated examples of LGCP-Strauss processes. In Section 3, our chosen method for ABC model fitting is specified. Section 4 contains ABC analyses for simulated data. Section 5 contains a real data example using a point pattern of oak trees which suffer from frost shake. Section 6 concludes with a brief summary and paths for future work.

The open source software R (R Core Team, 2019) is used for all statistical computations. Most plots are created with the R-package `ggplot2` (Wickham, 2016) and some of the functionalities of the R-package `spatstat` (Baddeley et al., 2015) are used to handle spatial point patterns.

2 Simulation study of the LGCP-Strauss process

Consider an LGCP-Strauss process $\mathbf{X}$ on the observation window W with density (3), which depends on the parameter vector $\theta = (\mu, \sigma^2, s, \gamma, R)$. We simulate data under this model in two steps: First, a realisation $\mathbf{z}$ of $\mathbf{Z}$ is simulated (see e.g. Schlather (1999)). In R, this can be done with the function `RFsimulate` from the R-package `RandomFields` (Schlather et al., 2015, 2019). Second, a realisation of $\mathbf{X}$ given $\mathbf{Z} = \mathbf{z}$ is simulated using an MCMC algorithm, namely a birth-death Metropolis-Hastings algorithm (Geyer and Møller, 1994, specifically, a birth is proposed with probability $1/2$ and otherwise a death is proposed; for a birth proposal, the new point is generated from a density proportional to $\exp(\mathbf{z})$; and for a death proposal, the point to die is selected uniformly from the current point pattern).

Figure 1 shows six examples of simulated realisations of the LGCP-Strauss process on the unit square (using a burnin of 20 000 in the MCMC algorithm) plotted on top of the corresponding realisation of $\mathbf{Z}$. The processes generating the first three point patterns only differ by the value of γ and the ones generating the last three only differ by the value of σ^2; the remaining parameters

are specified in the caption.

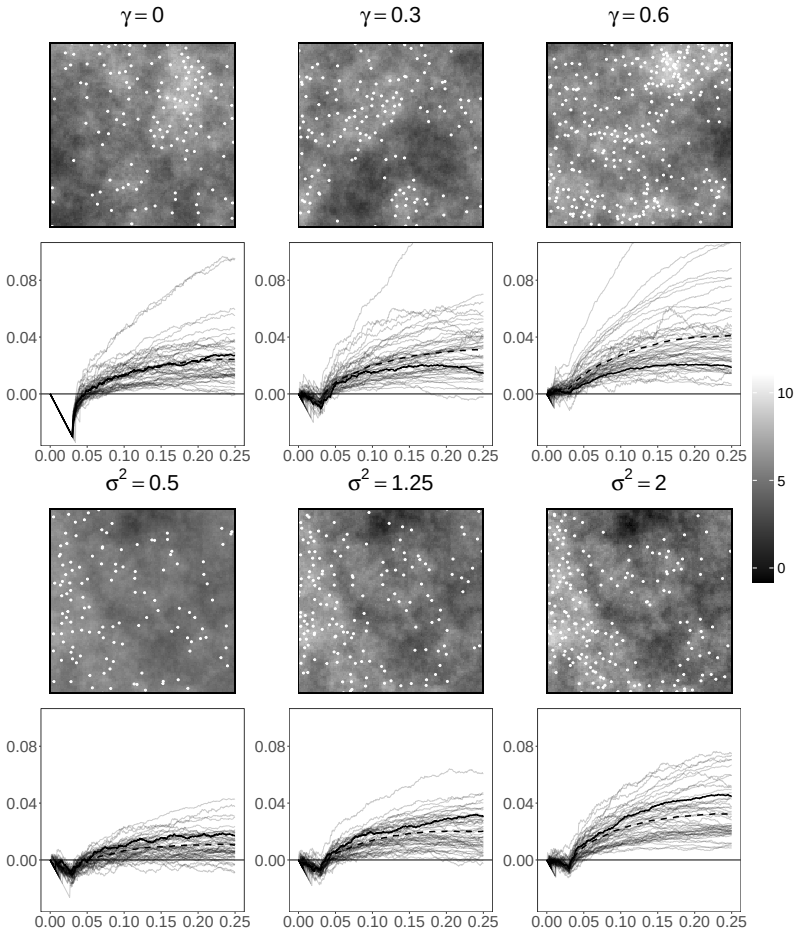

Fig. 1: First and third row: simulated LGCP-Strauss processes on the unit square (white points) and the corresponding realisation of $\mathbf{Z}$ (grey scale image). In the first row, the parameters are $\mu = 5,\ \sigma^2 = 2,\ R = 0.03,\ s = 0.3$, and γ is as specified at the top of each plot. In the third row, the parameters are $\mu = 5, R = 0.03, s = 0.2, \gamma = 0.3$, and σ^2 is as specified at the top of each plot. Second and fourth row: empirical L-function minus the identity for the point pattern directly above the plot (solid curve) and for 49 different simulations of the same process (grey curves) plus their mean (dashed curve).

To asses the degree of clustering and regularity we consider the L-function $L(r) = \sqrt{K(r)/\pi}$, where $r > 0$ denotes inter-point distance and K is Ripley's K-function (Ripley, 1976, 1977). The L-function is commonly used to summarise important aspects of the second order moment properties of spatial point processes. Since $L(r) = r$ for a Poisson process, one usually considers $T(r) \coloneqq L(r) - r$. The L-function is often used to make statements about clus-

tering/regularity as follows: If $L(r) < r$ ($L(r) > r$), this indicates that $\mathbf{X}$ is regular/repulsive (aggregated/clustered) at inter-point distances r (for more detailed explanations, see e.g. Baddeley et al. (2015)).

Figure 1 also shows plots of empirical estimates of the L-functions of the point patterns using Ripley's isotropic edge correction (Ripley, 1977)(an alternative to displaying the empirical L-functions would show the empirical pair correlation functions though they are sensitive to choice of bandwidth). Each of these plots also includes the empirical L-functions of 49 different realisations of the process in question plus the mean of these in order to assess the general behaviour of the estimator (note that the mean does not necessarily represent the behaviour of the theoretical L-function because the estimator is biased). As expected, the point patterns exhibit both regularity and aggregation. The first three point patterns show a decreasing degree of regularity at small to moderate distances as γ increases, but a similar degree of aggregation at large distances. However, the general behaviour of the empirical L-function suggests a tendency to a higher degree of clustering at large distances as γ increases. The last three point patterns show a similar degree of regularity at small to moderate distances and an increasing degree of clustering at large distances as σ^2 increases. We also see that the variance of the estimator of the L-function at large distances apparently increases as σ^2 increases. Note that the empirical L-function of the first point pattern, which is generated from the LGCP-Strauss process where $\sigma^2 = 0.5$, would not be uncommon for realisations of the LGCP-Strauss processes where $\sigma^2 = 1.25$ or $\sigma^2 = 2$ either. This suggest that it may be difficult to see the effect of σ^2 on the clustering when looking at a given realisation. Notice that in general the repulsive behaviour of the point patterns to some extent obscures the finer variations in Z (especially for strong repulsion), so overall we may expect that it will be difficult to make inference for the parameters of the GRF.

3 ABC for spatial point process models

ABC is a method used to make approximate Bayesian inference for a parametric model with an intractable likelihood by developing an approximate posterior sample of the parameters. Instead of having to evaluate the likelihood, it is only necessary to be able to simulate from the model in order to do ABC. The most basic ABC technique is ABC rejection sampling which goes as follows: for a parametric model with parameter θ and observed data x_{obs}, specify a summary statistic S and a distance function χ; repeat sampling θ' from its prior, and x' given θ' from the likelihood, until $\chi(S(x'), S(x_{\text{obs}})) < \varepsilon$ where ε is a chosen tolerance; then return θ'. If the inequality $\chi(S(x'), S(x_{\text{obs}})) < \varepsilon$ is replaced by $S(x') = S(x_{\text{obs}})$ and S is either the identity function or a sufficient statistic, θ' will be an exact sample from

the posterior distribution. When this ideal situation is not achievable, the sample will instead be from an approximation to the posterior distribution, referred to as the ABC posterior. The quality of this approximation will depend on the choice of ε and S. There exist many different ABC techniques and some even circumvent the need for summary statistics, see e.g. Jiang et al. (2018), Park et al. (2016), and Bernton et al. (2019). However, it does not seem to be straightforward to apply these methods to the setting of spatial point process models.

In this paper, we illustrate how ABC can be used to make inference for the LGCP-Strauss process. A relatively simple ABC method is successful for this illustration; it is specified in Section 3.1. So, we need not discuss or compare different ABC techniques. For a more detailed overview of ABC and some of the different techniques, see e.g. Beaumont (2010).

3.1 Specification of the ABC procedure

Consider a spatial point process $\mathbf{X}$ defined on a bounded region $W \subset \mathbb{R}^2$ and which follows a parametric model with parameter vector θ. Assume a realisation $\mathbf{x}_{\text{obs}}$ of $\mathbf{X}$ is observed. Our chosen procedure for ABC is specified in Algorithm 1 below. It is inspired by Shirota and Gelfand (2017) and the semi-automatic approach by Fearnhead and Prangle (2012). Shirota and Gelfand (2017) used a Markov chain Monte Carlo method for the ABC sampling whereas we choose ABC rejection sampling, because of its simplicity and ability to be run in parallel. The semi-automatic part refers to the fact that the user specified summary statistics are only used in a pilot run instead of directly in the ABC rejection step.

In Algorithm 1, $n(\mathbf{x})$ is the number of points in a point pattern $\mathbf{x}$, and in the first and last for loop we demand that $n(\mathbf{x}) > m$ for each simulated $\mathbf{x}$. This is not strictly necessary for ABC, but it is a way to insure that summary statistics are not calculated for point patterns with very few points. Most summary statistics for spatial point patterns can only be calculated or considered reliable if there is a reasonable number of points in the point pattern. In the examples of Sections 4 and 5, $m = 10$ was found to be sufficient.

In the second for loop of Algorithm 1, we choose to fit the linear models approximating the posterior means $\mathrm{E}[\theta_i \mid \mathbf{x}]$, $i = 1, \ldots, p$, with a special case of a relaxed Lasso (Meinshausen, 2007): First, a model is fitted with Lasso regression, where the penalty term is chosen based on a cross-validation argument using the 'one-standard-error rule' (see e.g. Hastie et al., 2015, Chapter 2). Let $\hat{\beta}_j^{i,\text{Lasso}}$, $j = 1, \ldots, d$, be the resulting estimate of β_j^i and set $T^{i,\text{Lasso}}(\mathbf{x}) = \left\{ T_j(\mathbf{x}) \mid \hat{\beta}_j^{i,\text{Lasso}} \neq 0, j = 1, \ldots, d \right\}$. Second, the summary statistics in $T^{i,\text{Lasso}}$ are used as predictors in a linear model fitted with ordinary least squares, which results in the final model for approximating $\mathrm{E}[\theta_i \mid \mathbf{x}]$.

Algorithm 1: Procedure for ABC

Input : Data $\mathbf{x}_{\text{obs}}$, a prior distribution $\pi(\theta)$ for $\theta = (\theta_1, \ldots, \theta_p)$, a procedure for simulating from the likelihood $\pi(\mathbf{x} \mid \theta)$, a summary statistic $T(\mathbf{x}) = (T_1(\mathbf{x}), \ldots, T_d(\mathbf{x}))$, positive integers k_{pilot} and k_{ABC}, and a non-negative integer m.

Output: A sample $\theta^{\text{ABC},1}, \ldots, \theta^{\text{ABC},k_{\text{ABC}}}$ from the ABC approximate posterior distribution.

Calculate $T_{\text{obs}} = T(\mathbf{x}_{\text{obs}})$.

Pilot run:

For $i = 1, \ldots, k_{\text{pilot}}$

  repeat

    sample $\theta^{\text{pilot},i} \sim \pi(\theta)$ and $\mathbf{x}^{\text{pilot},i} \sim \pi(\mathbf{x} \mid \theta^{\text{pilot},i})$

  until $n(\mathbf{x}^{\text{pilot},i}) > m$.

For $j = 1, \ldots, p$

  based on the sample $\left\{ \left(\theta^{\text{pilot},i}, \mathbf{x}^{\text{pilot},i} \right) \right\}_{i=1}^{k_{\text{pilot}}}$, fit a linear model for the posterior mean

$$\mathrm{E}[\theta_j \mid \mathbf{x}] \approx \theta_j(\mathbf{x}) \coloneqq \alpha^j + \beta^{j^T} (T(\mathbf{x}) - T_{\text{obs}})$$

  where $\mathbf{x}$ is a realisation of $\mathbf{X}$, $\alpha^j \in \mathbb{R}$, and $\beta^j = (\beta_1^j, \ldots, \beta_d^j) \in \mathbb{R}^d$. Let $\hat{\theta}_j(\mathbf{x})$ be the estimate of $\theta_j(\mathbf{x})$ when α^j and β^j are replaced by the estimates $\hat{\alpha}^j$ and $\hat{\beta}^j$.

Define the distance measure

$$\chi\left(\hat{\theta}(\mathbf{x}), \hat{\theta}(\mathbf{x}_{\text{obs}})\right) = \sum_{j=1}^{p} \frac{\left(\hat{\theta}_j(\mathbf{x}) - \hat{\theta}_j(\mathbf{x}_{\text{obs}})\right)^2}{\hat{\text{var}}\left(\hat{\theta}_j\right)} = \sum_{j=1}^{p} \frac{\left(\hat{\theta}_j(\mathbf{x}) - \hat{\alpha}^j\right)^2}{\hat{\text{var}}\left(\hat{\theta}_j\right)}$$

where $\hat{\theta}(\mathbf{x}) = \left(\hat{\theta}_1(\mathbf{x}), \ldots, \hat{\theta}_p(\mathbf{x})\right)$ and $\hat{\text{var}}\left(\hat{\theta}_j\right)$ is the empirical variance of $\left\{ \hat{\theta}_j(\mathbf{x}^{\text{pilot},i}) \right\}_{i=1}^{k_{\text{pilot}}}$.

Choose ε as the empirical 1% percentile of $\left\{ \chi(\mathbf{x}^{\text{pilot},i}, \mathbf{x}_{\text{obs}}) \right\}_{i=1}^{k_{\text{pilot}}}$.

ABC rejection sampling:

For $i = 1, \ldots, k_{\text{ABC}}$

  repeat

    repeat

      sample $\theta^{\text{ABC},i} \sim \pi(\theta)$ and $\mathbf{x}^{\text{ABC},i} \sim \pi(\mathbf{x} \mid \theta^{\text{ABC},i})$

    until $n(\mathbf{x}^{\text{ABC},i}) > m$.

  until $\chi(\hat{\theta}(\mathbf{x}^{\text{ABC},i}), \hat{\theta}(\mathbf{x}_{\text{obs}})) < \varepsilon$

We employ a Lasso regression approach because we want to use a relatively large number of summary statistics (see Section 3.2).

3.2 Choice of summary statistics in the case of LGCP-Strauss process models

An important part of ABC is the selection of appropriate summary statistics. It is not possible to find sufficient statistics for the LGCP-Strauss process, since it does not have a closed form density and, therefore, it is not obvious which summary statistics to use. We emphasize that there are limitless possibilities for choosing such summary statistics. The following describes one choice which, based on some theoretical knowledge and numerical experiments, we believe is an appropriate set of summary statistics when implementing ABC for LGCP-Strauss processes.

Recall from Section 1.1 that $\exp(\mu + \sigma^2/2)|W|$ provides an upper bound on the expected number of points for an LGCP-Strauss process on $|W|$. We may therefore expect that especially the parameters μ and σ^2 are related to the number of points in a point pattern generated by an LGCP-Strauss process. We therefore include the number of observed points as a summary statistic.

Recall also the L-function from Section 2 which is a theoretical tool commonly used to asses the degree of clustering and regularity. Since these properties are related to many of the parameters of the LGCP-Strauss process, we consider an empirical estimate of the L-function among the summary statistics for ABC (see (b)-(c) below). A simulation study suggested that for realisations of an LGCP-Strauss process, the empirical estimate of $L(r) - r$ often has a global minimum when r is close to the interaction radius R, at least when there is strong to moderate repulsion in the model. In this regard, see Figure 1 for some examples of empirical L-functions associated with realisations of LGCP-Strauss processes. We take this into consideration when choosing the summary statistics (see (b) below).

Furthermore, numerical experiments suggested that it may be particularly difficult to learn much about the GRF based on a realisation (see also the discussion in Section 2). The GRF mainly affects the clustering, so we would like to include some further summary statistics which can capture this. For this purpose, assume for ease of exposition that W is a square with side length h. Then we split W into q^2 squares $W_{i,j}$ of side length h/q, $i, j = 1, \ldots, q$, and let $n(\mathbf{x} \cap W_{i,j})$ be the number of points in $\mathbf{x}$ falling in $W_{i,j}$. We choose summary statistics which describe how $n(\mathbf{x} \cap W_{i,j})$ varies (see (d) below) and which are calculated for a user-specified finite range of q-values.

Specifically, for a point pattern $\mathbf{x}$ (either $\mathbf{x}_{\text{obs}}$ or one of the simulated point patterns in Algorithm 1), we chose the following summary statistics.

(a) $n_{\log} \coloneqq \log(n(\mathbf{x}))$.

(b) $L_{\max} \coloneqq \max(\hat{L}(r) - r)$,
$L_{\min} \coloneqq \min(\hat{L}(r) - r)$, and
$L_{\arg\min} \coloneqq \arg\min(\hat{L}(r) - r)$,
where $\hat{L}$ is a non-parametric estimate of the L-function evaluated over a user-specified finite range of r-values.

(c) $\hat{L}(r) - r$ evaluated at m equally spaced values of r between 0 and $0.2h$ referred to as $L_1, \ldots, L_m$.

(d) $C_{\max,q} \coloneqq \max_{i,j=1,\ldots,q} \left(\left\{n(\mathbf{x} \cap W_{i,j})/n(\mathbf{x})\right\}\right)$,
$C_{\min,q} \coloneqq \min_{i,j=1,\ldots,q} \left(\left\{n(\mathbf{x} \cap W_{i,j})/n(\mathbf{x})\right\}\right)$, and
$C_{\log \mathrm{var},q} \coloneqq \log\left(\hat{\mathrm{var}}\left(\left\{n(\mathbf{x} \cap W_{i,j})/n(\mathbf{x})\right\}_{i,j=1}^{q}\right)\right)$,
where again $\hat{\mathrm{var}}$ means empirical variance.

We have chosen these specific forms of the summary statistics based on some numerical experiments. In the examples of Sections 4 and 5, $m = 40$ and $q = 2, \ldots, 5$. This means that the vector of summary statistics T in Algorithm 1 has dimension equal to $1 + 3 + 40 + 3 \cdot 4 = 56$.

4 ABC for simulated realisations of LGCP-Strauss processes

4.1 Prior specification and numerical considerations

We will now illustrate how the procedure in Algorithm 1 can be used to make ABC for the simulated realisations of LGCP-Strauss processes in Figure 1. In order to do this, it is required to specify a (proper) prior distribution for the parameter vector $\theta = (\mu, \sigma^2, s, \gamma, R)$ of the LGCP-Strauss process. For the examples in this section, we considered three different prior distributions for θ which we refer to as P_1, P_2, and P_3. In each case, a priori we assume the five parameters μ, σ^2, s, γ, and R are independent.

1. P_1: $\mu \sim \mathrm{Unif}(3,6)$, $\sigma^2 \sim \mathrm{Unif}(0,4)$, $s \sim \mathrm{Unif}(0.01, 0.5)$, $\gamma \sim \mathrm{Unif}(0,1)$, and $R \sim \mathrm{Unif}(0, 0.05)$;

2. P_2: $\mu \sim \mathrm{Norm}(3.5, 1)$, $\sigma^2 \sim \mathrm{Gamma}(1,1)$, $s \sim \mathrm{Gamma}(1,6)$, $\gamma \sim \mathrm{Beta}(1,2)$, and $R \sim \mathrm{Gamma}(1, 50)$;

3. P_3: $\mu \sim \mathrm{Norm}(5,1)$, $\sigma^2 \sim \mathrm{Gamma}(10,4)$, $s \sim \mathrm{Gamma}(7,20)$, $\gamma \sim \mathrm{Beta}(2,1)$, and $R \sim \mathrm{Gamma}(10, 250)$.

Here, $\mathrm{Unif}(a,b)$ is the uniform distribution on the interval (a,b), $\mathrm{Norm}(a,b)$ is the normal distribution with mean a and variance b, $\mathrm{Gamma}(a,b)$ is the

gamma distribution with shape a and rate b, and $\text{Beta}(a, b)$ is the beta distribution with the shape parameters a and b. For computational reasons, we have chosen to truncate all the prior distributions for the parameters to the intervals of the uniform distributions of P_1. For example, the computational reasons include a consideration of the number of points in simulations. The more points a simulated point pattern has, the more computationally expensive the simulation procedure will be (see below). Recall that $\exp(\mu + \sigma^2/2)$ is an upper bound on the expected number of points in the unit square. By limiting the range of μ and σ^2 in their prior distributions, we ensure that point patterns simulated during the ABC procedure will not yield unreasonably many points compared to the number of points in our observed point patterns. Further, this ensures that the simulated point patterns can achieve regularity similar to that in the observed point patterns.

In order to use the MCMC algorithm when a realisation $\mathbf{z} = \{z(u)\}_{u \in W}$ of the GRF is given (see Section 2), it is necessary to choose a burn-in which can be used for all simulations in the ABC procedure. In order to choose this burn-in, we considered 30 samples of the parameters drawn from the prior distribution P_1; used the MCMC algorithm for all these samples; and considered trace plots of the number of points and R-close pairs. Figure 10 shows these plots for three different prior samples for illustration. We choose to initiate the MCMC algorithm at the empty point pattern and at a realisation of an inhomogeneous Poisson process on W with intensity function $\exp(z(u))$ (these initial states are extreme because of the coupling result mentioned in Section 1.1). It seems that the higher the number of points, the slower the convergence. The burn-in should be high enough for the MCMC algorithm to have converged given any prior sample, but increasing the burn-in will also increase the computation time. Considering all 30 examples, 20 000 appears to be an appropriate overall burn-in. All following simulations are iteration 20 001 of the MCMC algorithm initiated at the empty point pattern.

4.2 Posterior results

We used Algorithm 1 on the six point patterns in Figure 1 with $k_{\text{pilot}} = 10\,000$ and $k_{\text{ABC}} = 1\,000$ (the same choice as in Shirota and Gelfand, 2017). In some cases, one or two pilot samples had to be excluded afterwards because some summary statistics yielded infinite values. For a single point pattern, in our situation, it usually took about 10 hours to run the ABC procedure in parallel on 45 cores (evidently, run time will depend heavily on the given situation and software). Figures 2 and 3 show kernel density estimates of the resulting (approximate) marginal posterior distributions of the parameters, using a Gaussian kernel and a bandwidth chosen with the method by Sheather and Jones (1991).

From Figure 2, we see the following.

- As the true value of γ increases, the ABC posterior distributions of μ become more and more left skewed. The choice of prior seems to have small influence on the general behaviour of these. The ABC posteriors corresponding to the prior P_2 are very different from the prior in all three cases, whereas the ABC posteriors corresponding to the priors P_1 and P_3 seem to become increasingly different from their corresponding priors as γ increases.

- The ABC posterior distributions for σ^2 and s look rather similar to their prior distributions, except near zero in the situations of the priors P_1 and P_2 where the ABC posteriors are considerably smaller than their corresponding priors. This suggests that even though it may be difficult to infer with precision about the values of σ^2 and s, we are able to learn that they have a small probability of being near zero, which means that we can detect a clustering effect in the point patterns.

- When the true value of γ is 0, the ABC posteriors of γ are very concentrated near 0. We see a tendency for the spread of the ABC posteriors to increase as γ increases. The choice of prior seems to have small influence on the overall behaviour of the ABC posteriors in the first two cases. Especially for the prior P_1, the maxima of the ABC posteriors seem to be in good agreement with the true value in all three cases.

- The posterior distributions for R seem to approach the corresponding priors as γ increases. In all three cases, the maxima of the posterior distributions corresponding to the prior P_1 are in good agreement with the true value.

From Figure 3, we see the following.

- The overall behaviour of the ABC posteriors for μ seems to be rather unaffected by the choice of prior, and the ABC posteriors seem to be in good agreement with the true value. The spread of these ABC posteriors seems to increase slightly as σ^2 increases.

- For σ^2, the ABC posteriors corresponding to the prior P_3 are quite similar to the prior. The posteriors corresponding to the prior P_1 are getting closer to the prior as the true value of σ^2 increases. The spread of the posteriors corresponding to the priors P_1 and P_2 seems to be increasing as σ^2 increases. Again, we see that the posteriors corresponding to the priors P_1 and P_2 place less mass near zero than the corresponding priors.

- For s, the results are very similar to those in Figure 2.

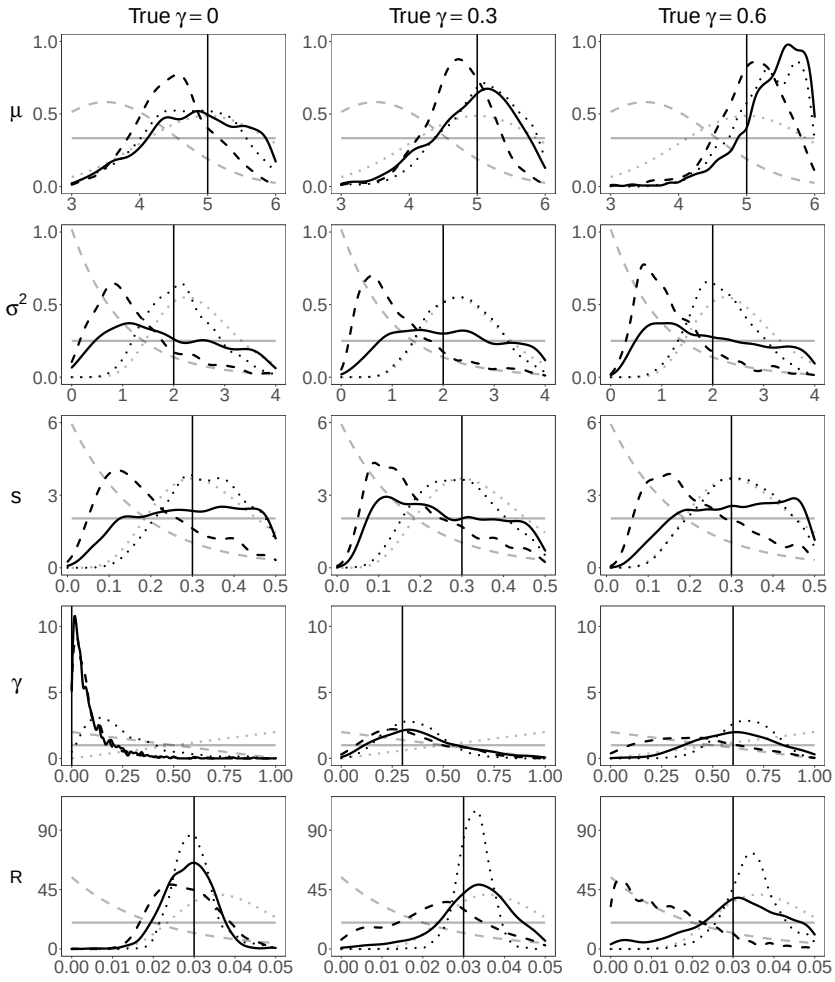

Fig. 2: Estimated marginal ABC posterior densities (black curves) when using different prior distributions (grey curves) for the parameters of the LGCP-Strauss model used for the first three point patterns in Figure 1. The vertical lines indicate the true parameter values. For each marginal ABC posterior distribution, the corresponding prior distribution is plotted using the same linetype (solid for P_1, dashed for P_2, and dotted for P_3). Each row represents a parameter (stated to the left of the row), and each column represents one of the three point patterns, as indicated by the true value of γ.

- For γ, the choice of prior seems to have little influence on the ABC posteriors, and overall the spread of the ABC posteriors seems to decrease slightly as σ^2 increases.
- For R, the prior seems to have some influence on the ABC posteriors and the spread of the ABC posteriors seems to be decreasing when σ^2 is increasing.

Figure 4 shows the means and medians of the ABC samples, where in all cases the posterior mean and median are close. Furthermore, we see the following:

- For μ, all the posterior means and medians for the different priors and point patterns look similar except in the case $\gamma = 0.6$ where they are somewhat higher than in the other cases. Overall, they are in fairly good agreement with the true values.
- For σ^2 and s, we again see that it is quite difficult to obtain much precision about these parameters from data.
- For γ, the prior has the smallest influence when the true value of γ is relatively low and the true value of σ^2 is relatively high, in which case the ABC posterior means and medians are also very close to the true value.
- For R, the prior has the smallest influence when the true value of γ is small. Considering the priors P_1 and P_3, the means and medians are generally close to the true value. For the prior P_2, the means and medians seem to become increasingly smaller than the true value of R as the true value of γ increases.

Overall, the ABC procedure seems to be most successful for estimating μ, γ and R, especially when the true value of γ is relatively small and the value of σ^2 is not too small. However, the success of the procedure will depend on the specific combination of the true parameters. Note that when fitting a Strauss process to a point pattern, Shirota and Gelfand (2017) first estimated R by maximum pseudo-likelihood and then used this value of R in their ABC procedure; in contrast, we found no need to fix R when fitting an LGCP-Strauss process with our ABC procedure.

Zhang (2004) demonstrated that some of the parameters in the Matérn model (which includes the exponential covariance function) may not be consistently identified in an increasing density infill asymptotics framework, but that the parameter σ^2/s may be consistently identified. This might explain why the ABC procedure is not so successful when it comes to identifying the scale and variance parameters of an exponential covariance function. Therefore, we made the same analysis as in Figure 2 using the prior P_1 when $s = 0.3$

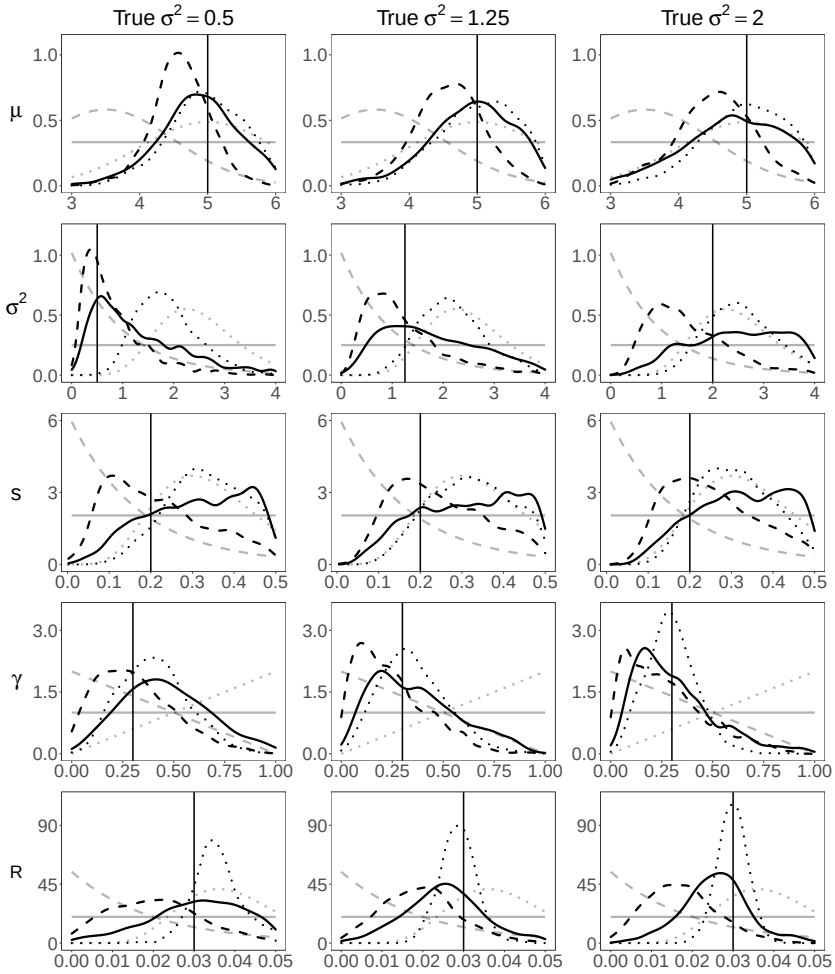

Fig. 3: Estimated marginal ABC posterior densities (black curves) when using different prior distributions (grey curves) for the parameters of the LGCP-Strauss model used for the last three point patterns in Figure 1. The vertical line indicates the true parameter value. For each posterior distribution, the corresponding prior distribution is plotted using the same linetype (solid for P_1, dashed for P_2, and dotted for P_3). Each row represents a parameter (stated to the left of the row), and each column represents one of the three point patterns, as indicated by the true value of σ^2.

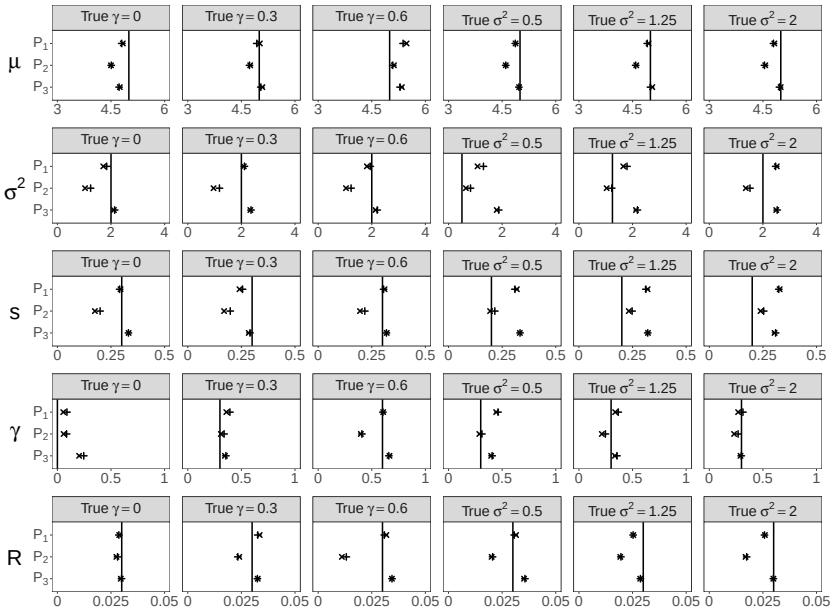

Fig. 4: Posterior means (indicated by +) and medians (indicated by x) of the ABC samples for the point patterns in Figure 1 (which are referred to by the true value of γ for the first three and the true value of σ^2 for the last three). Each row represent a parameter, which is stated to the left of the row. The prior distribution is stated on the y-axis and the x-axis correspond to the full range of each parameter.

is given. However, the posterior marginal distributions of the remaining parameters (not shown) looked very similar to those in Figure 2.

Table 1 summarises the estimated coefficients of the linear models fitted in the ABC procedure, cf. Algorithm 1. For each parameter, there are a total of 18 fitted linear models (one for each time the ABC procedure was run, that is, one for each combination of point pattern and prior). The table shows for each parameter the mean (over all 18 linear models) of the estimated coefficient for each summary statistic. In order to make the coefficients for different summary statistics comparable in this table, the linear models were fitted to a scaled version of the training data (the scaling was done by subtracting the mean and dividing by the standard deviation). As expected, $n_{\log}$ is most influential in the models for μ and σ^2. Of the summary statistics of the type $C_{\log \mathrm{var},q}$, $C_{\min,q}$, and $C_{\max,q}$, it appears that $C_{\log \mathrm{var},5}$ is the most important one. For all parameters, some values of the empirical L-function seem to have some influence in the linear models. The summary statistic $L_{\min}$ is most influential in the models for γ, which is in agreement with the fact that it describes the degree of regularity. Interestingly, $L_{\arg\min}$ does not appear to be very influential in the models related to R, in contrast to what

might be expected.

Summary statistic	γ	μ	R	s	σ^2
$n_{\log}$	-0.001	0.662	-0.168	0.05	0.337
$C_{\log \text{var},2}$	-0.043	0.005	-	0.046	-
$C_{\log \text{var},3}$	-	-	-	0.179	-
$C_{\log \text{var},4}$	0.035	-0.002	-0.046	0.046	0.088
$C_{\log \text{var},5}$	0.225	-0.133	-0.179	-0.017	0.171
$C_{\text{min},2}$	-	-0.008	-	-0.08	-0.009
$C_{\text{min},3}$	-0.004	0.001	-0.009	-0.036	-0.061
$C_{\text{min},4}$	-0.015	0.058	0.004	-	-0.029
$C_{\text{min},5}$	0.029	0.076	-0.033	0.038	-0.027
$C_{\text{max},2}$	0.008	-	-	0.006	-0.057
$C_{\text{max},3}$	0.023	0.025	-	-	-0.058
$C_{\text{max},4}$	0.031	-	-0.02	-	-0.077
$C_{\text{max},5}$	-	0.036	-	-	-0.146
L_2	0.093	-0.032	0.052	-0.067	0.054
L_3	0.064	-0.018	0.021	-0.039	0.053
L_4	0.055	-0.005	0.035	-0.044	0.014
L_5	0.059	-0.026	0.039	-0.024	0.023
L_6	0.091	-0.011	0.058	-0.024	0.007
L_7	0.117	-0.007	-0.035	-0.004	-0.013
L_8	0.072	-0.074	-0.145	-0.031	-0.075
L_9	0.041	-0.007	-0.201	-	-0.032
L_{10}	0.044	-0.058	-0.388	-0.044	-0.064
L_{11}	-	-	-0.008	-0.014	-
L_{12}	-0.039	-	-	-0.022	0.022
L_{13}	-0.036	-	-	-0.064	0.045
L_{14}	-0.045	-0.001	-	-0.055	0.06
L_{15}	-0.011	-	0.015	-0.14	0.078
L_{16}	-	-	-	-0.006	0.052
L_{17}	-0.027	-	0.103	-	0.023
L_{18}	-	-	0.009	-0.008	0.089
L_{19}	-0.01	-	-	-	0.062
L_{20}	-0.049	-	0.023	-	0.044
L_{21}	-0.023	-	-	-	0.008
L_{23}	-0.089	-	0.003	-	0.026
L_{24}	-	-	0.022	-	0.026
L_{26}	-0.011	-	0.056	-	-
L_{27}	-0.009	-	0.008	-	-
L_{28}	-	0.01	0.074	-	0.039
L_{29}	-0.01	0.027	-0.024	-	-
L_{30}	-	0.002	0.039	-	-
L_{31}	-0.036	-	-	-	-
L_{32}	-	-0.002	-	-	-
L_{33}	0.023	-	-	-	-
L_{34}	-0.043	-0.002	-0.001	-	-
L_{35}	-0.104	-	0.02	-	-
L_{37}	-	-	0.034	-	-
L_{38}	0.019	-	-	-	-
L_{39}	-	0.005	0.016	0.01	-
L_{40}	-0.021	-0.018	0.04	0.148	-
$L_{\arg\min}$	-0.087	-0.01	0.009	0.001	0.041
$L_{\min}$	0.12	0.027	0.017	0.085	-0.085
$L_{\max}$	-	-	0.035	0.161	-0.005

Table 1: Table of mean of estimated coefficients in the linear models fitted in the ABC procedure for each parameter. The data was scaled before the models were fitted in order to make the coefficients comparable. Summary statistics whose coefficients were zero in all models are not included.

We also investigated the estimated intercept (on the original scale) of the

linear models for each parameter, point pattern, and prior. According to the model specification in Algorithm 1, the intercept represents a linear approximation to the expected value of the parameter given the observed point pattern. Overall, these estimated intercepts were similar to the ABC posterior means in Figure 4 and are therefore not shown.

We now investigate how the ABC procedure for fitting an LGCP-Strauss process works when the data is generated from some of the special cases of this process. For this purpose, we simulated a realisation of an LGCP with parameters $\mu = 5$, $\sigma^2 = 2$, and $s = 0.3$, and a realisation of a Strauss process with parameters $\mu = 5$, $\gamma = 0.3$, and $R = 0.03$. Notice that when simulating under an LGCP, there is no need to employ the MCMC algorithm described at the beginning of Section 2. We used the faster method implemented in the function `rLGCP` from the package `spatstat` (Baddeley et al., 2015), which meant that we were able to run this ABC procedure for a single point pattern in about 40 minutes (again using 45 cores). We used the same ABC procedure as above with the specified uniform priors for fitting an LGCP-Strauss process to these point patterns and the posterior results can be seen in Figure 5. For the point pattern generated from an LGCP, the true value of μ seems to be identified well when fitting the LGCP-Strauss process. The posterior marginal distribution of γ is rather concentrated near 1, and a plot of the posterior samples of γ and R (not shown) shows that very small values of γ appear together with very small values of R. This indicates that the fitted LGCP-Strauss process is close to the special case of an LGCP, which is the true model. Again, it seems to be difficult to identify σ^2 and s.

For the point pattern generated from a Strauss process, the marginal ABC posterior distribution for σ^2 is very concentrated near zero, which is the true value. The true values of μ and R seems to be well identified, and the median and mean of the marginal posterior distribution of γ are quite close to the true value, but the maximum value of this ABC posterior distribution is somewhat smaller than the true value. For the Strauss process, s should be irrelevant, which is in agreement with the nearly uniform ABC posterior distribution for s.

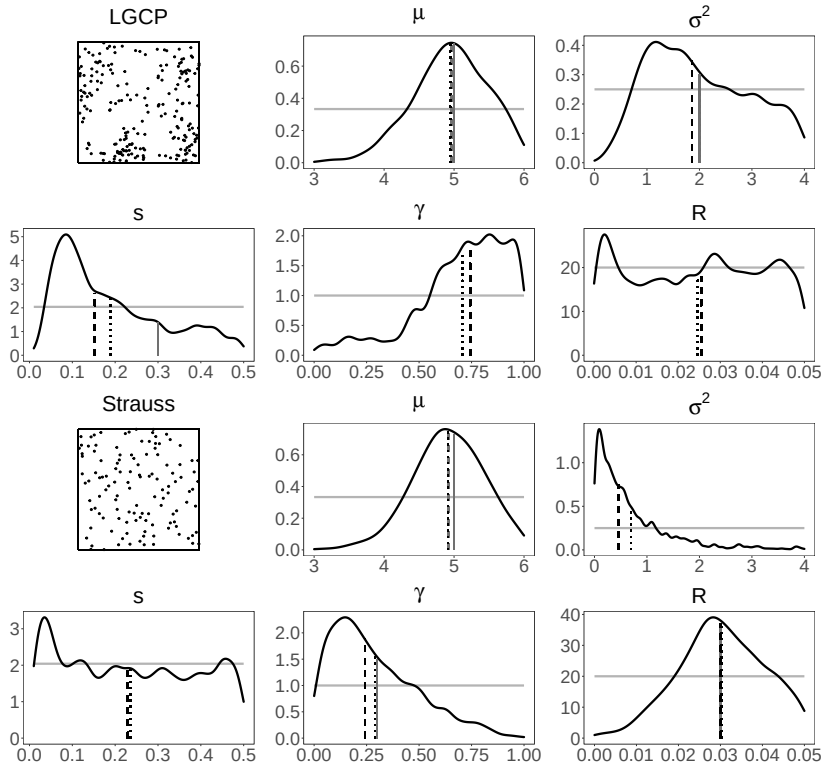

Fig. 5: Panels 2-6 and 8-12 show estimated marginal posterior distributions (black curves) and corresponding prior distributions (grey curves) for the parameters (as stated on the top) when fitting an LGCP-Strauss process to the realization of an LGCP in panel 1 (corresponding to panels 2-6) and the realization of a Strauss process in panel 7 (corresponding to panels 8-12). The point processes are defined on the unit square. The dashed and dotted vertical lines indicate the medians and means, respectively; the solid vertical lines indicate the true values, when relevant.

4.3 Model checking and comparison

We are interested in whether the point patterns in Figure 1 can be distinguished from realisations of an LGCP and a Strauss process, so for comparison we also fitted an LGCP and a Strauss process to each point pattern, using the ABC procedure in Algorithm 1. We used the same summary statistics as for the LGCP-Strauss process and the prior P_1 specified in 4.1 on the relevant parameters (that is, the parameters μ, σ^2, and s when fitting the LGCP, and the parameters μ, γ, and R when fitting the Strauss process). Again, when simulating under an LGCP, we used the faster method implemented in `spatstat`.

For model checking and comparison we first suggest to make global envelope tests based on posterior predictions as follows. For each ABC realisation

of θ, a realisation $\mathbf{x}$ of the process in question given θ is simulated. For each $\mathbf{x}$, a functional summary statistic is estimated. These empirical curves are then used to construct global envelopes and corresponding tests based on extreme rank lengths (Myllymäki et al., 2017, Mrkvička et al., 2020, note that we only used 1 000 simulations instead of the recommended 2499, because the ABC procedure is rather time consuming). The R-package `GET` (Myllymäki et al., 2017) was used for this purpose.

In order to compare the fitted LGCP-Strauss, LGCP, and Strauss process models, we used 95% global envelopes based on posterior predictions and the empirical L- and J-function, with $J(r) = (1 - G(r))/(1 - F(r))$ where F is the empty space function and G is the nearest-neighbour distribution function (see van Lieshout and Baddeley, 1996). We also tried to use the F- and G-functions for model validation but these functional summary statistics were unable to distinguish between the models (just reflecting the well-known fact that the J-function contains other information than each of the F- and G-functions).

Figures 6 and 7 show 95% combined global envelopes for the L- and J-function, meaning that, under the LGCP-Strauss process, the probability that both empirical curves are within their respective envelopes is approximately 95%. To combine the envelopes we have used the two-step combining procedure described in Myllymäki and Mrkvička (2019). The posterior predictions of the LGCP-Strauss processes are for the ABC samples from Section 4.2 obtained with the prior P_1. Note that the J-function can only be estimated reliably for all simulations for r-values in a relatively small interval, whereas the L-function can be estimated reliably on a larger interval.

In all cases, the p-values of the global envelope tests are highest in the situation of the LGCP-Strauss process, which may indicate that they provide the best fit to data. Considering Figure 6, the LGCP is rejected in the cases where $\gamma = 0$ and $\gamma = 0.3$ because the empirical J-functions in these cases are above the 95% global envelopes at small inter-point distances. This indicates that the point patterns are more regular at small inter-point distances than what would be expected under the fitted LGCPs. For the case $\gamma = 0.6$ (the case with weakest inhibition), the LGCP cannot be rejected. Notice that the p-values of these tests are increasing as γ increases which is in agreement with the fact that the LGCP-Strauss process approaches the special case of an LGCP. Considering Figure 7, the LGCP is only rejected in the case where $\sigma^2 = 1.25$, but the p-values are also rather small in the other two situations. In all three situations, the behaviour of the empirical J-function indicates that the point patterns are somewhat more regular than what is typical under the fitted LGCP.

The Strauss process model is rejected in all six cases because the empirical L-function clearly shows that the point patterns are more clustered at moderate to large inter-point distances than what can be modelled with a Strauss

process. In the cases $\gamma = 0.3$ in Figure 6 and $\sigma^2 = 2$ in Figure 7, the empirical J-functions also show this, but for the remaining cases, the J-function is contained completely within the envelopes.

Overall, it appears that the J-function is best at criticizing the LGCP and the L-function is best at criticizing the Strauss process. The later may have something to do with the fact that the J-function can only be estimated on a relatively small interval. So, it is less likely to capture the aggregation, which happens on a larger scale, than the L-function which can be estimated on a bigger interval. When we use the L-function for model validation we keep in mind that it was also used in the ABC procedure which might lead us to conclude that the model fits better to data than it actually does.

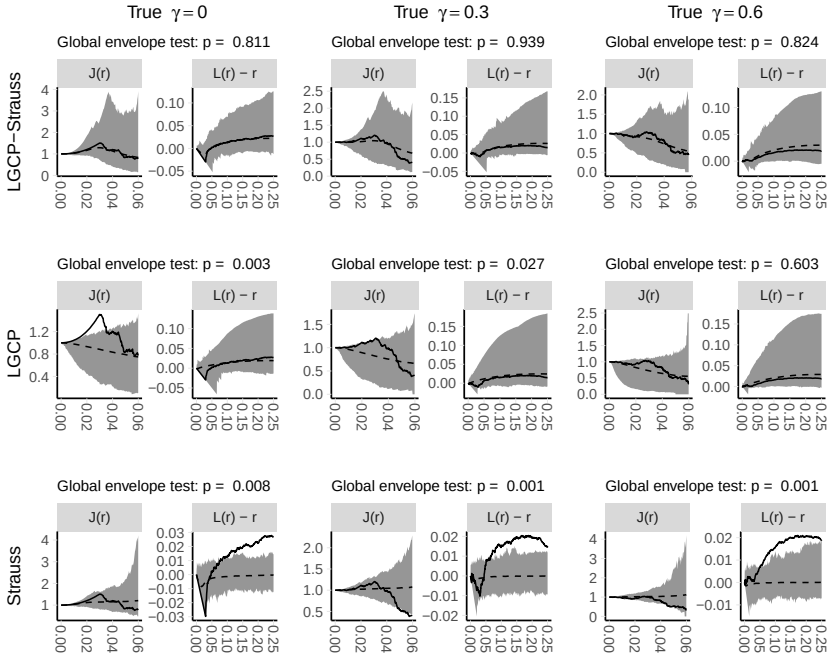

Fig. 6: Combined global envelopes based on the empirical J- and L-function for LGCP-Strauss, LGCP, and Strauss processes fitted with ABC to the three point patterns in Figure 1. The choice of the fitted model is stated to the left of each row and each column represents a different point pattern, as indicated by the true value of γ. The solid curves are the empirical functional summary statistics for the observed point patterns and the dashed curves are the means obtained from 1 000 posterior predictions. Each shaded area indicates a 95% global envelope based on the extreme rank length. At the top of each plot, the p-value of the corresponding global envelope test is stated.

The global envelope tests are mainly a method for model validation, but they may be used for model comparison by comparing p-values and concluding that the model with the highest p-value provides the best fit. However, it

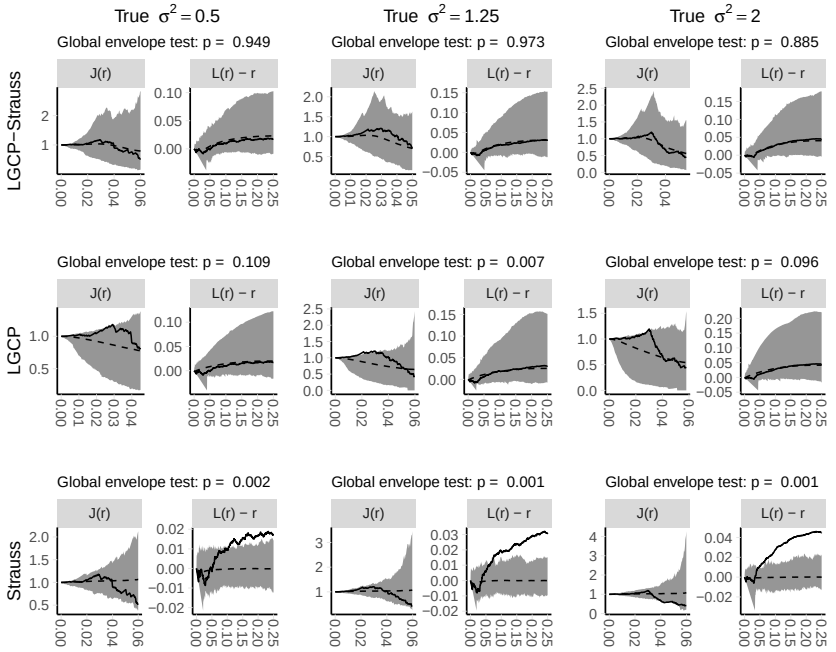

Fig. 7: Combined global envelopes based on the empirical J- and L-function for LGCP-Strauss, LGCP, and Strauss processes fitted with ABC to the last three point patterns in Figure 1. The choice of the fitted model is stated to the left of each row and each column represents a different point pattern, as indicated by the true value of σ^2. The solid curves are the empirical functional summary statistics for the observed point patterns and the dashed curves are the means obtained from 1 000 posterior predictions. Each shaded area indicates a 95% global envelope based on the extreme rank length. At the top of each plot, the p-value of the corresponding global envelope test is stated.

should be kept in mind that a higher p-value may be a result of overfitting.

In order to investigate this, we also fitted an LGCP to the first point pattern in Figure 5 and a Strauss process model to the second point pattern in Figure 5 and compare them to the fitted LGCP-Strauss process models (the global envelopes are not shown). For the realisation of an LGCP, the p-values of the 95% combined global envelope test for the fitted LGCP-Strauss and LGCP were 0.933 and 0.766, respectively. Since the data is generated from an LGCP, both models should fit the data equally well, so the higher p-value for the LGCP-Strauss process is probably a result of the fact that it is overfitting. For the realisation of a Strauss process model, the p-values of the 95% combined global envelope test for the fitted LGCP-Strauss and Strauss process were 0.46 and 0.766, respectively. In this example, the p-values do not reveal the fact that the LGCP-Strauss process is overfitting.

We also consider an ABC method for model comparison, using the method

of ABC model choice via random forests (ABC-RF) and the corresponding R-package `abcrf` from Pudlo et al. (2015). In short, the idea is to make a number of prior predictions (including a model index); calculate summary statistics for these; create a reference table with model indices and calculated summary statistics; and finally use this table to train a random forest classifier for predicting the model from the summary statistics. This classifier is then used on the summary statistics of the observed data to choose a model. Pudlo et al. (2015) also described how to approximate the posterior probability of the chosen model. According to Pudlo et al. (2015), the method is robust to the number and choice of summary statistics.

We used this method for the point patterns in Figures 1 and 5, using a uniform prior on the three models in consideration (LGCP-Strauss, LGCP, and Strauss), the uniform priors of P_1 for the relevant parameters, the summary statistics from Section 3.2, and 30 000 prior predictions (whereof eight were afterwards excluded because some summary statistics yielded infinite values). The number of prior predictions are in agreement with the recommendations in Pudlo et al. (2015). We used the default settings from the `abcrf` package for the remaining choices concerning the ABC-RF method.

The results are in Table 2. The true model is chosen in all cases, and the approximate posterior probabilities are very high in the cases where the true model is LGCP-Strauss or Strauss. When the true model is an LGCP, the approximate posterior probability is somewhat smaller.

Point pattern	Selected model	Approximate posterior probability
True $\gamma = 0$	LGCP-Strauss	0.97
True $\gamma = 0.3$	LGCP-Strauss	0.99
True $\gamma = 0.6$	LGCP-Strauss	0.88
True $\sigma^2 = 0.5$	LGCP-Strauss	0.96
True $\sigma^2 = 1.25$	LGCP-Strauss	0.98
True $\sigma^2 = 2$	LGCP-Strauss	0.99
True LGCP	LGCP	0.65
True Strauss	Strauss	0.94

Table 2: Selected model and its approximate posterior probability when using ABC-RF for the point patterns in Figures 1 and 5.

Note that this method is intended for choosing between different types of models. Whether a model of the chosen type actually fits the data is assessed by the global envelope test. So, a model choice method such as ABC-RF may then be particularly useful for choosing between different types of models which according to the global envelopes all fit the data.

5 Data example

The first panel in Figure 8 shows the locations of 256 oak trees which suffer from frost shake (frost shake refers to cracks in the trunk of the tree) in a 125×188 m rectangular region of Allogny in France. This data set is part of the Allogony data set from the R-package `ads` (Pélissier and Goreaud, 2015).

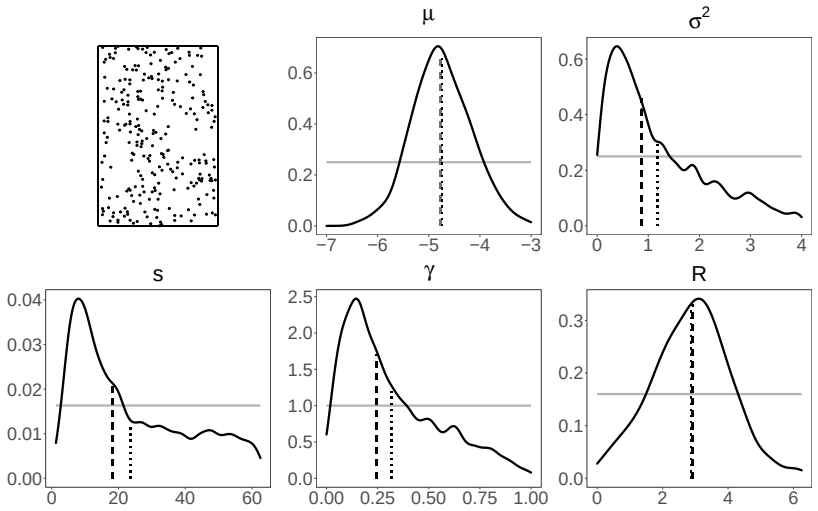

Fig. 8: The first panel shows the frost shake oak point pattern dataset where the observation window is a 125×188 m rectangle. The other panels show the estimated marginal ABC posterior distributions (black curves) and the prior distributions (grey curves) for the five parameters, with each parameter stated at the top of each plot. The dashed and dotted lines indicate the medians and means, respectively.

We used Algorithm 1 on this oak data set. Here, independent uniform prior distributions are chosen for μ on the interval $(-7, -3)$, σ^2 on $(0, 4)$, s on $(1.25, 62.5)$, γ on $(0, 1)$, and R on $(0, 6.25)$. Notice that the observation window for the oak data is much larger than the ones in Section 4.1, and the prior distributions are chosen to take this into account. Furthermore, when calculating the summary statistics for the ABC procedure, $W_{i,j}$, $i, j = 1, \ldots, q$, are now rectangular sets of the same size (see Section 3.2). Trace plots as those in Figure 10 (supplied in an appendix) suggested that 20 000 iterations of the MCMC algorithm is a sufficient burn-in for this example. Again, a pilot sample of 10 000 simulations was used and the resulting ABC posterior sample consists of 1 000 draws from the approximate posterior distribution.

The marginal posterior distributions, which are estimated from the ABC sample, can be seen in Figure 8. They are all clearly different from their uniform priors. The posterior distributions of μ and R look approximately normal, whilst the posterior distributions of σ^2, s, and γ are right skew. Note

that the posterior distribution of γ indicates strong repulsion between the points. The posterior distribution of σ^2, particularly its heavy tail, suggests some aggregation among the splited oaks.

We now consider the techniques for model checking from Section 4.3. The first plot in Figure 9 shows 95% combined global envelopes for the fitted LGCP-Strauss process as described in Section 4.3. The overall behaviour of the observed point pattern seems to be captured well by the LGCP-Strauss process, and the p-value is very high. For comparison, Figure 9 also shows the corresponding 95% envelopes for an LGCP and a Strauss process model fitted with the ABC procedure in Algorithm 1. The combined global envelopes indicate that the LGCP model provides a poor fit to data, but the Strauss process model also fits well. However, the p-value is lower than the corresponding p-value for the LGCP-Strauss process, indicating that the later may provide a better fit.

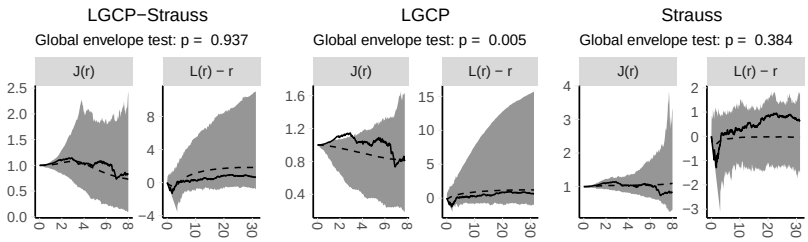

Fig. 9: Combined global envelopes based on the empirical J- and L-function for different fitted models to the splited oak point pattern (as indicated at the top of each plot). The solid curves correspond to the splited oak point pattern and the dashed curves are the means obtained from 1 000 posterior predictions. The shaded area indicate a 95% global envelope based on the extreme rank length. At the top of each plot, the p-value of the corresponding global envelope test is stated.

When we used the method of ABC-RF, the selected model is an LGCP-Strauss process and the approximate posterior probability is 0.7 showing a relatively high confidence in the chosen model.

All things considered, it seems that the fitted LGCP-Strauss process captures the behaviour of the splited oak point pattern very well, that it provides a much better fit than the LGCP process, and that it provides a somewhat better fit than the Strauss process.

6 Summary and future work

We have proposed a novel spatial point process model which enables capturing of regularity through pairwise interactions and aggregation through a Gaussian process realization. This doubly stochastic spatial point process

generalizes both the customary log Gaussian Cox process and the customary Gibbs process. Because the likelihood is intractable for this model we have developed model fitting through an ABC method. We have provided both simulation investigation and a real data application in order to reveal the behaviour of process realizations and also our ability to fit the model and do full inference for given point pattern realizations.

Future work may compare the quality of ABC inference with more traditional MCMC based posterior inference which indeed will be much more time consuming. By this, we mean using a missing data MCMC approach for the case of the LGCP (which is then included into the posterior) or the ancillary variable method by Møller et al. (2006) (see also Murray et al., 2006) for the Strauss process. However, this will be time consuming, especially when we have to make perfect simulations of the Strauss process for the ancillary variable method. Further future opportunities may consider inhomogeneous point patterns (e.g. by including covariate information into the mean function of the Gaussian process), and marked point patterns or so-called multitype versions of our model (see e.g. Møller and Waagepetersen, 2004). Such multi-type modelling may allow attraction or inhibition within types but also introduce attraction or inhibition between types. A different direction would consider space-time versions. That is, a realization of the process is seen as a spatial point pattern by *integrating* over a window of time.

Acknowledgements

The research of the first two authors was supported by The Danish Council for Independent Research | Natural Sciences, grant DFF – 7014-00074 'Statistics for point processes in space and beyond'. The second author was also supported by the 'Centre for Stochastic Geometry and Advanced Bioimaging', funded by grant 8721 from the Villum Foundation.

A Trace plots for accessing the burn-in for the simulation algorithm

Figure 10 shows trace plots of the number of points and R-close pairs for the MCMC algorithm when simulating an LGCP-Strauss process for different draws of the parameter vector θ from it's prior distribution P_1 which is described in Section 4.1. For each prior sample of θ, a realisation $\mathbf{z}$ of the GRF was simulated, and the MCMC algorithm was used to simulate the LGCP-Strauss process given $\mathbf{Z} = \mathbf{z}$. This analysis was used to choose an appropriate burn-in in Section 4.

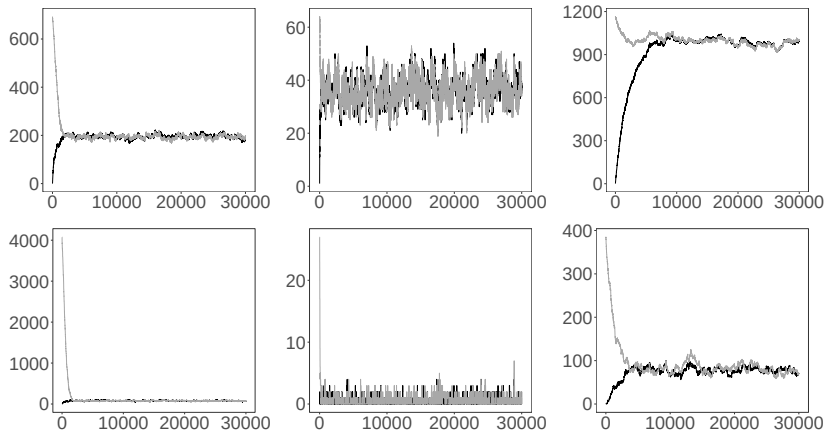

Fig. 10: Trace plots of the number of points (top) and R-close pairs (bottom) for 30 000 iterations of the MCMC algorithm for simulating an LGCP-Strauss process on the unit square with parameter vector θ drawn from the prior distribution P_1. Each column of images represent a different sample of θ and a corresponding realization $\mathbf{z} = \{z(u)\}_{u \in W}$ of the GRF $\mathbf{Z}$. For each column, the MCMC algorithm was initiated at the empty point pattern (black curves) or a realisation of an inhomogeneous Poisson process with intensity function $\exp(z(u))$ (grey curves).

References

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

Baddeley, A., Turner, R., Mateu, J., and Bevan, A. (2013). Hybrids of Gibbs point process models and their implementation. *Journal of Statistical Software*, 55:1–43.

Beaumont, M. A. (2010). Approximate Bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics*, 41:379–406.

Bernton, E., Jacob, P. E., Gerber, M., and Robert, C. P. (2019). Approximate Bayesian computation with the Wasserstein distance. *arXiv preprint arXiv:1905.03747*.

Berthelsen, K. K. and Møller, J. (2008). Non-parametric Bayesian inference for inhomogeneous Markov point processes. *Australian & New Zealand Journal of Statistics*, 50:257–272.

Cox, D. R. (1955). Some statistical methods connected with series of events. *Journal of the Royal Statistical Society: Series B (Methodological)*, 17:129–157.

Fearnhead, P. and Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian

computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74:419–474.

Geyer, C. J. and Møller, J. (1994). Simulation procedures and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, 21:359–373.

Goldstein, J., Haran, M., Simeonov, I., Fricks, J., and Chiaromonte, F. (2015). An attraction–repulsion point process model for respiratory syncytial virus infections. *Biometrics*, 71:376–385.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC, Boca Raton.

Jiang, B., Wu, T.-Y., and Wong, W. H. (2018). Approximate Bayesian computation with Kullback-Leibler divergence as data discrepancy. In *International Conference on Artificial Intelligence and Statistics*, pages 1711–1721.

Kelly, F. P. and Ripley, B. D. (1976). A note on Strauss's model for clustering. *Biometrika*, 63:357–360.

Kendall, W. S. and Møller, J. (2000). Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Advances in Applied Probability*, 32:844–865.

Lavancier, F. and Møller, J. (2016). Modelling aggregation on the large scale and regularity on the small scale in spatial point pattern datasets. *Scandinavian Journal of Statistics*, 43:587–609.

Meinshausen, N. (2007). Relaxed Lasso. *Computational Statistics & Data Analysis*, 52:374–393.

Møller, J., Pettitt, A. N., Berthelsen, K. K., and Reeves, R. W. (2006). An efficient markov chain monte carlo method for distributions with intractable normalising constants. *Biometrika*, 93:451–458.

Møller, J. and Waagepetersen, R. P. (2017). Some recent developments in statistics for spatial point patterns. *Annual Review of Statistics and Its Application*, 4:317–342.

Mrkvička, T., Myllymäki, M., Jilik, M., and Hahn, U. (2020). A one-way ANOVA test for functional data with graphical interpretation. *Kybernetika*, 56:432–458.

Murray, I., Ghahramani, Z., and MacKay, D. J. C. (2006). MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence*, page 359–366, AUAI Press.

Myllymäki, M. and Mrkvička, T. (2019). GET: Global envelopes in R. arXiv preprint arXiv:1911.06583.

Myllymäki, M., Mrkvička, T., Grabarnik, P., Seijo, H., and Hahn, U. (2017). Global envelope tests for spatial processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79:381–404.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25:451–482.

Møller, J. and Waagepetersen, R. P. (2004). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman and Hall/CRC, Boca Raton.

Park, M., Jitkrittum, W., and Sejdinovic, D. (2016). K2-abc: Approximate Bayesian computation with kernel embeddings.

Pélissier, R. and Goreaud, F. (2015). ads package for R: A fast unbiased implementation of the *K*-function family for studying spatial point patterns in irregular-shaped sampling windows. *Journal of Statistical Software*, 63:1–18.

Pudlo, P., Marin, J.-M., Estoup, A., Cornuet, J.-M., Gautier, M., and Robert, C. P. (2015). Reliable ABC model choice via random forests. *Bioinformatics*, 32:859–866.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ripley, B. D. (1976). The second-order analysis of stationary point processes. *Journal of Applied Probability*, 13:255–266.

Ripley, B. D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39:172–192.

Schlather, M. (1999). An introduction to positive definite functions and to unconditional simulation of random fields. Technical Report st 99-10, Department of Mathematics and Statistics, Lancaster University.

Schlather, M., Malinowski, A., Menck, P. J., Oesting, M., and Strokorb, K. (2015). Analysis, simulation and prediction of multivariate random fields with package RandomFields. *Journal of Statistical Software*, 63:1–25.

Schlather, M., Malinowski, A., Oesting, M., Boecker, D., Strokorb, K., Engelke, S., Martini, J., Ballani, F., Moreva, O., Auel, J., Menck, P. J., Gross, S., Ober, U., Ribeiro, P., Ripley, B. D., Singleton, R., Pfaff, B., and R Core Team (2019). RandomFields: Simulation and analysis of random fields. R package version 3.3.6.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53:683–690.

Shirota, S. and Gelfand, A. E. (2017). Approximate Bayesian computation and model assessment for repulsive spatial point processes. *Journal of Computational and Graphical Statistics*, 26:646–657.

Soubeyrand, S., Carpentier, F., Guiton, F., and Klein, E. K. (2013). Approximate Bayesian computation with functional statistics. *Statistical Applications in Genetics and Molecular Biology*, 12:17–37.

Stoica, R. S., Philippe, A., Gregori, P., and Mateu, J. (2017). ABC Shadow algorithm: A tool for statistical analysis of spatial patterns. *Statistics and Computing*, 27:1225–1238.

Strauss, D. J. (1975). A model for clustering. *Biometrika*, 62:467–475.

van Lieshout, M. N. M. and Baddeley, A. J. (1996). A nonparametric measure of spatial interaction in point patterns. *Statistica Neerlandica*, 50:344–361.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Zhang, H. (2004). Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association*, 99:250–261.

Paper B

Cellular 3D-reconstruction and analysis in the human cerebral cortex using automatic serial sections

Nick Y. Larsen, Xixia Li, Xueke Tan, Gang Ji, Jing Lin, Grazyna Rajkowska, Jesper Møller, Ninna Vihrs, Jon Sporring, Fei Sun, and Jens R. Nyengaard

The paper has been published in
Communications Biology (2021) **4**, 1030

The layout has been revised.

Abstract

Techniques involving three-dimensional (3D) tissue structure reconstruction and analysis provide a better understanding of changes in molecules and function. We have developed AutoCUTS-LM, an automated system that allows the latest advances in 3D tissue reconstruction and cellular analysis developments using light microscopy on various tissues, including archived tissue. The workflow in this paper involved advanced tissue sampling methods of the human cerebral cortex, an automated serial section collection system, digital tissue library, cell detection using convolution neural network, 3D cell reconstruction, and advanced analysis. Our results demonstrated the detailed structure of pyramidal cells (number, volume, diameter, sphericity and orientation) and their 3D spatial arrangement organized in a columnar structure. Our pipeline for these combined techniques provides a detailed analysis of tissue and cells in biology and pathology.

1 Introduction

Life science aims at a better understanding of multiple biological functions, such as healthy organ development with cellular proliferation, migration and organization, tumour formation and general pathology. Several techniques have been developed to study biological structure in 3D like serial blockface scanning Electron Microscopy (EM), focused ion beam scanning EM, serial-section transmission EM, Automatic Tape-collecting Ultra Microtome-scanning EM, and many types of light microscopy with or without tissue clearing (Denk et al., 2004, Escovitz et al., 1975, Hayworth et al., 2014, Burel et al., 2018). In life science, EM is widely used to explore the subcellular components, which are several orders of magnitude smaller than the spatial neuron networks. Clearing techniques can be very helpful in attempting to explain neuron networks in greater brain volumes, such as the adult mouse brain, in an entire state without disassembly (Ariel, 2017, Chung and Deisseroth, 2013). However, some of the disadvantages for tissue clearance are that immunostaining of archival tissues is usually complicated as a result of the antigen masking due to formaldehyde protein crosslinking (Lai et al., 2018). Furthermore, practicing immunostaining with tissue clearing remains difficult in human tissues due to factors such as inadequate antibody penetration depth, physicochemical properties, and tissue composition (Lai et al., 2018, Marx, 2016).

A typical human neuron has thousands of complex connections with neighboring neurons, which is essential for normal function, yet the organization of these neurons is still under debate (Hawkins and Ahmad, 2016). The cellular organization in the human neocortex has been described as a local network of vertical columns containing neurons. Neurons with similar

functions are grouped together and according to different theories, these cortical columns may contain smaller columns known as minicolumns, which are the smallest unit capable of processing information in the cerebral cortex (Mountcastle, 1957, Molnár and Rockland, 2020). Cortical columns are radially oriented cell bodies that span through the laminar pattern perpendicular to the pial surface and can be seen using regular Nissl preparations or another cell body-revealing histological procedures. The introduction of minicolumns was in response to studies of the patterning of apical dendrites of pyramidal cells with somata situated in layers II, III, and V (Fleischhauer et al., 1972, Peters and Walsh, 1972). Studies have attempted to characterize and analyze the morphology of minicolumns with a 2D computerized method designed to detect subtle differences among patient groups such as schizophrenia, autism, and Alzheimer (Chance et al., 2008, Casanova et al., 2003, McKavanagh et al., 2015, Raghanti, 2010). As a result, much of our understanding of cellular organization is focused on 2D histological images, which could potentially misrepresent biological structures and malpositioning of cells in 3D-space.

This paper aims to create a method that is accessible to the broader science community and analyze 3D tissue organization through the use of archival tissue to make detailed inferences about pathologies. In the present study, we developed Automatic Collector of Ultrathin Sections for Light microscopy (AutoCUTS-LM) to determine the neuronal cell morphology and their spatial organization in 3D-space of archived tissue. This is accomplished by modifying and adjusting the original AutoCUTS, which was designed for scanning EM array tomography (Shen et al., 2019, Liu et al., 2019, Li et al., 2017), to image archival human brain tissues($\sim$20 years) in layer III of Brodmann Area 46 (BA46). BA46 was chosen since it involves working memory, attention and has been the subject of studies related to mental disorders like schizophrenia and depression (Fuster, 1997, Selemon et al., 1998, Cruz et al., 2004, Dean et al., 2012, Udawela et al., 2017, Gibbons et al., 2009, Trojak et al., 2014, Dean et al., 2014). Myelinated axon bundles are potentially cortical efferents that originated in layer II/III pyramidal cells as these bundles descend to the white matter (Peters and Sethares, 1996). In both corticocortical and thalamocortical circuitry, layer III pyramidal neurons play a key role and have been suggested to be most affected by these disorders as it is the most prominent layer in BA46 (Levitt et al., 1993, Rajkowska et al., 1998, Glantz and Lewis, 2000).

Here, we report the applied technical workflow that is able to uncover morphological properties of pyramidal neurons in human brain autopsy tissue: First, we identified BA46 and applied advanced sampling procedures of biopsies. Next, we embedded biopsies in resin, AutoCUTS-LM sliced them automatically into semi-thin sections (300 and 400 *nm*), and on tape. Sections were stained and mounted on glass slides, and a library for each sample was

organized. The digitalization of sections was stored, aligned, and stacked as a volumetric structure. Then, we detected neurons using the UNetDense architecture. Finally, the 3D spatial arrangement and structural parameters of pyramidal neurons in layer III of BA46 were analyzed in three human brains applying recently developed methods.
Our findings present valuable insight into neuronal morphology and architecture by characterizing pyramidal neurons in 3D from old archived human brain tissue. We discovered that pyramidal cells are not randomly distributed but are clustered in small columnar structures, which may be relevant for understanding the formation and function of the cortical network.

2 Results

2.1 Sampling strategy and preparation of tissue

A block of tissue from the dorsolateral prefrontal cortex that contained all of BA46 was removed from each brain (Rajkowska and Goldman-Rakic, 1995b), see **Fig.1**. We used the Matlab method we developed to delineate BA46 and

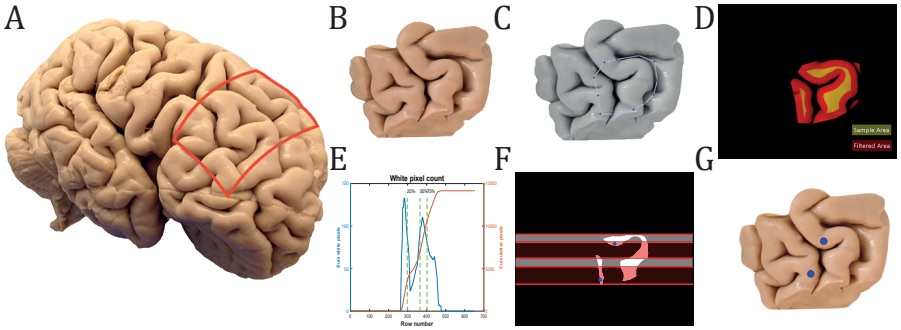

Fig. 1: Sampling of biopsies at BA46. **(A)** Formalin-fixed tissue from three human brains were selected from the brain collection at Aarhus University Hospital. The red box marks the excised area of the tissue. **(B)** Fixed coronal block of tissue containing BA46. **(C)** Manual delineation of BA46 performed by MATLAB. **(D)** Infused image between global threshold image and sample area. The filtered surface of a coronal block of tissue containing BA46 was marked with a red and yellow map that shows the available sample area. **(E)** Summation of all white pixels for each row of the binary picture of the sample area. The 1st, 2nd and 3rd quarter of pixels were marked with a green dashed line. **(F)** The two biopsies can only be sampled in either the 1st and 3rd quarter(white area) or the 2nd and 4th quarter(red area). In this case, random points in the 2nd and 4th quarters, marked with blue dots, were chosen by the algorithm. **(G)** The two chosen biopsies are marked with blue dots on the original block of tissue from (B).

sample two biopsies on a highly complex surface like the brain, with the second biopsy kept as a reserve. The cortical columns of neurons in the cortex could be successively extracted by a biopsy perpendicularly to the cortical

surface; therefore, only neurons from gyri and not sulci were analyzed, illustrated with the yellow color in **Fig.1D**. The sampling area was divided into four quarters to avoid any overlap or adjacent biopsies, see **Fig.1E-F**.
The brain tissue samples were obtained by using a biopsy punch positioned on the two sampling points with a diameter of 1.5 *mm* and a depth around 3-5 *mm*. This meant that a sample included all six layers of the cerebral cortex.

The tissue samples were fixed in resin and not stained with osmium since osmium fixation reduces the contrast of sections during light microscopy, see **Fig.S1**.

Our strategy to successfully find neurons in the supragranular layers (I-III) and layer IV, Region Of Interest (ROI), requires that the sample be positioned in such a manner that all six layers in the neocortex appear in every section during the AutoCuts-LM procedure. Consequently, the biopsy was placed at the bottom of the embedding form such that the pial surface was perpendicular to the cutting direction, see **Fig.S2**.
This orientation decreased the number of sections approximately 2-3 times relative to alternate orientations, thus shortened the time spent collecting and capturing images with the microscope. We used a light microscope to locate and measure the ROI by first examining and staining the outermost section of the block to delimit the area for collecting semi-thin sections (100-500 *nm*). We precisely trimmed the excess embedding resin around the sample to avoid wrinkles while cutting and removed layer V-VI, which resulted in an ROI of around 1.5 mm^2.

2.2 Collection and preparation of serial brain sections on tape

We have modified and adapted the automatic serial section technique, which was originally developed for scanning EM (Shen et al., 2019, Liu et al., 2019, Li et al., 2017), so it operates for light microscopy. The main changes were the replacement of the double-sided conductive tape with a plasma cleaned transparent tape and mounting tape strips containing the sections on glass slides instead of on a silicon wafer. The transparent polyester tape was 7 *mm* wide and was put through plasma treatment, which influenced the hydrophilicity of the tape. The hydrophilicity was found to not only reduce wrinkles of the brain sections produced during the collection process (Li et al., 2017), but also made the sections more adhesive to the tape. Thus, the sections would later stick to the tape during collection and staining.

Automated collection of the resin-embedded material was achieved using a combined ultramicrotome attached to a custom tape collection device (AutoCUTS-LM). We collected around 2400 serial sections with a total tissue depth of about 0.7 *mm* for each subject. The three human subjects' cutting thickness was chosen to be 400 *nm* for subject 1 and 300 *nm* for subjects 2 and 3 (sectional area was about 1.5 mm^2, and the total volume was approximately

1 mm^3). Sections were cut continuously with a diamond knife (Diatome, Switzerland) with an indoor humidity around 85%, see **Fig.2A**. The pulling

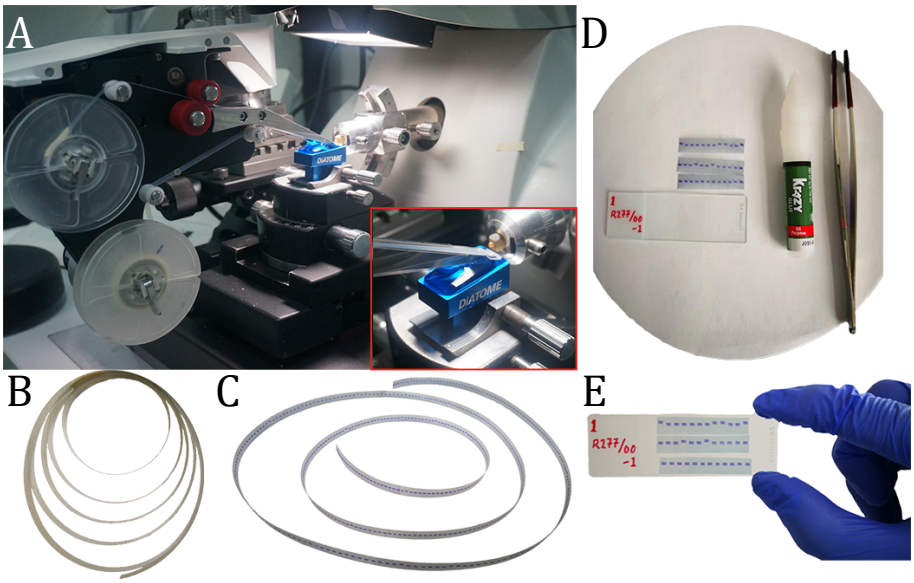

Fig. 2: Image of the AutoCUTS-LM and sample preparation on tape.(A) The supply-reel of the AutoCUTS-LM contains a transparent plastic tape that collects sections from the knife boat to the final take-up reel. (Red inset) Close-up view of the tape conveyor belt positioned in front of the knife boat with the mounted sample.**(B)** Collected sections on the transparent tape before staining. **(C)** Sections have been stained with toluidine blue. **(D)** Setup to glue the sections on the microscopic glass slide. **(E)** Stained sections glued on a microscopic slide and ready for image acquisition.

motion from the collecting tape moved the sections from the water onto the tape's surface, and its adhesiveness affected how flat these sections lay on the tape. Around 800 sections were collected per hour with our settings. Hence, we used less than three hours to cut a tissue sample that is around 0.7 mm thick. It is important to notice that the ultramicrotome calibrates itself after sectioning the maximum useful range of 200 μm and had to be manually reset. In our case, a tissue sample to a depth of 0.7 mm could be sectioned with only three interruptions, and it was possible to move the sample to a different knife-edge, as dulling of the knife affects the section quality. Collecting thousands of sections was possible without any loss of tissue. However, we observed different technical and environmental factors that could generate wrinkles and disfigure the sections during cutting. The indoor humidity level was one of the main factors (see Methods).

Following section-collection, we chose to stain our sections with toluidine blue, since it interacts with most cells in the brain (both neurons and glial

cells) and is thus excellent for revealing the neuronal patterns. However, to assess other biological results, methods such as immunolabeling or EM may also be used with the same AutoCUTS strategy, see **Fig.S3**. The spools holding the tape with attached sections were dried overnight in an oven at 50°C. Sections were then stained with toluidine blue, and the tape with attached sections was cut into three consecutive strips with approximately 60 serial sections and glued onto a standard microscope slide **(Fig.2B-E)**. Each sample resulted in a section library of approximately 40 glass slides, which were digitalized.

2.3 Data acquisition

Digital images were acquired using the Apiro Versa 200 platform from Leica. The scanning speed was approximately 15 min per glass slide and required about 10 hours to complete a section library of glass slides. We only sampled every second and third section because a distance of 800 (subject 1) and 900 (subject 2 and 3) *nm* between sections was considered sufficient for 3D-reconstruction of pyramidal cells, which have an average somal diameter of approximately 13 μm (Rajkowska and Goldman-Rakic, 1995a), see **Fig.3A**. The microscope included the commercial software Aperio ImageScope (Leica Biosystems Imaging, Inc., USA), which visualized the whole microscope slide image with high resolution. However, the user had to extract a region manually before an image could be exported as a TIF file. The uncompressed images were then automatically processed, ordered, and exported as individual images of each section using a MATLAB algorithm we developed, as shown in **Fig.3B**. These individual images were then aligned and stacked by sequential image registration using our custom MATLAB scripts (Rosenfeld and Kak, 1982). Next, the stack of images was prepared for further analysis by cropping tissue regions, see **Fig.3C**.

2.4 Pixelwise performance of deep learning model for segmenting pyramidal cells

We worked with UNetDense, a deep learning framework, which has provided promising results for image segmentation of pyramidal cells (Lin, 2019). The performance of the UNetDense model was measured by reporting sensitivity, precision, and F1-score. Neither recall nor accuracy was calculated because the number of cell pixels was dominated by background pixels rendering these measures less informative. These measures were calculated based on the confusion matrices shown in **Table 1**. In general, the output of using individual models per subject showed a better predictive result than predicting pyramidal cells using one combined model, see **Table 2**. In general, individual models for each subject performed well with sensitivity, accuracy

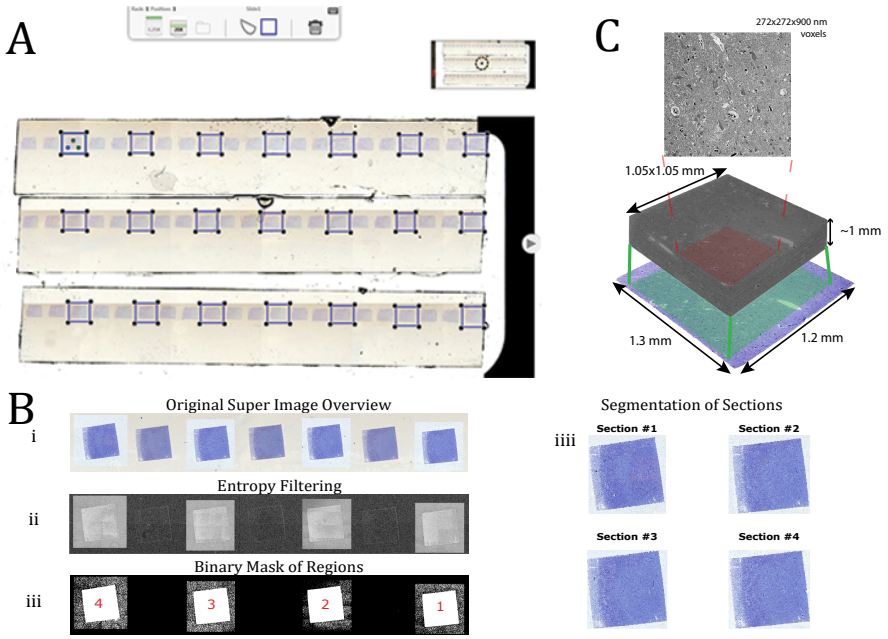

Fig. 3: Image acquisition and 3D-stacks of aligned sections. **(A)** Overview of a microscopic glass slide with the commercial Leica software. Systematic sampling of every third section was manually marked with three local points for the autofocus calibration. **(B)** In MATLAB, a TIF image was loaded where each second was of interest, which is different from (A) but easier to visualize as an example (i). Next, we used an entropy filter to detect sections of interest (ii). Binary masks have been computed for each section of interest and the images are ready for export (iii). The output of each exported section (iv). **(C)** Individual sections were aligned and stacked on top of each other. The stacked block of sections was then cropped down to a specific ROI (1.05x1.05 *mm*) which only contains tissue.

Model		1 (A)			0 (A)		
		Subject 1	Subject 2	Subject 3	Subject 1	Subject 2	Subject 3
Individual	1 (P)	128937 TP	83370 TP	91266 TP	11710 FP	14400 FP	10717 FP
Models	0 (P)	19923 FN	17810 FN	11512 FN	4033734 TN	4078724 TN	4080809 TN
Combined	1 (P)	119351 TP	77409 TP	92189 TP	9506 FP	12994 FP	12337 FP
Model	0 (P)	29509 FN	20717 FN	8931 FN	4035938 TN	4083184 TN	4080847 TN

Table 1: Pixelwise validation. Pixel to pixel comparison between an MS and UDP image (2048x2048) for each subject. The confusion matrix is used to validate the UNetDense model on images that the model has not encountered before. The result shows the performance of individual trained UNetDense models for each subject and a combined UNetDense model used for all subjects. Pixels belonging to the background have label 0, those belonging to the pyramidal cells have label 1, (A) stands for Actual (value in MS image), and (P) stands for Predicted (value in UDP image).

Model	Metrics	Subject 1	Subject 2	Subject 3
Individual Models	Sensitivity	0.87	0.82	0.89
	Precision	0.92	0.85	0.89
	F1	0.89	0.84	0.89
Combined Model	Sensitivity	0.80	0.79	0.91
	Precision	0.93	0.86	0.88
	F1	0.86	0.82	0.90

Table 2: Performance of pixelwise validation. Precision, sensitivity and F1 score of 2048x2048 pixel image for each subject with individual trained UNetDense models and one combined UNetDense model. Pixel evaluation is based on the confusion matrix of table 1.

and F1-score above 0.8.

2.5 Objectwise performance of 3D-reconstruction of segmented pyramidal cells

The performance of detecting pyramidal cells as 3D objects were evaluated by measuring the sensitivity, precision, and F1-score based on 3D-reconstructions of cells from a stack of 30 images. Manually Segmented (MS) and UNetDense Predicted (UDP) 3D-reconstructions were compared by checking whether estimated centroids from the MS cells fell within a cell profile of the UDP cells and vice versa. The sensitivity, precision and F1-score were 0.98, 0.93 and 0.95 respectively after removing data points from first and last three images, see **Table 3**. Thus, the 3D-reconstruction of pyramidal cells demonstrated a high performance across all three measurements.

3D-stack	TP	FN	FP	Sensitivity	Precision	F1
Original	472	19	33	0.96	0.93	0.95
Filtered	368	6	29	0.98	0.93	0.95

Table 3: Performance of objectwise validation. Validation of segmentation of pyramidal cells based on 3D-reconstruction. Precision, sensitivity, and F1-scores were calculated based on 491 reconstructed pyramidal cells from 30 stacked 2048x3840 manually marked images. Original refers to the case where centroid points from all 30 images are included. Filtered refers to the case where centroid points from the first and last three images of the stack were excluded.

2.6 3D-reconstruction and morphological analysis of pyramidal cells

Layer III was located using a density plot of a 2D-projection of the centroids of every 3D-reconstructed neuron, with yellow representing areas of high density and blue representing areas of low density, see **Fig.4**. Layer I got a very low density of neurons, while II and IV are denser than layer III in BA46. As Layer III has a smaller density than Layer II and Layer IV, the ROI is specified between the two dense yellow areas for our analysis. Further classifications into pyramidal or non-pyramidal neurons were needed as the UDP detects all neuronal shapes from 2D images. This is necessary because pyramidal cells' top and bottom image profiles can be mistaken for smaller neurons/glial cells, as seen in **Fig.S4**. The Gaussian mixture model(GMM) was used to classify the 3D-reconstructed pyramidal and non-pyramidal cells based on estimated sphericity and volume, see **Fig.5B**. Big objects/cells were classified as outliers if the maximum Feret Diameter in 2D or 3D measurements was three standard deviations from the mean. As a result, a total of 1, 19, and 28 datapoints for each subject were deemed outliers and hence excluded from the pyramidal cell data. The percentage of objects/neurons excluded from analysis using GMM and outlier detection accounted for 23%, 25%, 37%, and 23% of the total number of detected objects for each dataset, as shown in **Tab.S1**. The mean density of pyramidal cells in layer III of BA46 after filtering was 28155 mm^{-3}, and the GMM categorization and data containing outliers are shown in **Fig.S5-S6**. The measurements and calculations for each subject's entire stack were examined just for the classified pyramidal cells, with the number of cells and the size of the ROI window for each subject shown in **table S1**. **Table 4** provides information on the size, shape, and orientation of pyramidal cells of layer III in BA46 for all three subjects.

The average neuronal volume across all three subjects is 795 μm^3, and the shapes of pyramidal cells were assessed by approximating sphericity, giving an average value of 0.35 (Wadell, 1935). The orientations of pyramidal cells relative to the direction of the vector pointing towards the pial surface had an average of 29°, and some examples of orientation vectors are shown in **Fig.5E**.

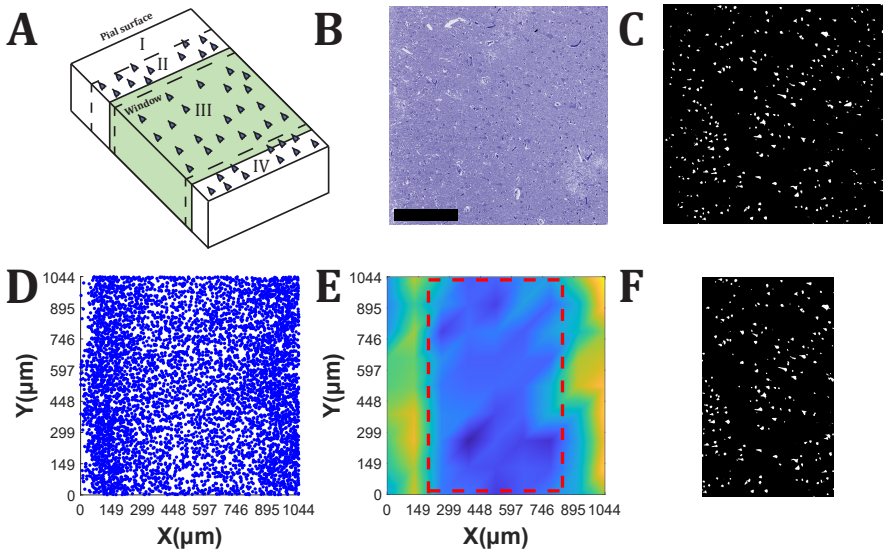

Fig. 4: Visualising the steps to identify layer III. **(A)** Illustration of four layers of BA46 and the observation window for the analysis is marked in green. **(B)** Raw image image of a section. Scale bar=300 μm.**(C)** Binary image of the output from the Deep learning model. **(D)** Position of centroids from 200 images projected to the x- and y-plane. **(E)** Density-map of the positions of neurons to visualize the different layers in the cerebral cortex and the yellow color represents areas of high density, whereas the blue color represents areas of low density. The density was high at the yellow areas, which indicate the position of layer II and layer IV (from left to right) and low at the blue areas, which indicate fewer neurons and show a part of layer I and layer III. The volumetric stack of images was cropped within the squared marked with the red dashed lines that was selected by a user. **(F)** The part of the image in C, which is chosen for analysis.

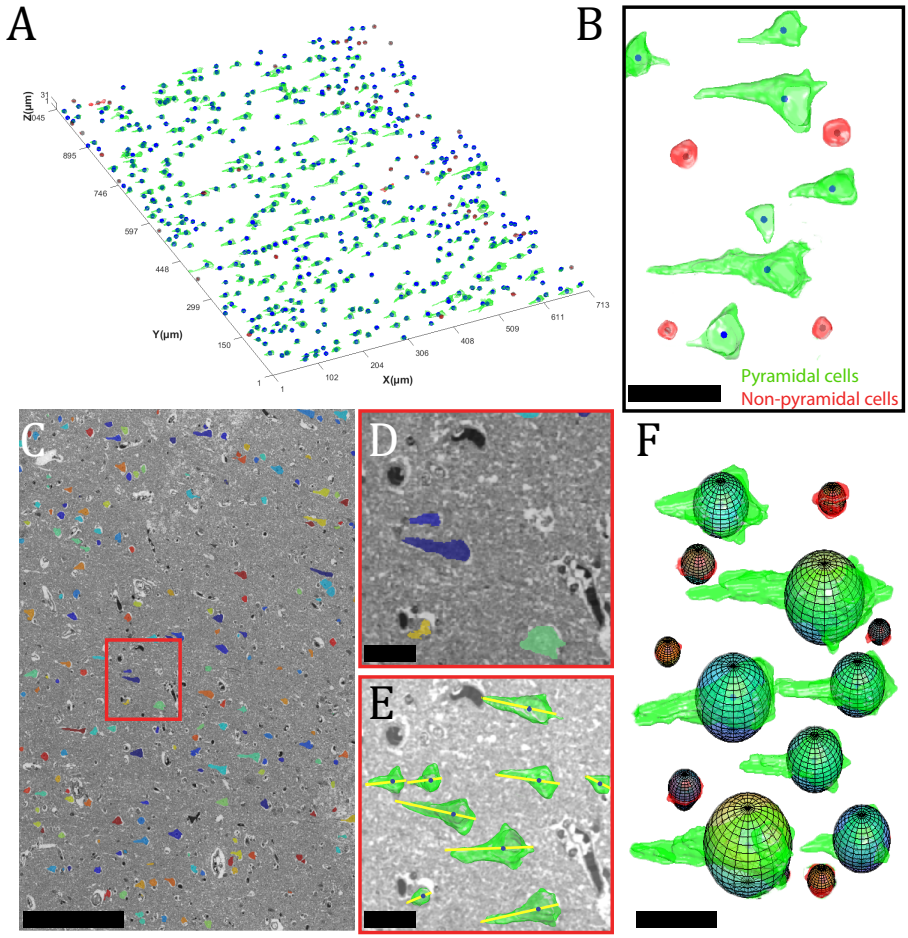

Fig. 5: Illustration of 3D-reconstruction of neurons and segmentation of pyramidal cells. **(A)** Visualisation of the 3D-reconstruction of one layer neurons with a window height of 31 μm. **(B)** Zoomed image of the segmented pyramidal and non-pyramidal cells. Scale bar=20 μm. **(C)** Overlay image of the binary segmented image with the gray-scale image of a section. Scale bar=200 μm. **(D-E)** Close-up view of the overlay picture and 3D-reconstruction of pyramidal cells, with yellow lines representing orientations. Scale bar=15 μm. **(F)** Measurement of diameter from the longest cell profile Dia$L$ was utilized to construct 3D spherical objects with their corresponding 3D-reconstructed cell. The graphic indicates that the difference in 3D volume space between the spherical approximation and the 3D-reconstructed item is similar. Scale bar=20 μm.

Brain	Volume (μm^3)	Orientation (°)	Sphericity	Dia$_L$ (μm)	Dia$_{All}$ (μm)
Subject 1	867 ± 960	34 ± 23	0.38 ± 0.08	11.13 ± 3.46	7.23 ± 3.26
Subject 2	709 ± 806	28 ± 22	0.33 ± 0.07	11.13 ± 3.74	6.49 ± 3.1
Subject 3	808 ± 893	24 ± 18	0.35 ± 0.06	11.27 ± 3.61	7.37 ± 3.1
Mean	795 ± 65	29 ± 4.1	0.35 ± 0.02	11.17 ± 0.07	7.03 ± 0.39
CV	0.08	0.14	0.06	0.01	0.06

Table 4: Quantitative values for pyramidal neurons from the entire stack of each subject. The table shows the neuronal volume, diameter, orientation and sphericity of pyramidal cells for each subject based on the data which is summarised in **Fig.S7**. The entries for each subject state the average and ± one standard deviation. Mean is the average measurement for each column, Coefficient of variation CV = SD/mean.

The diameter was calculated using the nucleator probe by measuring the segment length from the largest cell profile(Dia_L) and the average segment length from all cell profiles(Dia_{All}). The average neuronal diameter for Dia_L and Dia_{All} were 11.17 μm and 7.03 μm, respectively. Estimated spheres of length Dia$_L$ were constructed and displayed with their matching 3D-reconstructed cell in **Fig.5F**. Histograms of the different measurements can be found in **Fig.S7** and as log-normal transformed in **Fig.S8**.

2.7 2D vs 3D comparison of pyramidal cells sizes

The exact same neurons were used to compare the 2D and 3D analyses for each subject. The volumes of pyramidal cells from 2D images were approximated by constructing a spherical object based on the estimated radii measured by the nucleator probe.
The volumes of cells from the three subjects were calculated from 2D images using the estimated segment length from the largest cell profile (Vol_L) and all cell profiles (Vol_{All}). The average neuronal volume in 3D (Vol_{3D}), Vol_L and Vol_{All} are 795, 730, and 183 μm^3, respectively.
An approximation to the mean diameter of a cell Dia_{3D} was estimated from the mean volume Vol_{3D} under the assumption that it is the volume of a perfect sphere. Dia_{3D} was then assessed and compared to the diameter measured from the largest cell profile (Dia_L) and all cell profiles (Dia_{All}). Because the nucleator probe is derived from the mathematical fact that length of isotropic test lines between a unique point and the cell border, it provides the most accurate one-to-one comparison. The average diameter for Dia_{3D}, Dia_L and Dia_{All} are 11.48, 11.17, and 7.03 μm^3, respectively. The values for the 2D vs 3D comparison of volume and diameter can be seen in **Tab.S2-S3**.

2.8 Point pattern analysis of pyramidal cells

The coordinates of the centroids for the 3D analyzed pyramidal cells in layer III of BA46 form a spatial point pattern. We considered four such point

patterns, which we refer to as 1_1, 1_2, 2, and 3 and they correspond to the three subjects (Subject 1 was divided into two parts since it was collected over two different days).

In order to detect possible columnar structures in the data sets, we estimated the cylindrical K-function for each data set, and compared it to the 95% global envelope obtained by simulations under the null hypothesis of complete spatial randomness (CSR). The results can be seen in **Fig.6**. We considered the empirical cylindrical K-function in the directions of the three main axes. When the empirical cylindrical K-function is above the envelope, it indicates cylindrical clusters of points (centroids of cell locations) in the corresponding direction.

There were signs of columnar clusters in all three directions for all subjects. However, it was most pronounced in the direction of the x-axis, which is the expected direction of the possible columnar structure of pyramidal cells, especially when looking at radii between 5 and 20 μm and heights between 20 and 80 μm. There were areas where the empirical curves were below the envelopes suggesting some repulsive behavior in the data. This was not unexpected since the point patterns only represent the centroids of cells, since cells cannot overlap, it was thus natural to see some repulsion between the points. The global envelope tests corresponding to the situations in **Fig.6A** all yielded p-values below 0.05, and the tests corresponding to the situations in **Fig.6B** all yielded p-values below 0.001, indicating that all data sets exhibited large deviations from CSR.

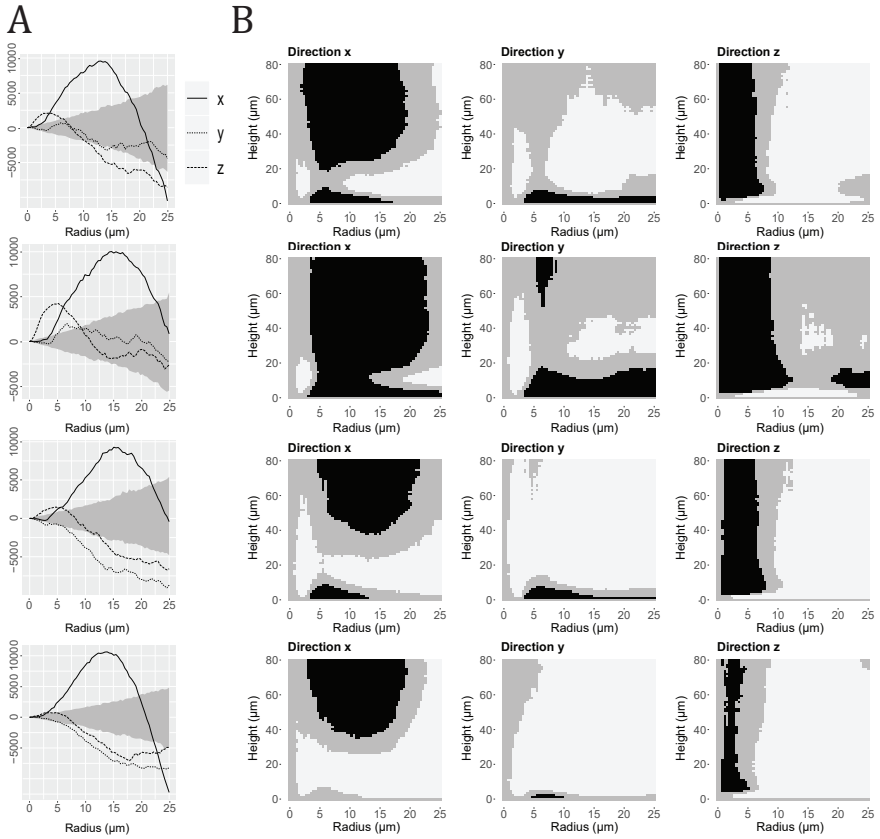

Fig. 6: Results of analyses with the cylindrical K-function for the spatial point patterns of centroids of pyramidal cells. Each row represents a dataset which from top to bottom are: Subject 1_1, 1_2, 2 and 3.**(A)** 95% global envelopes (shaded area) for the cylindrical K-function, with $t = 80$ fixed, and using 2000 CSR simulations. The theoretical value of the cylindrical K-function under CSR was subtracted from the curves for better visualization. The three curves correspond to the empirical cylindrical K-function for each dataset, in the direction of the x-axis (solid lines), y-axis (dotted lines) or z-axis (dashed lines). **(B)** Summary of 95% global envelopes for the four data sets based on the cylindrical K-function when varying both the height t and radius r. The envelopes are each based on 4000 CSR simulations. The direction of the cylinders is stated at the bottom of each plot. The plots indicate whether the empirical cylindrical K-function for the observed point pattern is above the envelope (black), within the envelope (grey) or below the envelope (white).

2.9 Tissue deformation

Every single-cell location research technique is vulnerable to tissue deformation. Epon embedding is used for the analysis in this paper to minimize shrinkage. We compared the tissue area of three Epon-embedded biopsies before and after processing. The human grey matter brain biopsies showed an areal shrinkage equal to 0.1%, 4.2% and 7.9%, respectively. Based on these data and our previous publications (Tang et al., 1997, Dorph-Petersen et al., 2001), it was decided not to correct our results for shrinkage.

3 Discussion

The presented method can reveal fundamental characteristics of specific brain areas, which can be used to improve our understanding of neuronal morphology and their spatial relationship. A variety of efforts have been made toward developing technologies for revealing the intricate patterns of neural circuits. By using different slicing or optical methods to recreate neuronal tissues, multiple studies have provided 3D data on the microscopic morphology or gene expression of neurons (Chung and Deisseroth, 2013, Knott et al., 2008, Mayerich et al., 2008, Li et al., 2010, Keller and Dodt, 2012, Becker et al., 2012, Amunts et al., 2013, Osten and Margrie, 2013). Due to a number of difficulties, such as size limitations of the slide scanner and microtome, low homogeneity of serial whole organ sectioning and staining, the time-consuming design of the operation, computer constraints, and limited digital processing capability, the entire human brain is difficult to recreate entirely on a mesoscopic scale. High resolution, 3D models of animal and human brains is difficult to acquire. Chemical clearance methods make the tissue transparent and enable the entire mouse brain to be reconstructed (Becker et al., 2012, Chung and Deisseroth, 2013), but 3D-reconstruction and digitization of fine neuronal morphology continue to be challenging. One explanation is that in the nervous system, cells are closely arranged, making it difficult to distinguish one from another. Applying Golgi-staining, a 3D structural dataset of the entire mouse brain was also collected by Micro-Optical Sectioning Tomography, which can conduct imaging and sectioning simultaneously on centimeter-sized tissues (Li et al., 2010).
However, such a section-based method demands costly specialized instruments, relatively long periods of sample treatment (weeks or months), and for a single brain imaging time can approach even one month (Gong et al., 2013). Based on the reconstruction of histological sections of a human brain preserved in paraffin, Amunts et al. developed a 3D model of the whole human brain called BigBrain with a spatial resolution of 20 μm (Amunts et al., 2013). The tissue was embedded in paraffin, which can shrink tissue volume up to 50-60% (Dorph-Petersen et al., 2001), and it is challenging to

study neuronal morphology with a spatial resolution of 20 μm. To date, no optimal approach to the analysis and processing of the whole human brain with micro-resolution has been achieved. The majority of the approaches discussed above share the use of immunolabeling in order to detect and localize antigens or proteins within a cell at a specific location.
The fundamental constraint of immunolabeling is that milder fixation conditions and a shorter detergent permeabilization incubation time are required to allow antibodies to penetrate the tissue. This is particularly important for soluble proteins in the cytoplasm, which are frequently damaged or destroyed during incubation. Hence, it is challenging to create neuronal antibodies that work in postmortem brains that have been in the fixative for months or years (Lyck et al., 2007). Another well-known concern is that many histological procedures can cause tissue samples to shrink and deform, potentially altering cell size, shape and organization. Keeping the dimensions of the tissue is especially essential if studies want to assess changes in the size and distances of any cells or organelles within the tissue. Our methodology provides an effective technique for imaging smaller pieces of most archival tissue from semi-thin serial sections into a functional dataset with hardly any tissue deformation. Researchers have with our method a unique opportunity to study archived tissue samples, enabling researchers to investigate tissue and disease development in great detail.
An automated section collection EM system has been altered into a unique 3D-reconstruction method for light microscopy, AutoCUTS-LM. We used histological staining to visualize cytoarchitecture and an autonomous slide scanner to produce high-resolution images of the brain tissue. Via deep learning, pyramidal neurons were characterized and their spatial distribution analyzed using advanced techniques. Our application provides valuable information on the neuronal 3D architecture from archived human brain tissue.
We systematically sampled the tissue for our setup to reduce the scanning time and the processing of data. A thickness of 800 and 900 nm provided sufficient axial spatial resolution to detect and reconstruct pyramidal cells. The UNetDense architecture was applied to detect the pyramidal cells in our images, as threshold-based segmentation techniques typically yield poor output in medical image analysis for low-contrast images, unexplained noise, blurred boundaries, and different light condition (Tsai, 2007, Malarvizhi, 2017). The variation between brains made it challenging to create a model that suited all three brains since flaws occurred mainly along the edges of the images, and where the contrast was low due to less staining absorption. Therefore, we trained a individual UNetDense model for each subject in this study since the sensitivity, precision, and F1-scores for both pixelwise and objectwise segmentation were above 0.82 and 0.93, respectively. Overall, the results of the combined model showed smilar precision and F1-scores but lower sensitivity compared to the individual models. That being said, for

this study, sensitivity was prioritized as it shows how well the model detects the neurons that are in data, which is essential for ensuing the point pattern analysis.
The risk of having three models is overfitting. Overall, the validation-findings from the sensitivity, precision and F1-scores showed that the outcomes of the three models are reliable. One reason for the difference between subjects was that the brains absorb staining differently, and this may be due to different ages, postmortem intervals and long fixation time. With this limitation in mind, one model might be adequate if the captured images did not alter too much or if we annotated more images to train the model.

We measured the volume, sphericity, orientation and diameter of pyramidal cells and used the cylindrical *K*-function to identify columnar structures. The average volume was 795 μm^3 with an equivalent diameter equal to 11.48 μm. The neuronal diameters of pyramidal neurons in layer III of BA46 have received little attention in the literature. Nonetheless, Rajkowska et al. examined 150-200 neurons in this region using stereological techniques and determined an average diameter of 13.45 μm. The process for measuring neuron radii was comparable to the nucleator probe as they measured the border outlines of cell profiles to compute the associated diameter. Similar to the procedure in this study, they only assessed neurons in the crown of a gyrus. The differences in radii could be related to the fact that we sampled pyramidal cells from a small concentrated area. In contrast, they quantified neurons in broader regions of BA46 using a succession of counting boxes.
Compared to 11.48 μm our method provides information of thousands of counted neurons per subject compared to 150-200 neurons counted by Rajkowska et al. (Rajkowska and Goldman-Rakic, 1995a). In BA46, counting fewer neurons from smaller sample sizes over a broader area may overestimate neuronal radii. This is due to the fact that smaller pyramidal cells dominate layer III over bigger ones, resulting in a stronger proclivity towards smaller radii. The difference in radii measurements across individuals might also be related to biological variance for subjects, brain storage period, or different embedding media, as the difference is roughly 17%.
Different factors such as age, disease, and toxicity have been reported to affect neuron volume in the cerebral cortex across other studies, however, they do not give any detail on how the shape of the neurons are affected (Janson and Møller, 1993, Bundgaard et al., 2001, Jansen et al., 2007, Rudow et al., 2008).
A literature search revealed that no quantitative values to quantify the sphericity and orientation of pyramidal cells had previously been published. As a result, such measurements were not comparable to those of other studies.
Sphericity is valuable as a general shape descriptor since it also applies to objects having holes, such as a torus. Yet, the main reasons sphericity is used for measuring shape are as follows: sphericity is a unit less number,

hence is scale invariant. Further, it is simple to compute since MATLAB easily calculates the surface area and volume of cells, which are fundamental descriptors of cells in their own right. Estimation of sphericity does not presume any shape before examining a 3D object, making it an appealing tool for differentiating between distinct forms. As a result, we implement sphericity estimates to comprehend a natural cell shape. The average sphericity of 0.35 found in this paper indicates that pyramidal cells appear elongated and far from spherical.
The average angle between the vector that represents the orientation of the pyramid cell and the x-axis, which points to the pial surface, is 29°, suggesting that pyramidal cells are focused perpendicular to the pial surface.
Quantities such as cell sphericity and orientation in follow-up studies with different disease groups could be utilized to understand the morphological alterations induced by these disorders.

The volumes of pyramidal cells calculated from 3D reconstruction images were compared to the results of the nucleator probe measurements for the same pyramidal cells, which estimate the mean cell volume from 2D images. It is crucial to keep in mind that the volume estimates are based on two separate methodologies, and the volume may be overestimated or underestimated with either approach.
The 3D approach measures every cell profile, and if a cell profile on the edge of a pyramidal cell is not recognized, this approach may underestimate the volume (top or bottom cell profile). On the other hand, the nucleator relies on measurements from a unique position (e.g. nucleolus), assuming that the section or cell is isotropic. If this criterion is not fulfilled, the estimate may be biased.
The average estimated volumes of pyramidal cells were 795, 730, and 183 μm^3 for Vol_{3D}, Vol_L, and Vol_{All}, respectively. The difference between Vol_{3D} and Vol_L is 8.7%, whereas the difference between Vol_{3D} and Vol_{All} is 77%. It appears that using the nucleator probe on the largest cell profile to estimate volume yields a comparable overall result, but employing the nucleator probe on all cell profiles causes a huge difference. The estimated average volumes varies greatly depending on the approach used.
The estimated diameter of pyramidal cells changed less than the volume estimates, with results of 11.48. 11.17, and 7.03 μm for Dia_{3D}, Dia_L, and Dia_{All}, respectively. The difference between Dia_{3D} and Dia_L is 2.9%, whereas the difference between Dia_{3D} and Dia_{All} is 39%. Hence, if we look at estimated radii when employing the largest cell profile compared to the 3D estimation, there is no substantial disagreement between those two approaches for calculating diameter because the difference is nearly equivalent to the 272 nm pixel size. In contrast, estimating the average radius from all cell profiles makes a notable difference. As the neurons are not spherical in reality, comparing neuronal size based on volume rather than diameter is a more

appropriate parameter to quantify changes in neuronal size in future references. Nevertheless, if only a 2D technique is available, adopting a spherical approximation by assuming general isotropy of these human pyramidal cells to estimate neuronal volume is not entirely inaccurate. It is vital to notice that a slight change in the radius can substantially change the volume value considering the radius is raised to the power of three while computing the volume.
Even though the nucleator is a 2D procedure, it is nevertheless based on a 3D sampling methodology since the biggest cell profile detected during the sample is required before the analysis can be completed. The nucleator is therefore a helpful tool to use when it is challenging to distinguish smaller cell profiles at the edges and only larger cell profiles are required.
Another important consideration is that the nucleator will be less sensitive to tissue shrinkage in the direction of the z-axis as it estimates the volume from a single 2D plane. The nucleator is advantageous for estimating volumes compared to other embedding materials that are sensitive to shrinkage, such as frozen sections or vibratome sections, which rarely deform in the x- and y-axes but shrink in the z-axis.

The observed pyramidal neuron density in layer III of BA46 after the classification between pyramidal and non-pyramidal cells is 28160 mm^{-3}. A stereology study by Francine M. Benes found a neuronal density of 36800 mm^{-3} in layer III of postmortem brains from nine healthy controls in the prefrontal cortex (Benes, 1986).
Cullen et al. estimated a neuronal density of 37000 mm^{-3} in the prefrontal cortex of 10 adults with no history of mental illnesses, similar to Benes et al. However, Cullen et al. also measured the density of pyramidal cells in layer III to be 25650 mm^{-3}, a difference of 30% compared to the total neuronal density.
Rajkowska et al. characterized and mapped the cytoarchitecture of BA46 in 17 healthy people. They were able to estimate a neuronal density of 51510 mm^{-3} by blending cortical layers I-III (Rajkowska and Goldman-Rakic, 1995a). Because the density in layer II varies from 48000-78000 mm^{-3} and covers a smaller area than layer III, it is reasonable to assume that the density in layer III would be about 36000-44000 mm^{-3} (Benes, 1986, Cullen et al., 2006). If the difference between neuronal and pyramidal density is roughly 30%, then the projected density should be approximately 25200-30800mm^{-3}, which is consistent with our findings.
One thing all of the three studies listed above have in common is that they solely measured prefrontal cortex neuronal density (Benes, 1986, Rajkowska and Goldman-Rakic, 1995a, Cullen et al., 2006). Number densities indicate changes in the number of cells as well as tissue volume under consideration, because they are ratios. It is commonly assumed that as the number of detected objects rises, so does the density. However, the density increases

as well if the number of identified objects remains constant, but the examined volume shrinks. As noted in other studies, making any definite claim regarding changes in 3D structures based purely on density estimates may lead to questionable findings (Brændgaard and Gundersen, 1986, Swaab and Uylings, 1987, Oorschot, 1994).
Because this assumption is so widespread, the phrase "reference trap" refers to circumstances in which wrong conclusions are drawn only on the basis of density. As a result, comparing densities should be handled very carefully.

Some studies have shown that pyramidal cells are organised in columns that are perpendicular to the pial surface of the cerebral cortex (Mountcastle, 1997, Casanova, 2013, Opris et al., 2017, Casanova and Casanova, 2018), other studies suggest otherwise (Slimp and Towe, 1990, Swindale, 1990). It is an attractive idea to explain the neuronal organization because interconnected neuron groups typically share similar physiological properties, and the conditions that excite a neuron are also likely to excite a considerable fraction of its afferent input. The loss or changes in the spatial organization of neurons may interfere with information processing between distributed networks, thereby promoting cognitive decline. The spatial distribution of pyramidal neurons was analyzed with the use of the cylindrical K-function, which does not assume any columnarity a priori, and where we applied the cylindrical K-function on much larger point pattern datasets than so far analyzed in the literature (Møller et al., 2016, Rafati et al., 2016, Christoffersen et al., 2021). Our results suggest that there is evidence of a columnar structure in the directions of the x-, y-, and z-axes for all three subjects, but it was most pronounced in the direction of the x-axis, which is pointing towards the pial surface. Thus, the results support the theory of a columnar pattern perpendicular to the pial surface. The method can be used to detect potential cytoarchitectural distortions that may impact the neuronal columnar organization in the brain cortex. However, more human brains need to be studied in order to draw any final conclusions.

Technical and environmental factors could affect the sampling quality of our sections. The plasma treatment required the transparent collection tape to be hydrophilic to ensure that the sections would adhere to the tape (Li et al., 2017, Kubota et al., 2018). If the tape was too hydrophilic, the sections would not have time to unfold before they landed on the tape and folds were unavoidable. Variations in density between tissue samples and resin meant that the remaining blank resin had to be trimmed off (Hildebrand et al., 2017, Li et al., 2017, Burel et al., 2018). We observed the advantage of higher indoor humidity, which provided a favorable environment for collecting sections. The impact of induced folds can be seen with a humidity level about 10%, 60% and 90% in **Fig.S10-S11**. The test showed that a stable indoor humidity level around 80% and 90% could help avoid folds.

If larger structures were to be explored in the future, the use of an op-

timized resin embedding protocol that could be used for thicker sections (> 500 *nm*) would ease the workload. Applying a resin ratio test to match the tissue density and select the most suitable one before collecting samples with AutoCUTS-LM would have a beneficial effect on reducing the folds in the section. Reducing the hardness of the resin to suit a particular tissue type can be achieved by controlling the ratio of two different anhydride curing agents(Dodecenyl Succinic Anhydride and NADIC Methyl Anhydride). This could avoid the excessive cutting of thousands of needless sections, which would result in the sharpness of the knife being extended, using less tape, and a reduced workload for data collection and image processing. Other labeling methods can also be carried out with the AutoCUTS-LM, as our sections can be combined with immunofluorescence labeling to identify the distribution and co-localization of proteins as shown in **Fig.S3A**, and it may also be combined with in situ hybridization in order to localize a specific DNA or RNA sequence (Baccetti et al., 2005). Due to the use of a hard resin, sections can also be viewed in an electron microscope, **Fig.S3B**, in which case cellular ultrastructure can be visualized. In essence, this methodology can be designed for both light microscopy and electron microscopy to bridge the localization of important molecules with cellular ultrastructure.

In conclusion, the AutoCUTS-LM method could benefit research in cell and developmental biology, model organism analysis to address mechanisms as cell migration, 3D tissue modeling, and morphological changes between animal/patient groups. This method is applicable for any disease and can potentially enhance the research of normal and disease processes, particularly those involving morphological alterations or in which the spatial interaction of disease features is essential.

4 Methods

4.1 Subjects

Three healthy human brains (two women, one male) aged 30, 53 and 58 (Subjects 1, 2 and 3) with no history of psychiatric or neurological diseases were selected from the brain collection at Core Centre for Molecular Morphology, Section for Stereology and Microscopy, Aarhus University Hospital, Aarhus, Denmark. These archived brains have been stored for 19, 21 and 21 years, respectively in 4% formaldehyde in phosphate buffer at neutral pH and were collected in compliance with Danish law and with approval from the Central Denmark Area Health Research Ethics Committees (license number: M-2017-91-17).

4.2 Sample extraction

We have developed an algorithm in MATLAB that can assist any user in delineating a ROI in a block of tissue and systematically sample two (or more) points for biopsies on the cortex, see **Fig.1**. The user had to capture a picture of the tissue block and define the biopsy diameter. Then the image was transformed into grayscale, and the ROI was delineated to produce the sample area. This picture was then transformed into a binary image indicating the sample area by means of a global threshold. Next, to prevent the biopsy from being placed on the crown of the gyrus, the edges of the binary image were removed. This was done by using morphological erosion with a filter size equivalent to the biopsy diameter and is shown with the red color in **Fig.1 D**. The sampling area was divided into four quarters. The two biopsy punches should either be performed in the first and third quarters or in the second and fourth quarters to avoid overlapping biopsies. The four quarters were determined by considering the accumulation of white pixels by rows and detect when they reached 25%, 50% and 75%, see **Fig.1 E**. A 1.5 *mm* diameter biopsy punch was used to sample brain tissue covering all six layers of the cerebral cortex.

4.3 Sample embedding and block preparation

The biopsies were first immersed in Phosphate-buffered saline (pH 7.3) with sucrose for one day and then washed two times in 0.05 Mol maleate buffer(pH 5.2), 5 min each time, at room temperature. Osmium is traditionally used to stain samples for epoxy resin embedding for EM. However, we discovered that it decreased the sample's signal-to-noise ratio (SNR) when it was applied for light microscopy, see **Fig.S1**. Samples were processed and embedded inside the Leica EM TP Automated Tissue Processor (Leica Microsystems, Brønshøj, Denmark). Here, they were stained with 1% uranyl acetate in maleate buffer for one hour and dehydrated through a graded ethanol series (70%, 86%, 96% and 99%, 20 min each). Following the completion of dehydration, samples were washed three times with 100% acetone for ten minutes, followed by infiltration in 100% acetone/epon 1:1 with constant rotation for 12 hours overnight. Infiltrated samples were incubated in pure Resin 812 for one hour and placed in embedding molds in a pre-warmed oven (60°C) to polymerize for 24 hours. The biopsies were placed in the bottom of the embedding form, such that the pial surface was perpendicular to the cutting direction of the knife, as sectioning all six layers in the neocortex was preferred. This reduced the number of sections by approximately 2-3 times compared to alternative orientations, which ease the time spend on capturing images for 3D-reconstruction. After the resin had fully cured, most of the white resin from the embedded sample was roughly trimmed by a high-

speed milling system (EM TRIM2, Leica) with an angle set to 60°. A glass knife was used for fine adjustment to trim around a 1.1× 1.4 mm rectangle with a depth of 0.7 mm (1 mm^3), resulting in a sample where only neurons in the supragranular layers and layer IV were included, see **Fig.S2**. It was essential to trim all the blank epoxy-resin away since the density difference between the tissue and epoxy can generate wrinkles while cutting. Supplementary Note 1 describes in detail how to prevent wrinkles and optimize section quality, as seen in **Fig.S9-S14**.

4.4 Transparent collection tape

Collection tape with different settings was tested to find the most suitable tape for our needs. A roll of PEN tape 300 mm wide, 45 m long and 50 μm thick was chosen for this study attributable to heat treatment and the feature of a protective coat on both sides, which prevents dirt(South China Science& Technology Co., Ltd, China). This tape was slit into 7 mm wide strips (Tianjian Xinhua Electronic Material Co. LTD. China). Adjusting the tape hydrophilicity was essential since it reduced wrinkles of the brain sections on the tape and made the sections more adhesive to the tape, so they did not fall off during collection and staining. The system parameters for the plasma treatment (Beijing Jiaruntongli Technology Co., Ltd., China) were set with the values: power 120 W, frequency 40 kHz, speed time 4 mm/s, and processing time of about 2 h for 20 m tape.

4.5 Automatic serial section collection

An ultramicrotome (EM UC7, Leica) connected to a custom-tape collection system (AutoCUTS) was used to automatically cut the resin-embedded material into serial sections. Serial sections with a thickness of 400 and 300 nm were cut by a 45° Histo diamond knife (Diatome, Switzerland) and floated onto the water surface, see **Fig.2A**. The tape's reel speed and cutting speed were set to 1, and 2 mm/s, respectively, which gives a distance of 1 mm between every section on the tape. The pulling motion from the collection tape brings the sections from the water to the surface of the tape, and the adhesiveness of the tape affects the flatness of these sections.
We collected about 800 sections per hour with our current settings. Hence, it takes less than three hours to finish around 0.7 mm of dense tissue. A video camera was attached to the AutoCUTS to monitor and record the process to ensure a more comfortable experience for the user by displaying the cutting process onto a computer screen. It was possible to collect thousands of sections without any loss of tissue.
Different technical and environmental factors were observed that could generate wrinkles and disfigure the sections during cutting as described in Sup-

plementary Note. We found that sections with a cutting thickness above 300 *nm* were more susceptible to generate these wrinkles during the cutting process. The impact of wrinkles was also affected by a combination of room humidity level and tape hydrophilicity.

4.6 Section Library

The spool that contains the tape with attached sections had to be completely dry before staining was applied to prevent any sections from falling off during this process. Hence, the spool was placed in a sealed plastic bag and placed inside a 50°C oven overnight. The toluidine blue staining was absorbed differently in each brain. Consequently, we had to check the optimal staining time for every brain before running the protocol. Some sections would be stained at room temperature for 20 minutes to decide which time the pyramidal cells had the best signal-noise ratio. Hereafter, the tape was segmented into smaller pieces and placed in a petri dish (20 cm diameter) filled with 1% toluidine blue without the tape sticking together**Fig.2B**. During the staining period, a cover was placed on top of the petri dish to condense moisture and prevent dust or dirt from mixing with the blue toluidine solution. Toluidine blue residues on the tape segment were washed away (once with 80% ethanol, twice with purified water), and the tape segment was then dried with a hairdryer **(Fig.2C)**. It was important not to let the tape dry naturally since water stains would then be developed on the plastic surface of the tape, which generates hazy white spots on the surface. The tape segment was cut into smaller strips and glued onto a typical 75 *mm* by 25 *mm* microscope slide**(Fig.2D,E)**. The sequence of the serial sections was numbered from the bottom right corner to the top left corner. Beforehand, each glass slide was cleaned of dust and other particles in pure acetone and alcohol. Different glues were bought and tested for their ability to adhere PEN tape to the microscopy glass slide. For this study, we chose the glue from Krazy glue (Krazy Glue All Purpose Super Glue Pen, Fine Tip, 3 Grams) due to its high adhesive level and because it was non-toxic and convenient to use. Krazy glue does not require a fume hood, which can generate turbulence and blow away the cutting strips. Besides, the pen shape made it easier to use than a standard plastic pipette**(Fig.2D)**. Note that sometimes air bubbles could be generated between the glass and tape, so it was essential to press the tape as flat as possible with a pair of rubber-tipped tweezers. The 7 *mm* wide tape was chosen for this specific reason since it was more convenient to glue 3x7 *mm* parallel strips onto a 25 *mm* glass side compared to the traditionally 8 *mm* wide tape.

4.7 Data acquisition

Images of the sections were acquired using Leica's Apiro Versa 200 Digital Pathology Scanner. The Apiro Versa 200 is equipped with a 200-slide autoloader and a robotic arm that allows any user to capture pictures unsupervised.
First, it captured a low-resolution overview image of the whole slide. Subsequently, three focal points were positioned manually on each observable section for autofocus adjustment. Next, the microscope collected images at higher magnifications (lens 20x, NA 0.8) with pixel size down to 272 *nm*, which did not lose any details because the pixel size was below the expected resolution for this system's optics. Systematic sampling of our sections was done by only chosen every second (subject 1) and every third section (subject 2 and 3). This choice was based on the chosen cutting thicknesses of 400 *nm* and 300 *nm*, which correspond to a sample interval of 800 and 900 *nm*, respectively.
The output files were named based on the positions of the glass slides in the loader, e.g., Slide1 for position 1, and the output files could be read from the commercial software Aperio ImageScope (Leica Biosystems Imaging, Inc., USA) that was part of the microscope interface. Aperio ImageScope could visualize the whole slide image and keep the high-resolution image. However, the user had to manually select a region before sections could be exported as individual image files. As a result, we built a script that could load large image files containing multiple sections and export the individual sampled sections as uncompressed TIF files in order, see **Fig.3B** (Larsen, 2020a).
The output images for each glass slide were roughly 2-4 GB in total if images for each stripe were exported, and it is thus recommended to break the photographs into smaller segments for each glass slide. First, the large image files were converted to grayscale, followed by a Gaussian blur. An entropy filter was then applied to measure randomness, which was used to characterize the texture of the input image. This could be used to detect sections of interest since pixel values of sections in focus varied a lot due to the presence of various fine details, while sections out of focus were blurry and showed more homogenous pixel values. After that, a binary mask for each section was created by replacing all values above the globally defined threshold with 1 and filtering out smaller connected components. Each section detected was then exported as an individual file and prepared for alignment.

4.8 Alignment of sections

The alignment of the segmented sections for each subject was accomplished by a sequential slice-to-slice image-based registration approach. Regions in image pairs were matched by translating and rotating images following a

precise registration. First, we converted the RGB image of a section into grayscale and then employed a median filter to remove the noise. Then, we did a rough registration followed by a fine-registration using rigid intensity-based image registrations with different optimizer and metric configuration properties (Rosenfeld and Kak, 1982). In MATLAB, we set the imregconfig function, which determines the optimizer and the metric configuration, to the multimodal configuration, as images may have varying intensity distributions. The function imregconfig was set to its default values for the rough registration, while the growth factor, initial radius and maximum iterations were updated to 1.02, 2×10^{-3} and 300 respectively for the fine registration. For the image registration, images were downscaled by a factor of four in all directions before any transformations were done in order to increase the registration speed. After the transformation matrix calculation, the images were upscaled again in order to recover the original scales. After all the images were aligned, a window in which only the tissue remained was chosen. The images were chopped to this window and prepared for analysis, see **Fig.3 C**.

4.9 Data analysis pipeline

The research pipeline for processing microscope images are depicted in **Fig.S15**, where the key steps in the pipeline are explained in the following subsections. First, we manually annotated microscope images and augmented those images to produce a sufficient amount of images for the training and validation set used to train the UNetDense model. After all aligned images were segmented with the UNetDense model, a density map was used to identify multiple layers of neurons in the neocortex, which made it possible to identify and crop out layer III for further study. Then, we performed 3D-reconstruction and calculated morphological parameters for all cells from the segmented neurons in layer III. On the basis of the 3D-reconstruction cells, pyramidal cells were detected and ready for analysis. Finally, 3D coordinates of the centroids of pyramidal cells were investigated for columnar patterns by using the cylindrical *K*-function.

Annotation

Data annotation is the method of labeling objects of interest that are detectable. The entire process for marking the pyramidal cells for our dataset was performed on 35 random cropped images from the aligned stack of each subject, is illustrated in **Fig.S16**. Here, the annotation was manually performed by an expert (NYL) using Photoshop's quick selection tool for each subject, but any image labelling program can be used, e.g. open-source tools such as VGG Image Annotator or ImageJ. Moreover, the Image Labeler program in MATLAB or employing the inbuilt function ginput to mark the bor-

der of a cell and then applying the imfill function to mask the cell may also be useful for labeling cells.
Next, we used MATLAB to read the Manually Segmented (MS) images as binary images where pixels in pyramidal cell profiles equaled one, and all other pixel values equaled zero.

Deep learning architecture

Deep neural networks, in particular, convolutional neural networks (CNN), are commonly used for tasks of image classification (Rawat and Wang, 2017). For the research of this paper, we have chosen to continue working with the network described in the thesis 'Statistical analysis of pyramidal cells in brain tissue' where the code was published on GitHub (Lin, 2019). The deep learning framework, UNetDense, is a modified version of the original UNet architecture and consists of several dense blocks, transition blocks and merging blocks adopted by the pre-trained DenseNet-121 to compute pixel-level predictions for neurons (Ronneberger et al., 2015, Huang et al., 2017). The Adam optimizer was used with a learning rate of 1×10^5, the loss function was Binary Cross Entropy plus Dice Loss, and the code was run on Google Colaboratory (Kingma and Ba, 2014, Lin, 2019). Cell profiles from 35 annotated images from each model were each sliced into 200 image patches of 256x256 pixels without redundancy, giving a total of 7000 (35x200) images. The 7000 images were augmented by adjusting brightness and contrast and were then divided into a training set of 5600 images and a validation set of 1400 images. The differences in tissue from different subjects make it be difficult to train one combined model to segment the data of all subjects, see **Table 2**. Hence, we compared a combined model with individual models trained for each subject. Only the model for Subject 1 was trained from development, after which transferred learning was applied to the other two models afterward.

Pixelwise validation

We classified our predicted pixel values into four categories: true-positive (TP), false-positive(FP), true-negative(TN), and false-negative(FN).

TP: The total number of pyramid-pixels correctly identified by UNetDense model.

FP: The total number of pyramid-pixels wrongly identified by UNetDense model.

TN: The total number of non-pyramid-pixels correctly identified by UNetDense model.

FN: The total number of non-pyramid-pixel wrongly identified by UNetDense model.

For validation, the model was used to segment neurons from images that had also been manually segmented. The difference was then measured between the MS image and the UDP image, see **Fig.S17-S19**. The performance of each UNetDense model for segmenting 2D images was assessed using metrics of sensitivity, precision, and F1-score.

$$Sensitivity = TP/(TP + FN) \tag{1}$$

$$Precision = TP/(TP + FP) \tag{2}$$

$$F1 - score = \frac{2 \cdot (Sensitivity \cdot Precision)}{Sensitivity + Precision} \tag{3}$$

Objectwise validation

For validating the predictions of 3D-reconstructed cells, a new validation set was produced of 30 stacked images (2048x3840x30), which took about one week to complete. The data obtained was just a small portion of stacked images from Subject 1 and had not been seen by the UNetDense model before. The 3D-reconstruction from a stack of binary images and performance of the objectwise validation was done using custom MATLAB scripts, where we used the built-functions bwconncomp and regionprops3 for the 3D-reconstruction. After being reconstructed as 3D objects, the manually marked and predicted pyramidal cells were compared.
For this comparison, observed structures that did not appear in more than three consecutive images were omitted, corresponding to a height below 3 μm. Then, for both the MS and UDP images, we calculated the centroid of all detected pyramidal cells. A case where a MS centroid fell within a cell profile of the UDP 3D-reconstruction was denoted TP, while a case was denoted as FN when an MS centroid did not fall within such a cell profile. The FP case was defined to be the situation where a centroid of the UDP 3D-reconstruction did not fall within a cell profile of the MS 3D-reconstruction since these are falsely detected cells in the UDP. Based on these definitions, sensitivity, precision and F1-scores were calculated.
The validation set contained 491 reconstructed and labeled pyramidal cells, which contains 5556 cell profiles in total, and we observed that the first and last three images of the data set predominantly contained false-negative neurons, see **Fig.S20**. The cause for this error was that centroids for cells extending beyond the image borders were poorly estimated. Thus, these were removed.

Defining layer III

The aligned segmented images from the output of the deep learning model were examined with custom MATLAB scripts. In this research, pyramidal cells in layer III were of primary interest. Thus, we identified a ROI containing only layer III by plotting a density map of a projection of estimated neuron centroids. These estimates were made from a total of 200 images loaded from the beginning, middle and end of our complete aligned stack. We used the built-in function regionprops3 to estimate neuron centroid values from such images. Segmented 3D objects that were smaller than eight voxels were considered as artifacts and were thus filtered out just like centroid points from the first and last three images.
The density map depicts where the majority of cells were located, with yellow representing regions of high density and blue representing areas of low density. Layer III of the neocortex has a lower density than layer II and layer IV, then the ROI was specified between the two dense yellow areas for our analysis**Fig.4D**. After a user had clicked on the top left and the bottom right corner to define the ROI, red dashed lines appeared to show the cropping frame, see **Fig.4D**. This semi-automatic approach was chosen due to its reproducibility and effectiveness.

3D-reconstruction and quantitative measurement of pyramidal cells

The use of custom MATLAB scripts completed the study of morphological features and visualized the 3D-reconstruction of pyramidal cells, as shown in **Fig.5**.
Quantitative analytical values for each pyramidal neuron were then approximated based on the entire stack of images, such as volume, centroid, diameter, maximum Feret diameter, orientation, surface area, and sphericity, where most values were calculated using the built in function regionprops3.
Each cell's shape was assessed by approximating the sphericity, which does not have any prior assumption of shape and is independent of cell size. The sphericity is a dimensionless ratio, and the formula is given in **Eq.4**, where V and A are the volume and surface area of the segmented cell. If a cell has sphericity equal to 1, it resembles a perfect sphere (Wadell, 1935).

$$\psi = \frac{\pi^{1/3}(6V)^{2/3}}{A} \tag{4}$$

It was necessary to classify the detected neurons into pyramidal or non-pyramidal neurons as the UDP detects all neuronal shapes. This is because the top and bottom parts of a pyramidal cell have smaller profiles and can appear to be part of minor neurons or glial cells, as seen in **Fig.S4**.

A Gaussian mixture model (GMM) clustering algorithm was used to differentiate pyramidal and non-pyramidal neurons based on 3D measurements. The data used for the GMM consisted of the estimated volume and sphericity in 3D. The choice of data for the GMM is based on the idea that spherical objects with a smaller volume resemble smaller neurons or glial cells and hence non-pyramidal cells. For the dataset used for the GMM, we decided to only use the observed cells below the average volume of the dataset to ensure that only small round cells were evaluated.
The data set was fitted to the GMM using the default settings in the in-built fitgmdist function in MATLAB. After the GMM separated the dataset into two, the cells in the cluster with the lowest measurements were considered non-pyramidal cells and were excluded for all three subjects, see **Fig.S5**. Sometimes two cells were very close to each other and such merged cells were detected as one cell. Non-cellular structures like big vessels may also look like one cell. Such large, undesirable items were detected by identifying elements that had a log-transformed maximum Feret diameter measurement in 2D and 3D were greater than three standard deviations from the mean.
This is particularly effective for distinguishing artifacts of unusual length because the maximum Feret diameter is the most extensive distance between two points in the convex hull, see **Fig.S6** for the filtered data.
Finally, the orientation vector of a pyramidal cell is defined to be the unit vector u in the direction of the maximum Feret Diameter, **Eq.5**. The maximum Feret diameter provides information about the most extended length of a cell in a particular direction, usually towards the apical dendrite. The orientation angle θ is then defined to be the angle between u and the unit vector $u_0 = (1, 0, 0)$ which points in the direction of the x-axis. This means that if a pyramidal cell has orientation angle 0, its orientation vector points towards the pial surface. The orientation vector for each pyramidal cell can be calculated by

$$u = \frac{d - c}{||d - c||}, \tag{5}$$

where c and d are the vectors of the coordinates that define the maximum Feret diameter of the cell.
The orientation angle can be calculated by

$$\theta = cos^{-1}\left(u_0 \cdot u\right) \tag{6}$$

Quantification of pyramidal cells sizes in 2D

The 2D analysis was performed on the precise same cells as the 3D analysis after removing non-pyramidal cells and outliers. Each identified cell consists of consecutive cell profiles that become a 3D cell entity after being combined,

see **Fig.S4**. The volume of a given cell in 2D can be estimated by using **Eq.7**, where $\bar{l}$ equals the mean segment length from the centroid to the cell boundary, and n equals the number of segments.

$$Volume = \frac{4}{3}\pi\bar{l}_n^3 \tag{7}$$

This method is based on the well-known nucleator probe in stereology, which is used in biological research to estimate mean cell volume for quantitative histology (Gundersen, 1988). This volume obtained by the nucleator probe relies on the mathematical fact that the mean intersection length, $\bar{l}_n$, between a unique point and the cell border by isotropic test lines can be viewed as a radius.
Two procedures were applied to test the 2D-analysis. For the first procedure, the largest cell profile was detected within the stack of profiles for each cell, which is usually near the middle of a cell.
The nucleator probe was then applied on the largest cell profile and five segment lengths were randomly positioned with a spacing between each other of 72° (360°/5), see **Fig.S21**.
For the second procedure, the nucleator probe was employed on every profile of a cell and the average segment length was used to calculate the cell volume.
The diameters for both procedures were estimated by **Eq.8**.

$$Diameter = \bar{l}_n \cdot 2 \tag{8}$$

Point pattern analysis

The statistical analysis was conducted with R (R Core Team, 2019). For each of Subjects 1_1, 1_2, 2 and 3, the 3D coordinates for locations of the centroids of pyramidal cells in layer III of BA46 form a 3D spatial point pattern that was analyzed using statistical methods (Møller and Waagepetersen, 2004). We used the cylindrical K-function (Møller et al., 2016), as was previously done in Rafati et al. (2016) and Christoffersen et al. (2021), to detect columnar structures in each 3D point pattern. In order to use the cylindrical K-function, we assumed that each point pattern was homogeneous. We assessed this assumption by looking at histograms of the projections of data onto the x-, y-, and z-axis and kernel smoothed intensity functions of the projections onto the xy-, xz, and yz-plane. Based on that, the point patterns seemed reasonably homogeneous. We denote the cylindrical K-function as $K_u(r,t)$ which makes it clear that it depends on a direction u (a unit vector in 3D space), a radius r, and a height t, and we estimated it by the non-parametric approximation defined in Møller et al. (2016). Let ρ be the intensity (mean number of points per volume unit). Then $\rho K_u(r,t)$ is interpreted as the expected number

of further points inside a cylinder centered at a 'typical point' (intuitively a randomly selected point of the point process) with direction u, base radius r, and height $2t$, as exemplified in **Fig.S22**. If there is a columnar structure in a point pattern, the estimate of $K_u(r,t)$ is expected to be particularly high for the direction of the columnar structure for a range of r and t values. In order to decide whether $K_u(r,t)$ was significantly high, we compared it to the situation of Complete Spatial Randomness (CSR), meaning that there is no structure in data (a so-called homogeneous Poisson process), using a test called the extreme rank length global envelope test with a corresponding 95% global envelope (Myllymäki et al., 2017, Mrkvička et al., 2020). This global envelope consists of a lower and upper curve such that the empirical cylindrical K-function for data falls entirely between these bounding curves if and only if the global envelope test cannot be dismissed at level 5% (more specifically at approximately 5% because we obtained the envelope based on simulations). When the empirical curve for $K_u(r,t)$ falls above the envelope, it means that it is higher than expected under CSR, which in turn indicates that there are cylinder-shaped clusters in the direction u. If the curve falls below the envelope, it indicates repulsive behavior between the points. In our analysis, we looked at the directions corresponding to the three main axes, and we expected to find a columnar structure in the direction of the x-axis. We considered two situations for the global envelopes: First, we allowed r and t to vary and estimated the cylindrical K-function on a 64×64 grid where $r \in [0,25]$ and $t \in [0,80]$. We used 4000 simulations under CSR for the envelopes in this situation. Second, we fixed $t = 80$, meaning that $K_u(r,t)$ only depends on $r \in [0,25]$. We estimated the function for 64 r-values and used 2000 simulations under CSR for the envelopes. (These numbers of simulations follow the recommendations in the references above)

4.10 Tissue deformation

One biopsy with a diameter of 1.5 *mm* was taken from the grey matter of three human autopsy brains. The tissue area was carefully measured before and after it was dehydrated, embedded, sectioned and stained. The area, A, of the tissue was estimated as:

$$A = \sum P \times (a/p)$$

where $\sum P$ is the number of test points hitting tissue and (a/p) the area associated with each test point (Tang et al., 1997, Dorph-Petersen et al., 2001). The areal shrinkage was estimated as:

$$\text{Areal shrinkage} = [(\text{area before}) - (\text{area after})]/(\text{area before}) \qquad (9)$$

4.11 Statistics and reproducibility

Biopsies were obtained and examined from three subjects (n=3), as stated throughout the article. The average, standard deviation and coefficient of variation were used for the morphological measurements of pyramidal cells in layer III of BA46 for each subject. Calculations of the pixelwise and objectwise performance of the UNetDense architecture are described in the Method section. Figures, tables and histograms of this data were done using custom code via MATLAB. Statistical analysis of the spatial point pattern for each subject was performed using R (R Core Team, 2019) and figures were produced using the package ggplot2 (Wickham, 2016). We employed the extreme rank length global envelope test with level 95% to determine whether $K_u(r,t)$ was significantly different from complete spatial randomness using 4000 simulations when constructing the envelopes. Statistical significance was defined as a p-value less than 0.05. All data sets presented in this work are available for download in our GitHub repository, https://doi.org/10.5281/zenodo.4287469 (Larsen, 2020a,b).

Data availability

The source data used to produce graphs and figures, and tables of quantitative measurements that support the current study's key findings are available from the corresponding author on reasonable request or in the GitHub repository, https://doi.org/10.5281/zenodo.4287469 (Larsen, 2020a,b).

Code availability

Source code of custom MATLAB and R scripts that support the findings of this study with image examples are available in the GitHub repository, https://doi.org/10.5281/zenodo.4287469 (Larsen, 2020a,b).

Acknowledgements

The brains analyzed in this paper are part of a brain series selected by Karl-Anton Dorph-Petersen. We want to thank Ute Hahn for providing us with the idea to sample the biopsies, Markus Kiderlen for help in estimating the orientation of pyramidal cells, and Hans Jacob Teglbjærg Stephensen for updating the deep learning code. We would also like to thank Jingyuan Yang and Mengyue Lou (FS lab) for their help in image acquisition. We are grateful to Junfeng Hao from Laboratory Animal Facility Center, Institute of Biophysics, Chinese Academy of Sciences, for her use in light microscope man-

agement. We also want to thank Andreas Dyreborg Christoffersen for providing us with an R-function for estimating the cylindrical K-function. This work has been supported by the Centre for Stochastic Geometry and Advanced Bioimaging, funded by the Villum Foundation, Sino-Danish Centre and The Danish Council for Independent Research | Natural Sciences, grant DFF – 7014-00074 'Statistics for point processes in space and beyond'. We also acknowledge the P30 GM103328 grant from the National Institutes of Health. We also thank grants from the National Natural Science Foundation of China (31925026) and the Chinese Academy of Sciences (QYZDB-SSW-SMC004).

Authors Contribution

Conception and design: NYL, FS, JRN; Sampling strategy and preparation: NYL, JRN; Optimising the sampling procedure for the AutoCUTS-LM: NYL, XL, XT, GJ, FS; Acquisition of data: NYL, XT; Software/scripts developed for analyzing data: NYL, JL, JS, NV; Analysed the data: NYL, JS, NV, JM; Supervised the project: NYL, GR, FS, JRN; NYL. took the lead in writing the manuscript, and NV contributed to the comprehensive revision of the article. All co-authors provided critical reviews and have accepted the final edition to be published.

A Supplementary Note 1

Improving section quality: avoiding wrinkles

Problems in sectioning can compromise both the sample quality and the reliability of the subsequent analysis. The presence of many wrinkles on the sections can make the data analysis challenging to quantify and can potentially destroy the image dataset. Two types of wrinkles were observed during section collection on tape with the AutoCUTS-LM denoted macro-folds and micro-folds. Macro-folds deform the section and may cover a large area of the ROI. Micro-folds, on the other hand, can appear anywhere on the section. Suggestions will be given on how to prevent and minimize the effect of factors that influence the creation of wrinkles in sections during cutting.

Sample size and tissue heterogeneity

If the sections are small, they develop fewer wrinkles since they have a smaller surface where wrinkles can be generated. In **Fig.S9**, large wrinkles around excess embedding resin surrounding the tissue were generated due to density differences between the sample and the empty resin. Therefore, we recommend removing all excess resin by trimming the block as close to

the sample as possible until there is only a rectangular block of embedded tissue. Nevertheless, it is challenging to remove resin around smaller non-homogeneous samples, such as C. elegans or Drosophila. For them, it is better to reduce the hardness of the resin so that the sections can be easily flattened as they are cut.

Section thickness

The quality of sections depends a lot on the cutting thickness. Ultra-thin sections (30-100 *nm*) generate few or no micro-folds compared to semi-thin (100-500 *nm*) sections. Sections with a cutting thickness above 300 *nm* were more susceptible to generate micro-folds compared to thinner sections when collected on tape, as shown in **Fig.S10**. Furthermore, thick sections generated more wrinkles but provided increased contrast of cells compared to thinner sections, which generated fewer micro-folds but also provided less contrast of cells.

Tape damage

Deformation on the tape, such as scratches or other damage caused by the AutoCUTS-LM reel-to-reel conveyor belt, could result in micro-folds on sections, **Fig.S12**. This problem was avoided by removing the protective coat on the section-collection side during the collection of sections. Other research groups have experienced wrinkles by using an automated tape-collecting ultramicrotome, and it could be caused by tape damage. In our case, we were using transparent tape, and it was easy to spot any damage compared to the solid coating tape used for Scanning EM.

Humidity and hydrophilicity

One of the main factors that could cause wrinkles on a section was the indoor humidity level. The tissue in **Fig.S10** was sampled with a humidity level around 10-20% because of the drying effect of the air condition in the room. If the tape's surface was exceptionally hydrophilic and the indoor environment was dry, it could cause any section to adhere without time to expand, see **Fig.S13**. Indoor temperatures above 20°C and a humidity level above 60% have a fundamental impact in decreasing the flexural and compressive strength of the epoxy. Hence, a softer resin reduces the amount of naturally generated wrinkles for semi-thin sections. We purchased a humidifier (YADU Company, China) to monitor and retain a humidity level of between 80-90% during section collections. Sections with a cutting thickness above 300 *nm* were more susceptible to generate wrinkles during the cutting process, as shown in **Fig.S11**. This was caused by a combination of the humidity in the room, section thickness and hydrophilicity of the tape. Nevertheless, a

too high humidity level (>90%) can have a devastating effect on the samples. This is shown in **Fig.S14**, where some areas of the section can adhere to the tape due to the hydrophilicity and other areas might move around on the tape. This will cause the section to be unstable and generate many folds. We suggest keeping the humidity level around 85% to avoid folds.

B Supplementary Tables

Brain	Window (μm)	Nr. Total	Nr. Pyramidal	Nr. Non-pyramidal	Nr. Outliers	Excluded (%)	Pyramidal cell density(mm^{-3})
Subject 1_1	$635 \times 1013 \times 334$	8872	6864	2008	0	23	31952
Subject 1_2	$507 \times 1001 \times 511$	12587	9501	3085	1	25	32569
Subject 2	$488 \times 1216 \times 682$	17534	11111	6404	19	37	27455
Subject 3	$664 \times 1055 \times 725$	16393	12586	3779	28	23	24764
Mean							28160±3101
CV							0.11

Table S1: Measurements and classification of identified objects for the entire stack of images for each subject. The dimensions of the observation windows (x,y,z), the total number of cells/objects, pyramidal cells, non-pyramidal cells, outliers. The density was calculated by the total number of pyramidal cells pr. volume for each subject of layer III. Excluded (%) is percentage of excluded datapoints for each subject, where Excluded (%)=(Non-pyramidal+outliers)/Total·100. The entries for the pyramidal cell density state the mean and ± one standard deviation. Mean is the average measurement at n=3, Coefficient of variation CV = SD/mean. The average for Subject 1_1 and Subject 1_2 were used for subject 1.

Brain	Vol_{3D} (μm^3)	Vol_L (μm^3)	Difference (%)	Vol_{3D} (μm^3)	Vol_{All} (μm^3)	Difference (%)
Subject 1	867	721	17	867	198	77
Subject 2	709	721	1.7	709	143	80
Subject 3	808	749	7.3	808	210	74
Mean	795 ± 65	730 ± 13	8.7 ± 6.3	795 ± 65	183 ± 29	77 ± 2.45
CV	0.08	0.18	0.72	0.08	0.06	0.03

Table S2: The table shows the volume comparison between 2D vs 3D approches from the values in Table 4. Vol_{3D} is the directly measured volume based on voxel counts. Vol_L and Vol_{All} are calculated from the average segment length measured from the nucleator probe. Difference (%) is the relative change in percentage Diff =$(Vol_x - Vol_{3D})/Vol_{3D} \cdot 100$ where Vol_x is either Vol_L or Vol_{All}, Mean is the average measurement for each column, Coefficient of variation CV = SD/mean.

Brain	Dia_{3D} (μ)	Dia_{L} (μ)	Difference (%)	Dia_{3D} (μ)	Dia_{All} (μ)	Difference (%)
Subject 1	11.83	11.13	5.9	11.83	7.23	39
Subject 2	11.06	11.13	0.6	11.06	6.49	41
Subject 3	11.56	11.27	2.5	11.56	7.37	36
Mean	11.48 ± 0.32	11.17 ± 0.07	2.9 ± 2	11.48 ± 0.32	7.03 ± 0.39	39 ± 2
CV	0.03	0.01	0.68	0.03	0.06	0.05

Table S3: The table shows the volume comparison between 2D vs 3D approaches from the values in Table 4. Dia_{3D} is the equivalent diameter of an approximated sphere equal to Vol_{3D}. Dia_{L} and Dia_{All} are the average segment length measured from the nucleator probe. Difference (%) is the relative change in percentage Diff $=(Dia_x - Dia_{3D})/Dia_{3D} \cdot 100$ where Dia_x is either Dia_{L} or Dia_{All}, Mean is the average measurement for each column, Coefficient of variation CV = SD/mean.

C Supplementary Figures

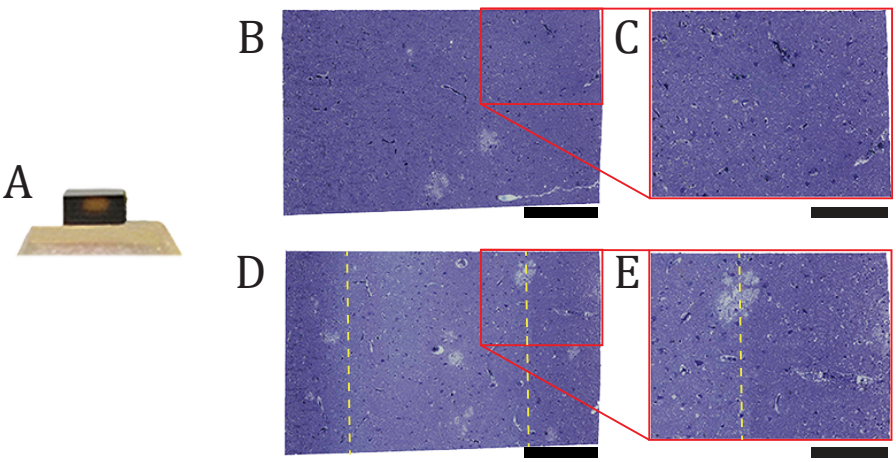

Fig. S1: The influence of osmium staining and section contrast. Osmium has not entirely infiltrated the entire biopsy and gives us the ideal opportunity to test its effect on sections that need to be later stained with toluidine blue. **(A)** Biopsy stained with osmium(black color) where no osmium was present in the middle part of the block. **(B)** Biopsy section with osmium from the top part of the block stained with toluidine blue. Scale bar=400 μm. **(C)** Magnified view of the red box where the contrast between the neurons and the background is not evident. Scale bar=200 μm. **(D)** Biopsy section stained with toluidine blue from the middle part of the block, where osmium only appears in the right and left side of the section (marked with yellow lines). Scale bar=400 μm.**(E)** Magnified view of the red box for comparing the contrast between neurons and background in areas with and without osmium (right and left side of the yellow line, respectively). Scale bar=200 μm.

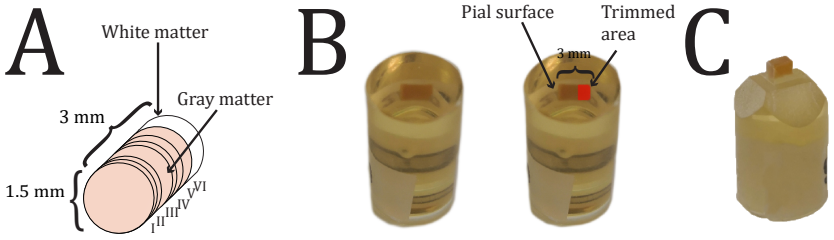

Fig. S2: Sample embedding. **(A)** Illustration of the biopsy with a diameter of 1.5 *mm* and a length of ∼3 *mm*. **(B)** Sample inside the epoxy resin. The red area indicates the trimmed part of the biopsy after the ROI was located from the outermost section. **(C)** Trimmed block of sample.

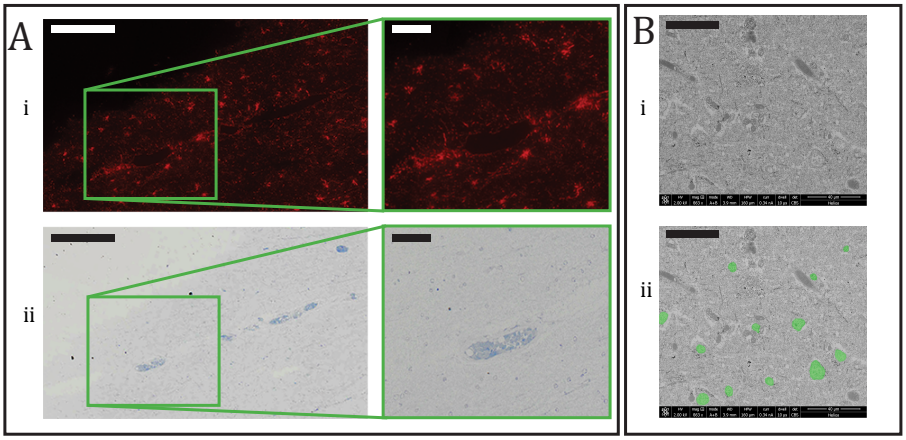

Fig. S3: Light microscopy and EM imaging on biopsy from BA46. **(A)** Immunolabeled and toluidine blue-stained section. **(i)** Fluorescently immuno-label staining of cells containing glial fibrillary acid protein was observed with a red channel.**(ii)** Toluidine blue staining on the same sections. The stained vessel served as an alignment reference to combine the first and second image. Overview image: scale bar=200 μm. Zoomed in image: scale bar=50 μm. **(B)** EM imaging on a section. **(i)** Overview image with a horizontal field width (HFW) that is 160 μm. **(ii)** Neurons were manually marked with green color as an overlay on the original image. Scale bar=40 μm

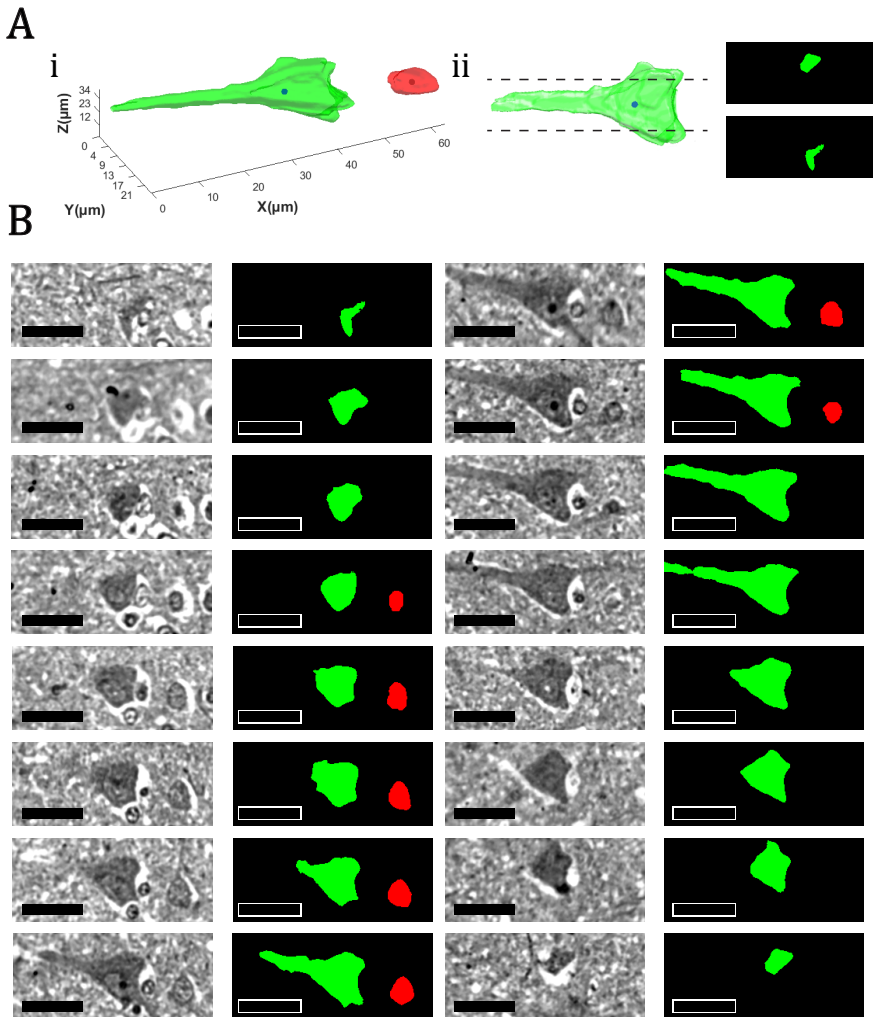

Fig. S4: Demonstrate that additional cell filtering is necessary based on 3D measurements. **(A)** 3D-reconstruct of a series of image profiles from pyramidal (green) and a non-pyramidal (red) cell (i). If the reconstruction is not accomplished, single image profiles near the edge can imitate sections of smaller cells/gial cells (ii). **(B)** Pyramidal (green) and non-pyramidal (red) cell image profiles were used to create the 3D reconstruction. Scale bar=20 μm.

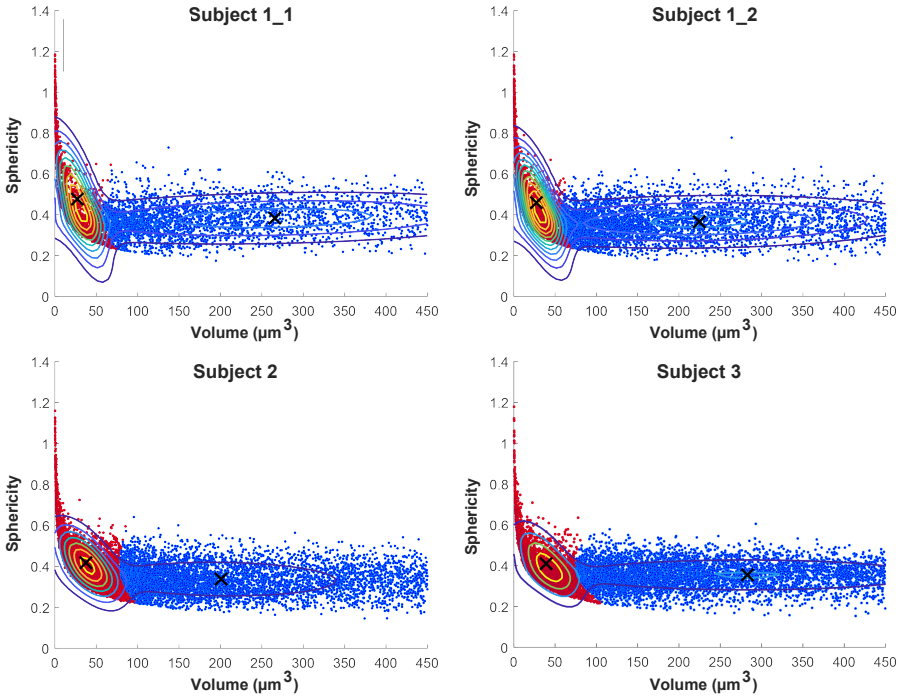

Fig. S5: Displaying filtration of data. K-means clustering algorithm was performed to filter smaller neurons and artefacts from the dataset. The data was segmented into non-pyramidal and pyramidal neurons employing estimations of neuronal volume (x-axis) against sphericity (y-axis). For each subject, The data points for non-pyramidal and pyramidal cells are represented by the red and blue points, respectively.

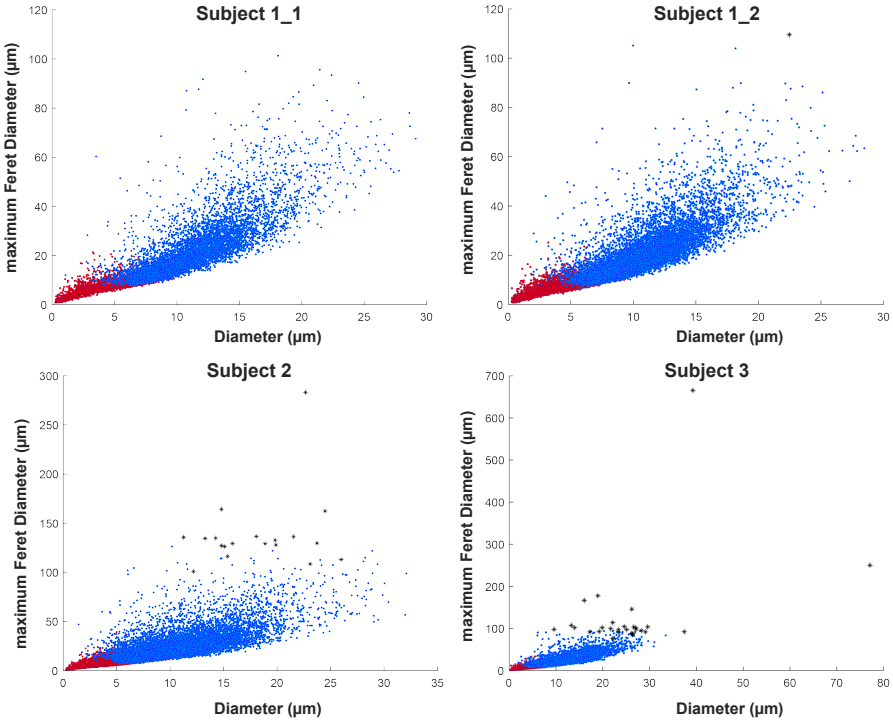

Fig. S6: Displaying points for pyramidal, non-pyramidal and outliers for each subject. The graphs show plots of the largest cell profile (x-axis) against the maximum Feret Diameter in 3D (y-axis) of each neuron. For each subject, the data points for pyramidal cells are shown as blue points, while the red and black points are non-pyramidal cells, as classified with the k-means method, and outliers, respectively.

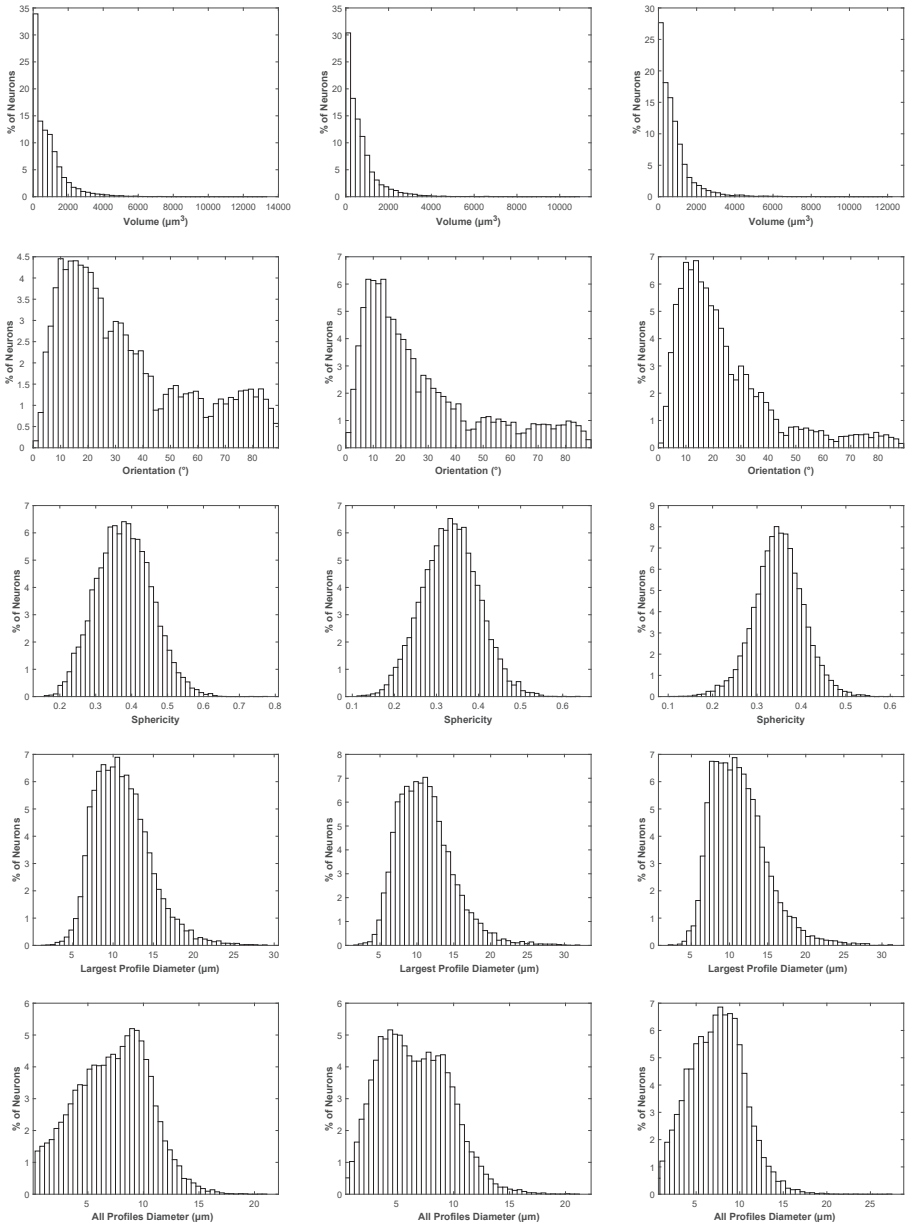

Fig. S7: Quantitative measurements of pyramidal neurons. Histograms showing neuronal volume, sphericity, orientation and diameter of pyramidal cells in BA46 in layer III for each subject. The columns correspond to Subjects 1, 2, and 3, respectively.

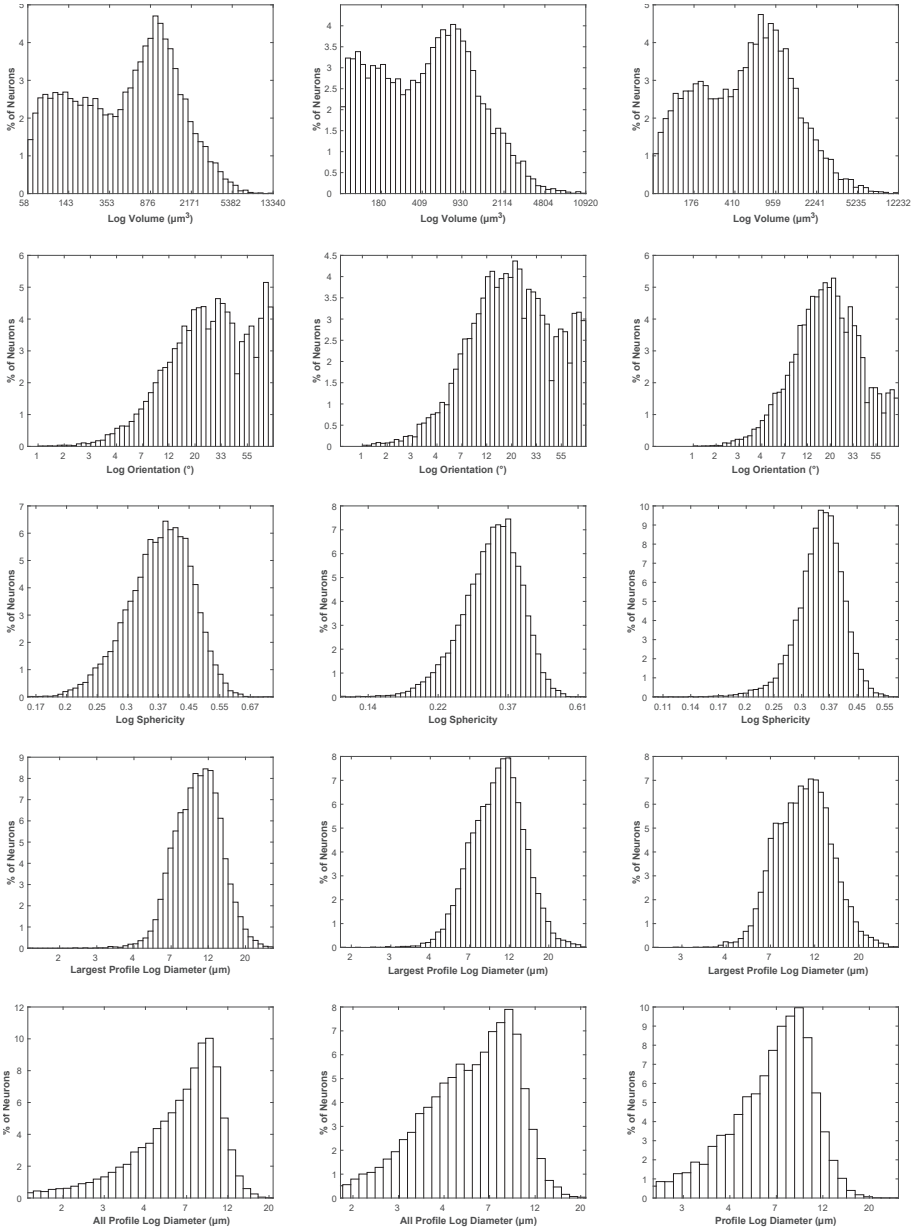

Fig. S8: Quantitative measurements of pyramidal neurons log-normal transformed. Histograms showing the log-normal transformed neuronal volume, sphericity, orientation and diameter of pyramidal cells in BA46 in layer III for each subject. The columns correspond to Subjects 1, 2, and 3, respectively.

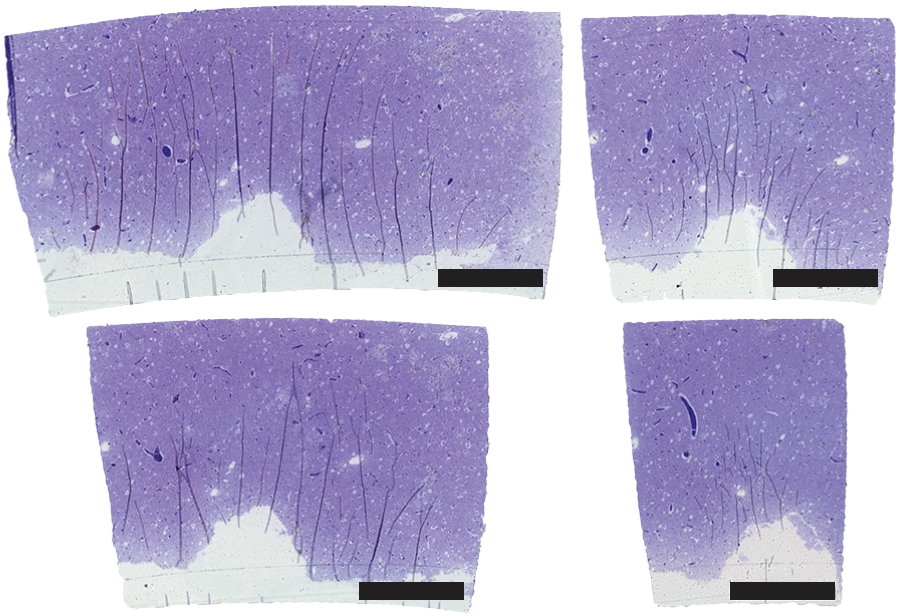

Fig. S9: The connection between cutting surface density and folds. A section (2.8x1.3 *mm*) was trimmed four times to visualise the relationship between section area and macro-folds. Macro-folds will most likely be generated in more extended sections and around the blank resin area. The room temperature and indoor humidity were, 20°C and 50%, respectively. Scale bar=500 μm.

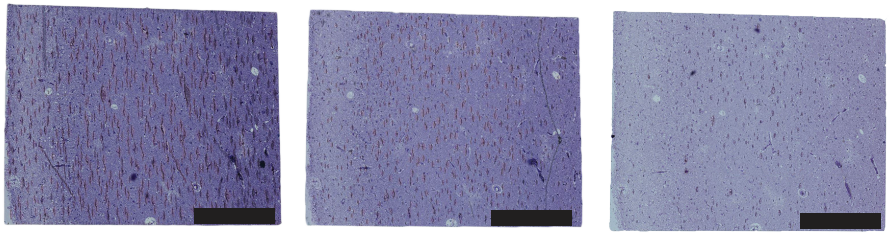

Fig. S10: The connection between section thickness and folds. The number of folds and staining penetration was reduced along with the cutting thickness. The sections were stained all together with toluidine for 15 mins. The room temperature and indoor humidity were, respectively, 20°C and 10%. Scale bar=400 μm.

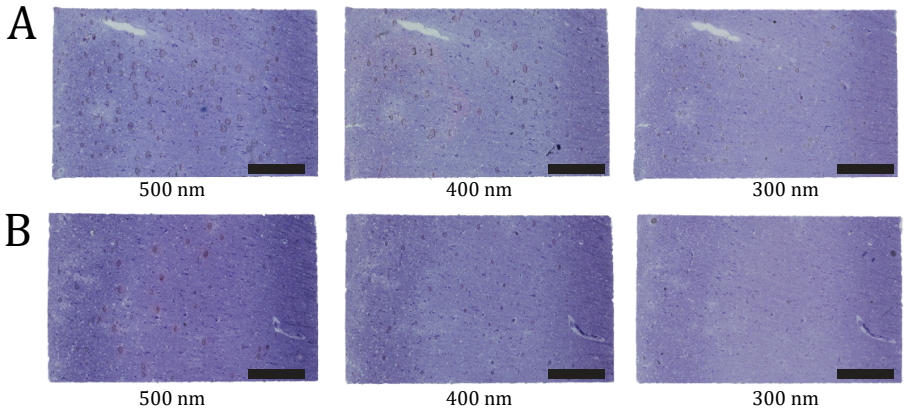

Fig. S11: Test sections with a section thickness that range from 500 to 300 *nm* with different indoor humidity. **(A)** Sections were cut with an indoor humidity of 60% and thinner sections generate fewer folds. **(B)** Sections were cut with a humidity between 85-90% and there are few to none folds on the 300 *nm* thick section. All sections were stained together with toluidine blue for 15 min in room temperature at the same time. Scale bar=400 μm.

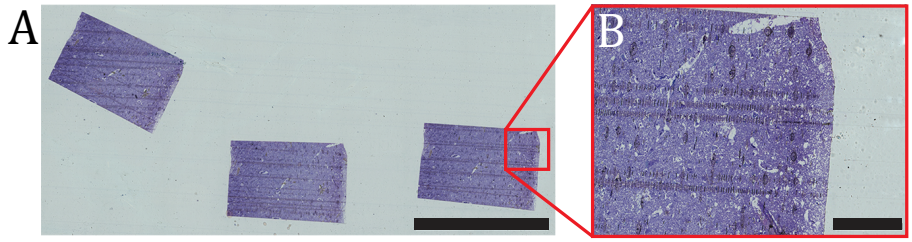

Fig. S12: Damage on tape caused by AutoCUTS-LM could generate folds on the sections due to the deformation of the tape. **(A)** The damage to the tape creates pronounced lines that impact the three collected sections. Scale bar=2 *mm*.**(B)** The three visible lines were on the underside of the tape and not on the upper-side where the sections were collected. Scale bar=200 μm.

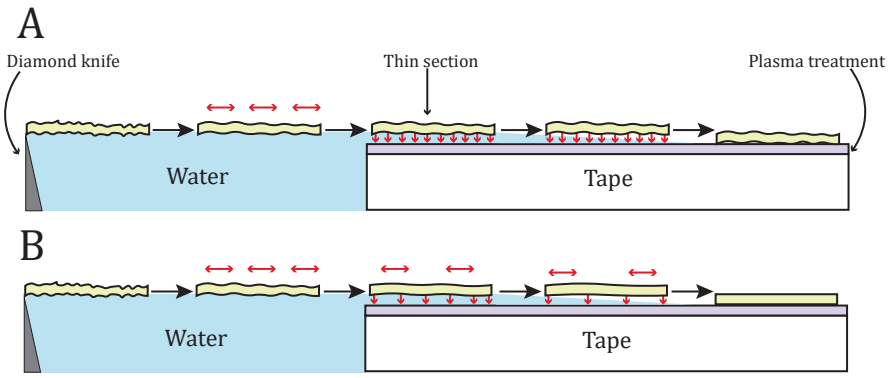

Fig. S13: Illustration of section flatness caused by humidity and tape adhesion. After a section was cut with the diamond knife, it would expand in the water. **(A)** In a dry environment, sections did not have enough time to expand since they adhere to the tape almost immediately after the cutting. **(B)** Condensation on the tape surface can develop when the indoor humidity is raised, allowing the sections to expand.

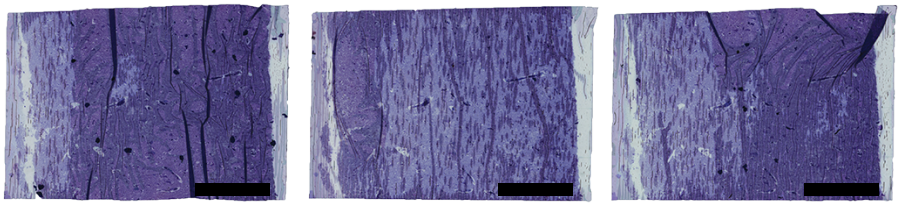

Fig. S14: The impact of sampling sections with too high humidity. High humidity above 90% made the sections unstable as they fell of the collection tape. One part of the section may have been attached to the more dry sections of the tape, and some pieces of the section may have moved around. Due to the unstableness of the sections during the set, several folds were formed. Scale bar=500 μm.

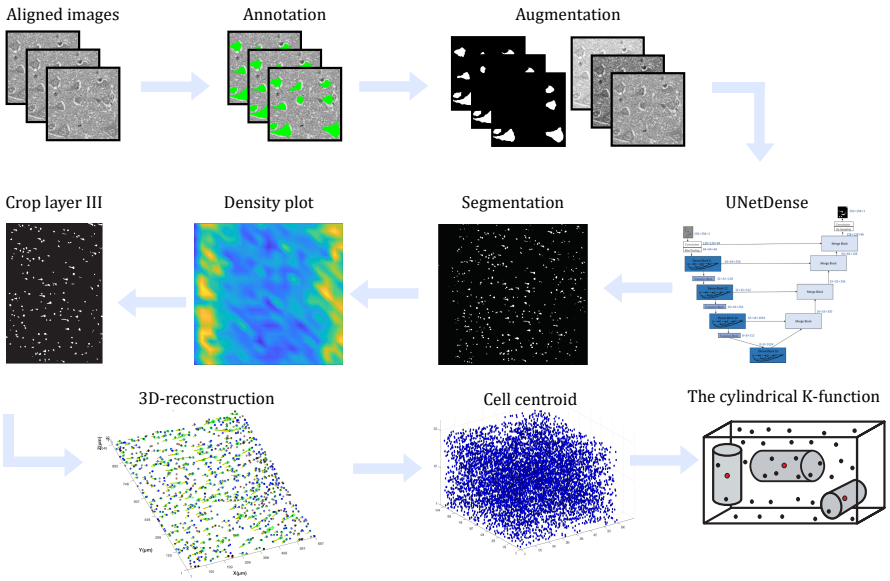

Fig. S15: The data analysis pipeline for this study. First, we would manually annotate pyramidal cells from 35 images. To produce the training (5600) and validation (1400) set, each of those 35 images was subsequently augmented into 200 images with a scale of 256x256 pixels. Based on those images, the UNetDense model trained itself and segmented neurons for each subject. A density map of the centroids was used to identify the ROI and crop the stack of images. Pyramidal cell parameters were then determined based on their 3D-reconstruction, and the 3D point pattern consisting of the centroids was analyzed with the cylindrical *K*-function in order to detect columnar structures.

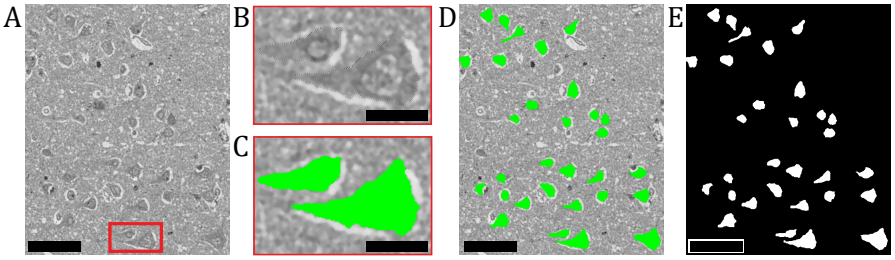

Fig. S16: Manual annotation of pyramidal cells. **(A)** displays the original picture and the area we want to annotate is displayed in the red boundary frame. Scale bar=35 μm. **(B)** Photoshop's fast selection tool was used to identify the pyramidal cell annotation region. Scale bar=10 μm. **(C)** A green-colored mask was placed on top of the original image of each pyramidal cell. Scale bar=10 μm. **(D)** The annotation of cells was repeated for the rest of the image in **A** until all cells were labeled in green. Scale bar=35 μm.**(E)** mask images were translated to binary images in MATLAB where the pixel value of the annotated cells was equal to 1. Scale bar=35 μm.

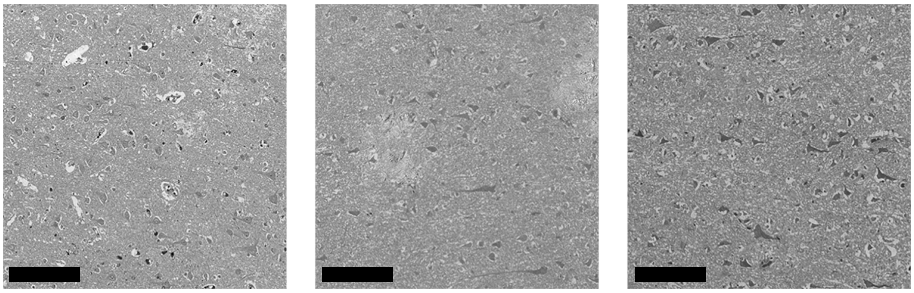

Fig. S17: Validation images that were used to test the combined and individual UNetDense models. These images show the differences between sections from different subjects after staining. The images from left to right correspond to Subjects 1, 2, and 3, respectively. Scale bar=140 μm.

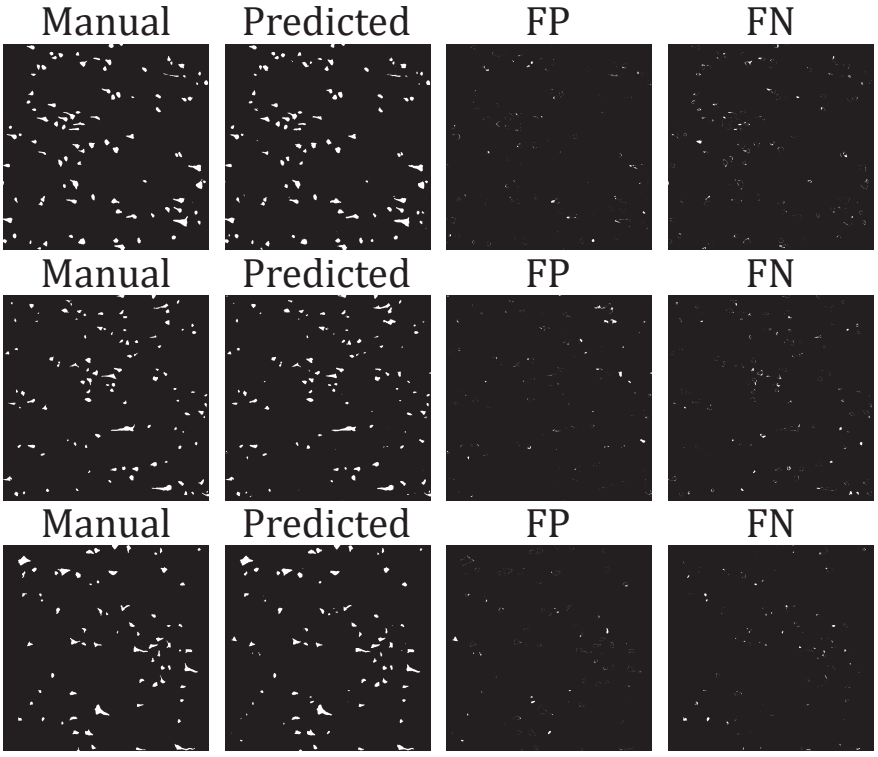

Fig. S18: Test of performance of individual UNetDense models on validation images. Pyramidal cells from **Fig.S17** were manually marked by (NYL), and the same images were processed using individual UNetDense models to predict and segment pyramidal cells. The figure shows pixels which have been measured as FP, FN, and quantitative comparisons can be found in **Table 1**. The rows correspond to Subjects 1, 2, and 3, respectively.

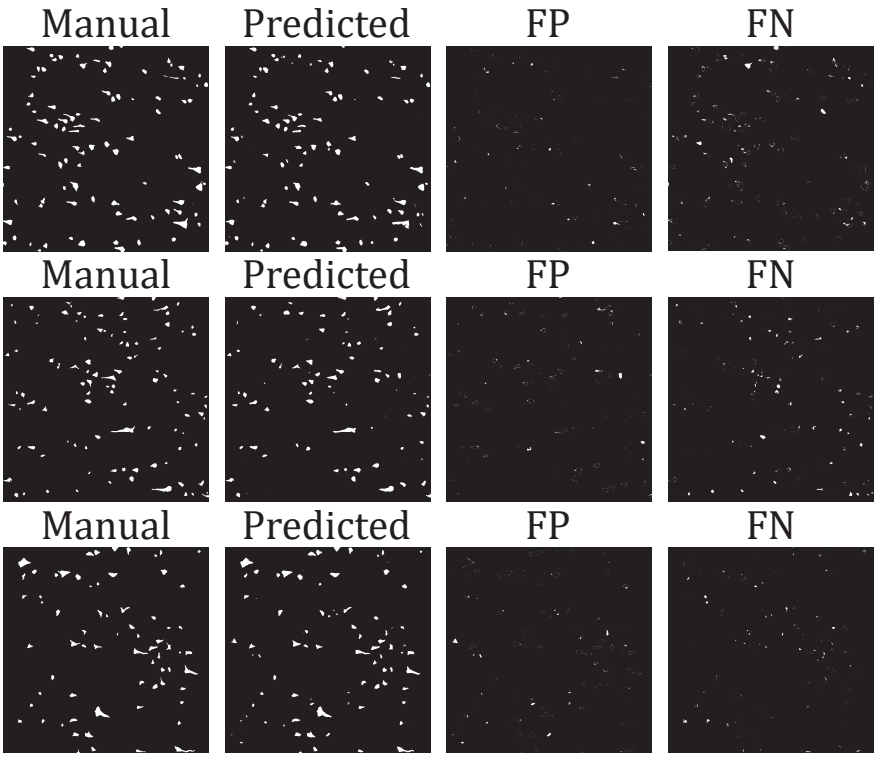

Fig. S19: Test of performance of combined UNetDense model on validation images. Pyramidal cells from **Fig.S17** were manually marked by (NYL), and the same images were processed using the combined UNetDense model to predict and segment pyramidal cells. The figure shows pixels which have been measured as FP, FN, and quantitative comparisons can be found in **Table 1**. The rows correspond to Subjects 1, 2, and 3, respectively.

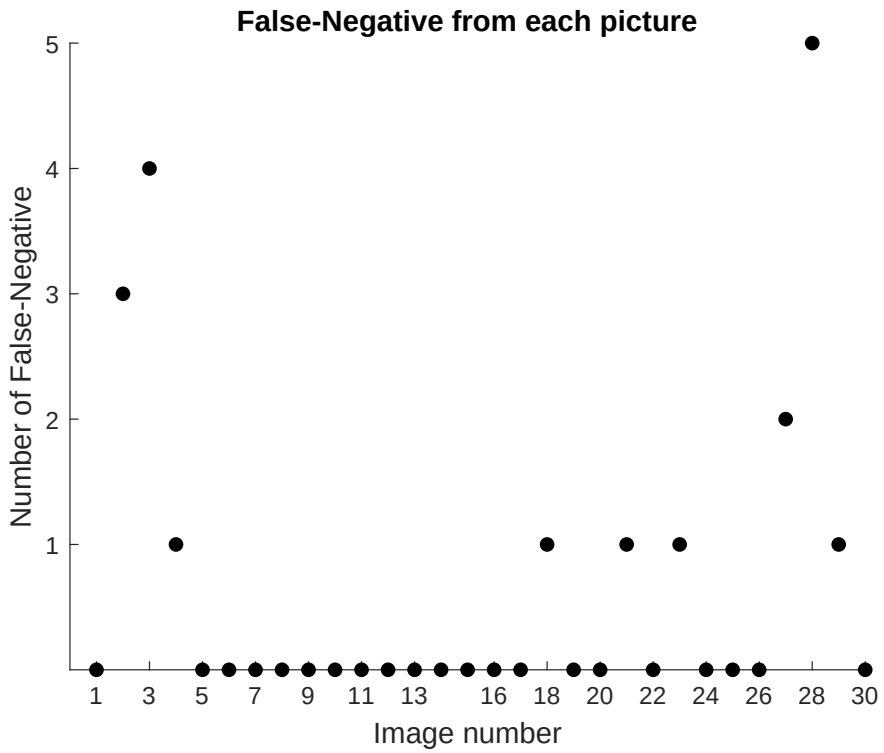

Fig. S20: Detection of FN centroids from the objectwise validation. Number of FN centroid points from 30 MS images that was not detected in 30 UDP images(FN cases). The graph shows the number of FN cases in each picture.

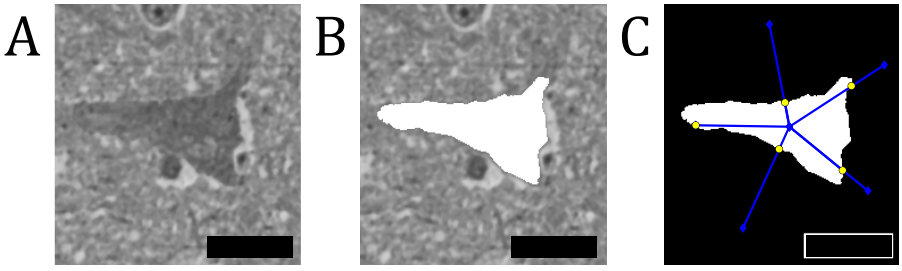

Fig. S21: 2D measurement of each neuron using the nucleator. **(A)** Gray-scale image of a cell profile. Scale bar=20 μm. **(B)** Overlay image of the gray-scale image and the segmented image output from UNetDens algorithm. **(C)**The nucleator probe was applied to a cell-profile. In this case, the reference point was the centroid. A random boundary point of the cell was selected, whereas five segments with a spacing between each other of 72° (360°/5) was superimposed on top of a cell. The average line segment was then calculated to estimate the cell average radius.

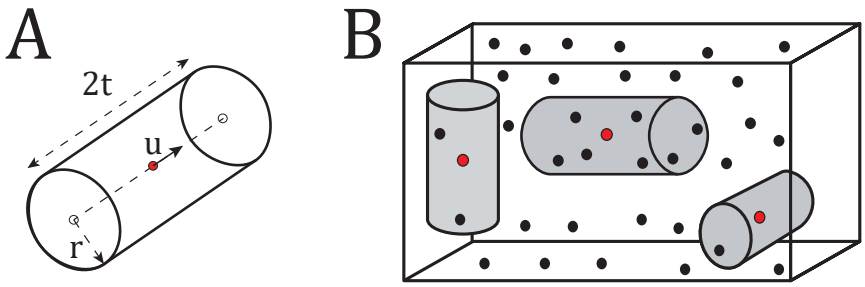

Fig. S22: Illustrations of the cylinders of the cylindrical *K*-function. **(A)** A cylinder with radius r, height $2t$, and direction u, which is the structuring element of the cylindrical K-function. **(B)** Three different cylinders of the same size directed along the x-, y-, and z-axis, respectively, and centered at a randomly selected point of the point pattern.

References

Amunts, K., Lepage, C., Borgeat, L., Mohlberg, H., Dickscheid, T., Étienne Rousseau, M., Bludau, S., Bazin, P.-L., Lewis, L. B., Oros-Peusquens, A.-M., Shah, N. J., Lippert, T., Zilles, K., and Evans, A. C. (2013). BigBrain: An ultrahigh-resolution 3d human brain model. *Science*, 340(6139):1472–1475.

Ariel, P. (2017). A beginner's guide to tissue clearing. *The International Journal of Biochemistry & Cell Biology*, 84:35–39.

Baccetti, B., Collodel, G., Gambera, L., Moretti, E., Serafini, F., and Piomboni, P. (2005). Fluorescence in situ hybridization and molecular studies in infertile men with dysplasia of the fibrous sheath. *Fertility and Sterility*, 84(1):123–129.

Becker, K., Jährling, N., Saghafi, S., Weiler, R., and Dodt, H. U. (2012). Chemical clearing and dehydration of GFP expressing mouse brains. *PLoS ONE*, 7(3):e33916.

Benes, F. M. (1986). Quantitative cytoarchitectural studies of the cerebral cortex of schizophrenics. *Archives of General Psychiatry*, 43(1):31.

Brændgaard, H. and Gundersen, H. J. G. (1986). The impact of recent stereological advances on quantitative studies of the nervous system. *Journal of Neuroscience Methods*, 18(1-2):39–78.

Bundgaard, M. J., Regeur, L., Gundersen, H. J. G., and Pakkenberg, B. (2001). Size of neocortical neurons in control subjects and in alzheimer's disease. *Journal of Anatomy*, 198(4):481–489.

Burel, A., Lavault, M.-T., Chevalier, C., Gnaegi, H., Prigent, S., Mucciolo, A., Dutertre, S., Humbel, B. M., Guillaudeux, T., and Kolotuev, I. (2018). A targeted 3d EM and correlative microscopy method using SEM array tomography. *Development*, 145(12):dev160879.

Casanova, M. (2013). Neural mechanisms in autism. In *Encyclopedia of Autism Spectrum Disorders*, pages 1994–2007. Springer New York.

Casanova, M. F., Buxhoeveden, D., and Gomez, J. (2003). Disruption in the inhibitory architecture of the cell minicolumn: Implications for autisim. *The Neuroscientist*, 9(6):496–507.

Casanova, M. F. and Casanova, E. L. (2018). The modular organization of the cerebral cortex: Evolutionary significance and possible links to neurodevelopmental conditions. *Journal of Comparative Neurology*, 527(10):1720–1730.

Chance, S. A., Casanova, M. F., Switala, A. E., and Crow, T. J. (2008). Auditory cortex asymmetry, altered minicolumn spacing and absence of ageing effects in schizophrenia. *Brain*, 131(12):3178–3192.

Christoffersen, A. D., Møller, J., and Christensen, H. S. (2021). Modelling columnarity of pyramidal cells in the human cerebral cortex. *Australian & New Zealand Journal of Statistics*, 63(1):33–54.

Chung, K. and Deisseroth, K. (2013). CLARITY for mapping the nervous system. *Nature Methods*, 10(6):508–513.

Cruz, D. A., Eggan, S. M., Azmitia, E. C., and Lewis, D. A. (2004). Serotonin1areceptors at the axon initial segment of prefrontal pyramidal neurons in schizophrenia. *American Journal of Psychiatry*, 161(4):739–742.

Cullen, T. J., Walker, M. A., Eastwood, S. L., Esiri, M. M., Harrison, P. J., and Crow, T. J. (2006). Anomalies of asymmetry of pyramidal cell density and structure in dorsolateral prefrontal cortex in schizophrenia. *British Journal of Psychiatry*, 188(1):26–31.

Dean, B., Gibbons, A. S., Tawadros, N., Brooks, L., Everall, I. P., and Scarr, E. (2012). Different changes in cortical tumor necrosis factor-α-related pathways in schizophrenia and mood disorders. *Molecular Psychiatry*, 18(7):767–773.

Dean, B., Tawadros, N., Seo, M. S., Jeon, W. J., Everall, I., Scarr, E., and Gibbons, A. (2014). Lower cortical serotonin 2a receptors in major depressive disorder, suicide and in rats after administration of imipramine. *The International Journal of Neuropsychopharmacology*, 17(06):895–906.

Denk, W., Horstmann, H., and Harris, K. M. (2004). Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. *PLoS Biology*, 2(11):e329.

Dorph-Petersen, K.-A., Nyengaard, J. R., and Gundersen, H. J. G. (2001). Tissue shrinkage and unbiased stereological estimation of particle number and size. *Journal of Microscopy*, 204(3):232–246.

Escovitz, W. H., Fox, T. R., and Levi-Setti, R. (1975). Scanning transmission ion microscope with a field ion source. *Proceedings of the National Academy of Sciences*, 72(5):1826–1828.

Fleischhauer, K., Petsche, H., and Wittkowski, W. (1972). Vertical bundles of dendrites in the neocortex. *Zeitschrift for Anatomie und Entwicklungsgeschichte*, 136(2):213–223.

Fuster, J. (1997). *The prefrontal cortex : anatomy, physiology, and neuropsychology of the frontal lobe*. Lippincott-Raven, Philadelphia.

Gibbons, A. S., Scarr, E., McLean, C., Sundram, S., and Dean, B. (2009). Decreased muscarinic receptor binding in the frontal cortex of bipolar disorder and major depressive disorder subjects. *Journal of Affective Disorders*, 116(3):184–191.

Glantz, L. A. and Lewis, D. A. (2000). Decreased dendritic spine density on prefrontal cortical pyramidal neurons in schizophrenia. *Archives of General Psychiatry*, 57(1):65.

Gong, H., Zeng, S., Yan, C., Lv, X., Yang, Z., Xu, T., Feng, Z., Ding, W., Qi, X., Li, A., Wu, J., and Luo, Q. (2013). Continuously tracing brain-wide long-distance axonal projections in mice at a one-micron voxel resolution. *NeuroImage*, 74:87–98.

Gundersen, H. J. G. (1988). The nucleator. *Journal of Microscopy*, 151(1):3–21.

Hawkins, J. and Ahmad, S. (2016). Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in Neural Circuits*, 10.

Hayworth, K. J., Morgan, J. L., Schalek, R., Berger, D. R., Hildebrand, D. G. C., and Lichtman, J. W. (2014). Imaging ATUM ultrathin section libraries with WaferMapper: a multi-scale approach to EM reconstruction of neural circuits. *Frontiers in Neural Circuits*, 8.

Hildebrand, D. G. C., Cicconet, M., Torres, R. M., Choi, W., Quan, T. M., Moon, J., Wetzel, A. W., Champion, A. S., Graham, B. J., Randlett, O., Plummer, G. S., Portugues, R., Bianco, I. H., Saalfeld, S., Baden, A. D., Lillaney, K., Burns, R., Vogelstein, J. T., Schier, A. F., Lee, W.-C. A., Jeong,

W.-K., Lichtman, J. W., and Engert, F. (2017). Whole-brain serial-section electron microscopy in larval zebrafish. *Nature*, 545(7654):345–349.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Jansen, P., Giehl, K., Nyengaard, J. R., Teng, K., Lioubinski, O., Sjoegaard, S. S., Breiderhoff, T., Gotthardt, M., Lin, F., Eilers, A., Petersen, C. M., Lewin, G. R., Hempstead, B. L., Willnow, T. E., and Nykjaer, A. (2007). Roles for the pro-neurotrophin receptor sortilin in neuronal development, aging and brain injury. *Nature Neuroscience*, 10(11):1449–1457.

Janson, A. M. and Møller, A. (1993). Chronic nicotine treatment counteracts nigral cell loss induced by a partial mesodiencephalic hemitransection: An analysis of the total number and mean volume of neurons and glia in substantia nigra of the male rat. *Neuroscience*, 57(4):931–941.

Keller, P. J. and Dodt, H. U. (2012). Light sheet microscopy of living or cleared specimens. *Current Opinion in Neurobiology*, 22(1):138–143.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

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

Kubota, Y., Sohn, J., Hatada, S., Schurr, M., Straehle, J., Gour, A., Neujahr, R., Miki, T., Mikula, S., and Kawaguchi, Y. (2018). A carbon nanotube tape for serial-section electron microscopy of brain ultrastructure. *Nature Communications*, 9(1).

Lai, H. M., Liu, A. K. L., Ng, H. H. M., Goldfinger, M. H., Chau, T. W., DeFelice, J., Tilley, B. S., Wong, W. M., Wu, W., and Gentleman, S. M. (2018). Next generation histology methods for three-dimensional imaging of fresh and archival human brain tissues. *Nature Communications*, 9(1).

Larsen, N. Y. (2020a). Autoctus-lm. GitHub repository: https://github.com/Nick7900/AutoCUTS-LM-Analysis.

Larsen, N. Y. (2020b). Autocuts-lm-analysis. Zenodo: https://doi.org/10.5281/zenodo.4287469.

Levitt, J. B., Lewis, D. A., Yoshioka, T., and Lund, J. S. (1993). Topography of pyramidal neuron intrinsic connections in macaque monkey prefrontal cortex (areas 9 and 46). *The Journal of Comparative Neurology*, 338(3):360–376.

Li, A., Gong, H., Zhang, B., Wang, Q., Yan, C., Wu, J., Liu, Q., Zeng, S., and Luo, Q. (2010). Micro-optical sectioning tomography to obtain a high-resolution atlas of the mouse brain. *Science*, 330(6009):1404–1408.

Li, X., Ji, G., Chen, X., Ding, W., Sun, L., Xu, W., Han, H., and Sun, F. (2017). Large scale three-dimensional reconstruction of an entire caenorhabditis elegans larva using AutoCUTS-SEM. *Journal of Structural Biology*, 200(2):87–96.

Lin, J. (2019). Pyramidal-cells-segmentation. GitHub repository: https://github.com/JingLin0/Pyramidal-Cells-Segmentation.

Liu, X., Guo, X., Niu, L., Li, X., Sun, F., Hu, J., Wang, X., and Shen, K. (2019). Atlastin-1 regulates morphology and function of endoplasmic reticulum in dendrites. *Nature Communications*, 10(1).

Lyck, L., Dalmau, I., Chemnitz, J., Finsen, B., and Schrøder, H. D. (2007). Immunohistochemical markers for quantitative studies of neurons and glia in human neocortex. *Journal of Histochemistry & Cytochemistry*, 56(3):201–221.

Malarvizhi, C. (2017). Segmentation by thresholding on medical imaging – a survey. *International Journal for Research in Applied Science and Engineering Technology*, V(IX):1218–1220.

Marx, V. (2016). Optimizing probes to image cleared tissue. *Nature Methods*, 13(3):205–209.

Mayerich, D., Abbott, L., and McCormick, B. (2008). Knife-edge scanning microscopy for imaging and reconstruction of three-dimensional anatomical structures of the mouse brain. *Journal of Microscopy*, 231(1):134–143.

McKavanagh, R., Buckley, E., and Chance, S. A. (2015). Wider minicolumns in autism: a neural basis for altered processing? *Brain*, 138(7):2034–2045.

Møller, J., Safavimanesh, F., and Rasmussen, J. G. (2016). The cylindrical *K*-function and poisson line cluster point processes. *Biometrika*, 103:937–954.

Møller, J. and Waagepetersen, R. P. (2004). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman and Hall/CRC, Boca Raton.

Molnár, Z. and Rockland, K. S. (2020). Cortical columns. In *Neural Circuit and Cognitive Development*, pages 103–126. Elsevier.

Mountcastle, V. (1997). The columnar organization of the neocortex. *Brain*, 120(4):701–722.

Mountcastle, V. B. (1957). Modality and topographic properties of single neurons of cat 's somatic sensory cortex. *Journal of Neurophysiology*, 20(4):408–434.

Mrkvička, T., Myllymäki, M., Jilik, M., and Hahn, U. (2020). A one-way ANOVA test for functional data with graphical interpretation. *Kybernetika*, 56:432–458.

Myllymäki, M., Mrkvička, T., Grabarnik, P., Seijo, H., and Hahn, U. (2017). Global envelope tests for spatial processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79:381–404.

Oorschot, D. E. (1994). Are you using neuronal densities, synaptic densities or neurochemical densities as your definitive data? there is a better way to go. *Progress in Neurobiology*, 44(3):233–247.

Opris, I., Chang, S., and Noga, B. R. (2017). What is the evidence for interlaminar integration in a prefrontal cortical minicolumn? *Frontiers in Neuroanatomy*, 11.

Osten, P. and Margrie, T. W. (2013). Mapping brain circuitry with a light microscope. *Nature Methods*, 10(6):515–523.

Peters, A. and Sethares, C. (1996). Myelinated axons and the pyramidal cell modules in monkey primary visual cortex. *The Journal of Comparative Neurology*, 365(2):232–255.

Peters, A. and Walsh, T. M. (1972). A study of the organization of apical dendrites in the somatic sensory cortex of the rat. *The Journal of Comparative Neurology*, 144(3):253–268.

Rafati, A. H., Safavimanesh, F., DORPH-PETERSEN, K.-A., Rasmussen, J. G., Møller, J., and Nyengaard, J. R. (2016). Detection and spatial characterization of minicolumnarity in the human cerebral cortex. *Journal of Microscopy*, 261:115–126.

Raghanti, M. A. (2010). A comparative perspective on minicolumns and inhibitory GABAergic interneurons in the neocortex. *Frontiers in Neuroanatomy*.

Rajkowska, G. and Goldman-Rakic, P. S. (1995a). Cytoarchitectonic definition of prefrontal areas in the normal human cortex: I. remapping of areas 9 and 46 using quantitative criteria. *Cerebral Cortex*, 5(4):307–322.

Rajkowska, G. and Goldman-Rakic, P. S. (1995b). Cytoarchitectonic definition of prefrontal areas in the normal human cortex: II. variability in locations of areas 9 and 46 and relationship to the talairach coordinate system. *Cerebral Cortex*, 5(4):323–337.

Rajkowska, G., Selemon, L. D., and Goldman-Rakic, P. S. (1998). Neuronal and glial somal size in the prefrontal cortex. *Archives of General Psychiatry*, 55(3):215.

Rawat, W. and Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, pages 234–241. Springer International Publishing.

Rosenfeld, A. and Kak, A. C. (1982). *Digital Picture Processing*. Academic Press, Inc., USA, 2nd edition.

Rudow, G., O'Brien, R., Savonenko, A. V., Resnick, S. M., Zonderman, A. B., Pletnikova, O., Marsh, L., Dawson, T. M., Crain, B. J., West, M. J., and Troncoso, J. C. (2008). Morphometry of the human substantia nigra in ageing and parkinson's disease. *Acta Neuropathologica*, 115(4):461–470.

Selemon, L. D., Rajkowska, G., and Goldman-Rakic, P. S. (1998). Elevated neuronal density in prefrontal area 46 in brains from schizophrenic patients: Application of a three-dimensional, stereologic counting method. *The Journal of Comparative Neurology*, 392(3):402–412.

Shen, W., Ma, L., Zhang, X., Li, X., Zhao, Y., Jing, Y., Feng, Y., Tan, X., Sun, F., and Lin, J. (2019). Three-dimensional reconstruction of picea wilsonii mast. pollen grains using automated electron microscopy. *Science China Life Sciences*, 63(2):171–179.

Slimp, J. C. and Towe, A. L. (1990). Spatial distribution of modalities and receptive fields in sensorimotor cortex of awake cats. *Experimental Neurology*, 107(1):78–96.

Swaab, D. F. and Uylings, H. B. M. (1987). Comments on review by coleman and flood 'neuron numbers and dendritic extent in normal aging and alzheimer's disease.' density measures: Parameters to avoid. *Neurobiology of Aging*, 8(6):574–576.

Swindale, N. (1990). Is the cerebral cortex modular? *Trends in Neurosciences*, 13(12):487–492.

Tang, Y., Nyengaard, J. R., Pakkenberg, B., and Gundersen, H. J. G. (1997). Age-induced white matter changes in the human brain: A stereological investigation. *Neurobiology of Aging*, 18(6):609–615.

Trojak, B., Meille, V., Jonval, L., Schuffenecker, N., Haffen, E., Schwan, R., Bonin, B., and Chauvet-Gelinier, J.-C. (2014). Interest of targeting either cortical area brodmann 9 or 46 in rTMS treatment for depression: A preliminary randomized study. *Clinical Neurophysiology*, 125(12):2384–2389.

Tsai, Y.-H. (2007). A new approach for image thresholding under uneven lighting conditions. In *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)*. IEEE.

Udawela, M., Scarr, E., Boer, S., Um, J. Y., Hannan, A. J., McOmish, C., Felder, C. C., Thomas, E. A., and Dean, B. (2017). Isoform specific differences in phospholipase c beta 1 expression in the prefrontal cortex in schizophrenia and suicide. *npj Schizophrenia*, 3(1).

Wadell, H. (1935). Volume, shape, and roundness of quartz particles. *The Journal of Geology*, 43(3):250–280.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Paper C

Layer III pyramidal cells in the prefrontal cortex reveal morphological changes in subjects with depression, schizophrenia, and suicide

Nick Y. Larsen, Ninna Vihrs, Jesper Møller, Jon Sporring, Xueke Tan, Xixia Li, Gang Ji, Grazyna Rajkowska, Fei Sun, and Jens R. Nyengaard

The paper has been submitted to
Translational Psychitry

The layout has been revised.

Abstract

Brodmann Area 46 (BA46) has long been regarded as a hotspot of disease pathology in individuals with schizophrenia (SCH) and major depressive disorder (MDD). Pyramidal neurons in layer III of the Brodmann Area 46 (BA46) project to other cortical regions and play a fundamental role in corticocortical and thalamocortical circuits.

The AutoCUTS-LM pipeline was used to study the 3D structural morphology and spatial organization of pyramidal cells. Using quantitative light microscopy, we used stereology to calculate the entire volume of layer III in BA46 and the total number and density of pyramidal cells. Volume tensors estimated by the planar rotator quantified the volume, shape, and nucleus displacement of pyramidal cells. All of these assessments were carried out in four groups of subjects: controls (C, n=10), SCH (n=10), MDD (n=8), and suicide subjects with a history of depression (SU, n=11).

SCH subjects had a significantly lower somal volume, total number, and density of pyramidal neurons when compared to C and tended to show a volume reduction in layer III of BA46. When comparing MDD participants with C, the measured parameters were inclined to follow SCH, although there was only a significant reduction in pyramidal total cell number. While no morphometric differences were observed between SU and MDD, SU had a significantly higher total number of pyramidal cells and nucleus displacement than SCH. Finally, no differences in the spatial organization of neurons were found among groups.

These results suggest that despite significant morphological alterations in layer III of BA46, which may impair prefrontal connections in people with SCH and MDD, the spatial organization of pyramidal cells remains the same across the four groups and implies no defects in neuronal migration. The increased understanding of pyramidal cell biology may provide the cellular basis for symptoms and neuroimaging observations in SCH and MDD patients.

(**The content of this paper has been excluded from this version due to potential copyright issues**)

Paper D

Should we condition on the number of points when modelling spatial point patterns?

Jesper Møller and Ninna Vihrs

The paper has been accepted for publication in
International Statistical Review

The layout has been revised.

Abstract

We discuss the practice of directly or indirectly assuming a model for the number of points when modelling spatial point patterns even though it is rarely possible to validate such a model in practice since most point pattern data consist of only one pattern. We therefore explore the possibility to condition on the number of points instead when fitting and validating spatial point process models. In a simulation study with different popular spatial point process models, we consider model validation using global envelope tests based on functional summary statistics. We find that conditioning on the number of points will for some functional summary statistics lead to more narrow envelopes and thus stronger tests and that it can also be useful for correcting for some conservativeness in the tests when testing composite hypothesis. However, for other functional summary statistics, it makes little or no difference to condition on the number of points. When estimating parameters in popular spatial point process models, we conclude that for mathematical and computational reasons it is convenient to assume a distribution for the number of points.

1 Introduction

Consider a spatial point process defined on the d-dimensional Euclidean space $\mathbb{R}^d$, and let $W \subset \mathbb{R}^d$ be a bounded region $W \subset \mathbb{R}^d$ within which realizations of the process are observed. In the literature on statistical analysis of spatial point processes, the data usually consist of a single realization $\{x_1, \ldots, x_n\} \subset W$, that is, a (finite) point pattern where both the number of points n and the locations $x_1, \ldots, x_n$ are considered to be random. For instance in the R-package `spatstat` (Baddeley et al., 2015), which is widely used for analysing spatial point patterns, there are more than fifty data examples of point patterns, but only seven of these consist of more than one point pattern. Although it is impossible from a single point pattern to validate the plausibility of any claimed model for the number of points, the practice is nonetheless to fit, validate and use spatial point process models which directly or indirectly assume a model for the number of points.

For the example of a stationary Poisson process, n is a realization of a Poisson distributed random variable, and conditioned on n, the points are realizations of n independent uniformly distributed random variables on W. Even for this simple example of a point process model, it is not possible to validate the assumed Poisson distribution for the number of points based on just one realization. A common procedure for testing whether a Poisson process model fits a given point pattern is to divide W into disjoint subsets of equal size and count the number of points falling within each subset. Conditioning on n, these observed counts constitute a realization from a multinomial model with equal probabilities, and the validity of this model

can be checked using a goodness-of-fit test, e.g. as implemented in the function quadrat.test from `spatstat`. However, even if we accept this multinomial model, to claim that n is a realization from a Poisson distribution amounts to assume independence for the counts (Moran, 1952), and indeed this assumption is hard to justify by a statistical test. Therefore, it will be impossible from a single point pattern to validate that n is a realization from a Poisson distribution.

We still believe that it is usually reasonable to think of the number of points as a single realization of a random variable. We merely point out that it would be inappropriate to make claims about the distribution of the number of points since we have no chance to validate these as illustrated for a stationary Poisson process above. If we are indeed willing to make some model assumptions and can make simulated point patterns under these assumptions, such simulations can of course be used to estimate the distribution of the number of points under the model if wanted, but why should we believe in such a distribution when we do not have the means to validate this? It may thus be more fair to accept that we do not know anything about the distribution of the number of points and therefore use a conditional approach instead. However, we have not found much places in the spatial point process literature where the role of conditioning on n is studied. Ripley (1977) provided a short discussion of conditional inference saying 'For Poisson models we can justify this conditioning by an appeal to a conditionality principle. In general all we can say is that conditional inference seems reasonable and that our revised procedures yield valid conditional tests'. Further, Ripley (1988) wrote 'Where we are interested in interactions, n may be approximately ancilliary'. He illustrated this for a Strauss process which on the unit square when ignoring boundary effects has density

$$f(\{x_1, \ldots, x_n\}) \propto \beta^n \gamma^{s(\{x_1,\ldots,x_n\})}$$

where $\beta > 0$, $0 \leq \gamma \leq 1$, $R > 0$ and $s(\{x_1, \ldots, x_n\})$ is the number of R-close pairs of points in the point pattern; we return to this process in Sections 2.3 and 4.2 and Appendix C. Ripley (1988) derived an approximation of the probability density function for the number of points:

$$p(n) \propto \frac{\beta^n}{n!} \exp\{(\gamma - 1)n(n-1)\pi R^2/2\}. \tag{1}$$

He noticed that this density depends on γ but wrote 'the dependence is quite weak in typical cases' where he referred to a plot of the cumulative distribution functions when $\beta = 50$, $R = 0.05$ and $\gamma = 0.0, 0.2, 0.4, 0.8, 1.0$. Mean values of n in this case and when $\beta = 200$ are shown in Table 1 where we see a good agreement between those means calculated from the approximation in (1) and those obtained by simulations. In our opinion, the mean values

in Table 1 depend much on γ, but we notice in Section 4.2 that maximum pseudo-likelihood estimates do not depend much on whether or not we condition on n.

	$\gamma = 0$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$	$\gamma = 1$
$\beta = 50$	37.33	39.13	41.19	43.61	46.48	50.00
	(36.91)	(38.91)	(41.24)	(43.78)	(46.56)	(49.87)
$\beta = 200$	95.00	104.11	115.92	132.12	156.45	200.00
	(88.37)	(100.72)	(114.53)	(132.35)	(157.36)	(200.32)

Table 1: Mean values for the number of points in a Strauss process on a two-dimensional unit square with $R = 0.05$ and different values of γ and β. The means are calculated from the approximate distribution in (1), and the numbers in parenthesis are the means obtained from 5000 simulations of the process

Apart from Ripley's study of the Strauss process considered above we are not aware of any thorough study of the effect of conditioning on n. Moreover, the practice is still to work with spatial point process models without conditioning on n. Therefore, the objective of this paper is to investigate the consequences of this practice and to explore the possible benefits of conditioning on n when considering various popular classes of spatial point process models. We investigate this through a comprehensive simulation study of model validation based on the widely used method of global envelopes and corresponding tests (Myllymäki et al., 2017), and by discussing the effect of conditioning on n when making frequentistic parameter estimation.

Our paper is organized such that Section 2 contains some preliminaries needed for our main contributions in Sections 3–5, where Section 5 summarises our findings. Technical details related to these sections are found in the appendices. All statistical analyses were made with `R` (R Core Team, 2019). We used the packages `spatstat` (Baddeley et al., 2015) for handling spatial point patterns, `GET` (Myllymäki and Mrkvička, 2019) to make global envelope tests and `ggplot2` (Wickham, 2016) for visualisation. Furthermore, we used our own implementations of simulation and estimation procedures when conditioning on the number of points, which can be found in the `R`-scripts in the ancillary files at `https://arxiv.org/abs/2108.10051`.

2 Preliminaries

2.1 Setting and notation

Throughout this paper we use the following point process setting and notation.

For a subset $x \subset \mathbb{R}^d$, let $n(x)$ denote its cardinality (setting $n(x) = \infty$ if x is not a finite set) and $x_B := x \cap B$ its restriction to any set $B \subseteq \mathbb{R}^d$. Let

Ω denote the set of all locally finite subsets $x \subset \mathbb{R}^d$, that is, $n(x_B) < \infty$ whenever $B \subset \mathbb{R}^d$ is bounded. A simple locally finite point process on $\mathbb{R}^d$ is a random variable taking values in Ω. Here and elsewhere, we omit measure theoretical details and refer instead to Møller and Waagepetersen (2004) and the references therein. The point process can also be specified in terms of the counting measure $N(B) := n(X_B)$ for bounded sets $B \subseteq \mathbb{R}^d$ (more precisely, B should also be a Borel set, but as mentioned we omit such details).

We assume that X is a stationary simple locally finite point process on $\mathbb{R}^d$ with intensity $\rho \in (0, \infty)$. This means that $X \in \Omega$, the distribution of X is invariant under translations in $\mathbb{R}^d$ and $\mathrm{E}N(B) = \rho|B|$ where $|B|$ denotes the d-dimensional volume (Lebesgue measure) of $B \subseteq \mathbb{R}^d$. Stationarity is a common assumption for spatial point processes, and it allows us to deal with frequently used functional summary statistics, see Section 2.2. We also assume that a single realization $x = \{x_1, \ldots, x_n\}$ of X_W has been observed where the observation window $W \subset \mathbb{R}^d$ is compact and $|W| > 0$.

Let $u, v \in \mathbb{R}^d$, $r \geq 0$ and $B \subset \mathbb{R}^d$. Then $\mathbb{1}(\cdot)$ is the indicator function; $\|u - v\|$ is the usual distance between u and v; $b(u, r)$ is the closed d-dimensional ball with centre u and radius r; $\mathrm{dist}(B, u) := \inf\{r > 0 \mid b(u, r) \cap B \neq \emptyset\}$ is the distance from u to B; $B_{\ominus r} := \{u \in \mathbb{R}^d \mid b(u, r) \subset B\}$ is B eroded by a ball of radius r; $B_{\oplus} := \cup_{u \in B} b(u, r)$ is B dilated by a ball of radius r; and $(B_{\ominus r})_{\oplus r} \subseteq B$ is the opening of B by a ball of radius r. Finally, we use the convention $0/0 := 0$.

2.2 Functional summary statistics

Functional summary statistics $\hat{K}, \hat{F}, \hat{G}$ and $\hat{J}$ which are non-parametric (empirical) estimates of the theoretical functions below are widely used for exploratory purposes, model fitting and model checking, see Baddeley et al. (2015) and the references therein.

For every $r > 0$ and an arbitrary point $u \in \mathbb{R}^d$, Ripley's K-function is defined by

$$\rho K(r) := \mathrm{E}(N(b(u, r) \setminus \{u\}) \mid u \in X), \tag{2}$$

the empty space function F (or spherical contact function) and the nearest-neighbour function G are

$$F(r) := \mathrm{P}(\mathrm{dist}(X, u) \leq r), \qquad G(r) := \mathrm{P}(\mathrm{dist}(X \setminus \{u\}, u) \leq r \mid u \in X) \tag{3}$$

and for $F(r) < 1$ the J-function is

$$J(r) := (1 - G(r))/(1 - F(r)). \tag{4}$$

These definitions do not depend on the choice of u since X is stationary, and as indicated by the notation we have conditioned on $u \in X$ in the definitions

of K and G, meaning that $X \setminus \{u\}$ then follows the reduced Palm distribution of X at u.

When estimating the K, F, G and J-functions by non-parametric methods, different types of edge correction methods have been suggested in order to adjust for the fact that $n(x_{W \cap b(u,r)})$ tends to be smaller for points $u \in W$ which are close to the boundary of W compared to points $u \in W$ which are far from the boundary of W. Baddeley et al. (2015) noticed that the choice of edge correction method is usually not very important. We choose to use a particular border (or minus-sampling) correction method which is available in `spatstat` and refer to Baddeley et al. (2015) for the concrete estimators.

When conditioning on n, we are not aware of how to modify the definitions of K, F, G and J. If we just condition in (2)–(4), the expressions will depend on u because stationarity no longer holds. Still, when only a single point pattern is observed and hence it is impossible to validate any claimed model of $N(W)$, it seems appropriate to condition on n when calculating global envelopes and tests as in Section 3.

2.3 Models

For the simulation studies in Sections 3 and 4.2, we consider four concrete examples of point process models on $\mathbb{R}^2$ as specified in M1–M4 below. Appendix B provides further details including how to make simulations both with and without conditioning on the number of points.

M1: X is a stationary Poisson process. This is the model of no spatial interaction or complete spatial randomness. When making simulations in Section 3, we let the intensity be $\rho = 100$.

M2: X is a stationary log-Gaussian Cox process. Then, X is driven by a stochastic intensity $Z = \exp(Y)$ where Y is a stationary Gaussian random field on $\mathbb{R}^2$, meaning that X conditioned on Y is a Poisson process with intensity function Z (Møller et al., 1998). We use an exponential covariance function $c(u, v) := \sigma^2 \exp(-\|u - v\|/\delta)$ for $u, v \in \mathbb{R}^2$ where $\sigma^2 > 0$ is a variance parameter and $\delta > 0$ is a scale parameter, and so the mean of Y is $\mu = \log(\rho) - \sigma^2/2$. Because of the positive correlation in Y, realizations of X exhibit clustered behaviour. When making simulations in Section 3, we let $\rho = 100$, $\sigma^2 = 1$ and $\delta = 0.1$.

M3: X is a stationary Strauss process. This process has parameters $\beta > 0$, $0 \leq \gamma \leq 1$ and $R > 0$. It is defined by a so-called local specification: Let $B \subset \mathbb{R}^2$ be a bounded set of positive area and define the R-close neighbourhood to B by $\partial B := B_{\oplus R} \setminus B$. Then, for every such B, X_B and $X_{\mathbb{R}^2 \setminus B_{\oplus R}}$ are conditionally independent given $X_{\partial B}$. Furthermore, for every finite $x_{\partial B} \subseteq \partial B$, X_B conditioned on $X_{\partial B} = x_{\partial B}$ has a density with respect to a Poisson process

of intensity 1 and restricted to B. This conditional density is

$$f_B(x_B \mid x_{\partial B}) \propto \beta^{n(x_B)} \gamma^{s(x)} \tag{5}$$

for finite $x_B \subseteq B$, $x = x_B \cup x_{\partial B}$ and $s(x) = \sum_{i<j} \mathbb{1}(\|x_i - x_j\| \leq R)$ if $x = \{x_1, \ldots, x_n\}$. The normalizing constant which is omitted in (5) depends on (β, γ, R) and $x_{\partial B}$, and it is intractable unless $\gamma = 1$. When $\gamma = 1$, X is just a stationary Poisson process with intensity β. As γ decreases, X becomes more and more inhibitory, and it is a Gibbs hard core model if $\gamma = 0$. When making simulations in Section 3, we let $\beta = 200$, $\gamma = 0.3$ and $R = 0.05$. Then, the intensity is approximately 100.

M4: X is a stationary determinantal point process. In brief, determinantal point processes (Macchi, 1975, Lavancier et al., 2015) are specified by a function $C : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{C}$ called the kernel, and they are repulsive at all scales, cf. Appendix B.4. We use a Gaussian kernel $C(u, v) = \rho \exp(-\|(u - v)/\kappa\|^2)$ where $\kappa > 0$ is a scale parameter and $\rho > 0$ is the intensity. It should be satisfied that $\kappa \leq 1/\sqrt{\rho\pi}$, and this upper limit corresponds to the most repulsive case when ρ is fixed. When making simulations in Section 3, we let $\rho = 100$ and $\kappa = 0.03 \leq 1/\sqrt{100\pi} \approx 0.056$.

3 Global envelopes and the effect of conditioning

In this section, we investigate the effect of conditioning on the number of points when using global envelopes for model validation. Section 3.1 first describes the set-up of the simulation study, and Section 3.2 describes and interprets the results.

3.1 Set-up

We investigate the effect of conditioning on the number of points when considering global envelopes for functional summary statistics and corresponding tests based on the extreme rank length as described in Myllymäki et al. (2017), Mrkvička et al. (2020) and Myllymäki and Mrkvička (2019). Briefly, a $(100 - \alpha)\%$ global envelope consists of a lower and an upper curve defining a region such that the observed functional summary statistic for data falls completely between these bounding curves if and only if the global envelope test cannot be rejected at level $\alpha\%$. There of course exist other tests which can be used in connection with spatial point processes, see e.g. Baddeley et al. (2015, chapter 10); however, since the use of a global envelope and its corresponding test statistic is by far the most popular method for performing model validation of a fitted spatial point process model, we restrict attention to this test procedure. There are ways to make a combined global envelope test based on several functional summary statistics, but for our purpose we

prefer to investigate the effect of conditioning on the number of points for each functional summary statistic.

We made the simulation study as follows. Under each of the four models M1–M4 we simulated 1000 independent point patterns within a two-dimensional unit square (the observation window W). For each of these point patterns, we fitted the parameters of the models as described in the last paragraph of this section. Under each fitted model and each true model, we made further 2500 simulations with and 2500 simulations without conditioning on the number of points. From each of these four cases, or three in the case of the Poisson process since the fitted and true model is the same when conditioning on the number of points, we used the 2500 simulations to calculate 95% global envelopes based on each of the functional summary statistics $\hat{F}, \hat{G}, \hat{J}$ and $\hat{K}$. Some further technical and practical details related to the set-up of the simulation study are deferred to Appendix A.

Clearly more narrow envelopes are preferable when comparing envelopes for the same type of functional summary statistic. For simplicity, in order to spot a general tendency in the width of envelopes, we considered for each envelope a numerical approximation of its area $\int_0^R (c_u(r) - c_l(r))\,\mathrm{d}r$ where c_u and c_l are the upper and lower curves of the envelope, respectively, and R is the highest r-value for which the considered functional summary statistic was estimated.

3.2 Results

Figure 1 shows boxplots of the approximated area of the envelopes. We see that it generally makes little difference in the area of the envelopes whether the parameters are fitted from data or not, except for $\hat{K}$ where there is less variation in the width of the envelopes when using the true parameters especially in the unconditional case. We also see that for $\hat{K}$ and $\hat{J}$ it makes no real difference in the areas of the envelopes to condition on the number of points either, but for $\hat{G}$ and especially $\hat{F}$ the envelopes are in general more narrow in the conditional case.

Figure 2 shows quantile-quantile plots comparing the distributions of the p-values of the global envelope tests for each fitted model to a uniform distribution on $[0, 1]$. We see that some of the tests are too conservative, which may be because the null hypothesis is composite except in the conditional case of the Poisson process. Using $\hat{F}$ overall gives very conservative tests in the unconditional case, especially for the Poisson, Cox and determinantal point process, and this behaviour is corrected very well by conditioning on the number of points. For $\hat{G}$, it makes little difference to condition on the number of points except in the case of the Cox process where it corrects the conservativeness in the unconditional case very well. For both $\hat{J}$ and $\hat{K}$, it makes little to no difference whether we condition on the number of points

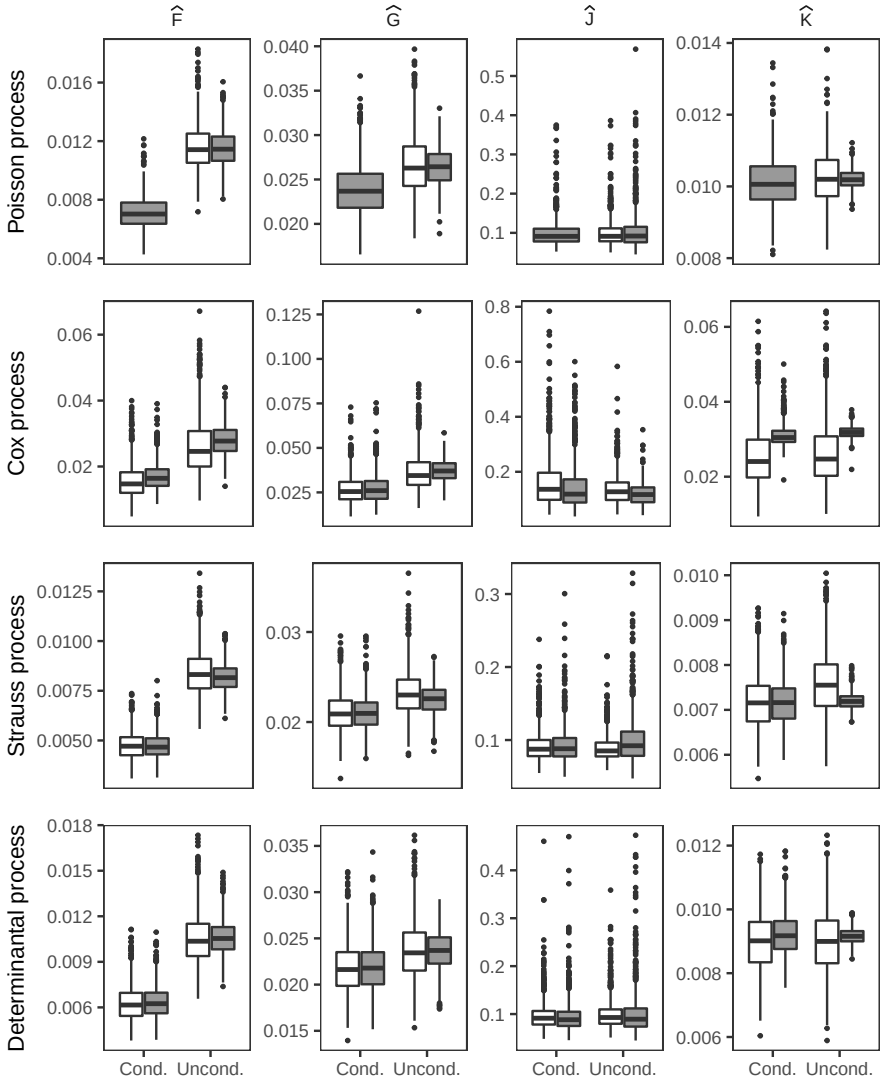

Fig. 1: Boxplots for the area between 95% global envelope curves for the functional summary statistic stated at the top of each column. The results were obtained from the simulation study described in the text. The grey level indicates whether the true parameters (dark grey) or fitted parameters (white) were used in the simulations. Whether simulations were made conditional on the number of points is stated at the bottom of each column. The type of model under consideration is stated to the left of each row.

or not, and in all cases the distributions of the p-values are in good agreement with the uniform distribution even though we also see some slight conservativeness in some tests. Since $\hat{K}$ was used to fit the parameters of the Cox and determinantal point processes, we usually do not want to use $\hat{K}$ for model validation as well.

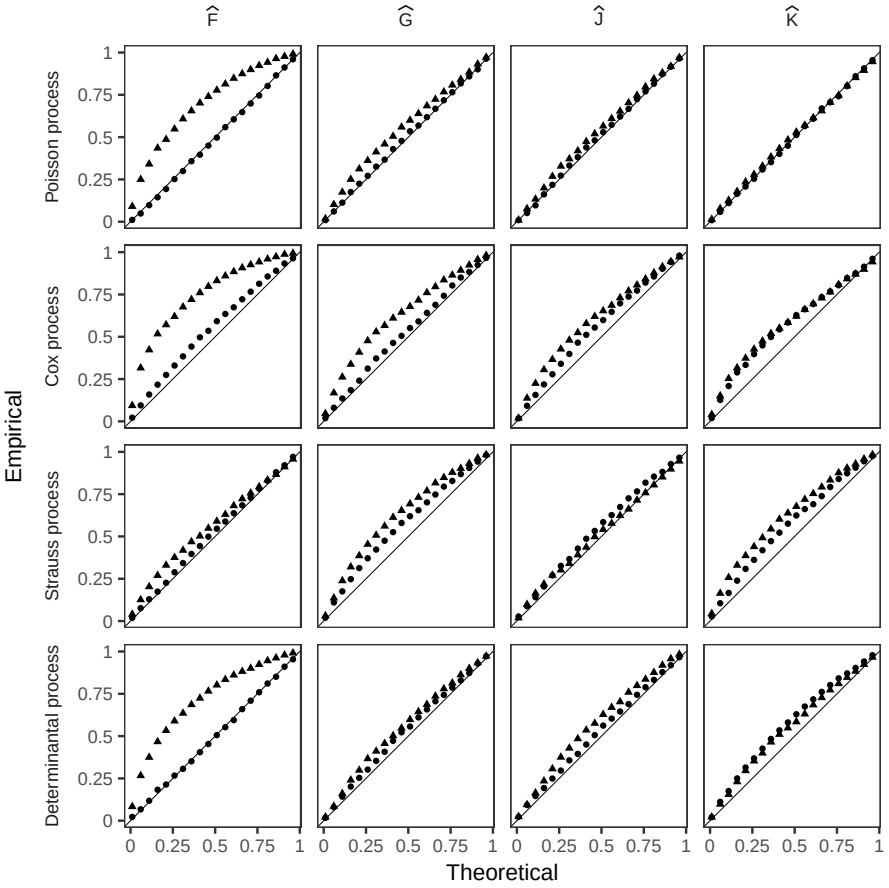

Fig. 2: Quantile-quantile plots comparing the empirical distributions of the p-values of global envelope tests to a uniform distribution on $[0, 1]$. The considered p-values are the results of the simulation study described in the text. The type of point indicates whether the global envelope tests were based on conditional (dots) or unconditional (triangles) simulations under the fitted model. The functional summary statistic used is stated at the top of each column, the type of model is stated to the left of each row and the straight line is the identity line.

Moreover, to see the effect of increasing the intensity, we made a simula-

tion study for the stationary Poisson process with $\rho = 200$. Our conclusions remained the same and the only real difference was that all envelopes were in general more narrow, which was to be expected since the summary statistics varies less when we observe more points. However, the simulation studies for the remaining three models are time consuming, and it will be even worse if we increase the intensity, but we do not believe the conclusions will be much different as we have indeed established for the example of a Poisson process.

4 Conditional estimation

In this section, we investigate the possibility to estimate parameters in spatial point process models conditional on the number of points. We consider Cox, Gibbs, and determinantal point processes and in each case discuss whether conditional estimation offers any advantages over the unconditional approach. There is of course no reason to consider conditional estimation in the case of a stationary Poisson process since the conditional case is a binomial point process with no unknown parameters.

4.1 Cox processes

Parametric models for a stationary Cox process X driven by a random field Z on $\mathbb{R}^d$ are usually of the form $Z(u) = \rho R(u)$ where $R = \{R(u)\}_{u\in\mathbb{R}^d}$ is a non-negative unit-mean stochastic process following a parametric model with a parameter ψ so that (ρ, ψ) has range $(0, \infty) \times \Psi$ for some set $\Psi \subseteq \mathbb{R}^p$. The process X_W then has a density

$$f(x) = æ^n \mathrm{E}\left[R(x_1)\dots R(x_n)\exp\left\{|W| - \rho\int_W R(u)\,\mathrm{d}u\right\}\right] \tag{6}$$

with respect to a Poisson process of intensity 1 and restricted to W. In general, this density is not expressible on closed form except for rather trivial cases, and so likelihood based inference is difficult although a missing-data Markov chain Monte Carlo approach can be used for approximate maximum likelihood estimation, see Møller and Waagepetersen (2004). Since second-order moments of the counts $N(B)$ are often expressible on closed form, moment-based and composite likelihood estimation procedures are usually preferred, see Møller and Waagepetersen (2017) and the references therein.

When conditioning on $N(W) = n$, any ordering $(X_1, \dots, X_n)$ of the n points in X_W has a density with respect to Lebesgue measure on W^n which is proportional to the right hand side in (6). Hence, it is also not expressible on closed form and depends on both ρ and ψ. Furthermore, moment-based estimation is no longer possible. Consequently, we do not see any advantage in conditioning on the number of points.

4.2 Gibbs processes

Suppose that X is a stationary Gibbs point process with known interaction radius $R > 0$ (details for this general setting are provided in Appendix B.3. Further, suppose that $X_W = x$ has been observed, and let $x_{W_{\ominus R}} = \{x_1, \dots, x_m\}$. We define the R-close neighbourhood to $W_{\ominus R}$ by $\partial W_{\ominus R} := (W_{\ominus R})_{\oplus R} \setminus W_{\ominus R}$ and base inference on the conditional distribution of $X_{W_{\ominus R}}$ given $X_{\partial W_{\ominus R}} = x_{\partial W_{\ominus R}}$. Thereby, we account for edge effects due to the unobserved points in $X_{\mathbb{R}^d \setminus W}$ because $X_{W_{\ominus R}}$ and $X_{\mathbb{R}^d \setminus W}$ are independent conditioned on $X_{\partial W_{\ominus R}} = x_{\partial W_{\ominus R}}$. Below, we discuss reasons and consequences of further conditioning on $N(W_{\ominus R}) = m$.

For specificity and simplicity, let us think of X as the stationary Strauss process given in Section 2.3. Then the likelihood function is of exponential family form with canonical parameter $(\log \beta, \log \gamma)$ and minimal sufficient statistic $(n(x_{W_{\ominus R}}), s(x_{W_{\ominus R}} \cup x_{\partial W_{\ominus R}}))$. However, estimation of (β, γ) is complicated by the fact that the normalizing constant is not expressible on closed form for $\gamma < 1$. Often, the interaction parameter γ is of main interest; if we also condition on $N(W_{\ominus R}) = m$, we obtain a likelihood function which only depends on γ. The normalizing constant of that likelihood function is also not expressible on closed form for $\gamma < 1$, but it is simpler to approximate, cf. Appendix C. In particular, as noticed at the end of Appendix C, simulations and computations will be faster when conditioning on $N(W_{\ominus R}) = m$. However, it is still slower and more difficult than using maximum pseudo-likelihood estimation, which will be described next.

Maximum pseudo-likelihood estimation is known to be a fast and often reliable alternative to maximum likelihood estimation, see e.g. Jensen and Møller (1991), Baddeley et al. (2014), and the references therein. The definition of the pseudo-likelihood function depends on the context and is specified in Appendix D both with and without conditioning on $N(W_{\ominus R}) = m$ and in a way which accounts for edge effects. In the conditional case, we consider Besag's original pseudo-likelihood function (Besag, 1975); in the unconditional case, we consider Besag's extension to spatial point processes (Besag, 1977, Jensen and Møller, 1991). The computational advantage of using the pseudo-likelihood functions is that they do not depend on the intractable normalizing constant which appears in maximum likelihood estimation.

We wanted to investigate whether it makes a real difference in the maximum pseudo-likelihood estimate of γ to condition on the number of points. In order to do this, we considered 1000 simulations on the unit square of a stationary Strauss process when $\beta = 200$, $R = 0.05$, and γ was sampled uniformly in the interval $[0.01, 1]$. For each simulation, we calculated the maximum pseudo-likelihood estimate of (β, γ) with the function `exactMPLEstrauss` in `spatstat`, which computes the maximum pseudo-likelihood estimate to a high accuracy. We also found the maximum pseudo-likelihood

estimate in the conditional case by implementing a function where we used the same numerical methods for optimization and evaluation of integrals as in `exactMPLEstrauss`. In both cases, we let $R = 0.05$ be given. The average absolute difference between the estimates of γ obtained with the two methods was 0.005, and the largest absolute difference was 0.01. So, it makes very little difference to condition on the number of points. Furthermore, the computations for the pseudo-likelihood function in the conditional case may be more cumbersome since more integrals have to be evaluated, cf. Appendix D. Therefore, there is no apparent reason to use the more complicated pseudo likelihood method of the conditional case.

4.3 Determinantal point processes

Parametric models for stationary determinantal point processes with intensity $\rho > 0$ are specified by a parametric class of kernel functions which are usually of the form $C_\theta(u, v) = \rho R_\psi(u - v)$ where $\theta = (\rho, \psi)$ and $(u, v) \mapsto R_\psi(u - v)$ is a (complex) correlation function. Under weak assumptions, e.g. that C_θ is a continuous complex covariance function, the kernel restricted to $W \times W$ has a spectral representation

$$C_\theta(u, v) = \sum_{i=1}^{\infty} \lambda_i \phi_i(u) \overline{\phi_i(v)}, \qquad u, v \in W,$$

where $\{\phi_i\}_{i=1,2,\ldots}$ is an orthonormal basis for the L^2-space of square-integrable complex functions on W and $\lambda_1, \lambda_2, \ldots$ are corresponding eigenvalues. We have $\lambda_i = \rho \lambda_i'$ where (ϕ_i, λ_i') depends only on ψ and the condition $0 \leq \lambda_i \leq 1$ is needed to ensure existence of the process. Therefore, the parameter space Ψ_ρ of ψ will depend on the value of ρ; briefly speaking, there is a trade-off between intensity and repulsion, and the set Ψ_ρ decreases as ρ increases, cf. Lavancier et al. (2015). For parameter estimation based on maximum likelihood and moment-based methods, see Lavancier et al. (2015).

Conditioned on $N(W) = n$, any ordering $(X_1, \ldots, X_n)$ of the n points in X_W has probability density function

$$\frac{\sum_{i_1 < \cdots < i_n} \left(\prod_{j=1}^n \lambda_{i_j} \right) \left(\prod_{j \notin \{i_1, \ldots, i_n\}} (1 - \lambda_j) \right) \frac{1}{n!} \det \left\{ \sum_{k=1}^n \phi_{i_k}(x_i) \overline{\phi_{i_k}(x_j)} \right\}_{i,j=1,\ldots,n}}{\sum_{i_1 < \cdots < i_n} \left(\prod_{j=1}^n \lambda_{i_j} \right) \left(\prod_{j \notin \{i_1, \ldots, i_n\}} (1 - \lambda_j) \right)}$$

for $(x_1, \ldots, x_n) \in W^n$. For parametric models as considered above, it follows that the conditional distribution of X_W given $N(W) = n$ depends on both ρ and ψ in a complicated way; in fact it is more complicated than the likelihood in the unconditional case, cf. Lavancier et al. (2015). Therefore, we do not see any advantage in conditioning on $N(W) = n$ when making parameter estimation.

5 Concluding remarks

It is worth repeating that any claimed model for the number of points cannot be justified based on just one realization. However, in case of Cox, Gibbs and determinantal point processes, it is convenient for computational reasons to assume a distribution for the number of points when estimating parameters.

Regarding global envelope tests based on $\hat{K}$ or $\hat{J}$, it made little or no difference in our simulation study whether we condition on the number of points. However, when we instead used $\hat{F}$ or $\hat{G}$, conditioning on the number of points gave more narrow envelopes and hence stronger tests, and it corrected for conservativeness in the tests.

Global envelopes and tests are usually calculated from simulations of a single point process model. For a composite null hypothesis, it is possible to make an adjusted global envelope test but at the expense of many simulations(Myllymäki and Mrkvička, 2019). Conditioning on the number of points may offer an alternative which requires fewer simulations, but whether this will be faster in practice depends on the actual speed of the simulation procedures. We leave this for future research.

Regarding conditional estimation, we concluded that it is impractical and offers no clear advantage to consider conditional estimation for Cox and determinantal point process models. For Gibbs point process models, it simplifies maximum likelihood estimation but complicates the more commonly used and faster method of maximum pseudo-likelihood estimation where there is also little difference in the estimates achieved with and without conditioning. We therefore overall have found no apparent reason to use conditional estimation.

Acknowledgements

The research of the authors was supported by The Danish Council for Independent Research | Natural Sciences, grant DFF – 7014-00074 'Statistics for point processes in space and beyond'.

A Details regarding the simulation study in Section 3

The following gives some practical and technical details regarding the simulation study described in Section 3.1.

When calculating functional summary statistics, we always used the default range of r-values in `spatstat`. For details on simulation procedures, especially conditional simulation, see Appendix B. When fitting parameters,

we always did it without conditioning on the number of points since we argue in Section 4 that there is no apparent reasons to use conditional estimates. We used the natural estimate $n(x)$ of the intensity in the case of the Poisson process, used the method of minimum contrast estimation based on Ripley's K-function (see Baddeley et al. (2015)) in the cases of the log-Gaussian Cox process and the Gaussian determinantal point process and used the method of profile maximum pseudo likelihood in the case of the Strauss process (where we considered 41 equally spaced values of R in the interval $(0.03, 0.07)$). When fitting the parameters of the log-Gaussian Cox process, we found that sometimes the scale parameter δ was seriously overestimated, which caused the conditional simulation procedure to be extremely slow. Therefore, we decided only to use realizations of the log-Gaussian Cox process M2 where the fitted scale parameter was below 0.3, which left 953 realizations in the simulation study. When fitting parameters in the case of the Gaussian determinantal point process, we found that κ was seriously underestimated for a few realizations, which either slowed down the simulation procedures considerably or caused them to fail. We therefore excluded realizations with a fitted value of κ less than 0.001 after which 996 realizations remained.

B Stationary point process models and conditional simulation

In the following, we describe some popular classes of spatial point process models: Poisson, Cox, Gibbs and determinantal point processes. It is well known how to simulate such point processes within the compact observation window W without conditioning on the number of points; for Poisson, Cox and Gibbs point processes, see Møller and Waagepetersen (2004) and the references therein, and for determinantal point processes, see Lavancier et al. (2015). It is also well-known how to simulate Poisson processes when we condition on the number of points falling in W. For finite Gibbs point processes defined on W (and thus not the infinite stationary Gibbs processes which we consider), Ripley (1979) described how to make simulations conditioned on the number of points but without accounting for edge effects. Below, we suggest methods for simulation of Cox, infinite Gibbs and determinantal point processes conditioned on $N(W)$ and where we account for edge effects when needed (the case of infinite Gibbs point processes).

B.1 Poisson processes

Suppose that $\rho : \mathbb{R}^d \to [0, \infty)$ satisfies that $\nu(B) := \int_B \rho(u)\,\mathrm{d}u < \infty$ for bounded (Borel) sets $B \subset \mathbb{R}^d$. A point process X is a Poisson process with

intensity ρ if for any $B \subset \mathbb{R}^d$ with $0 < \nu(B) < \infty$, $N(B)$ follows a Poisson distribution with mean $\nu(B)$, and for any $n \in \mathbb{N}$, conditioned on $N(B) = n$, the n points in X_B are independent and identically distributed with a density proportional to ρ. If ρ is constant, X is a stationary Poisson process. It is usually easy to simulate a Poisson process on W conditioned on n, in particular in the stationary case where the n points are just independent and uniformly distributed on W.

B.2 Cox processes

Let $Z = \{Z(u)\}_{u \in \mathbb{R}^d}$ be a non-negative stochastic process such that (almost surely) for every bounded set $B \subset \mathbb{R}^d$, $\int_B Z(u)\,\mathrm{d}u$ exists and is finite. Assuming X conditioned on Z is a Poisson process with intensity function Z, we call X a Cox process driven by Z. When Z is stationary with finite mean, X is stationary with intensity $\rho = \mathrm{E}Z(u)$ for any $u \in \mathbb{R}^d$.

For simulating X_W conditioned on $N(W) = n$, we can use the following simple acceptance-rejection sampling procedure. Defining $Z_W \coloneqq \{Z(u)\}_{u \in \mathbb{W}}$, repeat simulating a realization $Z_W = z$ and an independent uniform variable U on $[0, 1]$ until

$$U \leq \frac{1}{n!} \left(\int_W z(u)\,\mathrm{d}u \right)^n \exp\left(- \int_W z(u)\,\mathrm{d}u \right).$$

Then, simulate n points from W independently with a density proportional to z. For a log-Gaussian Cox process, $Z = \exp(Y)$ with Y a Gaussian random field defined on $\mathbb{R}^d$. In this case, the conditional simulation procedure can be refined by considering a subdivision of W as described in Møller et al. (1998).

B.3 Gibbs point processes

The definition of a stationary Gibbs point process is rather technical. In the following definition using a local specification, we omit for simplicity not only measure theoretical details but also technical conditions ensuring existence of the process; for such details, we refer instead to Georgii (1988) or the review in Møller and Waagepetersen (2004).

We need the following notation. Let R be a given non-negative number and $B \subset \mathbb{R}^d$ a bounded set. As a reference measure we consider the stationary Poisson process on $\mathbb{R}^d$ with intensity 1 and denote its distribution μ. The restriction of μ to B is denoted μ_B. The R-close neighbourhood to B is defined by $\partial B \coloneqq B_{\oplus R} \setminus B$. Let $\Omega_{\text{fin}} \subset \Omega$ be the set of all finite subsets of $\mathbb{R}^d$. Consider functions $s : \Omega_{\text{fin}} \mapsto \mathbb{R}^p$ and $a : \Omega_{\text{fin}} \mapsto [0, \infty)$ satisfying the following conditions: a is hereditary, that is, $a(x) > 0$ implies $a(y) > 0$ for $y \subset x$; a and s are invariant under translations in $\mathbb{R}^d$ (this will be needed to

ensure that X is stationary); and for every $x \in \Omega_{\text{fin}}$ and every $u \in \mathbb{R}^d \setminus x$, setting

$$a(x,u) \coloneqq a(x \cup \{u\})/a(x), \qquad s(x,u) \coloneqq s(x \cup \{u\}) - s(x),$$

then

$$a(x,u) = a(x \cap b(u,R), u), \qquad s(x,u) = s(x \cap b(u,R), u). \tag{7}$$

Finally, we consider a parameter $\theta = (\beta, \psi)$ with $\beta > 0$ and $\psi \in \Psi$ so that $\Psi \subseteq \mathbb{R}^p$ and $\Theta \coloneqq (0,\infty) \times \Psi$ is the parameter space.

Now, we consider a local specification which is given by the following assumptions (i)–(ii). For every bounded set $B \subset \mathbb{R}^d$ and every $\theta \in \Theta$, we have: (i) X_B and $X_{\mathbb{R}^d \setminus B_{\oplus R}}$ are conditionally independent given $X_{\partial B}$; (ii) for ($\mu_{\partial B}$ almost) every finite $x_{\partial B} \subseteq \partial B$, X_B conditioned on $X_{\partial B} = x_{\partial B}$ has a density with respect to μ_B given by

$$f_{B,\theta}(x_B \mid x_{\partial B}) = a(x)\beta^{n(x_B)} \exp(\psi^T s(x))/c_{B,\theta}(x_{\partial B}) \qquad \text{for finite } x_B \subseteq B, \tag{8}$$

where $x = x_B \cup x_{\partial B}$, $s(x)$ and ψ are viewed as column vectors, ψ^T is the transpose of ψ and $c_{B,\theta}(x_{\partial B})$ is a normalizing constant. Then, we call X an infinite Gibbs (or Markov) point process with parameter θ and interaction radius R.

Above, we have imposed more structure than usual when defining a Gibbs point process: the assumption in (7) is a local Markov property, which is in line with the spatial Markov properties specified in (i), and in (8), we have imposed an exponential family structure. In applications, we typically interpret ψ as an interaction parameter, and β usually controls the intensity, which also depends on ψ. For example, a stationary Strauss process has $a = 1$, $\Psi = (-\infty, 0]$ and

$$s(x) = s_R(x) \coloneqq \sum_{i<j} \mathbb{1}(\|x_i - x_j\| \le R) \qquad \text{for } x = \{x_1, \dots, x_n\} \in \Omega_{\text{fin}}.$$

For conditional simulation of X_W given $N(W) = n$, we consider an extended compact region $W_{\text{ext}} \supseteq W_{\oplus R}$ in order to account for edge effects. Then, we let Y denote a process on W_{ext} with a density with respect to $\mu_{W_{\text{ext}}}$ given by

$$f_\theta(x) = a(x)\beta^{n(x)} \exp(\psi^T s(x))/c_\theta, \quad x \subset W_{\text{ext}},$$

assuming the normalising constant c_θ is finite. Simulations of Y_W will then be approximate simulations of X_W if W_{ext} is large enough. Let $(Y_1, \dots, Y_n)$ be an arbitrary ordering of the n points in Y_W, and define $A \coloneqq W_{\text{ext}} \setminus W$. The conditional density of $(Y_1, \dots, Y_n)$ given both $Y_A = y_A$ and $n(Y_W) = n$ is

$$f_\psi(y_1, \dots, y_n \mid y_{\partial W}, n) \propto a(\{y_1, \dots, y_n\} \cup y_{\partial W}) \exp(\psi^T s(\{y_1, \dots, y_n\} \cup y_{\partial W})) \tag{9}$$

with respect to Lebesgue measure on W^n, and it only depends on y_A through $y_{\partial W}$ and on θ through ψ. Moreover, the conditional density of Y_A given both $(Y_1, \ldots, Y_n) = (y_1, \ldots, y_n)$ and $n(Y_W) = n$ depends only on $(y_1, \ldots, y_n)$ through those $y_i \in \partial A$. It is

$$\begin{aligned} &f_\theta(y_A \mid \{y_i : y_i \in \partial A\}, n) \propto \\ &\quad a(y_A \cup \{y_i : y_i \in \partial A\})\beta^{n(y_A)} \exp(\psi^T s(y_A \cup \{y_i : y_i \in \partial A\})) \end{aligned} \tag{10}$$

with respect to μ_A.

We use a Gibbs within Metropolis-Hastings algorithm where we alternate between simulating from (9) and (10). In case of (9), a single point updating Metropolis algorithm is used (specifically, Algorithm 7.2 in Møller and Waagepetersen (2004) where a proposal consists in replacing a uniformly selected existing point y_i with another point y_i' which is uniformly distributed on W). In case of (10), we use the birth-death Metropolis-Hastings algorithm in Geyer and Møller (1994) (Algorithm 7.4 in Møller and Waagepetersen (2004)).

In practice, it is necessary to choose an appropriate burnin when using Markov chains for simulation. For our simulations, based on various trace plots, we chose to use a burnin of 4000 and 1000 iterations for unconditional and conditional simulation of Strauss processes, respectively. It is also necessary to decide what W_{ext} should be. We used the default in the function `rmh` from `spatstat`, which was also the function we used to make unconditional simulations.

B.4 Determinantal point processes

We say that X is a determinantal point process with kernel $C : \mathbb{R}^d \times \mathbb{R}^d \mapsto C$ if for $n = 1, 2, \ldots$ and any pairwise disjoint bounded (Borel) sets $A_1, \ldots, A_n \subset \mathbb{R}^d$,

$$\mathrm{E}[N(A_1) \cdots N(A_n)] = \int_{A_1} \cdots \int_{A_n} \det\{C(u_i, u_j)\}_{i,j=1,\ldots,n} \, \mathrm{d}u_1 \cdots \mathrm{d}u_n \tag{11}$$

where $\det\{C(u_i, u_j)\}_{i,j=1,\ldots,n}$ is the determinant of the $n \times n$ matrix where entry (i, j) is $C(u_i, u_j)$. For $1 < k < n$ we have

$$\mathrm{cov}(N(A_1) \cdots N(A_k), N(A_{k+1}) \cdots N(A_n)) \leq 0,$$

and the process is therefore said to be repulsive at all scales. It is often assumed that C is a continuous complex covariance function so that $C(u, v) = C_0(u - v)$ is translation invariant; we also make that assumption, which implies that X is stationary. For details on the many nice properties of determinantal point processes including those given below, see Lavancier et al. (2015) and the references therein.

The kernel restricted to $W \times W$ has a spectral representation

$$C(u,v) = \sum_{i=1}^{\infty} \lambda_i \phi_i(u)\overline{\phi_i(v)}, \qquad u,v \in W \tag{12}$$

where $\{\phi_i\}_{i=1,2,\ldots}$ is an orthonormal basis for the L^2-space of square-integrable complex functions on W and each λ_i is an eigenvalue. Existence of the determinantal point process restricted to W is equivalent to that each $\lambda_i \in [0,1]$ and $\sum_{i=1}^{\infty} \lambda_i < \infty$. Now, let $B_1, B_2, \ldots$ be independent Bernoulli variables with parameters $\lambda_1, \lambda_2, \ldots$. Then, X_W is distributed as a determinantal point process on W with kernel

$$K(u,v) \coloneqq \sum_{i=1}^{\infty} B_i \phi_i(u)\overline{\phi_i(v)}, \qquad u,v \in W,$$

and $N(W)$ is distributed as $\sum_{i=1}^{\infty} B_i$, which is finite with probability 1. It can be shown that X_W conditional on a realization $b_1, b_2, \ldots$ of $B_1, B_2, \ldots$ has $n = \sum_{i=1}^{\infty} b_i$ points, and any ordering $(X_1, \ldots, X_n)$ of these points has probability density function

$$p_{i_1,\ldots,i_n}(x_1,\ldots,x_n) \coloneqq \frac{1}{n!} \det\left\{ \sum_{k=1}^{n} \phi_{i_k}(x_i)\overline{\phi_{i_k}(x_j)} \right\}_{i,j=1,\ldots,n}$$

on W^n where $i_1, \ldots, i_n$ are the indices for which the Bernoulli variables are 1. It is well-known how to make simulations from this distribution.

Apart from special cases, only the existence of the spectral representation but not its exact form is known in practice. Lavancier et al. (2015) provided efficient ways of obtaining a useful approximation.

For conditional simulation of X_W given $N(W) = n$, we first simulate $B_1, B_2, \ldots$ conditional on $N(W) = n$ and hereafter simulate X_W conditional on $B_1, B_2, \ldots$, which then does not depend on $N(W)$. Since it is well known how to simulate from the density $p_{i_1,\ldots,i_n}$, it suffices to discuss how to simulate $B_1, B_2, \ldots$ conditioned on $N(W) = n$. Define $I_0 \coloneqq 0$ and $I_k \coloneqq \inf\{j > I_{k-1} \mid B_j = 1\}$ $(k = 1, 2, \ldots)$, setting $\inf \emptyset \coloneqq \infty$. So, I_k is the kth time that a Bernoulli variable is 1, that is, $B_i = 1$ if $i \in \{I_1, \ldots, I_n\}$ and $B_i = 0$ otherwise. We simulate a realization of $(I_1, \ldots, I_n)$ as follows. For any integers $0 \le \ell < k$ such that $\prod_{\ell<j<\infty}(1-\lambda_j) < 1$, define

$$p(k \mid \ell) \coloneqq \frac{\lambda_k \prod_{\ell<j<k}(1-\lambda_j)}{1 - \prod_{\ell<j<\infty}(1-\lambda_j)},$$

where $\prod_{\ell<j<k}(1-\lambda_j) \coloneqq 1$ if $k = \ell + 1$. Set $i_0 \coloneqq 0$. Then, for $k = 1, \ldots, n$ and any integers $i_n > \ldots > i_1 \ge 1$,

$$\begin{aligned} &\mathrm{P}(I_k = i_k \mid I_0 = i_0, \ldots, I_{k-1} = i_{k-1}, I_k < \infty) = \\ &\quad \mathrm{P}(I_k = i_k \mid I_{k-1} = i_{k-1}, I_k < \infty) = p(i_k \mid i_{k-1}) \end{aligned}$$

and

$$\mathrm{P}(I_{n+1} = \infty \mid I_0 = i_0, \ldots, I_n = i_n) = \prod_{k>i_n} (1 - \lambda_k).$$

Hence, a simulation of $(I_1, \ldots, I_n)$ can be generated by the following acceptance-rejection algorithm: (i) For $k = 1, \ldots, n$, generate a proposal i_k from the probability mass function $p(\cdot \mid i_{k-1})$. (ii) Return $(I_1, \ldots, I_n) = (i_1, \ldots, i_n)$ with probability $\prod_{k>i_n}(1 - \lambda_k)$, else go to (i).

For the generation of the proposal in (i), we use inversion sampling: set $\lambda_0 \coloneqq 1$ and $F(m|\ell) \coloneqq \sum_{\ell<k\leq m} p(k|\ell)$ for integers $0 < \ell < k \leq m$, which may be computed using the recursion

$$p(\ell + 1 \mid \ell) = \frac{\lambda_{\ell+1}}{1 - \prod_{\ell<j<\infty}(1 - \lambda_j)}, \qquad p(k \mid \ell) = p(k-1 \mid \ell)\frac{\lambda_k(1 - \lambda_{k-1})}{\lambda_{k-1}}$$

if $k > \ell + 1$. Then, generate U from a uniform distribution on $[0, 1]$ and return $\inf\{m > \ell \mid F(m|\ell) \geq U\}$ as a simulation of the proposal.

As we need to truncate the infinite products $\prod_{k>i_n}(1 - \lambda_k)$ and $\prod_{\ell<j<\infty}(1 - \lambda_j)$ by only considering a finite number of eigenvalues, we only get an approximate simulation. For the choice of truncation, we used the default in the function `simulate.detpointprocfamily` from `spatstat`, which was also the function we used to make unconditional simulations.

C Details for maximum likelihood estimation for Gibbs point processes

Estimation of θ is complicated by the fact that the normalizing constant is in general not expressible on closed form: the conditional density of $X_{W_{\ominus R}}$ given $X_{\partial W_{\ominus R}} = x_{\partial W_{\ominus R}}$ has normalizing constant

$$c_{W_{\ominus R},\theta}(x_{\partial W_{\ominus R}}) = \sum_{n=0}^{\infty} \frac{\exp(-|W_{\ominus R}|)}{n!} \int_{W_{\ominus R}} \cdots \int_{W_{\ominus R}} a(y \cup x_{\partial W_{\ominus R}}) \beta^n \exp(\psi^T s(y \cup x_{\partial W_{\ominus R}}))\, \mathrm{d}y_1 \cdots \mathrm{d}y_n$$

where $y = \{y_1, \ldots, y_n\}$ and the term for $n = 0$ is interpreted as $\exp(-|B|)$. Often, the interaction parameter ψ is of main interest in which case, following Ripley (1977), it may be reasonable to further condition on $N(W_{\ominus R}) = m$. Conditional on $N(W_{\ominus R}) = m$, let the random vector $(X_1, \ldots, X_m)$ be any ordering of the m points in $X_{W_{\ominus R}}$, which conditioned on both $X_{\partial W_{\ominus R}} = x_{\partial W_{\ominus R}}$ and $N(W_{\ominus R}) = m$ has probability density function

$$f_\psi(x_1, \ldots, x_m \mid x_{\partial W_{\ominus R}}, m) \coloneqq \frac{a(\{x_1, \ldots, x_m\} \cup x_{\partial W_{\ominus R}}) \exp(\psi^T s(\{x_1, \ldots, x_m\} \cup x_{\partial W_{\ominus R}}))}{c_\psi(x_{\partial W_{\ominus R}}, m)}$$

on $W_{\ominus R}^m$ where

$$c_\psi(x_{\partial W_{\ominus R}}, m) = \int_{W_{\ominus R}} \cdots \int_{W_{\ominus R}} a(y \cup x_{\partial W_{\ominus R}}) \exp(\psi^T s(y \cup x_{\partial W_{\ominus R}})) \, \mathrm{d}y_1 \cdots \mathrm{d}y_m$$

with $y = \{y_1, \ldots, y_n\}$. This conditional density does not depend on β and has $s(x)$ as a sufficient statistic for ψ. The m-fold integral above may be hard to compute, but at least $c_\psi(x_{\partial W_{\ominus R}}, m)$ is simpler than $c_{W_{\ominus R},\theta}(x_{\partial W_{\ominus R}})$. Thus, it seems appealing to condition on both $X_{\partial W_{\ominus R}} = x_{\partial W_{\ominus R}}$ and $N(W_{\ominus R}) = m$ when using maximum likelihood estimation. However, in general, $c_{W_{\ominus R},\theta}(x_{\partial W_{\ominus R}})$ considered as a function of θ cannot be written as a product of two functions with one depending on β only and the other depending on ψ only, and so we cannot appeal to one of the known conditioning principles: in general, using a terminology as in Barndorff-Nielsen (1978), n will not be an S-ancillary statistic for ψ, and $s(x)$ will not be an S-sufficient statistic for ψ.

No matter if we condition on $N(W_{\ominus R}) = m$ or not, the likelihood function is log-concave. An approximate maximum likelihood estimate of ψ (and β if we do not condition on $N(W_{\ominus R}) = m$) can be found by combining simulations with importance sampling to obtain an approximate likelihood function which is log-concave too, see Møller and Waagepetersen (2004) and the references therein. Typically, unless X is 'close' to a Poisson process, long runs of Markov chains are needed for the simulations. The simulations and the computations will be faster when conditioning on $N(W_{\ominus R}) = m$ because the normalizing constant is simpler to approximate and since a single point updating Metropolis algorithm can be used for simulations; in the unconditional case, the more advanced birth-death Metropolis-Hastings algorithm (Geyer and Møller, 1994) is used.

D Details for maximum pseudo-likelihood estimation for Gibbs point processes

When defining pseudo-likelihood functions below, we need the Papangelou conditional intensity for the density in (8), which is

$$\lambda_{B,\theta}(x_B, u \,|\, x_{\partial B}) \coloneqq \frac{f_{B,\theta}(x_B \cup \{u\} \,|\, x_{\partial B})}{f_{B,\theta}(x_B \,|\, x_{\partial B})} = \beta a(x_B \cup x_{\partial B}, u) \exp(\psi^T s(x_B \cup x_{\partial B}, u))$$

for $u \in B \setminus x_B$. Furthermore, in order to account for edge effects, we let $B = W_{\ominus R}$ and consider

$$\lambda_{W_{\ominus R},\theta}(x_{W_{\ominus R}}, u \,|\, x_{\partial W_{\ominus R}}) = \lambda_\theta(x, u) \coloneqq \beta a(x, u) \exp(\psi^T s(x, u)) \tag{13}$$

for the observed point pattern $x \subset W$.

First, consider the case where we do not condition on $N(W_{\ominus R}) = m$. Then, the log pseudo-likelihood function is

$$pl(\theta) \coloneqq -\int_{W_{\ominus R}} \lambda_\theta(x,u)\,\mathrm{d}u + \sum_{i=1}^{m} \log \lambda_\theta(x \setminus \{x_i\}, x_i). \tag{14}$$

If we fix ψ and insert (13) into (14), we see that

$$\hat{\beta}(\psi) \coloneqq m / \int_{W_{\ominus R}} a(x,u) \exp(\psi^T s(x,u))\,\mathrm{d}u$$

is the maximum pseudo-likelihood estimate of β; hence, the profile log pseudo-likelihood function for ψ becomes $pl(\hat{\beta}(\psi), \psi)$. Assuming that we can interchange differentiation and integration, the pseudo-score function is

$$\begin{aligned} s(\psi) \coloneqq \frac{\partial}{\partial \psi} pl(\hat{\beta}(\psi), \psi) = &- m \frac{\int_{W_{\ominus R}} a(x,u) s(x,u) \exp(\psi^T s(x,u))\,\mathrm{d}u}{\int_{W_{\ominus R}} a(x,u) \exp(\psi^T s(x,u))\,\mathrm{d}u} \\ &+ \sum_{i=1}^{m} s(x \setminus \{x_i\}, x_i), \end{aligned} \tag{15}$$

which has a negative definite derivative. Thus, the profile log pseudo-likelihood function is concave (and strictly concave under mild conditions), so the maximum pseudo-likelihood estimate of ψ (provided it exists) can be found by a numerical optimization method where evaluating (14) and (15) involves approximating the integrals by numerical methods (Baddeley et al., 2014).

Second, we condition on both $X_{\partial W_{\ominus R}} = x_{\partial W_{\ominus R}}$ and $N(W_{\ominus R}) = m$. For $i = 1, \ldots, m$, define $X_{-i} \coloneqq (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_m)$ and $x_{-i} \coloneqq (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m)$. The random vectors $X_{-1}, \ldots, X_{-m}$ are identically distributed. Further, X_i conditioned on both $X_{\partial W_{\ominus R}} = x_{\partial W_{\ominus R}}$, $N(W_{\ominus R}) = m$ and $X_{-i} = x_{-i}$ depends only on $x \setminus \{x_i\}$ and has probability density function

$$f_\psi(u \mid x \setminus \{x_i\}) \coloneqq a(x \setminus \{x_i\}, u) \exp(\psi^T s(x \setminus \{x_i\}, u)) / c_\psi(x \setminus \{x_i\})$$

for $u \in W_{\ominus R}$ where

$$c_\psi(x \setminus \{x_i\}) = \int_{W_{\ominus R}} a(x \setminus \{x_i\}, u) \exp(\psi^T s(x \setminus \{x_i\}, u))\,\mathrm{d}u.$$

Now, the logarithm of Besag's pseudo-likelihood function (Besag, 1975) as defined by the product of the 'full conditionals' $f_\psi(u \mid x \setminus \{x_i\})$ $(i = 1, \ldots, m)$ becomes

$$pl_m(\psi) \coloneqq \psi^T \sum_{i=1}^{m} s(x \setminus \{x_i\}, x_i) - \sum_{i=1}^{m} \log c_\psi(x \setminus \{x_i\}) \tag{16}$$

when we omit the term $\sum_{i=1}^{m} \log a(x \setminus \{x_i\}, x_i)$, which only depends on the data. Assuming we can interchange differentiation and integration, the pseudo-score corresponding to (16) is

$$s_m(\psi) \coloneqq \frac{\partial}{\partial \psi} pl_m(\psi) = \sum_{i=1}^{m} s(x \setminus \{x_i\}, x_i) - \sum_{i=1}^{m} \int_{W_{\ominus R}} s(x \setminus \{x_i\}, u) f_\psi(u \mid x \setminus \{x_i\}) \, \mathrm{d}u,$$

and its derivative is

$$\begin{aligned} \frac{\partial}{\partial \psi^T} s_m(\psi) &= -\sum_{i=1}^{m} \left(\int_{W_{\ominus R}} s(x \setminus \{x_i\}, u) s(x \setminus \{x_i\}, u)^T f_\psi(u \mid x \setminus \{x_i\}) \, \mathrm{d}u \right. \\ &\quad \left. - \int_{W_{\ominus R}} s(x \setminus \{x_i\}, u) f_\psi(u \mid x \setminus \{x_i\}) \, \mathrm{d}u \int_{W_{\ominus R}} s(x \setminus \{x_i\}, u)^T f_\psi(u \mid x \setminus \{x_i\}) \, \mathrm{d}u \right) \\ &= -\sum_{i=1}^{m} \mathrm{Var}_\psi \left[s(x \setminus \{x_i\}, X_i) \mid x \setminus \{x_i\} \right]. \end{aligned}$$

So, the log pseudo-likelihood is again concave (and strictly concave under mild conditions) and can be optimized numerically using numerical evaluation of integrals, but the computations may be more cumbersome compared to (14) and (15) because we need to evaluate more integrals. However, it is still easier than using maximum likelihood estimation.

References

Baddeley, A., Coeurjolly, J.-F., Rubak, E., and Waagepetersen, R. (2014). Logistic regression for spatial Gibbs point processes. *Biometrika*, 101:377–392.

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

Barndorff-Nielsen, O. E. (1978). *Information and Exponential Families in Statistical Theory*. Wiley, New York.

Besag, J. (1975). Statistical analysis of non-lattice data. *The Statistician*, 24:179–195.

Besag, J. (1977). Some methods of statistical analysis for spatial data. *Bulletin of the International Statistical Institute*, 47:77–91.

Georgii, H.-O. (1988). *Gibbs Measures and Phase Transition*. Walter de Gruyter, Berlin.

Geyer, C. J. and Møller, J. (1994). Simulation procedures and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, 21:359–373.

Jensen, J. L. and Møller, J. (1991). Pseudolikelihood for exponential family models of spatial point processes. *Annals of Applied Probability*, 1:445–461.

Lavancier, F., Møller, J., and Rubak, E. (2015). Determinantal point process models and statistical inference. *Journal of Royal Statistical Society: Series B (Statistical Methodology)*, 77:853–877.

Macchi, O. (1975). The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7:83–122.

Moran, P. (1952). A characteristic property of the Poisson distribution. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48:206–207.

Mrkvička, T., Myllymäki, M., Jilik, M., and Hahn, U. (2020). A one-way ANOVA test for functional data with graphical interpretation. *Kybernetika*, 56:432–458.

Myllymäki, M. and Mrkvička, T. (2019). GET: Global envelopes in R. arXiv preprint arXiv:1911.06583.

Myllymäki, M., Mrkvička, T., Grabarnik, P., Seijo, H., and Hahn, U. (2017). Global envelope tests for spatial processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79:381–404.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25:451–482.

Møller, J. and Waagepetersen, R. (2017). Some recent developments in statistics for spatial point patterns. *Annual Review of Statistics and Its Applications*, 4:317–342.

Møller, J. and Waagepetersen, R. P. (2004). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman and Hall/CRC, Boca Raton.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ripley, B. D. (1977). Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39:172–212.

Ripley, B. D. (1979). Algorithm AS 137: simulating spatial patterns: dependent samples from a multivariate density. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28:109–112.

Ripley, B. D. (1988). *Statistical Inference for Spatial Processes*. Cambridge University Press, Cambridge.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Paper E

Using neural networks to estimate parameters in spatial point process models

Ninna Vihrs

The paper has been published in
Spatial Statistics (2022) **51**, 100668

The layout has been revised.

Abstract

In this paper, I show how neural networks can be used to simultaneously estimate all unknown parameters in a spatial point process model from an observed point pattern. The method can be applied to any point process model which it is possible to simulate from. Through a simulation study, I conclude that the method recovers parameters well and in some situations provide better estimates than the most commonly used methods. I also illustrate how the method can be used on a real data example.

1 Introduction

Briefly, a point process X may be defined as a countable random subset of $\mathbb{R}^d$. Usually, a realisation of the process is only observed within a bounded set $W \subset \mathbb{R}^d$. A common problem is to fit a parametric spatial point process model to a realization x. This can be a difficult problem since the likelihood function is intractable except in the very simple case of a Poisson process. Many alternative approaches have thus been suggested including estimation based on pseudo-likelihood, composite likelihood, and minimum contrasts (see the review in Møller and Waagepetersen, 2017). However, it is possible to define meaningful spatial point process models for which both the intensity and other moment characteristics of X, the density, and the Papangelou conditional intensity (see e.g. Møller and Waagepetersen, 2004) are not expressible in closed form. Then, the above methods are not feasible. An example of such a point process model is the LGCP-Strauss process presented in Vihrs et al. (2022) where the authors found it necessary to consider parameter estimation in a Bayesian setup because it was then possible to use the method of approximate Bayesian computations (ABC) which is based entirely on the ability to simulate under the model (see e.g. the overview of some ABC methods in Beaumont (2010)).

The purpose of this paper is to explore the possibility to estimate parameters in spatial point process models by using neural networks. The idea is to consider the estimation problem as a prediction problem where parameters of the model are to be predicted from a realization. This prediction problem can be handled with machine learning methods, such as neural networks, trained on a suitable training data set. Thus the only requirement for this approach is to be able to construct a number of training cases consisting of values for the unknown parameters and realizations of the model corresponding to these parameter values. If it is known how to simulate from the model, the training data set for the chosen machine learning method can be constructed from simulations of the model. Thus the approach, like ABC, only requires the model to be equipped with a feasible simulation procedure. The idea is somewhat similar to the concept behind the ABC technique in

Marin et al. (2019) where random forests are used to predict mean, variance, and quantiles in the posterior distribution. However, I instead use neural networks for the prediction task, and is furthermore only interested in obtaining point estimates for the unknown parameters and do not attempt to get knowledge about a posterior distribution. Gabrielli et al. (2017) have used neural networks to predict parameters in acoustic physical modelling, but the idea has to the best of my knowledge not been explored in relation to spatial point process models. I explain the suggested approach for using neural networks to obtain point estimates of unknown parameters of spatial point process models in Section 2.

Neural networks have proven useful in many different prediction problems and are well suited to handle many different types of input including images and curves. Their ability to handle different input formats is an advantage when attempting to pass information about a relatively complex data structure like a point pattern. In Section 2.1, I discuss how to pass information about a point pattern to a neural network aiming at predicting unknown parameters; I decide on summarising important aspects of the point pattern by means of a functional summary statistic and then passing this information to the neural network, thereby using the possibility to handle input data in the form of curves.

As I mention above, there are many different estimation procedures for spatial point process models, and which one it is preferable to use depends on the type of model and the theoretical knowledge available for that class of models. A clear advantage of simulation based methods, like the neural network approach in this paper, is that they are generally applicable to all point process models for which it is possible to generate realizations. Since the ability to simulate from the model must be considered necessary for any model of practical value, this requirement is not very restrictive. Good simulation based methods thus allow us to use almost any type of spatial point process model without being limited by lack of theoretical knowledge when it comes to parameter estimation. Another clear advantage of the suggested neural network approach in this paper is that all unknown parameters can always be estimated simultaneously, which is not always the case in traditional estimation procedures. For instance, I consider the example of a Strauss process in Section 3.2 where parameters are usually fitted with the method of maximum pseudo likelihood estimation, but the Strauss process contains an interaction radius R, and this parameter cannot be estimated alongside the other parameters when using maximum pseudo likelihood estimation. Finally, I show through the simulation study in Section 3 that the suggested neural network approach recovers parameters well, and compared to the most commonly used estimation procedures it gives either better or similar results.

All statistical computations in this paper were made with the open source software `R` version 4.0.2 (R Core Team, 2020). The `R`-packages `ggplot2` ver-

sion 3.3.2 (Wickham, 2016), `spatstat` version 2.1-0 (Baddeley et al., 2015), and `keras` version 2.3.0.0 (Allaire and Chollet, 2020, Chollet, 2018) were used to make plots, handle spatial point patterns, and train neural networks, respectively. The *R*-scripts I wrote for the simulation studies and data example in this paper are available in the Supplementary material which can be found at `https://www.sciencedirect.com/science/article/pii/S2211675322000422`.

2 The neural network approach

In this section, I explain the suggested neural network approach to parameter estimation in spatial point process models. I restrict attention to models without covariates and leave the case of inhomogeneous models to future research. The objective is to train a neural network to predict the values of parameters in a chosen point process model based on a realisation from the process. When the neural network has been trained, it can be used to estimate the parameters of the point process model based on an observed point pattern x_{obs}.

2.1 Considerations regarding training data

In order to train the neural network, training data is constructed by simulating a number of point patterns from the chosen model for different values of the parameters. Neural networks are known to be able to take input data in many forms including pixel images and sequences. One way to pass a point pattern dataset to a neural network would be to represent it as a pixel image where the pixel values corresponds to the number of points within the pixel. I tried to send data in this form through a 2-dimensional convolutional neural network, which is a good choice for handling image input, but this method seemed to be less successful than summarising the point pattern dataset with functional summary statistics as explained below. This may be because the behaviour of a point pattern at a very small scale is important for estimating some parameters accurately, and such information was lost in the discretization of the pattern but not when summarising aspects of the pattern with a functional summary statistic. The need to choose an appropriate summary statistic brings some arbitrariness and subjectivity to the method which would not have been the case if using the entire point pattern as input, but it is not uncommon to estimate parameters in spatial point process models based on some functional summary statistic as this is also done in the popular method of minimum contrast estimation (see Section 3.1). In this paper, I therefore choose to use some appropriate summaries calculated from the point pattern as input to the neural network.

A common way to summarise many important aspects of a point pattern is by means of functional summary statistics where I briefly describe some common choices here and refer the reader to Baddeley et al. (2015) for more details. A common choice is Ripley's K-function which depends on an inter point distance r. Assuming stationarity of the point process, if ρ is the intensity of the process, the interpretation of K is that $\rho K(r)$ is the expected number of further points falling in a ball with radius r centered at a typical point of the process. One often considers its transformation $L(r) = \sqrt[d]{K(r)/\omega_d}$ where ω_d is the volume of a d-dimensional unit sphere. It is known that $L(r) = r$ in case of a stationary Poisson process, which is the case of complete spatial randomness. If $L(r) - r < 0$ ($L(r) - r > 0$), it is usually interpreted as the point process exhibiting regularity/repulsion (clustering/attraction) at inter-point distances r. Non-parametric estimates of K and L from a point pattern $x = \{x_1, \ldots, x_n\}$ on an observation window W are

$$\widehat{K}_x(r) = \frac{|W|}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} 1[\|x_i - x_j\| \leq r] e_{ij}(r), \quad \widehat{L}_x(r) = \sqrt[d]{\widehat{K}_x(r)/\omega_d}$$

where $|W|$ is the Lebesgue measure of W and $e_{ij}(r)$ is an edge correction weight to account for the unobserved points outside W. Baddeley et al. (2015) noted that it is not so important which edge correction method to use as long as some correction is used; I used Ripley's isotropic correction (Ripley, 1988, Ohser, 1983). I use $\hat{L}(r) - r$ as input to the neural network since parameters of point processes are usually related to regularity and clustering, and it was found to give better results than using Ripley's K-function or the L-function directly, which may suggest that this transformation of K allows the neural network to learn more efficiently. Note however that the suggested neural network approach can easily be used with a different functional summary statistic as input.

Other popular summary functions for point processes include the so-called F-, G-, and J-functions defined for a stationary point process X by

$$F(r) = P(X \cap b(0, r) \neq \emptyset), \tag{1}$$

$$G(r) = P((X \setminus \{u\}) \cap b(u, r) \neq \emptyset \mid u \in X), \quad \text{and} \tag{2}$$

$$J(r) = \frac{1 - G(r)}{1 - F(r)}, \quad F(r) < 1, \tag{3}$$

where $b(u, r)$ is the ball centered at u with radius r, see e.g. Møller and Waagepetersen (2004) and the references therein. Stationarity implies that the definition of $G(r)$ does not depend on the choice of u. The F-, G-, and J-functions are however not considered further as input to the neural network, since they can usually only be estimated reliably for a smaller range of r-values than K, and this was found to be a disadvantage for the example in Section 3.3 where large scale properties had to be summarised.

The number of points in the point pattern was also included in the input to the neural network, since this is important knowledge regarding some parameters of most point process models which the L-function is generally not able to provide.

2.2 The suggested neural network approach

The suggested method for estimation is as follows:

1. Choose a homogeneous spatial point process model $M(\theta)$ with unknown parameters $\theta = (\theta_1, \ldots, \theta_k)$, a number of training cases n_{train}, and optionally a number of test cases n_{test}.

2. Construct training data:

   (a) For $i = 1, \ldots, n_{\text{train}}$, sample the parameters $\tilde{\theta}^i = (\tilde{\theta}^i_1, \ldots, \tilde{\theta}^i_k)$ from some pre-chosen distribution for θ. In this paper, I sample the $\tilde{\theta}^i_j$'s independently and uniformly on bounded intervals.

   (b) For $i = 1, \ldots, n_{\text{train}}$, sample $\tilde{x}^i$ from $M(\tilde{\theta}^i)$.

   (c) Choose some values $r_1, \ldots, r_m$. For $i = 1, \ldots, n_{\text{train}}$ calculate the functional summary statistic $L_i = (\widehat{L}_{\tilde{x}^i}(r_1) - r_1, \ldots, \widehat{L}_{\tilde{x}^i}(r_m) - r_m)$ and $n(\tilde{x}^i)$ where $n(\cdot)$ is the number of points in a point pattern.

   (d) Standardize each component of $\{L_i, n(\tilde{x}^i), \tilde{\theta}^i_1, \ldots, \tilde{\theta}^i_k\}_{i=1}^{n_{\text{train}}}$, by subtracting the mean and dividing by the standard deviation (for $\{L_i\}_{i=1}^{n_{\text{train}}}$ the mean and standard deviation were calculated both over all n_{train} simulations and over all m values for r meaning that all values of $\{L_i\}_{i=1}^{n_{\text{train}}}$ were scaled by the same amount.) After standardization the training data is $\{L_i, n(\tilde{x}^i), \tilde{\theta}_1, \ldots, \tilde{\theta}_k\}_{i=1}^{n_{\text{train}}}$.

3. (Optional) Construct test data:

   (a) Construct n_{test} test cases $\{L_i, n(\tilde{x}^i), \tilde{\theta}^i_1, \ldots, \tilde{\theta}^i_k\}_{i=1}^{n_{\text{test}}}$ with the same procedure as in items 2a–2c.

   (b) Scale the test data according to item 2d, i.e. subtract the means and divide by the standard deviations calculated in item 2d.

4. Use the training data to train a neural network to predict θ.

5. (Optional) Send the test data through the trained neural network, and asses its predictive performance.

6. Calculate $L_{\text{obs}} = (\widehat{L}_{x_{\text{obs}}}(r_1) - r_1, \ldots, \widehat{L}_{x_{\text{obs}}}(r_m) - r_m)$ and $n(x_{\text{obs}})$; scale these according to item 2d; feed them to the trained neural network; and return the (rescaled) prediction $\hat{\theta}$ as the estimated vector of parameters.

Regarding the choice of the values $r_1, \ldots, r_m$ in item 2c there is a sensible default in the `spatstat` implementation for estimating $L(r)$, which I used. Even though it is optional to construct test data, I strongly recommend to do this in order to asses the performance of the method in a given situation.

As I write in item 2a, I sample each parameter in the training data uniformly on a bounded interval, in which case there should be strong reasons to believe that the parameters corresponding to the observed point pattern fall within these intervals. Otherwise, the trained neural network cannot be expected to do well for the observed point pattern. Note however, that it is possible to consider wide intervals of the parameters, so it is not necessary to have very specific knowledge about the ranges of the parameters. It will usually be possible to obtain some range for each parameter by combining knowledge about the effect of the parameters in the model with a preliminary investigation of the point pattern, which may include interpreting some functional summary statistics and looking at some simulations. I give an idea of how this could be done for the example in Section 4.

The neural network architecture which I chose to use in item 4 is illustrated in Figure 1. The functional summary statistic $\widehat{L}(r) - r$, which constitutes a sequence, is send through a number of 1-dimensional convolution layers and max pooling operations, which is a good way to handle sequenced data.

A 1-dimensional convolution layer takes as input a number of sequences say $s^i = (s^i_1, \ldots, s^i_k)$, $i = 1, \ldots, m$, and returns p sequences of the form $\tilde{s}^i = (\tilde{s}^i_1, \ldots, \tilde{s}^i_{k-(q-1)})$, $i = 1, \ldots, p$, where p is some chosen number and $\tilde{s}^i_j = f(b_i + \sum_{l=0}^{q-1} \sum_{h=1}^{m} a^i_{lh} s^h_{j+l})$ for some activation function f, chosen size q, constant b_i, and weights $a^i_{lh}, l = 0, \ldots, q-1, h = 1, \ldots, q$. I used $p = 64$ in each convolution layer and $q = 7$, so in the first convolution layer $m = 1$ (the functional summary statistic is just one curve) and in the subsequent layers $m = 64$ (the output from the previous layer constitutes 64 curves).

The max pooling operation used between the convolution layers splits every input sequence into sub-sequences of a specified length (I chose 5) and replaces each sub-sequence with its highest value yielding a new sequence usually of much smaller size.

After the convolution layers, the output is fed to two densely connected layers (the output is now considered as individual values instead of sequences) together with the number of points in the observed point pattern. In a densely connected layer which gets input values $I_1, \ldots, I_n$, the output is $O_1, \ldots, O_h$ where h is some chosen number of hidden units and $O_i = f(b_i + \sum_{j=1}^{n} a^i_j I_j)$ for some activation function f, constant b_i, and weights $a^i_j, j = 1, \ldots, n$. The final output of the network is a prediction of the unknown parameters of the spatial point process model based on the functional summary statistic and number of points which was given as input to the model.

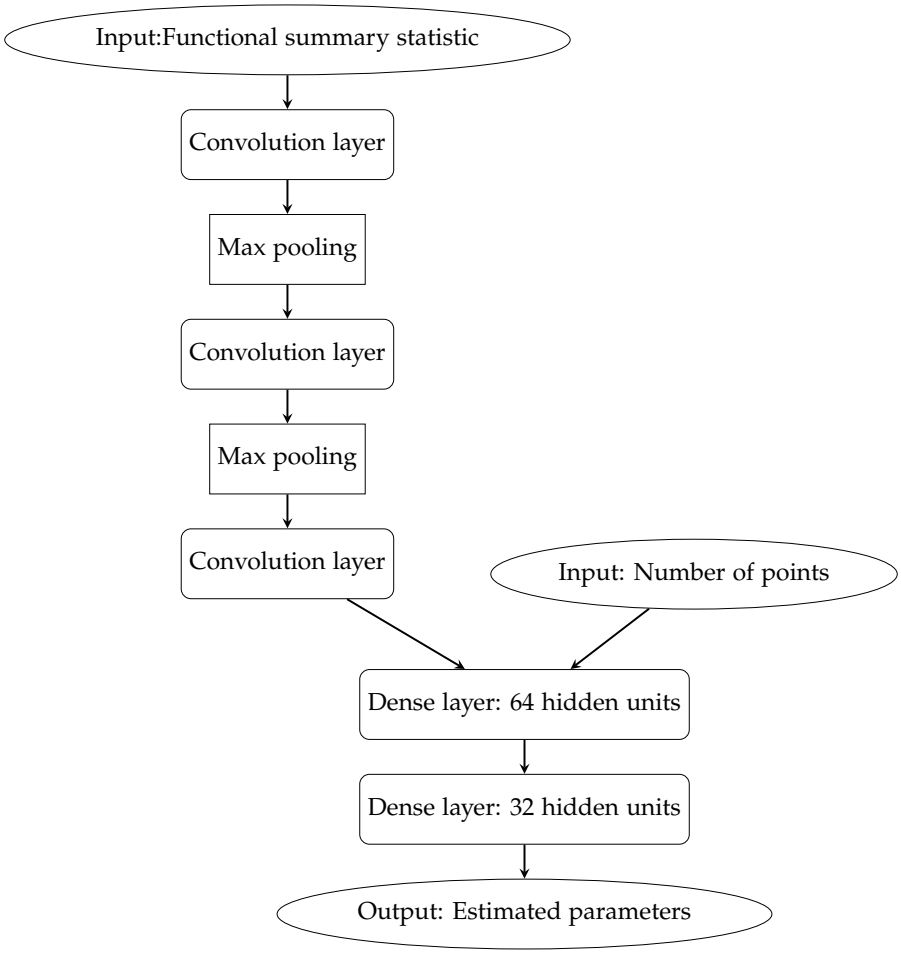

Fig. 1: Visual overview of the neural network architecture.

This final layer is actually also a dense layer where the activation function is just the identity and the number of output values corresponds to the number of unknown parameters in the spatial point process model which is to be estimated.

As the activation function I used $f(\cdot) = \text{relu}(\cdot) = \max(0, \cdot)$ both for the convolution and dense layers, which is a very common choice for the activation function in neural networks. All the above mentioned weights of the linear combinations taken in the neural network and the constants b_i constitute the unknown parameters of the neural network, which should be learned based on the training data. For some details about how these unknown parameters of the neural network were learned see Appendix A.2. For more information about neural networks and how to use them in `R` see e.g. Chollet (2018).

I also tried to use a network only with densely connected layers, which is much faster to train, but it generally gave poorer results than including the convolution layers.

3 Simulation study for examples of point process models

In this section, I consider three classes of parametric spatial point process models as examples: log-Gaussian Cox processes (LGCP) (Møller et al., 1998), Strauss processes (Strauss, 1975, Kelly and Ripley, 1976), and LGCP-Strauss processes (Vihrs et al., 2022). I briefly define these in the following subsections and refer to the above references for more details about these models. The preferred method for estimating parameters in spatial point process models depends on the type of model. Through simulations, I assess the accuracy of estimates obtained with the neural network approach and compare this to the most commonly used estimating procedure in each case, which I briefly describe in each of the following subsections. I will not go into details about simulation methods and instead refer to Møller and Waagepetersen (2004) and Baddeley et al. (2015). In this section, W is always a 2-dimensional unit square. Considerations about how many simulations to use for the training data in each example are provided in Appendix A.1, which also shows histograms of the number of points in the simulations in the training data sets.

3.1 LGCP processes

An LGCP is a popular process for modelling aggregation in spatial point patterns. It is driven by a stochastic intensity $Z = \exp(Y)$ where Y is a Gaussian random field with mean function m and covariance function $c(u, v)$.

I use $m = \mu$ for a constant μ and an exponential covariance function $c(u, v) = \sigma^2 \exp(-\|u - v\|/s)$ with unknown parameters σ^2 and s. The model is then stationary, and the vector of unknown parameters which are to be estimated is (μ, σ^2, s). Note that if $\sigma^2 = 0$, s becomes irrelevant, so any estimation procedure may be expected to struggle with estimating s if σ^2 is small.

Seemingly, the most common way to estimate parameters in Cox processes is to use the method of minimum contrast estimation (Diggle, 1983, Diggle and Gratton, 1984), although composite likelihood estimation (Guan, 2006) is a popular alternative. I have chosen to compare my method to that of minimum contrast estimation. In minimum contrast estimation, the theoretical value of a summary function (e.g. Ripley's K-function) is compared to a non-parametric estimate of it. I used this method to estimate σ^2 and s based on an observation x and Ripley's K-function, which depends on σ^2 and s, by finding the values of σ^2 and s which minimize

$$\int_{a_1}^{a_2} |K(r)^q - \widehat{K}_x(r)^q|^p \, \mathrm{d}r$$

for some user specified $0 \leq a_1 < a_2$ and exponents p and q. Subsequently, μ can be estimated from the unbiased estimation equation of the intensity $\hat{\rho} = n(x)/|W|$ by using that $\rho = \exp(\mu + \sigma^2/2)$ for my considered model. For finding the minimum contrast estimates I used the function `kppm` from `spatstat` with the default settings, which include $p = 2$ and $q = 1/4$.

I made the training data for the neural network approach based on 10,000 simulations of an LGCP with parameters sampled uniformly in the intervals $\mu \in (4, 6)$, $\sigma^2 \in (0, 4)$, and $s \in (0.001, 0.1)$. For justification of the number of simulations in the training data see Appendix A.1. I made further 5,000 simulations for a test set, and Figures 2–4 show some plots for the estimated parameters obtained with the neural network approach and the method of minimum contrast estimation. In the case of minimum contrast estimation, some extreme estimates were omitted in Figures 2 and 3, see the captions for more details. Overall, there is less variation in the error of the estimates obtained with the neural network approach compared to the method of minimum contrast estimation, especially when the true parameter is high. Furthermore, the neural network approach does not give the same kind of extremely wrong estimates as sometimes seen with minimum contrast estimation, probably because it has only seen training data with parameters in the same intervals as in the test set.

For μ, both methods recover the parameter well, but the neural network approach has unlike the method of minimum contrast estimation a slight tendency to overestimate the parameter when the true value is small.

For σ^2, the neural network approach recovers the parameter well when the true value is less than three, especially when the true value is very small where it also performs considerably better than minimum contrast estimation

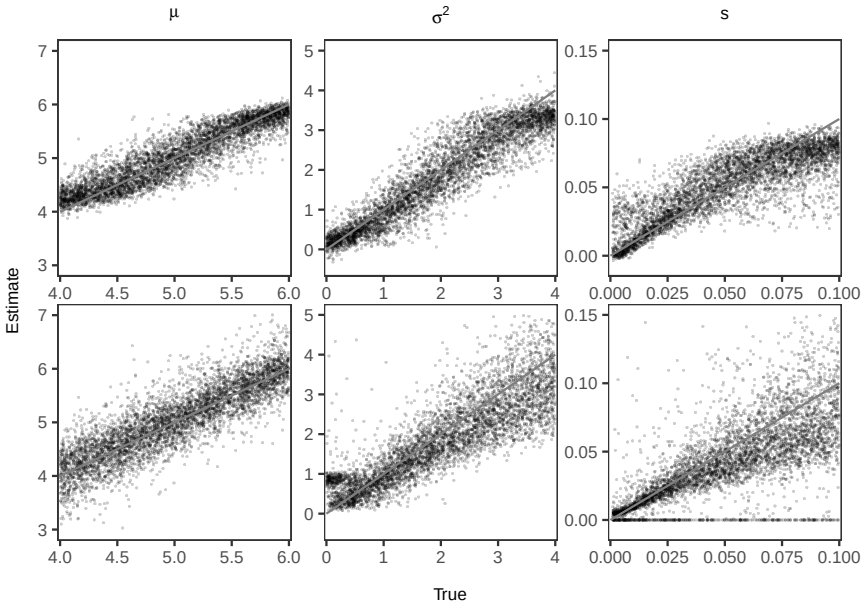

Fig. 2: Estimated parameters obtained with the neural network approach (top row) and minimum contrast estimation (bottom row) plotted against true parameters in an LGCP. The solid gray line is the identity line. In the case of minimum contrast estimation, 15, 43, and 77 cases where the estimate of μ was below 3, σ^2 was above 5, and s was above 0.15, respectively, were omitted from the respective plots; the smallest estimate of μ was -0.174, the highest estimate of σ^2 was 10.7, and the highest estimate of s was 78,738.35.

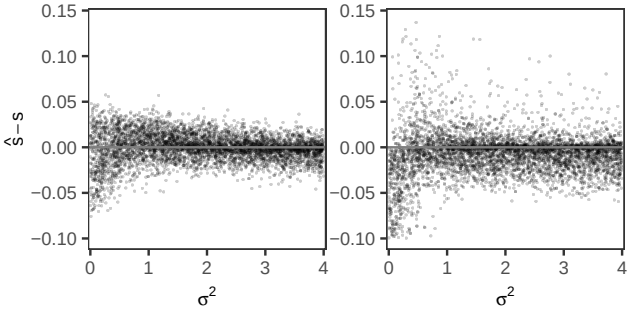

Fig. 3: Estimate of s minus the true value plotted against σ^2 for a log-Gaussian Cox process. In the plot to the left, estimates were obtained with the neural network approach; in the plot to the right, estimates were obtained with minimum contrast estimation. In the case of minimum contrast estimation, 34 cases where the error fell outside the showed range were omitted.

since the latter quite often seems to estimate the parameter to be near one when it is in fact near zero. When the true parameter is high, both methods

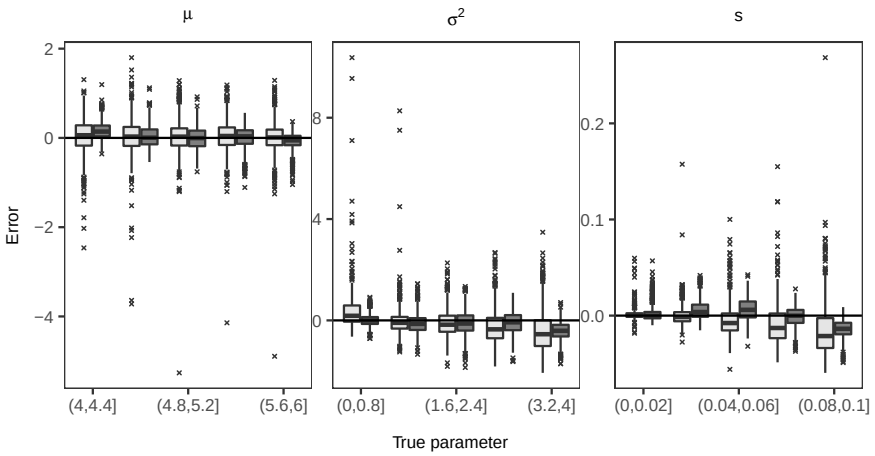

Fig. 4: Boxplot of the errors (estimated minus true value) for the parameter of an LGCP stated at the top of each plot. The estimates were obtained with the neural network approach (dark gray) and minimum contrast estimation (light gray). For the parameter s, cases where $\sigma^2 > 1$ were omitted.

have a tendency for underestimation. However, in the case of minimum contrast estimation, this tendency starts to be clear when the true value gets above circa 2.5 whereas it for the neural network approach only starts to be clear when the true value gets above circa 3.5.

For s, Figure 3 shows that both methods as expected struggle to recover s when σ^2 is near 0. In this case, the neural network approach has a tendency to estimate s to be near the mean in the training data whereas the method of minimum contrast has a tendency to estimate it to be near 0. There is no reason to prefer any of these strategies above the other, so for a more fair comparison, cases where $\sigma^2 < 1$ has been excluded from the plot for s in Figure 4 and we see that the excluded cases include the most extreme estimates of s achieved with minimum contrast estimation. Both methods recover s well when the true value is small (and σ^2 is not near 0). When the true value of s gets above circa 0.3, the method of minimum contrast develops a tendency for underestimation, which gets more severe as s increases, and the neural network approach starts to slightly overestimate s until the true value gets above circa 0.8 after which it also underestimates s, but not as severely as minimum contrast estimation.

In order to asses how the method performs on point patterns with few points, I made a second simulation study where I considered $\mu \in (3, 4)$ for the training and test data. I used 1,000 simulations in the test set and everything else was as above. Figure 5 shows the errors of the estimates obtained with the neural network approach and minimum contrast estimation for each

parameter for the test cases where the number of points was below 200. This

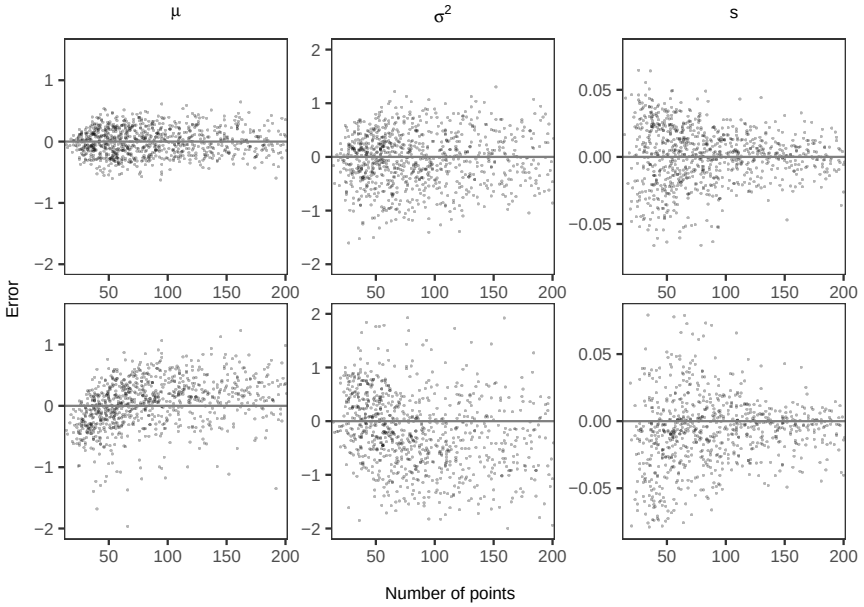

Fig. 5: Estimated minus true value plotted against the number of points in the point pattern. The parameter of the LGCP is stated at the top of each column. In the top row, estimates were obtained with the neural network approach; in the bottom row, estimates were obtained with minimum contrast estimation. In the case of minimum contrast estimation, 2 cases of μ, 19 cases of σ^2, and 100 cases of s where the error fell outside the showed ranges were omitted.

shows that it is mainly the estimation of s which benefits from more points in the point pattern. It is also seen that minimum contrast estimation has a tendency to underestimate μ and s and overestimate σ^2 if there are very few points in the point pattern, but the neural network approach shows no such tendencies.

3.2 Strauss processes

A Strauss process is a popular model for regularity. A Strauss process defined on a bounded set $S \subset \mathbb{R}^d$ has density $f(x) \propto \beta^{n(x)}\gamma^{S_R(x)}$ with respect to a unit rate Poisson process for $x = \{x_1, \ldots, x_n\} \subset S$ where $n(x)$ is the number of points in x, $S_R(x) = \sum_{i<j} 1[\|x_i - x_j\| \leq R]$ is the number of R-close pairs, and the unknown parameters are $\beta > 0$, $\gamma \in [0, 1]$, and $R \geq 0$. I assume that $W \subset S$ but that S is unknown, so when simulating from the Strauss process, I simulate it on an extended window determined by the default settings in the function `rmh` from `spatstat`, which in this case is to add a margin of width $2R$ around all sides of the square W. Note that if $\gamma = 1$, R becomes irrelevant;

and if $R = 0$, γ becomes irrelevant. Both these special cases collapses into the same model namely a homogeneous Poisson process.

The density of the Strauss process involves an intractable normalising constant, so instead of using maximum likelihood estimation it is more common to use maximum pseudo likelihood estimation (Besag, 1975, Ripley, 1988, Jensen and Møller, 1991, Baddeley and Turner, 2000), which is known to be a fast and reliable alternative. The pseudo likelihood function for an observed point pattern x is

$$PL_A(\theta) = \exp\left(-\int_A \frac{f(x \cup \{u\})}{f(x)}\,\mathrm{d}u\right) \prod_{u \in x \cap A} \frac{f(x)}{f(x \setminus \{u\})}$$

for some set $A \subset S$ chosen in order to account for edge effects. The pseudo likelihood function is maximised in order to find the maximum pseudo likelihood estimate of an unknown parameter vector θ of the density f. This method is particularly tractable if the model is on exponential family form, that is the unnormalised density h is of the form $h(x) = \exp(t(x)\theta^T)$ for a canonical parameter vector θ and canonical statistic $t(x)$. This is the case for the Strauss process if R is given with $\theta = (\log(\beta), \log(\gamma))$ and $t(x) = (n(x), S_R(x))$. Thus, maximum pseudo-likelihood estimation can easily be used to obtain estimates of β and γ. In order to also obtain an estimate for R, the method of profile maximum pseudo likelihood can be used in the following way: consider a finite set $R_1, \ldots, R_k$ of possible values for R, find the maximum pseudo likelihood estimates $\hat{\beta}_i$ and $\hat{\gamma}_i$ of β and γ given $R = R_i$, and choose the combination of parameters $(\hat{\beta}_i, \hat{\gamma}_i, R_i)$ which gives the highest value of PL_A. For finding the profile maximum pseudo likelihood estimates I used the function `profilepl` from `spatstat` where I forced the method to yield a valid model ($\gamma \in [0, 1]$) and considered 50 equally spaced values of R in the interval $[0.001, 0.05]$.

For the neural network approach, I used 5,000 simulations of a Strauss process with parameters sampled in the intervals $\beta \in (200, 900)$, $\gamma \in (0, 1)$, and $R \in (0, 0.05)$ for the training data and made further 5,000 simulations for a test data set. For justification of the number of simulations in the training data see Appendix A.1. The simulation of Strauss processes which I used involves Markov chains, and a shared burnin for all simulations was chosen based on trace plots of the number of points and R-close pairs for certain combinations of the parameters believed to require the most iterations. Based on this, I used 100,000 iterations of the Markov chain. Figures 6–8 show some plots for the estimated parameters for the point patterns in the test set. The estimates were obtained with either the neural network approach or profile maximum pseudo likelihood estimation. A clear advantage of the neural network approach is that all parameters can be estimated simultaneously, and the estimate of R is thus not restricted to a finite set of values.

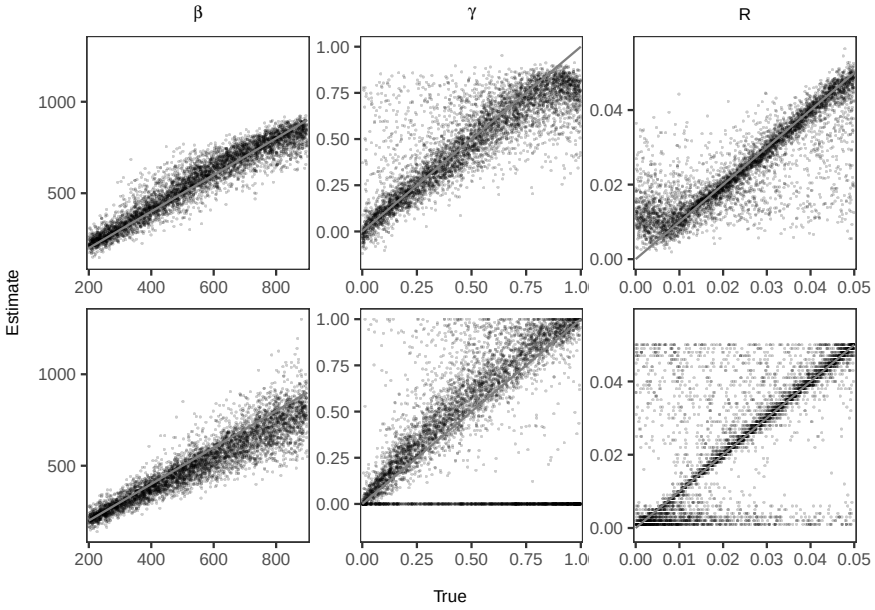

Fig. 6: Estimated parameters obtained with the neural network approach (top row) and profile maximum pseudo likelihood (bottom row) plotted against true parameters in a Strauss process. The parameter is stated at the top of each column. The solid gray line is the identity line.

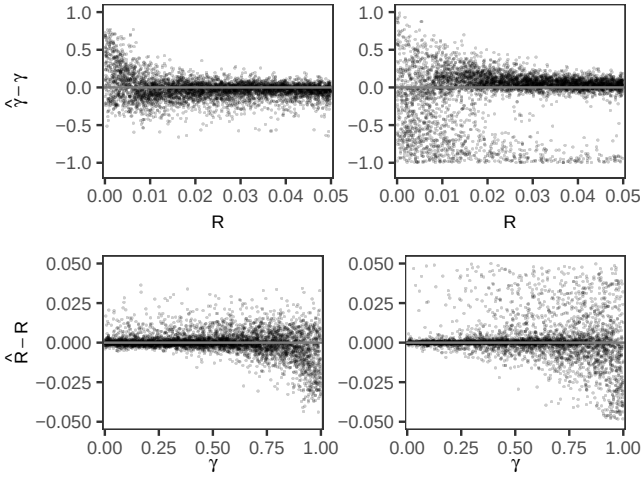

Fig. 7: Estimates of γ and R minus their true value plotted against R and γ, respectively, for a Strauss process. Estimates were obtained with the neural network approach (left) and maximum profile pseudo likelihood estimation (right).

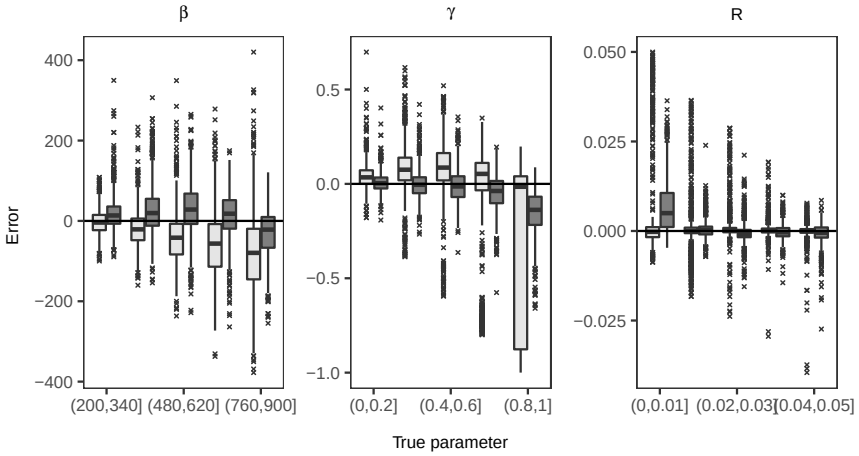

Fig. 8: Boxplot of the errors (estimated minus true value) for the parameter of a Strauss process stated at the top of the plots. The estimates were obtained with maximum profile pseudo likelihood estimation (light gray) and the neural network approach (dark gray). For the parameters γ and R, cases where $R < 0.01$ and $\gamma > 0.7$, respectively, were omitted.

For β, there is overall less variation in the error obtained with the neural network approach compared to the method of profile maximum pseudo likelihood estimation where the variation increases with the true value of β, something which does not happen with the neural network approach. The method of profile maximum pseudo likelihood estimation have a tendency to underestimate β, which gets worse as the true value increases whereas the neural network approach has an overall tendency to slightly overestimate it.

For γ, both methods struggle when R is small as seen by Figure 7, but the neural network approach seems to overall handle it better than profile maximum pseudo likelihood estimation since it can apparently handle the estimation of γ well for smaller values of R than profile maximum pseudo likelihood. For a better comparison of the methods, cases where $R < 0.01$ are excluded from the plot for γ in Figure 8. The neural network approach recovers γ very well if the true value is not above circa 0.8, and in this case it also performs better than profile maximum pseudo likelihood estimation which has a tendency for overestimating γ. If the true value of γ is above circa 0.8, the neural network approach in general underestimates γ whereas profile maximum pseudo likelihood estimation either estimates it to be near 1, as it should, or near 0.

For R, both methods struggle when γ is high, so cases where $\gamma > 0.7$ are excluded from the plot for R in Figure 8. The neural network approach has difficulties recovering R when the true value is small in which case profile maximum pseudo likelihood estimation shows better performance; however,

if the true value is above circa 0.01, the neural network approach recovers R very well and the results are comparable to those obtained with profile maximum pseudo likelihood estimation. Remember though, that the performance of profile maximum pseudo likelihood estimation depends much on how fine a grid of R-values one considers.

In order to asses how the method performs on point patterns with few points, I made a second simulation study where I considered $\beta \in (20, 200)$ for the training and test data. I used 1,000 simulations in the test set, and in this case I only trained the network for 10 epochs because a plot like in Figure 19 revealed problems with overfitting when training the network for longer. Everything else was as above. Figure 9 shows the errors of the estimates obtained with the neural network approach and profile maximum pseudo likelihood estimation for each parameter for the test cases where the number of points was below 200. The plots reveal no clear tendencies in

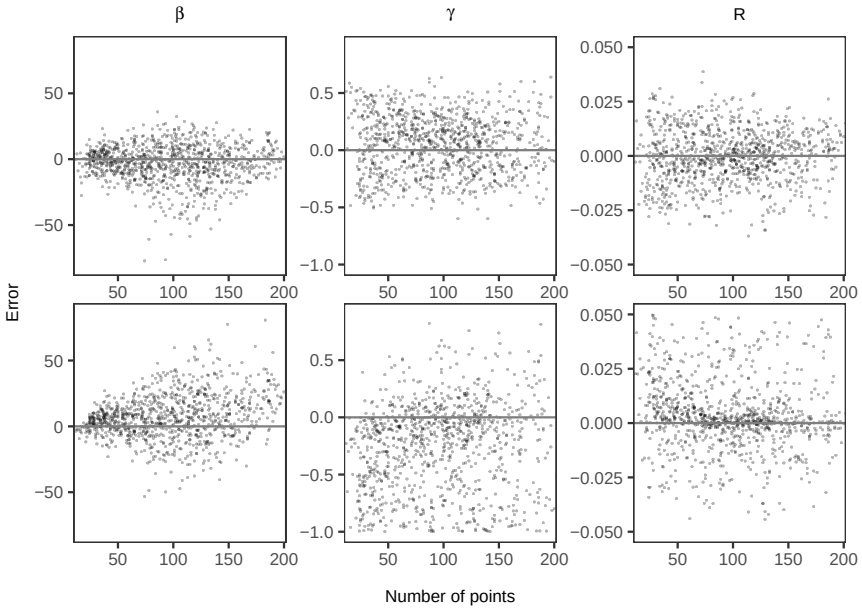

Fig. 9: Estimated minus true value plotted against the number of points in the point pattern. The parameter of the Strauss process is stated at the top of each column. In the top row, estimates were obtained with the neural network approach; in the bottom row, estimates were obtained with profile maximum pseudo likelihood estimation.

the estimates obtained with the neural network approach. In the case of maximum profile pseudo likelihood estimation there are no clear tendencies for β, but for γ there is a general tendency to underestimate and for R there is a tendency to overestimate if there are very few points in the point pattern.

3.3 LGCP-Strauss processes

An LGCP-Strauss process is a model for repulsion at small scale and clustering at a larger scale. It is a combination of an LGCP and a Strauss process, and, defined on W, it has density

$$f(\mathbf{x}) = \mathrm{E}\left[\frac{1}{C_\theta(Y)} \exp\left(\sum_{i=1}^{n} Y(x_i)\right) \gamma^{S_R(x)}\right]$$

for $x = \{x_1, \ldots, x_n\} \subset W$ with respect to the unit rate Poisson process where θ is the parameter vector; $Y = \{Y(u)\}_{u \in W}$ is a Gaussian random field; the expectation is with respect to Y; and $C_\theta(Y)$ is the normalising constant obtained when conditioning on Y. For Y, I used a parametrization as in Section 3.1 with parameters μ, σ^2, and s, so $\theta = (\mu, \sigma^2, s, \gamma, R)$ where $\gamma \in [0, 1]$. If $\gamma = 1$ or $R = 0$, it collapses to an LGCP; if $\sigma^2 = 0$, it collapses to a Strauss process.

I made the training data for the neural network based on 40,000 simulations of an LGCP-Strauss process with parameters sampled in the intervals $\mu \in (4.5, 6)$, $\sigma^2 \in (0, 4)$, $s \in (0.001, 0.1)$, $\gamma \in (0, 1)$, and $R \in (0, 0.05)$. For justification of the number of simulations in the training data see Appendix A.1. I made further 5,000 simulations for a test set for which I estimated the parameters with the neural network approach. The simulation of LGCP-Strauss processes which I used involves Markov chains, and a shared burnin for all simulations was chosen based on trace plots of the number of points and R-close pairs for certain combinations of the parameters believed to require the most iterations. Based on this, I used 200,000 iterations of the Markov chain.

Regarding estimating the parameters of an LGCP-Strauss process Vihrs et al. (2022) noted that the usual methods for estimating parameters in point process models are intractable for this model and thus used ABC. I therefore compare the estimates obtained with the neural network approach to approximate posterior means obtained with an ABC technique. Specifically, I used the method of ABC via random forests as implemented in the R-package `abcrf` version 1.8.1 (Marin et al., 2019). In short, this method trains a regression random forest on a reference table consisting of chosen summary statistics calculated for a number of prior predictions with the aim of predicting posterior expectations, variances and quantiles for a parameter. A regression random forest consists of a number of regression trees trained on bootstrap samples of the training data. In each regression tree the input is subjected to a number of binary decision rules after which a leaf of the tree will be reached. The prediction made by this regression tree is then the mean of the response variables from its training data which are associated to this leaf. The prediction of the random forest is then the mean of the predictions from each individual tree. I refer to Raynal et al. (2019) for more details about ABC via random forests.

In this paper, I am only interested in the approximate posterior means obtained with ABC via random forests, and these are the predictions of the parameters made by trained random forests. The approach is thus quite similar to the neural network approach except that the training data is used to train random forests instead of a neural network. As recommended in Raynal et al. (2019), I trained an independent random forest for each parameter, and the parameters are thus estimated separately instead of simultaneously as with the neural network approach. I used random forests with 500 trees (the default in `abcrf`) and made the check recommended in Raynal et al. (2019) for whether this was sufficient. As a reference table, I used the same training data as for the neural network since both methods are based on machine learning techniques and I wanted to compare their performance when given the exact same information; for the same reason I did not investigate whether the ABC technique would benefit from more simulations in the training data. So in the ABC approach, the independent uniform distributions used to sample the parameters for the training data serve as prior distributions.

The estimates obtained with the neural network approach and the posterior means obtained with ABC via random forests are plotted against the true parameter values in Figures 10–11. Figure 12 shows boxplots of the

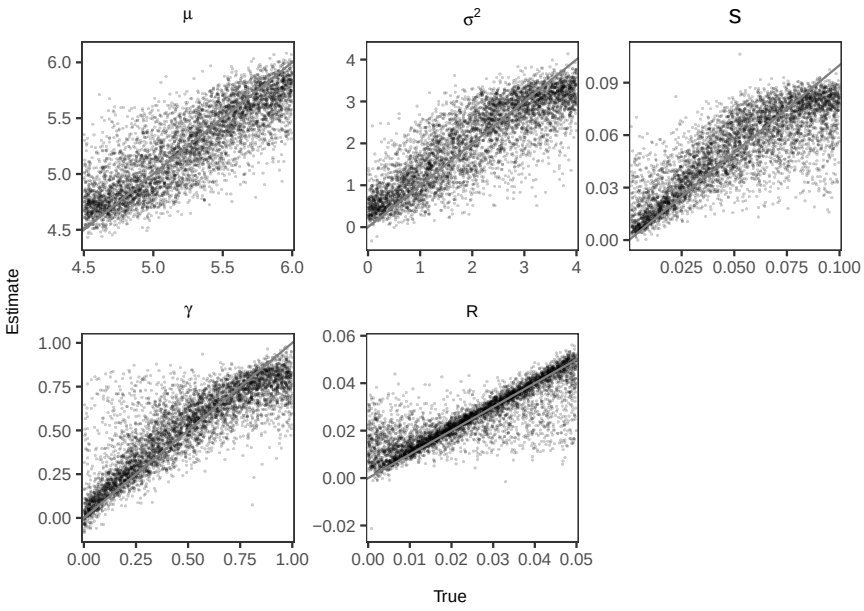

Fig. 10: Estimated parameters obtained with the neural network approach plotted against true parameters in an LGCP-Strauss process. The parameter is stated at the top of each plot. The solid gray line is the identity line.

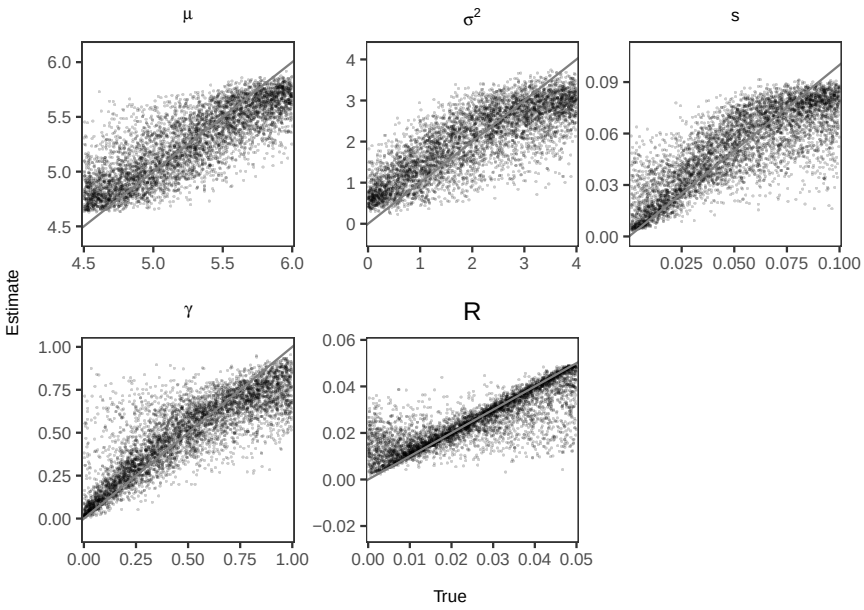

Fig. 11: Approximate posterior means obtained with ABC via random forests plotted against true parameters in an LGCP-Strauss process. The parameter is stated at the top of each plot. The solid gray line is the identity line.

errors where some cases are omitted in the plots for s, γ, and R due to arguments similar to those in Sections 3.1 and 3.2. The results obtained with the two methods are very similar, except that the neural network approach performs slightly better for μ and σ^2 near the endpoints of the considered intervals. It is apparently easiest to estimate γ and R, which are recovered very well except that there is again a tendency to underestimate γ when the true value is high and to overestimate R when the true value is small. The estimate for s is again best when the true value is small. There is a tendency for overestimating σ^2 unless the true value is above circa 3 in which case it is usually underestimated. Vihrs et al. (2022) also found it to be difficult to make inference about the parameters of the Gaussian random field in an LGCP-Strauss process and related it to the fact that it can be difficult to see the effect of changes in the Gaussian random field from a realization of the process because it is obscured by the small scale regularity.

The LGCP-Strauss process models quite complex behaviour in point patterns, so I do not think it is appropriate to fit it to point patterns with few points. I therefore do not consider a second simulation study focusing on point patterns with few points as I did in Sections 3.1–3.2.

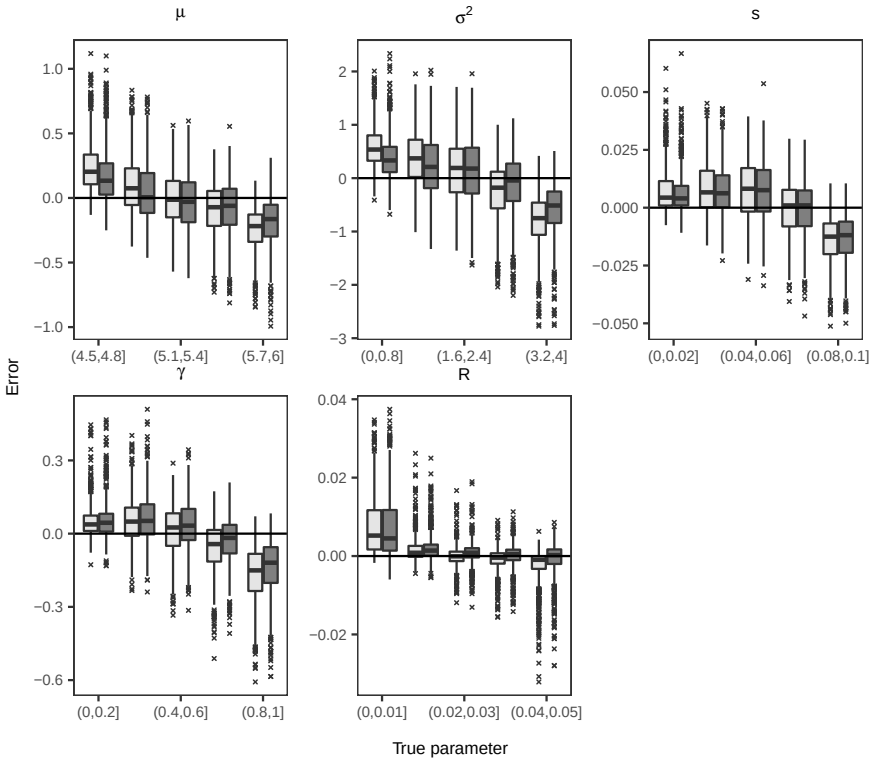

Fig. 12: Boxplot of the errors (estimated minus true value) for the parameter of an LGCP-Strauss process stated at the top of each plot. The estimates were obtained with an ABC technique (light gray) and the neural network approach (dark gray), respectively. For the parameters s, γ, and R, cases where $\sigma^2 > 1$, $R < 0.01$ and $\gamma > 0.7$, respectively, were omitted.

3.4 Some remarks about speed

The purpose of this section is to give an idea of how time consuming the neural network approach is even though this will of course depend heavily on implementation, software, the data a model should be fitted to etc. All the below timings were made using just a single core, but some of the calculations can also be run in parallel.

In the situations in Sections 3.1–3.3, it took about 5.4, 17, and 115 minutes, respectively, to make the training data and 2.6, 1.5, and 9.7 minutes, respectively, to train the neural network. The most time consuming part of the procedure is to make the training data, especially to make the simulations. However, the process of making simulations and calculating summary statistics can easily be parallelized if multiple cores are available. I also recommend to always take the extra time to make a test data set which can be used to asses the performance of the method in a given situation. After the

network was trained, it took about 1 second in all three cases to fit the point process model to the 5,000 simulations in the test data.

With the method of minimum contrast estimation it took about 100.8 minutes to fit LGCP models to the 5,000 simulations in the test data in Section 3.1, and it thus took 1.21 seconds on average to fit one model; with the method of profile maximum pseudo likelihood estimation it took about 154 minutes to fit Strauss process models to the 5,000 simulations in the test data in Section 3.2, and it thus took 1.85 seconds on average to fit one model (these timings of course depend heavily on how many values of the parameter R one considers). When fitting a single model, it is thus much faster to use minimum contrast estimation or profile maximum pseudo likelihood estimation than to use the neural network approach. However, after the neural network has been trained, it can be used to fit the spatial point process model to multiple point patterns as long as they are well represented in the training data, and this can be done very fast. If a model is to be fitted to multiple point patterns and it is possible to train a neural network which is suitable for all these cases, the neural network approach can be faster. With the ABC method in Section 3.3 it took about 93.4 minutes to fit the random forest objects and 34 minutes to make the predictions which include predictions of the posterior means. The ABC procedure also needs the time for making the training data. Thus, the neural network approach was faster in this case, but it is possible to use parallelization in the ABC method in order to speed it up.

4 Data example

The left panel in Figure 13 shows the part of the Allogony data set from the R-package ads version 1.5-5 (Pélissier and Goreaud, 2015) which contains the locations of 256 oak trees which suffer from frost shake in a 125×188 m rectangular region of Allogny in France (this rectangular region is W). The right panel shows $\hat{L}(r) - r$ together with a 95% global envelope for the null hypothesis that data comes from a homogeneous Poisson process. Briefly, a 95% global envelope is a region for which the functional summary statistic calculated from the observed data will fall completely within if and only if the null hypothesis cannot be rejected at level approximately 5%. The envelope was calculated from 2499 simulations (thereby following the recommended number of simulations in Myllymäki et al. (2017)) of a homogeneous Poisson process and based on the extreme rank length (see Myllymäki et al. (2017), Mrkvička et al. (2020), Myllymäki and Mrkvička (2019) for more information about global envelopes and the R-package GET (version 0.2-4), which I used to calculate them). The plots indicate that the point pattern exhibit repulsive behaviour at a small scale and some clustering at a larger scale. As an example, I now show how the neural network approach can be used to fit an

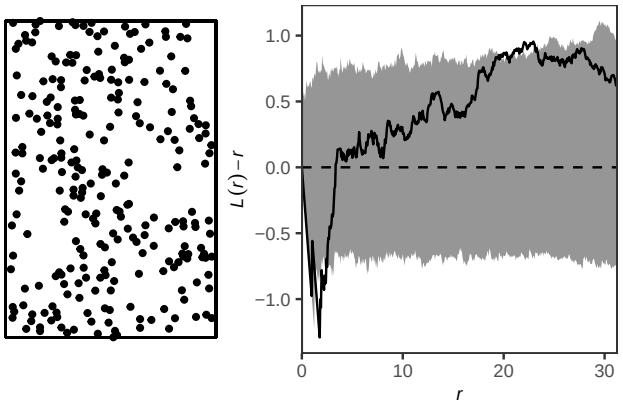

Fig. 13: Left: Point pattern of the locations of 256 oak trees which suffer from frost shake in a 125×188 m rectangular region of Allogny in France. Right: $\hat{L}(r) - r$ together with a 95% global envelope calculated from 2499 simulations of a homogeneous Poisson process.

LGCP-Strauss process to this point pattern (an LGCP-Strauss process model was previously fitted to this data in Vihrs et al. (2022)).

Regarding the ranges of parameters to use in the training data for the neural network, I used $\gamma \in (0, 0.7)$ and $R \in (1, 5)$ since Figure 13 shows clear evidence of repulsion in the observed point pattern and Vihrs et al. (2022) noted that the interaction radius R is often near the r-value which gives the smallest value of $\hat{L}(r) - r$. For the parameters of the Gaussian random field, I decided to use $\mu \in (-5.6, -3)$, $\sigma^2 \in (0, 2)$, and $s \in (0.001, 15)$ after having looked at some simulations of LGCP-Strauss processes. I then used 40,000 simulations on W where the parameters were sampled uniformly on the above intervals to construct the training data for the neural network approach. Figure 14 shows a histogram of the number of points in the point patterns in the training data and a 95% global envelope calculated from the 40,000 estimates of $L(r) - r$ in the training data. The same summaries obtained from the oak point pattern are also shown in the plots, where we see that both the observed number of points and the behaviour of $\hat{L}(r) - r$ are well represented in the training data, which is crucial in order to get reliable estimates with the neural network approach. A check like this may both reveal if the intervals for the parameters have been chosen inappropriately or if the considered class of model is ill-suited for fitting the observed point pattern.

I also made 5,000 simulations for a test data set, and Figure 15 shows the estimated parameters for these plotted against the true values. This shows that in this situation μ, γ, and R are recovered well whereas there is more uncertainty in the estimates of σ^2 and s.

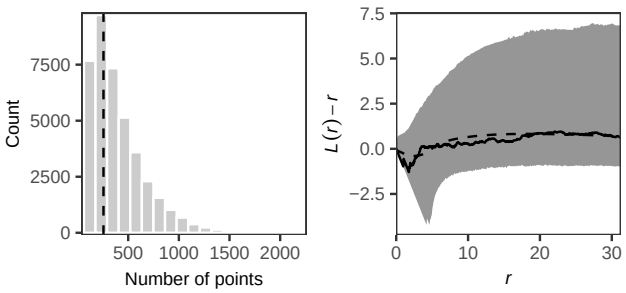

Fig. 14: Left: Histogram of the number of points in the simulations in the training data where the vertical dashed line indicates the number of points in the observed point pattern of oak trees. Right: A 95% global envelope calculated from the 40,000 estimates of $L(r) - r$ in the training data (gray area), the mean (dashed curve) and $\hat{L}(r) - r$ obtained from the observed point pattern of oak trees (solid curve).

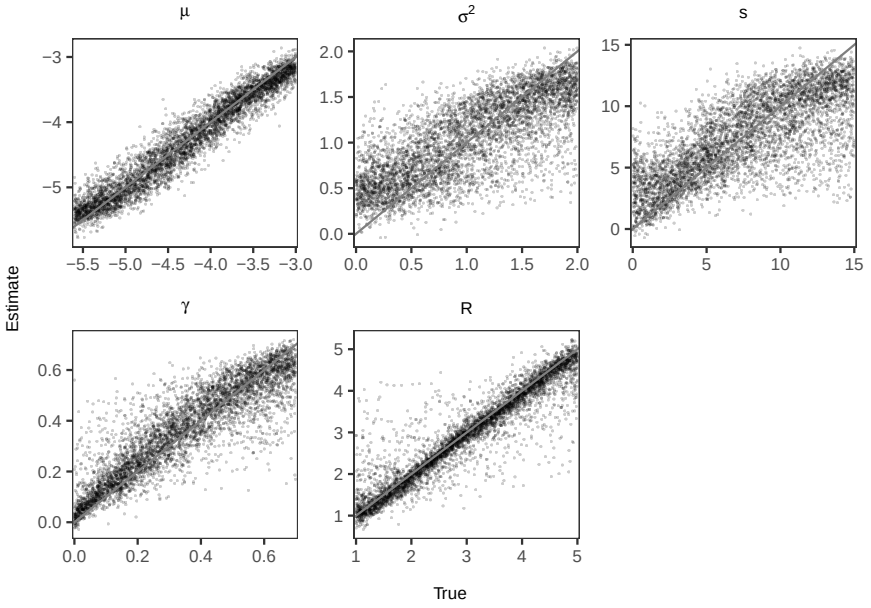

Fig. 15: Estimated parameters of the test data obtained with the neural network approach aimed at fitting an LGCP-Strauss process to the oak point pattern plotted against true parameters. The parameter is stated at the top of each plot.

When using the trained neural network to estimate the parameters for the point pattern of oak trees, I got the estimates $\hat{\mu} = -4.54, \hat{\sigma}^2 = 0.32, \hat{s} = 10.93, \hat{\gamma} = 0.21$, and $\hat{R} = 1.91$. The most popular way to validate a fitted point process model is to consider global envelopes and corresponding tests

calculated for some functional summary statistic. I did not want to use the L-function for this global envelope and test since it plays a major part in the estimating procedure. I therefore used the J-function given in (3) instead. I used the non-parametric estimate $\widehat{J}(r) = (1 - \widehat{G})/(1 - \widehat{F})$ where $\widehat{G}$ and $\widehat{F}$ are the so-called Kaplan-Meier estimators of G and F, which account for edge effects, see Baddeley et al. (2015, Section 8.11.4) for how these estimators are given. Regarding the considered range of r-values for $\widehat{J}(r)$, the function `Jest` from `spatstat` which is used to estimate J gives a recommendation, which I have followed.

Figure 16 shows a 95% global envelope and the p-value of the corresponding global envelope test based on the J-function and calculated from 2499 simulations under the fitted model. This indicates that the fitted model describes the point pattern of oak trees very well.

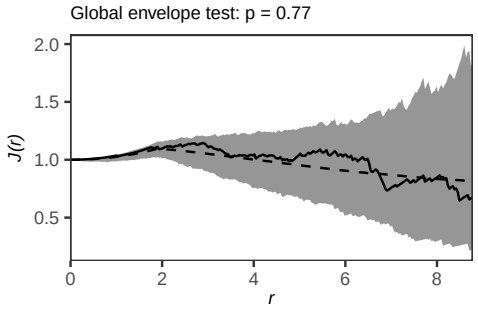

Fig. 16: A 95% global envelope based on the J-function calculated from 2499 simulations under the LGCP-Strauss process model fitted to the oak point pattern (gray area), the mean obtained from the simulations (dashed curve), and the estimate calculated from the observed point pattern (solid curve). The p-value of the corresponding global envelope test is stated at the top.

5 Discussion and future research

I have presented a method which is generally applicable to estimate parameters in all spatial point process models which it is possible to simulate from. The method recovers parameters well compared to common estimating techniques since it gives either better or similar results. The advantages of the method are that the only necessary information about the model is a tractable simulation procedure and that all unknown parameters can be estimated simultaneously. The method is more time consuming than minimum contrast estimation and profile maximum pseudo likelihood estimation when it comes to fit a single model. However, the most time consuming part of the method is to make training data and to train the neural network, so if it is possible to train a neural network which can be reused to fit a model to many point

patterns, the neural network approach can be faster than using minimum contrast estimation or maximum pseudo likelihood estimation on each point pattern. Compared to ABC, the neural network approach is also potentially faster.

Future research may include how to use the neural network approach to estimate parameters in inhomogeneous point process models which include covariate information. It could also be interesting to explore the possibility to pre-train large neural networks which could be applicable to a wide range of point patterns which are often encountered in practice thereby obtaining a very fast estimation procedure for such point patterns.

Acknowledgements

The research of the author was supported by The Danish Council for Independent Research | Natural Sciences, grant DFF – 7014-00074 'Statistics for point processes in space and beyond'. I would also like to thank Jesper Møller for his helpful comments regarding this paper.

A Details for the neural network approach

A.1 Training data

It is possible to get as much training data as desired since it is merely a matter of making more simulations, but simulation procedures may be time consuming. Figure 17 shows the mean squared errors obtained with the neural network approach for test sets with 5,000 simulations in the situations of the first simulation studies in Sections 3.1–3.3 plotted against the number of simulations in the training data. The necessary number of simulations

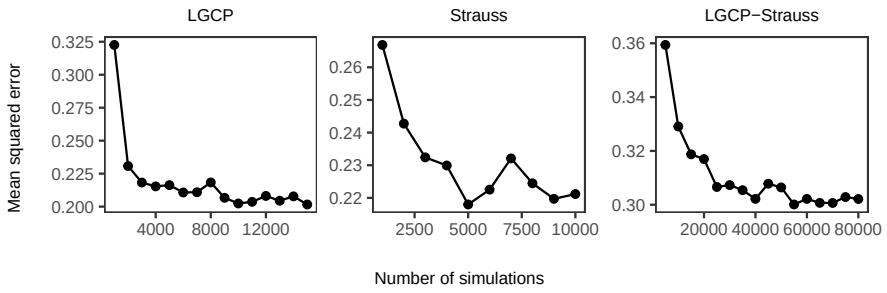

Fig. 17: Mean squared errors calculated for test sets with 5,000 simulations for the situations of the first simulation studies in Sections 3.1–3.3 as stated at the top plotted against the number of simulations in the training data.

depends on how complicated the model is. Based on Figure 17 I used 10,000 simulations in the case of an LGCP in Section 3.1; 5,000 simulations in the case of a Strauss process in Section 3.2; and 40,000 simulations in the case of an LGCP-Strauss process in Section 3.3.

Figure 18 shows histograms of the number of points in the training data sets used for the first simulation studies in Sections 3.1–3.3.

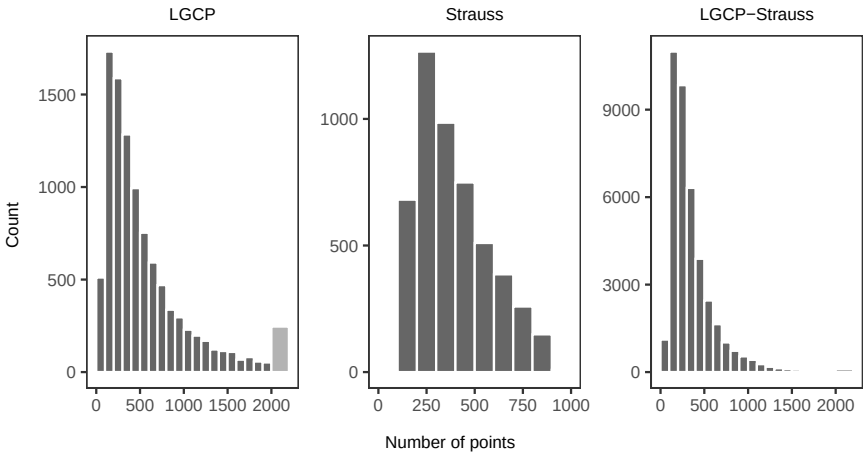

Fig. 18: Histogram of the number of points in the point patterns in the training data used in the first simulation studies in Sections 3.1–3.3 as stated at the top of each plot. The light grey column indicates the count of point patterns where the number of points is above 2000, the highest value being 6116 and 6497 in the cases of the LGCP and LGCP-Strauss processes, respectively.

A.2 Network training

The unknown parameters of the neural network should be learned based on the training data. This is done by minimizing a loss function with some optimization technique. I used the mean squared error as loss function and the Adam optimizer (Kingma and Ba, 2014) for optimization. During training, the training data was send through the network in smaller batches of size 100. An iteration over the entire training data is referred to as an epoch. During training, I also monitored the mean squared error of a test set again constructed from simulations of the point process model as described in Section 2.2. The mean squared error of the test set was among other things used to decide on the number of epochs where the choice in general fell on 20 epochs based on Figure 19, which also revealed that with the choices I made, there is no apparent problem with overfitting.

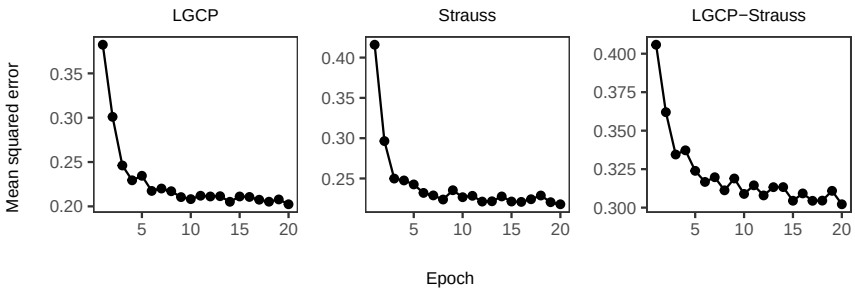

Fig. 19: Mean squared error calculated from test sets with 5,000 simulations plotted against number of epochs for the situations of the first simulation studies in Sections 3.1–3.3 as stated at the top of each plot. The number of simulations in the training data were as in Sections 3.1–3.3.

References

Allaire, J. J. and Chollet, F. (2020). *keras: R Interface to 'Keras'*. R package version 2.3.0.0.

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

Baddeley, A. and Turner, R. (2000). Practical maximum pseudolikelihood for spatial point patterns. *Australian and New Zeland Journal of Statistics*, 42:283–322.

Beaumont, M. A. (2010). Approximate Bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics*, 41:379–406.

Besag, J. E. (1975). Statistical analysis of non-lattice data. *The Statistician*, 24:179–195.

Chollet, F. (2018). *Deep Learning with R*. Manning Publications, Shelter Island, NY.

Diggle, P. J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

Diggle, P. J. and Gratton, R. J. (1984). Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society, series B*, 46:193–2012.

Gabrielli, L., Tomassetti, S., Squartini, S., and Zinato, C. (2017). Introducing deep machine learning for parameter estimation in physical modelling. In *Proceedings of the 20th International Conference on Digital Audio Effects*.

Guan, Y. (2006). A composite likelihood approach in fitting spatial point process models. *Journal of the American Statistical Association*, 101:1502–1512.

Jensen, J. L. and Møller, J. (1991). Pseudolikelihood for exponential family models of spatial point processes. *Annals of Applied Probability*, 3:445–461.

Kelly, F. P. and Ripley, B. D. (1976). A note on Strauss's model for clustering. *Biometrika*, 63:357–360.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Marin, J.-M., Raynal, L., Pudlo, P., Robert, C. P., and Estoup, A. (2019). *abcrf: Approximate Bayesian Computation via Random Forests*. R package version 1.8.1.

Møller, J. and Waagepetersen, R. P. (2017). Some recent developments in statistics for spatial point patterns. *Annual Review of Statistics and Its Application*, 4:317–342.

Mrkvička, T., Myllymäki, M., Jilik, M., and Hahn, U. (2020). A one-way ANOVA test for functional data with graphical interpretation. *Kybernetika*, 56:432–458.

Myllymäki, M. and Mrkvička, T. (2019). GET: Global envelopes in R. arXiv preprint arXiv:1911.06583.

Myllymäki, M., Mrkvička, T., Grabarnik, P., Seijo, H., and Hahn, U. (2017). Global envelope tests for spatial processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79:381–404.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25:451–482.

Møller, J. and Waagepetersen, R. P. (2004). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman and Hall/CRC, Boca Raton.

Ohser, J. (1983). On estimators for the reduced second moment measure of point processes. *Mathematische Operationsforschung und Statistik, series Statistics*, 14:63–71.

Pélissier, R. and Goreaud, F. (2015). ads package for R: A fast unbiased implementation of the *K*-function family for studying spatial point patterns in irregular-shaped sampling windows. *Journal of Statistical Software*, 63:1–18.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Raynal, L., Marin, J.-M., Pudlo, P., Ribatet, M., Robert, C. P., and Estoup, A. (2019). ABC random forests for bayesian parameter inference. *Bioinformatics*, 35(10):1720–1728.

Ripley, B. D. (1988). *Statistical Inference for Spatial Processes*. Cambridge University Press, Cambridge.

Strauss, D. J. (1975). A model for clustering. *Biometrika*, 62:467–475.

Vihrs, N., Møller, J., and Gelfand, A. E. (2022). Approximate Bayesian inference for a spatial point process model exhibiting regularity and random aggregation. *Scandinavian Journal of Statistics*, 49:185–210.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Paper F

Determinantal shot noise Cox processes

Jesper Møller and Ninna Vihrs

The paper has been submitted to
Stat

The layout has been revised.

Abstract

We present a new class of cluster point process models, which we call determinantal shot noise Cox processes (DSNCP), with repulsion between cluster centres. They are the special case of generalized shot noise Cox processes where the cluster centres are determinantal point processes. We establish various moment results and describe how these can be used to easily estimate unknown parameters in two particularly tractable cases, namely when the offspring density is isotropic Gaussian and the kernel of the determinantal point process of cluster centres is Gaussian or like in a scaled Ginibre point process. Through a simulation study and the analysis of a real point pattern data set we see that when modelling clustered point patterns, a much lower intensity of cluster centres may be needed in DSNCP models as compared to shot noise Cox processes.

1 Introduction

This paper studies a cluster point process model defined as follows. Let Y be a simple locally finite point process defined on the d-dimensional Euclidean space $\mathbb{R}^d$; we can view Y as a random subset of $\mathbb{R}^d$ (for background material on spatial point processes, see Møller and Waagepetersen (2004)). Assume Y is stationary, that is, its distribution is invariant under translations in $\mathbb{R}^d$. Conditioned on Y, let X be a Poisson process on $\mathbb{R}^d$ with intensity function

$$\rho(x \mid Y) = \gamma \sum_{y \in Y} k_\alpha(x - y), \quad x \in \mathbb{R}^d, \tag{1}$$

where $\gamma > 0$ and α are parameters and k_α is a probability density function (pdf) on $\mathbb{R}^d$; in our specific models α will play the role of a band width (a positive scale parameter). We can identify X by a cluster process $\cup_{y \in Y} X_y$ where conditioned on Y, the clusters X_y are independent finite Poisson processes on $\mathbb{R}^d$ and X_y has intensity function $\rho_y(x) = \gamma k_\alpha(x - y)$ (depending on the 'offspring' density k_α relative to the cluster center y).

In the special case where Y is a stationary Poisson process, X is a shot noise Cox process (SNCP), see Møller (2003). Then there may be a large amount of overlap between the clusters unless the intensity of Y is small as compared to the band width α. In this paper, we will instead be interested in repulsive point process models for Y. This may be an advantage since the repulsiveness of Y implies less overlap of clusters. Thereby it may be easier to apply statistical methods for cluster detection, and when modelling clustered point pattern data sets a much lower intensity of Y may be needed as compared to the case of a SNCP. The idea of using a repulsive point process Y is not new, where Van Lieshout and Baddeley (2002) suggested to use a Markov point process. However, we are in particular interested in the case

where Y is a stationary determinantal point process (DPP) in which case we call X a *determinantal shot noise Cox process (DSNCP)*. Briefly, a DPP is a model with repulsion at all scales, cf. Lavancier et al. (2015), Møller and O'Reilly (2021), and the references therein. There are several advantages of using a DPP for Y: In contrast to a Markov point process, there is no need of MCMC when simulating a DPP, and as we shall see Y and hence X possess nice moment results, which can be used for estimation.

The cluster point process X given by (1) is a special case of a stationary generalized shot noise Cox process (GSNCP), see Møller and Torrisi (2005), and it may be extended as follows. Suppose X conditioned on both Y and positive random variables $\{\Gamma_y\}_{y\in\mathbb{R}^d}$ and $\{A_y\}_{y\in\mathbb{R}^d}$ is a Poisson process with intensity function

$$\rho(x \mid Y, \{\Gamma_y\}_{y\in\mathbb{R}^d}, \{A_y\}_{y\in\mathbb{R}^d}) = \sum_{y\in Y} \Gamma_y k_{A_y}(x-y), \quad x \in \mathbb{R}^d,$$

where k_{A_y} is a pdf on $\mathbb{R}^d$. In addition, assume that $\{\Gamma_y\}_{y\in\mathbb{R}^d}$, $\{A_y\}_{y\in\mathbb{R}^d}$, and Y are mutually independent, the Γ_y are independent identically distributed with mean γ and has finite variance, and the A_y are independent identically distributed. Then X is still a stationary GSNCP and if also Y is a stationary DPP we may call X a DGSNCP. In fact, all results and statistical methods used in this paper will apply for the DGSNCP when k_α is replaced by $\mathrm{E}k_{A_y}$ in all expressions to follow. The DGSNCP may most naturally be treated in a MCMC Bayesian setting using a similar approach as in Beraha et al. (2022) and the references therein.

In the present paper, we study and exploit for statistical inference the nice moment properties for various DSNCP models as follows. In Section 2 we describe further what it means that Y is a DPP and present two specific cases of DSNCP models where we let k_α be an isotropic Gaussian density as in the Thomas process (Thomas, 1949), which is the most popular example of a SNCP. Section 3 considers general results for so-called pair correlation and K-functions for first the GSNCP model and second the DSNCP model. Section 4 discusses how the results in Section 3 may be used when fitting a parametric DSNCP model in a frequentist setting, and we illustrate this on a real data example. In Section 5 we investigate the ability to distinguish between DSNCPs and Thomas processes through a simulation study. Finally, in Section 6 we summarize our results.

All statistical analyses were made with `R` (R Core Team, 2019), and all plots were made using the package `ggplot2` (Wickham, 2016).

2 Determinantal shot noise Cox process models

In this section we consider the DSNCP model for X and suggest some specific models. In brief, the DPP Y is specified by a so-called kernel which is usually assumed to be a complex covariance function $c(u, v)$ defined for all $u, v \in \mathbb{R}^d$; for details, see Appendix A. We assume Y is a stationary DPP with intensity $\rho_Y > 0$, meaning two things: First, if $A \subset \mathbb{R}^d$ is a bounded Borel set, $Y(A)$ denotes the cardinality of $Y \cap A$, and $|A| = \int_A \mathrm{d}u$ is the Lebesgue measure of A, then $\mathrm{E}Y(A) = \rho_Y |A| < \infty$. Second, $|c(u, v)| = |c(u - v, 0)|$ for all $u, v \in \mathbb{R}^d$ where $|s|$ denotes the modulus of a complex number s. We denote the corresponding complex correlation function by $r(u, v) = c(u, v)/\rho_Y$ and assume it depends on a correlation/scale parameter $\beta > 0$ so that $r = r_\beta$ with

$$r_\beta(u, v) = r_1(u/\beta, v/\beta). \tag{2}$$

The correlation parameter β cannot vary independently of ρ_Y since there is a trade-off between intensity and repulsiveness in order to secure that a DPP model is well defined (Lavancier et al., 2015). For instance, for many DPP models r_β is real, continuous, and stationary, that is, $r_\beta(u, v) = r_{\beta,\mathrm{st}}(u - v)$ where $r_{\beta,\mathrm{st}} : \mathbb{R}^d \to [-1, 1]$ is a continuous, symmetric, and positive semi-definite function with $r_{\beta,\mathrm{st}}(0) = 1$. Then, if $r_{\beta,\mathrm{st}}$ is square integrable and has Fourier transform $\varphi_\beta(u) = \int r_{\beta,\mathrm{st}}(v) \cos(2\pi u \cdot v)\, \mathrm{d}v$ where $\cdot$ is the usual inner product, the DPP is only well-defined for $\rho_Y \sup \varphi_\beta \leq 1$, cf. Lavancier et al. (2015). In case of (2), this existence condition of the DPP means that $0 < \beta \leq 1/(\rho_Y^{1/d} \sup \varphi_1)$, where for a fixed value of ρ_Y, most repulsiveness is obtained when $\beta = 1/(\rho_Y^{1/d} \sup \varphi_1)$.

Consider the special case where Y is a jinc-like DPP, that is, $d = 2$ and $r(u, v) = J_1(2\sqrt{\pi}\|u - v\|)/(\sqrt{\pi}\|u - v\|)$ where J_1 is the first order Bessel function of the first kind and $\|\cdot\|$ denotes usual distance. So, the distribution of Y depends only on the intensity, and Y is a most repulsive DPP in the sense of Lavancier et al. (2015), see also Biscio and Lavancier (2016) and Møller and O'Reilly (2021). Christoffersen et al. (2021) used this special case of a DSNCP model in a situation where a realization of X but not Y was observed within a bounded region. They estimated ρ_Y with a minimum contrast procedure based on the pair correlation function (pcf) given by (5) in Section 3, where the pcf had to be approximated by numerical methods. Instead we consider more tractable cases, as we shall see in Section 3.

Note that X is stationary with intensity $\rho_X = \gamma\rho_Y$. We will consider two specific DSNCP models of X where we let k_α be the pdf of $N_d(0, \alpha^2 I)$, the zero-mean isotropic d-dimensional normal distribution, and Y is given by one of the following two DPPs.

1. If $r_\beta(u, v) = \exp\left(-\|(u - v)/\beta\|^2\right)$ is the Gaussian correlation function, then Y is a *Gaussian DPP* (Lavancier et al., 2015).

2. If $d = 2$ and we identify $\mathbb{R}^2$ with the complex plane $\mathbb{C}$, and if $r_\beta(u, v) = \exp((u\overline{v} - |u|^2/2 - |v|^2/2)/\beta^2)$ where $\overline{v}$ and $|v|$ denote the complex conjugate and the modulus of the complex number v, then Y is a *scaled Ginibre point process* (Deng et al. (2014), Miyoshi and Shirai (2016); Y is the standard Ginibre point process (Ginibre, 1965) if $\rho_Y = 1/\pi$ and $\beta = 1$).

Both these DPP models are well defined if and only if $0 < \beta \leq 1/(\rho_Y^{1/d}\sqrt{\pi})$ (Lavancier et al., 2015). For a fixed value of ρ_Y, Y becomes in both cases less and less repulsive as β decreases from $1/(\rho_Y^{1/d}\sqrt{\pi})$ to 0, where in the limit Y is a stationary Poisson process and X is a Thomas process. Therefore, we call X a *Gaussian-DPP-Thomas process* in the first case and a *Ginibre-DPP-Thomas process* in the second case. In both cases, Y and hence also X are stationary and isotropic, although r_β is only stationary when it is the Gaussian correlation function (Appendix B.1 verifies that the distribution of a scaled Ginibre point process Y is invariant under isometries).

The two first columns of plots in Figure 1 show simulated realizations of a Ginibre-DPP-Thomas process and a Gaussian-DPP-Thomas process (see Appendix C for some details about simulation procedures) within a 20×20 square region, where $\alpha = 1$, $\rho_X = 1$ (so we expect to see about 400 points in each simulated point pattern), and in the three rows of plots we have $\beta = 2, 3, 4$ (from bottom to top). In each case, $\rho_Y = 1/(\pi\beta^2)$ is as large as possible. For comparison, the third column of plots in Figure 1 shows simulations of Thomas processes with the same values of (ρ_X, ρ_Y) as for the two first columns of plots. So, in each row the three processes have the same expected number of clusters, the same expected cluster sizes, and the same offspring density. For all processes, as β increases (that is, ρ_Y decreases and γ increases) we see that the point patterns look more clustered, since we get less and less cluster centres but larger and larger clusters. We also see that the eye detects less diffuse clusters in the DPP-Thomas processes compared to the Thomas processes, which is in agreement with the fact that cluster centres are repulsive in DPP-Thomas processes whereas they are completely random in Thomas processes. From Figure 1 it can be difficult to make any conclusions about the differences between Gaussian- and Ginibre-DPP-Thomas processes, but we will make further comparisons between these in Sections 3–4.

3 Pair correlation and K-functions

3.1 The general setting of stationary GSNCPs

Consider again the general setting in Section 1 of a stationary GSNCP. Henceforth we assume the stationary point process Y has pair correlation function

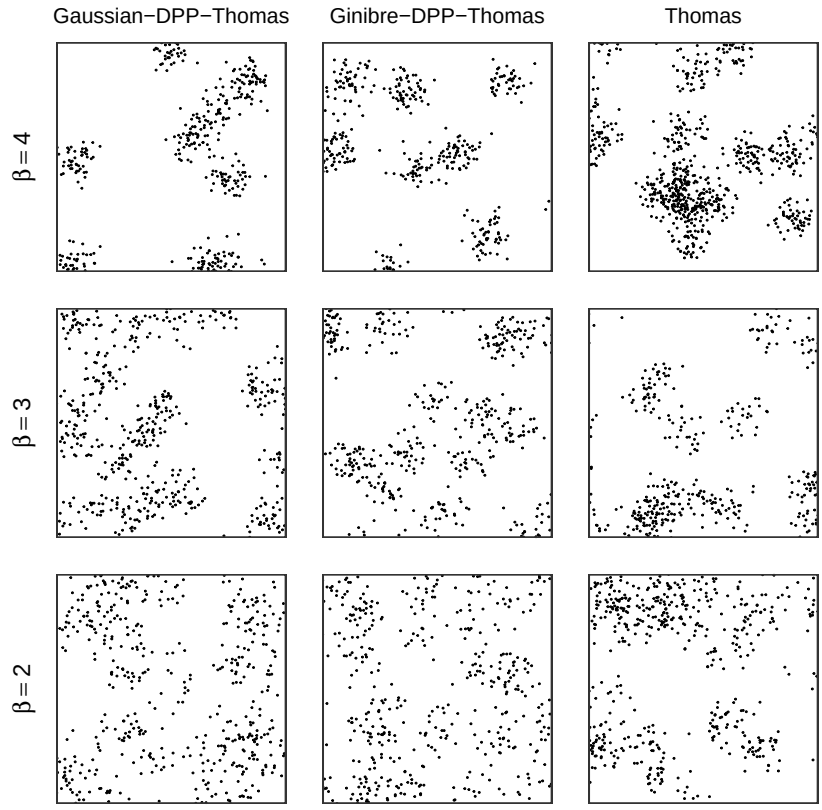

Fig. 1: Simulations of Gaussian-DPP-Thomas processes, Ginibre-DPP-Thomas processes, and Thomas processes (as stated at the top of each column) within a square with side lengths 20 when $\alpha = 1$, $\rho_X = 1$, $\beta = 2, 3, 4$ (stated to the left of each row), and $\rho_Y = 1/(\pi\beta^2)$ in all processes. Note that β is not a parameter of the Thomas process and is thus only used to calculate ρ_Y in this case.

(pcf) g_Y. This means that if $A, B \subset \mathbb{R}^d$ are disjoint bounded Borel sets, then

$$\mathrm{E}[Y(A)Y(B)] = \rho_Y^2 \int_A \int_B g_Y(u,v)\,\mathrm{d}u\,\mathrm{d}v < \infty.$$

By stationarity, $g_Y(u,v) = g_{Y,\mathrm{st}}(u-v)$ depends only on the lag $u-v$ almost surely (with respect to Lebesgue measure) and for ease of presentation we can assume this is the case for all $u, v \in \mathbb{R}^d$.

The stationary GSNCP X given by (1) has intensity $\rho_X = \gamma\rho_Y$ and a stationary pcf $g_X(u,v) = g_{X,\mathrm{st}}(x)$ where $x = u - v$ and

$$g_{X,\mathrm{st}}(x) = k_\alpha * \tilde{k}_\alpha * g_{Y,\mathrm{st}}(x) + k_\alpha * \tilde{k}_\alpha(x)/\rho_Y, \quad x \in \mathbb{R}^d, \tag{3}$$

where $*$ denotes convolution and $\tilde{k}_\alpha(x) = k_\alpha(-x)$ is the reflection of k_α, cf. Møller and Torrisi (2005). Thus, g_X decreases as ρ_Y increases; this makes good sense since ρ_Y is the intensity of clusters and the first term in (3) corresponds to pairs of points from different clusters whilst the second term is due to pairs of points within a cluster. Furthermore, from (3) we obtain Ripley's K-function (Ripley, 1976, 1977)

$$K(r) = \int_{\|x\| \le r} g_{X,\mathrm{st}}(x)\,\mathrm{d}x, \quad r > 0,$$

which will be used in Section 4 for parameter estimation.

In the special case where Y is a stationary Poisson process (i.e., X is a SNCP), we have $g_Y = 1$ and (3) reduces to $g_{X,\mathrm{st}} = 1 + k_\alpha * \tilde{k}_\alpha/\rho_Y$. Thus $g_X \ge 1$ and $g_X \ne 1$ which is usually interpreted as X being a model for clustering. This is of course also the situation if $g_Y \ge 1$. However, such models for clustering may cause a large amount of overlap between the clusters unless ρ_Y is small as compared to the band width α.

3.2 The special setting of DSNCPs

When Y is a DPP and we let $R_\beta(y) = |r_\beta(y,0)|^2$, we have

$$g_{Y,\mathrm{st}}(y) = 1 - R_\beta(y), \quad y \in \mathbb{R}^d, \tag{4}$$

cf. Lavancier et al. (2015). Thus $g_Y \le 1$, which reflects that a DPP is repulsive. From (3) and (4) we get

$$g_{X,\mathrm{st}}(x) = 1 - k_\alpha * \tilde{k}_\alpha * R_\beta(x) + k_\alpha * \tilde{k}_\alpha(x)/\rho_Y, \quad x \in \mathbb{R}^d. \tag{5}$$

This is in accordance to intuition: As R_β increases, meaning that g_Y decreases and hence that Y becomes more repulsive, it follows from (5) that g_X decreases; and as the band width α tends to 0, we see that $g_{X,\mathrm{st}}(x)$ tends to $g_{Y,\mathrm{st}}(x)$ for every $x \in \mathbb{R}^d$. Below, we let k_α be the pdf of $N_d(0, \alpha^2 I)$ and consider the pcfs and K-functions in the special cases of Gaussian/Ginibre-DPP-Thomas processes.

Let X be a Gaussian-DPP-Thomas process. Then Y is a Gaussian DPP and

$$R_\beta(y) = \exp\left(-2\|y/\beta\|^2\right), \quad y \in \mathbb{R}^d. \tag{6}$$

Thus we obtain from (5) that $g_{X,\mathrm{st}}(x) = g_{X,\mathrm{iso}}(\|x\|)$ is isotropic with

$$g_{X,\mathrm{iso}}(r) = 1 + \frac{\exp\left(-\frac{r^2}{4\alpha^2}\right)}{(4\pi\alpha^2)^{d/2}\rho_Y} - \frac{\left(\beta^2/2\right)^{d/2}\exp\left(-\frac{r^2}{4\alpha^2+\beta^2/2}\right)}{(4\alpha^2+\beta^2/2)^{d/2}}, \quad r > 0. \tag{7}$$

We have

$$g_{X,\mathrm{iso}}(r) \gtreqless 1 \;\Leftrightarrow\; r^2 \gtreqless \frac{\ln\left(\rho_Y\left(\frac{2\pi\alpha^2\beta^2}{4\alpha^2+\beta^2/2}\right)^{d/2}\right)}{\frac{1}{4\alpha^2+\beta^2/2}-\frac{1}{4\alpha^2}}, \tag{8}$$

where

$$\frac{\ln\left(\rho_Y\left(\frac{2\pi\alpha^2\beta^2}{4\alpha^2+\beta^2/2}\right)^{d/2}\right)}{\frac{1}{4\alpha^2+\beta^2/2}-\frac{1}{4\alpha^2}} > 0$$

since $\rho_Y \leq \left(\pi\beta^2\right)^{-d/2}$. Furthermore, if ω_d denotes the volume of the d-dimensional unit ball and $F_{d/2}$ is the CDF of a gamma distribution with shape parameter $d/2$ and scale parameter 1, we obtain

$$K(r) = \omega_d r^d + \frac{1}{\rho_Y}F_{d/2}\left(\frac{r^2}{4\alpha^2}\right) - \left(\pi\beta^2/2\right)^{d/2}F_{d/2}\left(\frac{r^2}{4\alpha^2+\beta^2/2}\right), \quad r>0. \tag{9}$$

Let X be a Ginibre-DPP-Thomas process. Then Y is a scaled Ginibre point process which has some similarity to the Gaussian-DPP, since β has the same range in the two processes and

$$R_\beta(y) = \exp(-|y/\beta|^2), \quad y \in \mathbb{C}, \tag{10}$$

in the case of a scaled Ginibre point process. It thus follows from (4), (6), and (10) that the pcfs of the scaled Ginibre point process and the Gaussian-DPP are of the same form, but β^2 in the scaled Ginibre point process corresponds to $\beta^2/2$ in the Gaussian-DPP. This shows that the scaled Ginibre point process is more repulsive than the Gaussian-DPP when using the same parameters ρ_Y and β, and therefore it will be possible to obtain a larger repulsion between the clusters in a Ginibre-DPP-Thomas process than in a Gaussian-DPP-Thomas process. In fact, if $\beta = 1/\sqrt{\pi\rho_Y}$ when Y is a scaled Ginibre point process, then Y is a most repulsive DPP in the sense of Lavancier et al. (2015). Because β^2 in the scaled Ginibre point process corresponds to $\beta^2/2$ in the Gaussian-DPP, (7)–(9) give that $g_{X,\mathrm{st}}(x) = g_{X,\mathrm{iso}}(\|x\|)$ is isotropic with

$$g_{X,\mathrm{iso}}(r) = 1 + \frac{\exp\left(-r^2/(4\alpha^2)\right)}{4\pi\alpha^2\rho_Y} - \frac{\beta^2\exp\left(-r^2/\left(4\alpha^2+\beta^2\right)\right)}{4\alpha^2+\beta^2},$$

$$g_{X,\mathrm{iso}}(r) \gtreqless 1 \;\Leftrightarrow\; r^2 \gtreqless \frac{\ln\left(\rho_Y\frac{4\pi\alpha^2\beta^2}{4\alpha^2+\beta^2}\right)}{\frac{1}{4\alpha^2+\beta^2}-\frac{1}{4\alpha^2}},$$

and

$$K(r) = \pi r^2 + \frac{1}{\rho_Y}\left(1-\exp\left(-\frac{r^2}{4\alpha^2}\right)\right) - \pi\beta^2\left(1-\exp\left(-\frac{r^2}{4\alpha+\beta^2}\right)\right).$$

where $r > 0$.

Figure 2 shows plots of $g_{X,\mathrm{iso}}(r)$ and $K(r) - \pi r^2$ for Gaussian- and Ginibre-DPP-Thomas processes for different values of β when $\rho_Y = 1/(\pi\beta^2)$ corresponds to the most repulsive case and without loss of generality we let $\alpha = 1$. For comparison the plots also include the case of a Thomas process

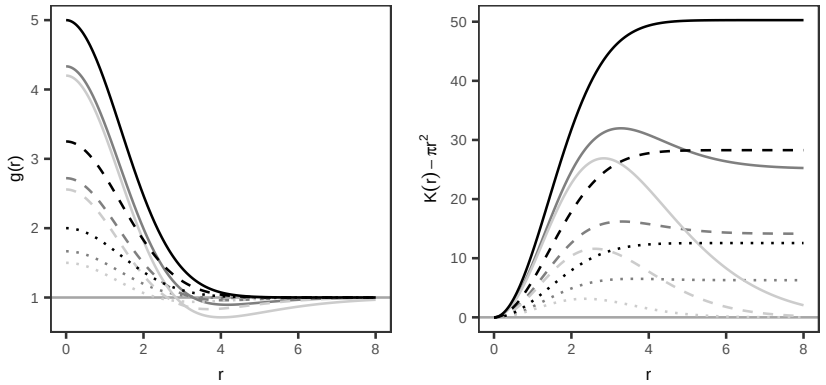

Fig. 2: Plots of $g_{X,\mathrm{iso}}(r)$ (left) and $K(r) - \pi r^2$ (right) for the Thomas process (black curves), Gaussian-DPP-Thomas process (dark grey curves), and Ginibre-DPP-Thomas process (light grey curves) when $d = 2$, $\alpha = 1$, $\beta = 2, 3, 4$ (dotted, dashed, and solid curves, respectively), and $\rho_Y = 1/(\beta^2\pi)$. The plots also shows the constant lines corresponding to the cases of a stationary Poisson process.

with the same values for γ and ρ_Y. Note that $K(r) - \pi r^2 = 0$ in case of a planar stationary Poisson process, and the figure shows that as β increases, the processes behave less and less like a planar stationary Poisson process. The pair correlation functions in the cases of the DSNCP processes show an increasing degree of clustering at small scales and regularity at larger scales as β increases, whereas Ripley's K-function only reveals an increasing degree of clustering. We furthermore see that the Ginibre-DPP-Thomas processes are overall more regular than the corresponding Gaussian-DPP-Thomas processes, especially at larger scales, which again reflects that the cluster centres are more regular in the Ginibre-DPP-Thomas processes. The considered Thomas processes are more clustered than the corresponding DSNCP processes and show no signs of repulsive behaviour.

4 Statistical inference

Suppose $d = 2$, $W \subset \mathbb{R}^2$ is a bounded observation window, $X \cap W = \{x_1, \ldots, x_n\}$ is a point pattern data set, and we want to fit a parametric

DSNCP model given by either the Gaussian-DPP-Thomas or Ginibre-DPP-Thomas process. There is a trade-off between ρ_Y and γ because of the relation $\rho_X = \gamma\rho_Y$. Therefore, when modelling the data, we choose to let $\rho_Y = 1/(\pi\beta^2)$, which means that Y will be as repulsive as possible. That is, $\gamma > 0$ and $\theta = (\alpha, \beta) \in (0, \infty)^2$ are the unknown parameters.

4.1 Estimation

Likelihood based inference is complicated because of the unobserved process of cluster centres.
Møller and Waagepetersen (2004) showed how a missing data MCMC approach can be used for maximum likelihood estimation in the special case of the Thomas process, and it may be simpler but still rather complicated to use a MCMC Bayesian setting along similar lines as in Beraha et al. (2022). We propose instead to exploit the parametric expressions of the intensity and of the pcf or K-function given in Section 3.2 when estimating γ and θ. In this paper, we use a minimum contrast procedure and leave it for future research to investigate the alternative approaches of composite likelihood (Guan, 2006) and Palm likelihood (Tanaka et al., 2008) using the expressions of ρ_X and $g_{X,\mathrm{st}}$, see the review in Møller and Waagepetersen (2017) and the references therein.

Specifically, we use a minimum contrast procedure for estimating θ, where it is preferable to consider Ripley's K-function, since it is easier to estimate K than $g_{X,\mathrm{st}}$ by non-parametric methods, see e.g. Møller and Waagepetersen (2004). Since K does not depend on γ, we need to estimate γ separately. Writing $K = K_\theta$ to stress the dependence of θ and $\hat{K}$ for a non-parametric estimate based on $\{x_1, \ldots, x_n\}$, the minimum contrast estimate of θ is given by

$$\hat{\theta} = \arg\min_{\theta} \left\{ \int_{r_{\min}}^{r_{\max}} |\hat{K}(r)^q - K_\theta(r)^q|^p \, \mathrm{d}r \right\}$$

where we use the R-package `spatstat` (Baddeley et al., 2015) for calculating $\hat{K}$ and the minimum contrast estimate by using default settings for the choice of $r_{\min}, r_{\max}, q$, and p. Finally, having estimated θ, we estimate γ from the unbiased estimation equation $\rho_X = \gamma/(\pi\beta^2) = \gamma\rho_Y = n/|W|$.

4.2 Model checking

When checking a fitted model, we prefer to use other functional summary statistics than $\hat{K}$ since this was used as part of the estimation procedure. The standard is to consider empirical estimates of theoretical functions known as the empty space function (or spherical contact function) F, the nearest-neighbour function G, and the J-function, which are defined for a stationary

point process X as follows. Consider any number $r > 0$ and an arbitrary point $u \in \mathbb{R}^d$. Then,

$$\begin{aligned} F(r) &= \mathrm{P}(\mathrm{dist}(X, u) \leq r), \\ G(r) &= \mathrm{P}(\mathrm{dist}(X \setminus \{u\}, u) \leq r \,|\, u \in X), \\ J(r) &= (1 - G(r))/(1 - F(r)). \end{aligned}$$

Here J is only defined for $F(r) < 1$, $\mathrm{dist}(X, u) = \inf\{r > 0 \,|\, b(u, r) \cap X \neq \emptyset\}$ is the distance from u to X, and in the definition of G when conditioning on $u \in X$ it means that $X \setminus \{u\}$ follows the reduced Palm distribution of X at u, see e.g. Møller and Waagepetersen (2004). Since X is stationary, the definitions of F, G, and J do not depend on the choice of u.

We have not been able to derive the expressions of F, G, and J for Gaussian- and Ginibre-DPP-Thomas processes; to the best of our knowledge, these expressions are not even known for a Thomas process. We refer to empirical estimates of these theoretical functions as functional summary statistics and use the relevant functions in `spatstat` to calculate such non-parametric estimates (always using the default settings including settings which account for boundary effects). Figure 3 concerns means of non-parametric estimates $\hat{F}$, $\hat{G}$, and $\hat{J}$ calculated from simulations of Thomas processes, Ginibre-DPP-Thomas processes and Gaussian-DPP-Thomas processes for the parameters stated in the caption. In agreement with Figure 2 the plots show an increas-

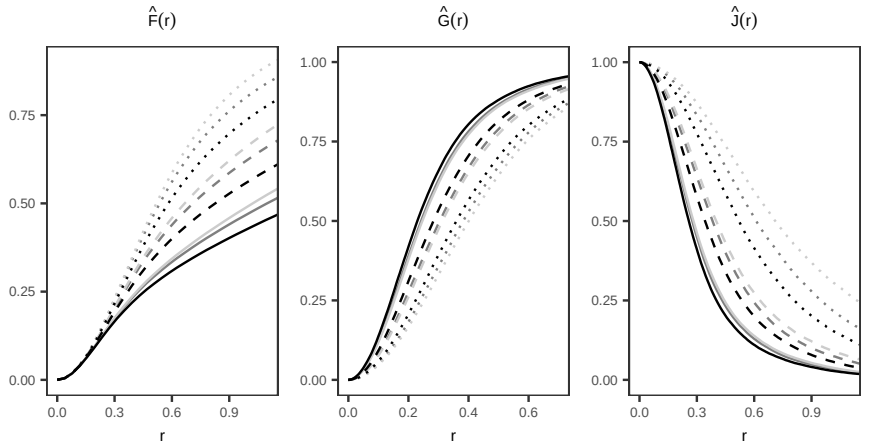

Fig. 3: Means of $\hat{F}, \hat{G}$, and $\hat{J}$ calculated from 500 simulations of Thomas processes (black curves), Gaussian-DPP-Thomas processes (dark grey curves), and Ginibre-DPP-Thomas processes (light grey curves) on a square with side lengths 20. In all types of processes, $\alpha = 1$, $\beta = 2, 3, 4$ (dotted, dashed, and solid curves, respectively), $\rho_Y = 1/(\pi\beta^2)$, and $\gamma = 1/\rho_Y$. Note that β is not a parameter of the Thomas process and is thus only used to calculate ρ_Y in this case.

ing degree of clustering as β increases and that the Thomas processes are more clustered than the corresponding DSNCP processes. The plots also indicate that the Gaussian-DPP-Thomas processes are more clustered and exhibit more empty space than the corresponding Ginibre-DPP-Thomas processes, and the difference becomes more apparent as β decreases.

In order to validate a fitted model, we use 95% global envelopes and global envelope tests based on the extreme rank length as described in Myllymäki et al. (2017), Mrkvička et al. (2020), and Myllymäki and Mrkvička (2019), which is implemented in the R-package `GET` (Myllymäki and Mrkvička, 2019). These envelopes are based on functional summary statistics calculated from a number of simulations under the fitted model. We use 2499 simulations as recommended in the above references.

4.3 An application example

The first point pattern in Figure 4 shows the positions of 448 white oak trees in a square region (scaled to a unit square) of Lansing Woods, Clinton County, Michigan USA, which is part of the `lansing` data set which is available in `spatstat`. We will refer to this point pattern as x_{obs}. By using

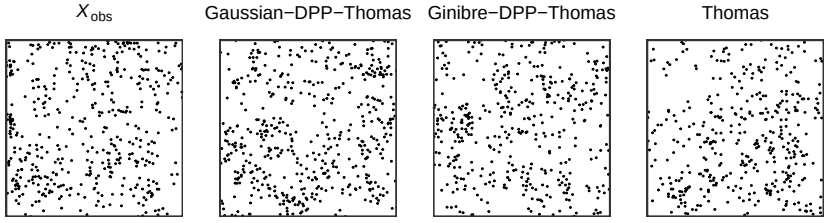

Fig. 4: Plots of the whiteoak point pattern (x_{obs}) and a simulation from fitted models of the type stated at the top of each plot.

the method of minimum contrast estimation as described in Section 4.1, we fitted a Gaussian-DPP-Thomas process, Ginibre-DPP-Thomas process, and Thomas process to x_{obs}. The obtained estimates are given in Table 1. We see

Model	β	ρ_Y	γ	α
Gaussian-DPP-Thomas	0.05	105.36	4.25	0.03
Ginibre-DPP-Thomas	0.09	35.32	12.68	0.05
Thomas	-	204.11	2.19	0.03

Table 1: Estimated parameters when fitting models to x_{obs}. Note that the Thomas process does not have the parameter β.

that the fitted DPP-Thomas processes expect much fewer clusters than the Thomas process and thus also more points in each cluster. As we expected, the fitted Ginibre-DPP-Thomas process is the one which expects the fewest clusters. Because of its expected 35 clusters with about 12 points on average in each it also seems to be a more sensible cluster process model than the other processes, which have many clusters with only a few points in each cluster. Figure 4 also shows a realization of each fitted model. The behaviour of these realizations are apparently in good agreement with x_{obs}. In order to check whether the models fit to data, we made 95% global envelope tests as described in Section 4.2. Figure 5 shows the results, which indicate that all three models fit well, but the Ginibre-DPP-Thomas process has a much higher p-value than the other two processes.

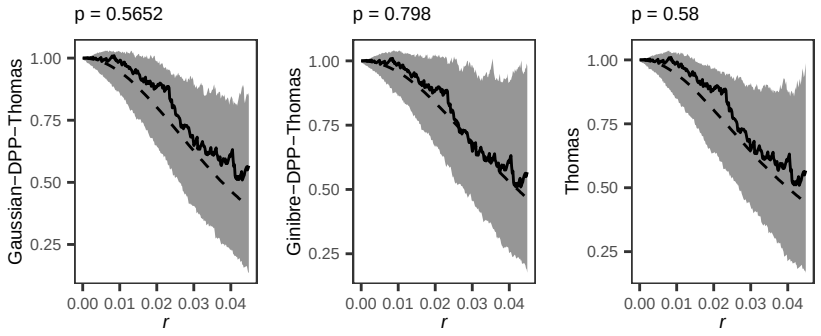

Fig. 5: Plots of 95% global envelopes (grey area) and tests (p-value stated above each plot) based on $\hat{J}(r)$ from 2499 simulations from the fitted model stated to the left of each plot. The solid curves show $\hat{J}$ for X_{obs}, and the dashed curves show the mean of $\hat{J}$ calculated from the simulations.

In connection with this paper we considered over 100 examples of point pattern data sets and found 20 point patterns which the considered DSNCP models describe well, including the application example in this section. For all of these we also found that the Thomas process fits well, but that the fitted Thomas process models expected more clusters than the corresponding fitted DSNCP models. Thus the situation exemplified in this section where all three of the considered models can be used to model data but the DSNCP models expect fewer clusters appears to be a typical situation.

5 Simulation study

Section 4.3 suggests that it may be difficult to distinguish between realizations of Thomas, Gaussian-DPP-Thomas, and Ginibre-DPP-Thomas pro-

cesses. To investigate this further, we in this section describe a simulation study where we considered the parameters $\rho_Y = 10, 30, 50$, $\gamma = 10, 30, 50$, and $\alpha = 0.03, 0.04, 0.05$ for the three considered cluster point process models (in the DPP-Thomas processes we as always used the relation $\rho_Y = 1/(\pi\beta^2)$ or equivalently $\beta = \sqrt{1/(\pi\rho_Y)}$). The values of α are like those in Table 1, and the values of ρ_Y and γ are like those from the fitted Ginibre-DPP-Thomas process in Table 1. For each combination of parameters and each model we made 100 simulations on a unit square. For each of these simulations, we fitted the two models which were not the true one and made a global envelope test as described in Section 4.3 for validating the fitted models. Since this simulation study is time consuming, we only used 1999 simulations to calculate each global envelope test in order to save some time, but this is still in agreement with the recommendations regarding the number of simulations in global envelope tests.

Table 2 shows the proportion of tests which yielded a p-value below 0.05 for each combination of parameters, true model, and fitted model. We overall see that in order to distinguish between the models, the ideal situation is when γ is large and α is small, meaning that the realization consists of small clusters with many points in each. Overall, it also seems to be an advantage if there is a moderate number of clusters since the rejection rates are generally higher when $\rho_Y = 30$, especially for small α. It appears to be most difficult to distinguish between the Gaussian-DPP-Thomas process and the two remaining processes, especially the Thomas process, whereas it is easier to distinguish between Thomas and Ginibre-DPP-Thomas processes. Figure 6 shows a realization under each model with parameters $\alpha = 0.03$, $\gamma = 50$, and $\rho_Y = 30$, which the simulation study suggests is a good situation when it comes to distinguishing between the models.

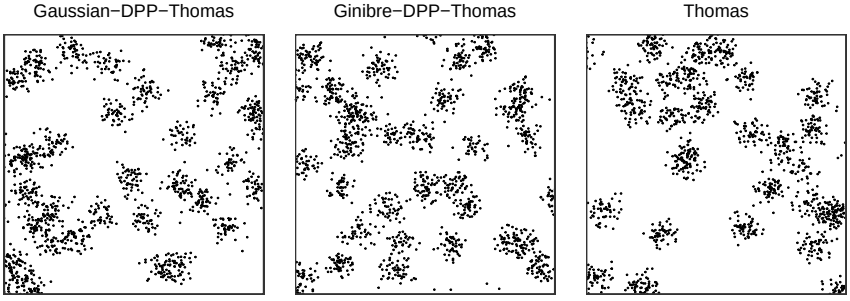

Fig. 6: Simulation of a realization under the model stated at the top with parameters $\alpha = 0.03$, $\gamma = 50$, and $\rho_Y = 1/(\pi\beta^2) = 30$.

		$\rho_Y = 10$			$\rho_Y = 30$			$\rho_Y = 50$		
Fitted model:	α \ γ	10	30	50	10	30	50	10	30	50
True model is a Ginibre-DPP-Thomas process										
Thomas	0.03	0.02	0.35	0.37	0.05	0.56	0.79	0.06	0.26	0.52
	0.04	0.03	0.11	0.30	0.02	0.06	0.22	0.05	0.06	0.00
	0.05	0.01	0.09	0.20	0.02	0.01	0.01	0.03	0.01	0.03
Gaussian	0.03	0.02	0.09	0.14	0.02	0.33	0.55	0.05	0.33	0.51
	0.04	0.06	0.09	0.10	0.02	0.11	0.24	0.04	0.10	0.06
	0.05	0.02	0.05	0.18	0.05	0.03	0.05	0.03	0.02	0.02
True model is a Gaussian-DPP-Thomas process										
Thomas	0.03	0.02	0.17	0.22	0.03	0.20	0.19	0.07	0.09	0.13
	0.04	0.02	0.06	0.09	0.04	0.02	0.05	0.02	0.04	0.02
	0.05	0.02	0.06	0.10	0.03	0.01	0.00	0.01	0.03	0.04
Ginibre	0.03	0.02	0.10	0.08	0.18	0.31	0.44	0.21	0.35	0.44
	0.04	0.03	0.07	0.10	0.09	0.15	0.15	0.04	0.14	0.15
	0.05	0.04	0.06	0.06	0.05	0.03	0.02	0.04	0.02	0.07
True model is a Thomas process										
Ginibre	0.03	0.08	0.19	0.25	0.25	0.45	0.60	0.19	0.48	0.56
	0.04	0.06	0.11	0.12	0.07	0.15	0.18	0.06	0.09	0.08
	0.05	0.03	0.05	0.05	0.03	0.06	0.10	0.02	0.05	0.02
Gaussian	0.03	0.02	0.08	0.06	0.06	0.07	0.15	0.08	0.08	0.19
	0.04	0.03	0.07	0.04	0.06	0.06	0.06	0.04	0.06	0.04
	0.05	0.02	0.04	0.01	0.05	0.02	0.03	0.02	0.04	0.05

Table 2: Table of the proportion of global envelope tests in the simulation study for for which the p-value was below 0.05. Concerning the column with the fitted model, for short the models Thomas process, Gaussian-DPP-Thomas process, and Ginibre-DPP-Thomas process are written as Thomas, Gaussian, and Ginibre, respectively.

We also used this simulation study to investigate the apparent tendency for fitted Thomas processes to expect more clusters than fitted DPP-Thomas processes. Table 3 shows the mean of the fitted value of ρ_Y divided by the value of ρ_Y in the true model for each combination of parameters, model, and fitted model. We see that when the true model is a DPP-Thomas process, the fitted Thomas processes expect more clusters than the true model, especially when the true model is a Ginibre-DPP-Thomas process; this behaviour gets more extreme as α and ρ_Y increases. This is also the behaviour of the fitted Gaussian-DPP-Thomas processes when the true model is a Ginibre-DPP-Thomas process, although it is not as extreme as for the fitted Thomas processes. If the true model is a Thomas process, we similarly see that the fitted DPP-Thomas processes expect fewer clusters than the true model, especially the Ginibre-DPP-Thomas process; this behaviour gets more extreme as ρ_Y increases, whereas in this case it seems that α has only little influence on

		$\rho_Y = 10$			$\rho_Y = 30$			$\rho_Y = 50$		
Fitted model:	α \ γ	10	30	50	10	30	50	10	30	50
True model is a Ginibre-DPP-Thomas process										
Thomas	0.03	1.92	1.85	1.84	2.83	2.77	2.78	3.71	3.55	3.54
	0.04	2.10	2.10	2.08	3.78	3.52	3.65	5.11	4.71	4.85
	0.05	2.50	2.23	2.54	4.93	4.66	4.32	6.73	6.50	6.68
Gaussian	0.03	1.26	1.22	1.22	1.60	1.57	1.57	1.98	1.91	1.90
	0.04	1.32	1.32	1.31	2.03	1.89	1.96	2.63	2.41	2.50
	0.05	1.49	1.35	1.51	2.55	2.42	2.25	3.41	3.29	3.38
True model is a Gaussian-DPP-Thomas process										
Thomas	0.03	1.52	1.59	1.71	1.89	1.93	1.81	2.03	2.02	2.04
	0.04	1.72	1.73	1.61	2.00	2.01	2.12	2.41	2.19	2.17
	0.05	1.87	1.95	1.98	2.41	2.36	2.18	2.31	2.49	2.45
Ginibre	0.03	0.85	0.89	0.93	0.70	0.71	0.67	0.59	0.59	0.59
	0.04	0.85	0.86	0.81	0.61	0.61	0.63	0.55	0.51	0.52
	0.05	0.83	0.86	0.87	0.62	0.60	0.57	0.45	0.49	0.49
True model is a Thomas process										
Ginibre	0.03	0.70	0.65	0.64	0.44	0.45	0.43	0.37	0.38	0.38
	0.04	0.66	0.59	0.64	0.44	0.42	0.43	0.34	0.34	0.35
	0.05	0.59	0.59	0.61	0.38	0.40	0.40	0.33	0.31	0.33
Gaussian	0.03	0.83	0.77	0.75	0.65	0.67	0.62	0.62	0.64	0.64
	0.04	0.81	0.72	0.78	0.71	0.66	0.69	0.64	0.65	0.66
	0.05	0.76	0.76	0.78	0.67	0.71	0.70	0.69	0.64	0.68

Table 3: Table of the mean of the fitted value of ρ_Y divided by the value of ρ_Y in the true model. Concerning the column with the fitted model, for short the models Thomas process, Gaussian-DPP-Thomas process, and Ginibre-DPP-Thomas process are written as Thomas, Gaussian, and Ginibre, respectively.

this behaviour. This is also the behaviour of the fitted Ginibre-DPP-Thomas processes when the true model is a Gaussian-DPP-Thomas process. The parameter γ in the true model has no apparent effect on the expected number of clusters in the fitted models.

6 Conclusion

We have presented the new class of cluster point process models called determinantal shot noise Cox processes which have repulsion between cluster centres. For the two special cases which we have called Gaussian-DPP-Thomas processes and Ginibre-DPP-Thomas processes we have derived closed form expressions for the pair correlation function and Ripley's K-function. The ability to actually derive such closed form parametric expressions for these theoretical summary functions is a huge advantage compared to using Mar-

kov point processes for the cluster centres, which has previously been done for cluster point processes with repulsion between clusters, since easy and fast parameter estimation can then be achieved with the method of minimum contrast estimation or other methods based on the pair correlation or K-function, cf. Section 4.1.

We have seen that the fitted DPP-Thomas process models in Sections 4.3 and 5 expect much fewer clusters than a Thomas process and thus they also expect much more points in each cluster, especially the Ginibre-DPP-Thomas model. In many situations it will be intuitively more pleasing to fit a cluster point process with few clusters consisting of many points compared to many clusters consisting of very few points. We have also seen through a simulation study that the ideal situation for distinguishing between the considered three types of cluster point process models is if the realization has small clusters with many points in each.

Acknowledgements

We would like to thank Rasmus Plenge Waagepetersen for providing us with many examples of point pattern data sets which we considered in connection with the application example.

A Definition of a DPP and some properties

Let Y be a simple point process defined on $\mathbb{R}^d$ and c be a complex function defined on $\mathbb{R}^d \times \mathbb{R}^d$ so that for every integer $n > 0$ and pairwise disjoint bounded Borel sets $A_1, \ldots, A_n \subset \mathbb{R}^d$, we have

$$\mathrm{E}\left[Y(A_1) \cdots Y(A_n)\right] = \int_{A_1} \cdots \int_{A_n} \det\{c(u_i, u_j)\}_{i,j=1,\ldots,n} \, \mathrm{d}u_1 \cdots \mathrm{d}u_n < \infty$$

where $\det\{c(u_i, u_j)\}_{i,j=1,\ldots,n}$ is the determinant of the $n \times n$ matrix with ij'th entry $c(u_i, u_j)$. Then Macchi (1975) defined Y to be a DPP with kernel c. Note that Y must be locally finite and the function

$$\rho^{(n)}(u_1, \ldots, u_n) = \det\{c(u_i, u_j)\}_{i,j=1,\ldots,n} \tag{11}$$

is the so-called n'th order intensity function $\rho^{(n)}$ of Y.

In fact for the DPP Y, its distribution is unique and completely characterized by the intensity functions of all order, cf. Lemma 4.2.6 in Hough et al. (2009). Thus stationarity of Y is equivalent to that $\rho^{(n)}(u_1, \ldots, u_n) = \rho^{(n)}(u_1 + v, \ldots, u_n + v)$ for all $v \in \mathbb{R}^d$ and (Lebesgue almost) all $u_1, \ldots, u_n \in \mathbb{R}^d$, and isotropy of Y means that $\rho^{(n)}(u_1, \ldots, u_n) = \rho^{(n)}(\mathcal{O}u_1, \ldots, \mathcal{O}u_n)$ for all $n \times n$ rotations matrices $\mathcal{O}$ and (Lebesgue almost) all $u_1, \ldots, u_n \in \mathbb{R}^d$.

For later use, consider any numbers $\beta > 0$ and $0 \le p \le 1$, and the scaled point process $\beta Y = \{\beta y \mid y \in Y\}$. Let $Y_{\beta,p}$ be an independent p-thinning of βY (that is, the points in βY are independently retained with probability p and $Y_{\beta,p}$ consists of those retained points). It is easily seen that $Y_{\beta,p}$ is a DPP with kernel

$$c_{\beta,p}(u,v) = (p/\beta)^d c(u/\beta, v/\beta). \tag{12}$$

B Some results for the scaled Ginibre point processes

In the following assume $d = 2$ and Y is a standard Ginibre point process as defined in Section 2, so we identify $\mathbb{R}^2$ with the complex plane $\mathbb{C}$. Let $Y_{\beta,p}$ be as above and let $\lambda = \rho_{Y_{\beta,p}}$ be its intensity. By (12), $Y_{\beta,p}$ is the DPP with kernel

$$c_{\beta,p}(u,v) = \lambda \exp((u\overline{v} - |u|^2/2 - |v|^2/2)/\beta^2), \quad u, v \in \mathbb{C},$$

and $\lambda = (p/\beta)^2/\pi$. In Section 2 we used the variation dependent parametrization $(\rho_{Y_{\beta,p}}, \beta)$, which is in one-to-one correspondence to (β, p). For the following it is convenient to let $\nu = p^2$ and use the variation independent parametrization $(\nu, \lambda) \in (0,1] \times (0,\infty)$, which is also in one-to-one correspondence to (β, p). Using this parametrization, with a slight abuse of notation we write $Y_{\nu,\lambda}$ for the DPP $Y_{\beta,p}$ and

$$c_{\nu,\lambda}(u,v) = \lambda \exp\Big((\lambda\pi/\nu)(u\overline{v} - |u|^2/2 - |v|^2/2)\Big) \tag{13}$$

for its kernel.

B.1 Invariance under isometries

Below we show that the n'th order intensity function is invariant under translations and rotations, and therefore Y is stationary and isotropic. In the same way, it can be shown that $\rho^{(n)}$ is invariant under reflections and glide reflections. So the distribution of Y is invariant under isometries (mappings of the form $z \to az + b$ and $z \to a\bar{z} + b$ where $a, b \in \mathbb{C}$ with $|a| = 1$; these mappings correspond to translations, rotations, reflections, and glide reflections).

Denote S_n the set of all permutations of $\{1, 2, \ldots, n\}$ and $\operatorname{sgn}(\sigma)$ the sign

of a permutation $\sigma \in S_n$. From (11) and (13) we get

$$\begin{aligned}\rho^{(n)}(u_1,\ldots,u_n) &= \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} c_{\nu,\lambda}(u_i,u_{\sigma(i)}) \\ &= \lambda^n \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma) \exp\left((\lambda\pi/\nu)\sum_{i=1}^{n}\left(u_i\overline{u}_{\sigma(i)} - |u_i|^2/2 - |u_{\sigma(i)}|^2/2\right)\right) \\ &= \lambda^n \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma) \exp\left((\lambda\pi/\nu)\sum_{i=1}^{n}\left(u_i\overline{u}_{\sigma(i)} - |u_i|^2\right)\right).\end{aligned}$$

Hence, for any $a, b, u_1, \ldots, u_n$, a straightforward calculation gives

$$\rho^{(n)}(au_1+b,\ldots,au_n+b) = \rho^{(n)}(u_1,\ldots,u_n),$$

so $\rho^{(n)}$ is invariant under translations and rotations.

B.2 Spectral decompositions

Spectral representations of the kernel restricted to compact regions are needed for simulation as well as other purposes, cf. Lavancier et al. (2015). The simplest case occurs when we consider $Y_{\nu,\lambda}$ restricted to a closed disc around zero. So for $r > 0$, let $b(0,r) \subset \mathbb{C}$ be the closed disk around zero with radius $r \in (0,\infty)$ and $Y_{\nu,\lambda,r} = Y_{\nu,\lambda} \cap b(0,r)$ the restriction of $Y_{\nu,\lambda}$ to $b(0,r)$. Because $Y_{\nu,\lambda}$ is a DPP, $Y_{\nu,\lambda,r}$ is a DPP with kernel

$$c_{\nu,\lambda,r}(u,v) = \begin{cases} c_{\nu,\lambda}(u,v) & \text{if } (u,v) \in b(0,r), \\ 0 & \text{otherwise.}\end{cases}$$

The integral operator corresponding to the kernel $c_{\nu,\lambda}$ has only one eigenvalue, namely ν, and the eigenfunctions are

$$\phi^i_{\nu,\lambda}(u) = \frac{\sqrt{\lambda}(\lambda\pi)^{(i-1)/2}}{\sqrt{(i-1)!\nu^i}} \exp(-\lambda\pi|u|^2/(2\nu))u^{i-1}, \quad u \in \mathbb{C}, \quad i = 1,2,\ldots.$$

This follows easily by exploiting the moment properties of two independent zero-mean complex normally distributed random variables and the definition of the complex exponential function ($\exp(z) = \sum_{k=0}^{\infty} z^k/k!$ for $z \in \mathbb{C}$) for the term $\exp((\lambda\pi/\nu)u\overline{v})$ in (13). In other words, we have the spectral representation

$$c_{\nu,\lambda}(u,v) = \sum_{i=1}^{\infty} \nu\phi^i_{\nu,\lambda}(u)\overline{\phi^i_{\nu,\lambda}(v)}.$$

Similarly, we see that the integral operator corresponding to $c_{\nu,\lambda,r}$ has eigenfunctions

$$\phi^i_{\nu,\lambda,r}(u) = \phi^i_{\nu,\lambda}(z)/\sqrt{F_i(\lambda\pi r^2/\nu)}, \quad u \in b(0,r), \quad i = 1,2,\ldots,$$

with corresponding eigenvalues

$$\xi^i_{\nu,\lambda,r} = \nu F_i(\lambda\pi r^2/\nu), \quad i = 1, 2, \ldots, \tag{14}$$

and the spectral representation is

$$c_{\nu,\lambda,r}(u,v) = \sum_{i=1}^{\infty} \xi^i_{\nu,\lambda,r}\phi^i_{\nu,\lambda,r}(u)\overline{\phi^i_{\nu,\lambda,r}(v)}.$$

C Simulation procedures

For simulating determinantal point processes, we use the algorithm described in Lavancier et al. (2015) which is a specific case of the simulation algorithm of Hough et al. (2006). We refer to these references for specific details. The algorithm is implemented in `spatstat` for the models suggested in Lavancier et al. (2015), which include Gaussian DPPs. For these models, it is necessary to approximate the kernel because the spectral representation is unknown. In the case of a scaled Ginibre point process, this approximation is however unnecessary for simulating it on a disc because the spectral representation is known, see Appendix B.2. The simulation is still only approximate because the procedure also involves other approximations including approximating the upper bound for rejection sampling chosen in Lavancier et al. (2015) (an approximation which is in fact not necessary for the models they consider since the expression simplifies in those cases). For simulating a scaled Ginibre point process on a window W, we thus use the spectral representation on a disc to simulate the process on $b(0,r) \supseteq W$ and thereafter extract the part which is in W.

For simulating DPP-Thomas processes on a window W, we first simulate the DPP Y_{ext} obtained by restricting Y to an extended window in order to account for boundary effects. Regarding the extension, we decided to use the default setting from the function `rThomas` in `spatstat` which simulates a Thomas process. Given the cluster centers Y_{ext} on the extended window, we simulate the clusters of the DPP-Thomas process X independently as finite Poisson processes with intensity functions $\rho_y(x) = \gamma K_\alpha(x-y)$ for each $y \in Y_{\text{ext}}$. That is, first simulate the number of points n_y in a cluster X_y centered at $y \in Y$ from a Poisson distribution with rate γ. Then sample the n_y independent points in X_y from the d-dimensional normal distribution $N_d(y, \alpha^2 I)$. Finally, the simulation of X on W is the part of $\cup_{y\in Y} X_y$ which falls in W.

References

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

Beraha, M., Argiento, R., Møller, J., and Guglielmi, A. (2022). MCMC computations for Bayesian mixture models using repulsive point processes. *Journal of Computational and Graphical Statistics*. To appear. Available at arXiv:2011.06444.

Biscio, C. A. N. and Lavancier, F. (2016). Quantifying repulsiveness of determinantal point processes. *Bernoulli*, 22:2001–2028.

Christoffersen, A. D., Møller, J., and Christensen, H. S. (2021). Modelling columnarity of pyramidal cells in the human cerebral cortex. *Australian and New Zealand Journal of Statistics*, 63:33–54.

Deng, N., Zhou, W., and Haenggi, M. (2014). The Ginibre point process as a model for wireless networks with repulsion. *IEEE Transactions on Wireless Communications*, 14:107–121.

Ginibre, J. (1965). Statistical ensembles of complex, quaternion, and real matrices. *Journal of Mathematical Physics*, 6:440–449.

Guan, Y. (2006). A composite likelihood approach in fitting spatial point process models. *Journal of American Statistical Association*, 101:1502–1512.

Hough, J. B., Krishnapur, M., Peres, Y., and Virág, B. (2006). Determinantal processes and independence. *Probability Surveys*, 3:206–229.

Hough, J. B., Krishnapur, M., Peres, Y., and Virág, B. (2009). *Zeros of Gaussian Analytic Functions and Determinantal Point Processes*. American Mathematical Society, Providence.

Lavancier, F., Møller, J., and Rubak, E. (2015). Determinantal point process models and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77:853–877.

Macchi, O. (1975). The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7:83–122.

Miyoshi, N. and Shirai, T. (2016). Spatial modeling and analysis of cellular networks using the Ginibre point process: A tutorial. *IEICE Transactions on Communications*, 99:2247–2255.

Møller, J. (2003). Shot noise Cox processes. *Advances in Applied Probability*, 35:614–640.

Møller, J. and O'Reilly, E. (2021). Couplings for determinantal point processes and their reduced Palm distributions with a view to quantifying repulsiveness. *Journal of Applied Probability*, 58:469–483.

Møller, J. and Torrisi, G. L. (2005). Generalised shot noise Cox processes. *Advances in Applied Probability*, 37:48–74.

Møller, J. and Waagepetersen, R. (2004). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman & Hall/CRC, Boca Raton, Florida.

Møller, J. and Waagepetersen, R. (2017). Some recent developments in statistics for spatial point patterns. *Annual Review of Statistics and Its Applications*, 4:317–342.

Mrkvička, T., Myllymäki, M., Jilik, M., and Hahn, U. (2020). A one-way ANOVA test for functional data with graphical interpretation. *Kybernetika*, 56:432–458.

Myllymäki, M. and Mrkvička, T. (2019). GET: Global envelopes in R. arXiv preprint arXiv:1911.06583.

Myllymäki, M., Mrkvička, T., Grabarnik, P., Seijo, H., and Hahn, U. (2017). Global envelope tests for spatial processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79:381–404.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ripley, B. D. (1976). The second-order analysis of stationary point processes. *Journal of Applied Probability*, 13:255–266.

Ripley, B. D. (1977). Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39:172–212.

Tanaka, U., Ogata, Y., and Stoyan, D. (2008). Parameter estimation and model selection for Neyman-Scott point processes. *Biometrical Journal*, 50:43–57.

Thomas, M. (1949). A generalization of Poisson's binomial limit in the use in ecology. *Biometrika*, 36:18–25.

Van Lieshout, M. N. M. and Baddeley, A. J. (2002). Extrapolating and interpolating spatial patterns. In Lawson, A. B. and Denison, D., editors, *Spatial Cluster Modelling*, pages 61–86. Chapman and Hall/CRC, Boca Raton, FL.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

ISSN (online): 2446-1636
ISBN (online): 978-87-7573-877-9

AALBORG UNIVERSITY PRESS