Accurate Approximation Series for Optimal Targeting Regions in a Neural Growth Model with a Low –branching Probability

Bernardo Nieto

Follow this and additional works at: https://scholarworks.gsu.edu/math_theses

Recommended Citation

This Thesis is brought to you for free and open access by the Department of Mathematics and Statistics at ScholarWorks @ Georgia State University. It has been accepted for inclusion in Mathematics Theses by an authorized administrator of ScholarWorks @ Georgia State University. For more information, please contact scholarworks@gsu.edu.
ACCURATE APPROXIMATION SERIES FOR OPTIMAL TARGETING REGIONS IN A NEURAL GROWTH MODEL WITH A LOW –BRANCHING PROBABILITY

by

BERNARDO NIETO

Under the Direction of Remus Osan, PhD

ABSTRACT

Understanding the complex growth process of dendritic arbors is essential for the medical field and disciplines like Biology and Neurosciences. The establishment of the dendritic patterns has received increasing attention from experimental researchers that seek to determine the cellular mechanisms that play a role in the growth of neural trees. Our goal in this thesis was to prove the recurrence formula for the probability distribution of all possible neural trees, as well as the formulas of the expected number of active branches and their variances. We also derived formulas for the spatial locations of the optimal targeting region for a tree with branching probability. These formulas were necessary for the simplified stochastic computational model that Osan et al have developed in order to examine how changes in branching probability influence the success of targeting neurons located at different distances away from a starting point.

INDEX WORDS: Growth of neural trees, Computational model, Stochastic branching probability, Expected number of active branches, Variances, Recurrence formula.
ACCURATE APPROXIMATION SERIES FOR OPTIMAL TARGETING REGIONS IN A NEURAL GROWTH MODEL WITH A LOW –BRANCHING PROBABILITY

by

BERNARDO NIETO

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in the College of Arts and Sciences

Georgia State University

2015
ACCURATE APPROXIMATION SERIES FOR OPTIMAL TARGETING REGIONS IN A NEURAL GROWTH MODEL WITH A LOW –BRANCHING PROBABILITY

by

BERNARDO NIETO

Committee Chair: Remus Osan

Committee: Vladimir Bondarenko

Yi Jiang

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
December 2015
DEDICATION

To my son Emiliano
ACKNOWLEDGEMENTS

I would first like to thank my thesis advisor, Dr. Remus Osan, for his invitation to participate in this team project where he and his collaborators have been working for several years and have obtained many interesting theoretical and analytical results about neural arborization. I greatly appreciate the time he spent with me to explain the details of the arborization model he used in his previous work. The paper notes he produced when working with me were very helpful. I also want to thank Jun Xia for his time spent with me answering my questions. Those talks were useful for me to carry out the analytical formulation of my work which I present in this thesis. I also appreciate his help with the Matlab codes for the recurrence formulas.

I would like to thank my other committee members Dr. Yi Jiang and Dr. Vladimir Bondarenko for taking the time to review and evaluate this thesis.

I would also like to thank, Dr. Gensheng Qin, Dr. Alexandra Smirnova, and Ms. Sandra Ahuama-Jones for all the administrative support during these years.

Finally I would specially like to thank Dr. Guantao Chen for his unwavering support along the way.
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ........................................................................................................... v

LIST OF TABLES ...................................................................................................................... vii

LIST OF FIGURES ..................................................................................................................... ix

1 INTRODUCTION .................................................................................................................. 1

1.1 Overview: Neurons, Dendrites, Structure and Function ................................................. 1

1.2 Previous research in Mathematical and Computational Models of Neural Arborization ................................................................................................................................. 9

2 METHODS AND RESULTS ................................................................................................. 13

2.1 Discrete Probability Distribution for the Evolving Family of Trees ......................... 13

2.2 Recurrence Formula for the Probability Distribution of all Possible Neural Trees ................................................................................................................................. 16

2.3 Describing Length Constraints ...................................................................................... 18

2.4 Approximating the Location of Optimal Targeting Performances ............................. 20

2.5 Methods for Neuronal Culture and Analysis ................................................................. 22

2.5.1 Cell Culture, Transfection, and Immunostaining: .................................................. 23

2.5.2 Imaging and Assessment of Dendrite Number ......................................................... 23

2.6 Analytical Methods ........................................................................................................ 25

2.6.1 Derivation of the Recurrence Formula for all Possible Neural Trees ...... 25

2.6.2 Derivation of the Formula for the Expected Number of Active Branches. 26
2.6.3 Derivation of the Formula for the Variance of the Expected Number of
Active Branches ................................................................. 28

2.6.4 Optimal Targeting Region for Trees with Size Restrictions...................... 29

2.6.4.1 Determining the Approximation for the Peak Location .................. 30

3 CONCLUSION .................................................................................................................. 32

REFERENCES ..................................................................................................................... 35

APPENDICES ...................................................................................................................... 38

Appendix A Calculations .................................................................................................. 38

Appendix A.1 Obtaining the Expected Value for Step 2 ........................................... 38

Appendix A.2 Obtaining the Variance for step 2  n=2 ............................................. 38

Appendix A.3 Proof of Theorem 4.4 ............................................................................. 38

Appendix A.4 Complementary steps for the proof of theorem 4.4 ......................... 42

Appendix A.5 Obtaining the number of active branches for step 2 using the
formula: ............................................................................................................................. 46

Appendix A.6 Obtaining the total number of possible trees in step 3 using ......... 46

Appendix B Matlab Code to Obtain the Expected Number of Active Branches
using the Derivative of the Recurrence Formula. ......................................................... 47
LIST OF TABLES

Table 1 Average number of active branches and the associated variances for the first two time steps.................................................................................................................................................................................. 15

Table 2 Expected Value and Variance for the active (terminal) branches after three time steps . 16

Table 3 Maximum length of the tree is equal to 7 ............................................................................................................................................. 30
LIST OF FIGURES

Figure 1 The parts of a Neuron (Nerve Cell): Cell Body, Dendrites, Axon, Myelin Sheath, Node of Ranvier, End Feet/Axon Terminal. [4] ..................................................................................................................... 2

Figure 2 The transmission of nerve impulses or Action Potentials (AP’s) is unidirectional. The impulse (AP) crosses the synapse from the end feet (axon terminal) of cell A into the dendrites of cell B. From the dendrites of cell B the impulse (AP) travels to the cell body and then out again along the axon to its end feet (axon terminal). Once there, it jumps across the synapse, helped by chemical messengers, to the dendrites of cell C. This process continues until the impulse (AP) reaches either the brain or the muscle/organ involved. [3] .......................................................................................................................... 3

Figure 3 Diversity of dendritic morphology. Different dendritic morphologies illustrating their wide diversity in neural systems. Dendrites are laid out on the same scale: (red) rat cortical pyramidal cell; (cyan) fly lobula plate HSN cell ;(orange) rat thalamic relay neuron;(yellow) rat hippocampal pyramidal cell; (green) rat cerebellar Purkinje cell;(pink) rat neocortical neuroglia form cell. Note the differences in size, overall shape, and diameters. [5] ........................................................................................................................................ 6

Figure 4 Time sequences showing branching and pruning of dissociated E11 chick dorsal root ganglion neurites. (a) Branching (red arrow) and extension (blue arrowheads) of primary axons. (b) Extension and retraction (blue arrowheads) of neurite tip. (c) Tertiary branching and pruning (encircled). Cultures are grown in the presence of glia in 5% CO2/ 37uC on Poly-L-lysine/laminin in N3 complete serum-free media. Phase-contrast live imaging at 28 hrs post-plating. Time interval between acquisitions for
each time series is as follows: (a) 30 mins, (b) 75 mins, (c) 75 mins. Snapshots are contrast enhanced for visual clarity of the neurites. [6]............................................ 8

Figure 5 Diagram of all possible tree instantiation after 3 time steps. (reproduced from [6]). ... 14

Figure 6 All possible trees with lengths below a maximum value of 7........................................... 18

Figure 7 Expected number of branches as a function of distance for families of trees with maximum length 7 and branching probability p = 0.5. Close to the origin, the expected value of searching sites increases as a geometrical series (step 2). However, this picture changes at large distances from the origin. There is an optimal targeting region, indicated by the peak at step 3, and a long tail of decaying values for steps 4 to 7. There are no possible trees for steps 8 or larger since a tree of maximal size of 7 cannot reach beyond a total distance of 7. As a matter of fact, only one tree, the one that does not branch at all and has a probability $q^6$, will reach a distance $d = 7$ with only one active branch. ........... 19

Figure 8 Sholl plots for different branching probabilities p. Expected number of dendrites at each time step is plotted for a tree of maximum possible length of 15, for the branching probabilities in the set \{0, 0.2, 0.4, 0.6, 0.8, 1\}. The tree with $p = 0$, showed in blue, simply extends until it runs out of resources, and generates a flat line of 1 expected branch at each time step. In contrast, the tree with $p = 1$, doubles the number of branches at each time step and runs out of resources at step 3, achieving a maximum at the time it stops. In between these two extremes, families of trees shift their optimal targeting regions farther away as the branching probability decreases, at the cost of reducing the overall amplitude of the corresponding peak values (success rates). ........... 20

Figure 9 Comparison of naïve vs exact results for a tree of maximum length $L = 31$. ............... 22

Figure 10 Two Samples Neurons A-B.................................................................................................. 24
Figure 11 Arborization profiles for two sample neurons (A and B) result in typical profile curves (blue curve) for the family of these neurons (n = 30). Fits of the probabilistic neural growth model are in excellent agreement (red curve).

Figure 12 Comparison of equal length trees under different branching scenarios.
1 INTRODUCTION

1.1 Overview: Neurons, Dendrites, Structure and Function

Spinal cord injury is a devastating medical condition that in many cases requires lifelong healthcare. The movement loss contributes to other health complications that accelerate the physical and mental deterioration of the patient. The development of strategies that facilitate the regrowth and reconnection of nerve cells (neurons) is an important area of research in Neuroscience and the medical field. Understanding the biological mechanisms by which growing neurites seek out and synapse with viable targets is fundamental for the development of appropriate regenerative therapies. It is important to mention that there are other strategies, like the neural decoding approach which is concerned with the reconstruction of sensory and other stimuli from information that has already been encoded and represented in the brain by networks of neurons. These signals can then be used to stimulate the muscle neurons and thus to generate movement through an external loop. The central focus of this neural decoding approach is to characterize how the electrical activity of neurons generates activity and responses in the brain. [1] Nicolelis’ Lab at Duke University Medical Center is one of the pioneers in this approach and, with his Brain Machine Interfaces and Neuroprosthetics, might provide an alternative to the strategy of regrowth and reconnection of nerve cells.

Neurons, like all cells, have a membrane containing a nucleus and a cytoplasm but they look quite different from other cells due to the many long cellular processes that extend from their central cell-body (See figure 1) [2]. Unlike other cells in the body, neurons are not replaced when they die; however, some neurons have the ability to regenerate. [3]
The neurons communicate within the body by transmitting electrochemical signals. They have small tree-like structures called dendrites that extend from the cell body of the neuron to pick up stimuli from the environment, other neurons, or sensory receptor cells. They also have long transmitting processes called axons that extend from the cell body to send signals onward to other neurons or effector cells in the body. (See figure 2)
The transmission of nerve impulses or Action Potentials (AP’s) is unidirectional. The impulse (AP) crosses the synapse from the end feet (axon terminal) of cell A into the dendrites of cell B. From the dendrites of cell B the impulse (AP) travels to the cell body and then out again along the axon to its end feet (axon terminal). Once there, it jumps across the synapse, helped by chemical messengers, to the dendrites of cell C. This process continues until the impulse (AP) reaches either the brain or the muscle/organ involved.

There are three basic classes of neurons: the afferent neurons, the efferent neurons, and the interneurons. The afferent neurons transmit sensory signals to the central nervous system from receptors in the body. The efferent neurons transmit signals from the central nervous system to effectors in the body such as muscles and glands. Finally, the interneurons, which form complex networks within the central nervous system, i.e., the spinal cord and the brain, and whose function is to integrate the information received from the afferent neurons and to direct the function of the body through the efferent neurons.
Neurons function through the generation and propagation of electrochemical signals known as action potentials (nerve impulses). An AP is created by the movement of sodium and potassium ions through the membrane’s neuron. At rest, neurons retain a concentration of sodium ions outside of the cell and potassium ions inside of the cell. This concentration is retained by the sodium-potassium pump of the cell membrane which pumps 3 sodium ions out of the cell for every 2 potassium ions that are pumped into the cell. The ion concentration results in a resting electrical potential of -70 millivolts (mV), which means that the inside of the cell has a negative charge compared to its surroundings [2, 3].

If a stimulus permits enough positive ions to enter a region of the cell to cause it to reach -55 mV, that region of the cell will open its voltage-gated sodium channels and allow sodium ions to diffuse into the cell. The threshold potential for activation of the sodium current is -55 mV, as this is the “trigger” voltage that they must reach to cross the threshold into forming an action potential. [3]

Sodium carries a positive charge that causes the cell to become depolarized, that means, positively charged compared to its normal negative charge. The neural cells can be depolarized up to +30 mV. The depolarization of the cell is the AP that is transmitted by the neuron as a nerve signal. The positive ions spread into neighboring regions of the neural cell, initiating a new AP in those regions as they reach threshold voltage of -55 mV. The AP continues to spread down the cell membrane of the neuron until it reaches the end of an axon [2, 3]. After the depolarization voltage is reached, voltage-gated potassium ion channels open, allowing positive potassium ions to diffuse out of the cell.

The axon terminal is separated from the next cell by a small gap known as the synaptic cleft. It is in this place where a chemical synapse occurs. A synapse is the junction between a
neuron and another cell. Synapses may form between two neurons or between a neuron and an effector cell. There are two types of synapses found in the body: chemical synapses and electrical synapses.

When an AP reaches the axon terminal, it opens voltage-gated calcium ion channels. Calcium ions cause vesicles containing chemicals known as neurotransmitters to release their contents by exocytosis into the synaptic cleft. The neurotransmitters molecules cross the synaptic cleft and bind to receptor molecules on the cell, forming a synapse with the neuron. These receptor molecules open ion channels that may either stimulate the receptor cell to form a new action potential or may inhibit the cell from forming an action potential when stimulated by another neuron [2].

Electrical synapses are formed when two neurons are connected by small holes called gap junctions. The gap junctions allow electric current to pass from one neuron to the other, so that an AP in one cell is passed directly on to the other cell through the synapse [2].

The most remarkable feature of a neuron is its characteristic morphology: dendritic and axonal processes sprout as intricate arboreal structures to enable connection with other neurons. Dendrites play an important role in neuronal function and connectivity. It is through their dendrites that neurons receive signals from other neurons, and via their axons they transmit those signals to other neurons. The study of neuronal morphology, historically, has been focused more significantly on dendrites than in axon structures, although, recently, more research of axon structures has become available. With the increase in the quantity and quality of neuronal staining and microscopy methods the interest on dendritic morphological analysis has been renewed. The knowledge obtained through the morphological analysis of dendritic tree structures is extremely useful for resolving the circuitry and function of the nervous system [5].
Dendritic and axon branches appear in many shapes and sizes (See figure 3) [5]. Their total length ranges from a few tens of micrometers to a few millimeters. In some cases we find neurons having only one main dendritic branch, while others could show up to 15 or more. Some branches have a curvy form while others are approximately straight.

Figure 3 Diversity of dendritic morphology. Different dendritic morphologies illustrating their wide diversity in neural systems. Dendrites are laid out on the same scale: (red) rat cortical pyramidal cell; (cyan) fly lobula plate HSN cell ;(orange) rat thalamic relay neuron;(yellow) rat hippocampal pyramidal cell; (green) rat cerebellar Purkinje cell;( pink) rat neocortical neuroglia form cell. Note the differences in size, overall shape, and diameters. [5]
Dendritic morphologies vary significantly even within one neuronal class. Added to this structural diversity, the molecular composition of ion channels in the membrane have remarkable differences along the stretch of one dendrite, and more pronounced differences even exist between different types of neurons.

Dendrites clearly play two critical roles in the process of signal integration. First, neuronal morphology defines and is defined by the circuitry. The most relevant element of neuronal connectivity is the synaptic contact between the output axon of one neuron and the input dendrite of another. At this point, a precise morphology is crucial to establish the connectivity required for the nervous system to operate normally. Secondly, the precise morphology of a dendrite and its membrane’s ion channel composition set the computation that a neuron performs on its inputs, i.e., the propagation and integration of synaptic input signals along the dendritic membrane up to the axon initial segment, the location where the neuronal output is typically generated. [5]

Neurons of different types serving different functions differ in the morphology and/or physiology of their dendrites. In fact, dendritic morphology is a defining feature of neuronal classes upon which neurons can be categorized. Up to now, dendrite morphology represents one of the main criteria for classification of neurons into individual types. [5]

In the normal development of a neuron a controlled growth and elaboration of its extensions (axons, dendrites, and synaptic connections) are fundamental for the setting up of a functional nervous system [6]. Any deficit or alteration caused by trauma in the programmed neural architecture leads to impaired functioning of the nervous system. For example in spinal cord injuries, there is a loss of the motor and sensory capabilities of the body. A successful post-traumatic repair of the nervous system implies the re-establishment of its functional connections
The connections between neurons, through axons and dendrites are a complex and diverse phenomenon that involves the morphologies that facilitate those connections to different types of targets. In many cases there is a regulated differentiation of neurites so that beneficial branches are elongated while aberrant branches are eliminated [6]. In other cases some trajectories are abruptly and purposefully eliminated once their collaterals have reached an appropriate target. This phenomenon has been documented and studied in a variety of developmental neural systems (See figure 4) [6].

**Figure 4** Time sequences showing branching and pruning of dissociated E11 chick dorsal root ganglion neurites. (a) Branching (red arrow) and extension (blue arrowheads) of primary axons. (b) Extension and retraction (blue arrowheads) of neurite tip. (c) Tertiary branching and pruning (encircled). Cultures are grown in the presence of glia in 5% CO2/37°C on Poly-L-lysine/laminin in N3 complete serum-free media. Phase-contrasts live imaging at 28 hrs post-plating. Time interval between acquisitions for each time series is as follows: (a) 30 mins, (b) 75 mins, (c) 75 mins. Snapshots are contrast enhanced for visual clarity of the neurites. [6]
1.2 Previous research in Mathematical and Computational Models of Neural Arborization

Mathematical and computational models of dendritic arborization have been used in previous studies. For example, quantitative models of the detailed branching patterns in dendritic trees have investigated the impact of network topology on firing patterns and neuronal signal processing [7–10].

Those models can be categorized in two groups: Growth Models and Reconstruction Models. Growth Models are based on principles of dendritic development, which use rules of outgrowth associated with dynamic growth-cone behavior, microtubule-mediated neurite elongation, and actin meshwork branch formation [11-14]. In contrast, Reconstruction Models use an algorithm based on a canonical set of elementary properties which are originally derived from the characterization of an existing dendritic structure [15,16]. Those neural arbor structures obtained using Reconstruction Models, although generated from minimalistic rules, are statistically indistinguishable from a sample of real neurons. A Reconstruction Model is a purely descriptive approach that uses minimal rules to “synthesize” topologically-realistic neurons. On the other hand Growth Models adopts an exploratory approach by using biological rules of development and observations of the outgrowth process to explain or predict variations in full-grown arbor structures [17].

The Osan et al model is a conceptually new Growth Modeling approach which incorporates a pruning function into the algorithm and evaluates the growth of the neurons in the context of a target-search problem [6]. Growth and Reconstructionist models focus on the finalized structure of a neuron, whereas the Osan et al new approach examines the evolution of a neuron through its time-steps of development and addresses the potential for its intermediate
morphologies to establish connections. As a result, the focus of the research shifted from faithfully mimicking the neural structures obtained in the in-vitro experiments to asking the question: how successful are neurons with similar growth properties in reaching their targets?

In Osan et al [6] the focus was to understand the functional role of the interplay between the neurites branching formation and branch elimination (pruning) processes on the optimization of a neural target search. In other words, they studied how the main parameters, especially branching and pruning probabilities, determine an optimal dendritic structure. The optimization problem they formulated for their work was this: given a neural tree of total constrained length \( L_{\text{max}} \), defined as the sum of the lengths of all of its branches, what is the fixed branching probability \( P_{\text{branch}} \) that would maximize the number of search sites out to a radius, \( D \), from the originating cell body? They defined the optimal neural tree as the one that maximizes the number of hits at the set distance \( D \), which is equivalent to increasing the chance of success for finding a single target located at distance \( D \) away from the origin. Correspondingly, the optimal class of neurons is the set of neural trees generated with the same set of parameters that on average achieve maximal performances at distance \( D \). [6]

To examine the targeting efficiency of growing neurites subject to limited resources Osan et al used a computational model [18] and theoretical tools. To simplify the model for the analytical derivation of their results, they assumed that the neurites grow in a straight line, bifurcate at fixed time intervals and branch at angles very close to zero, thus doubling the amount of search in the same spatial location, after each successful branching event.

They found that in order to efficiently reach a particular target, growing neurites must achieve a balance between branching and pruning since rapidly growing neurites that do not
prune will exhaust their resources, and frequently pruning neurites will fail to explore spatial distance effectively. They also found that the optimal branching/pruning balance must shift as the distance changes; thus demanding different strategies to reach nearby versus far away targets. They conclude that these results suggest the existence of a higher-level regulatory factor, still unidentified, that controls the neural arborization dynamics. They also were convinced that the manipulation of the neural arborization behavior could be useful in future neuro-regenerative therapies.

1.3 Objectives

While successful, the previous work of Osan et al has some shortcomings that will be addressed here. First of all, the model is mainly computational and many of the statistical results come from extensive numerical simulations. Second, that model did not produce analytical results that would enhance our understanding of the probability distribution of optimal targeting rates. Finally, although the models are qualitatively in agreement with the consensus in this area of research, no efforts were made to systematically match experimental data with the predictions of our model. We aim to address these issues in the current work.

In the present work we start with the computational model that is also used to examine how the average number of active neural branches changes as a function of distance. This is carried out for neurons following stochastic branching while subject to a maximum length constraint. We then examine what are the properties of this model, under the assumption of low branching angles. Our goal in this thesis was to prove the recurrence formula for the probability distribution of all possible neural trees, as well as the formulas of the expected number of active branches and their variances. We also proved the formula to determine the optimal region for a
tree with branching probability. These formulas were necessary for the simplified stochastic computational model that Osan et al have developed to examine how changes in branching probability influence the success of targeting neurons located at different distances away from a starting point.

It was found that the short distance targeting performances follow an exact geometric series law. In addition further refinements of this approximation, as well as another geometric series formula, were used to determine the location of the maximum of the probability distribution, which is the optimal targeting area. This is the area where, on average, a maximum number of branches are expected to be produced. The results obtained show that the model is in excellent agreement with experimental data.
2 METHODS AND RESULTS

2.1 Discrete Probability Distribution for the Evolving Family of Trees

In this work a simplified stochastic model is applied to investigate how the use of different branching probabilities influence the success of targeting neurons located at different distances from an initiation point. The rules for generating a simplified neural tree were described by Osan et al [6] as follows. Initially there is one active branch that at each discrete time step can split into two branches with probability $p$ or further extend with probability $q = 1 - p$. Furthermore, each active branch thus formed acts then independently of the others and can further extend or split out to create a new set of branches. As a result, multiple branching processes shape the structure of the resulting neural tree. In order to facilitate the statistical estimates for the resulting probability distribution, the spatial structure of these branches is ignored, and it is assumed that they evolve linearly after branching at small angles. These simplifications allow to obtain precise analytical results, while taking into account the most important feature of the neural tree that evolves stochastically, namely the random generation of extra neurites. The advantage of this approach is that for each step, the full probability distribution for each possible outcome can be generated. In Figure 5, reproduced from Osan et al 2011 [6], the process is illustrated up to three time steps.
The full probability distribution can be generated by examining all possible tree configurations that can be achieved after each time step. For example at $t = 3$, the simplest tree is a single evolving branch of length 4 that is obtained with a probability of $q^3$. At the opposite end of the spectrum, the most complex tree contains 8 active branches each of length 1, obtained with probability $p^7$. The associated probabilities can be determined by computing the products of individual probabilities along the arrows. Each tree has an associated probability and some of the trees listed have multiple replicates (shaded boxes).

An example of the probabilities thus generated is shown in Table 1 for the first two time steps. It is assumed here that the parameter that most reflects the probability of finding a target at a distance $R$ from the starting point is the number of active branches at that distance. Then the stepwise mean and variance of the expected number of active branches at a distance $R$ from the origin can be determined as a function of the branching probability $p$. The expected value and variance of the discrete probability distributions, listed in Table 1 for the first two time steps, are obtained using the following formulas:

$$\bar{x} = \sum_{i=1}^{n} x_i \cdot p_i$$

$$\sigma^2 = \sum_{i=1}^{n} (x_i - \bar{x})^2 \cdot p_i$$
Table 1 Average number of active branches and the associated variances for the first two time steps.

<table>
<thead>
<tr>
<th>Tree</th>
<th>Probability</th>
<th># active branches</th>
<th>Total length</th>
<th>Expected value</th>
<th>Variance σ²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 0</td>
<td></td>
<td></td>
<td></td>
<td>(1+p)⁰</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Step 1</td>
<td></td>
<td></td>
<td></td>
<td>(1 + p)¹</td>
<td>- p² + p</td>
</tr>
<tr>
<td>1</td>
<td>Q</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>P</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td></td>
<td></td>
<td></td>
<td>(1 + p)²</td>
<td>- p⁴ - 2p³ + p² + 2p</td>
</tr>
<tr>
<td>1</td>
<td>q-q</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>q-p</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>p-q²</td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>p·(2-p-q)</td>
<td>Two possible trees</td>
<td>3</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>p·p³ = p³</td>
<td>4</td>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It was noticed that the pattern could be expressed clearly with the formula

\[ E(n) = (1 + p)^n \] (1).

The formula gives the expected value of the active branches after the n time step. This formula is proved analytically using the discrete probability distribution of possible neural trees. The proof for the formula of the variance and the proof for the recurrence formula for the probability distribution of all possible neural trees are provided in the Methods section.
The expected values of active branches, as well as the associated variances, for the first three time steps, are listed in Table 2. The exact procedure of how to obtain the values for the expected value and the variance in step 2 of table 2 is shown in appendices A.1 and A.2.

<table>
<thead>
<tr>
<th>N</th>
<th>Expected Value</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 + p</td>
<td>- p^2 + p</td>
</tr>
<tr>
<td>2</td>
<td>(1 + p)^2</td>
<td>- p^4 - 2<em>p^3 + p^2 + 2</em>p</td>
</tr>
<tr>
<td>3</td>
<td>(1 + p)^3</td>
<td>- p^6 - 4<em>p^5 - 5</em>p^4 + p^3 + 6<em>p^2 + 3</em>p</td>
</tr>
</tbody>
</table>

### 2.2 Recurrence Formula for the Probability Distribution of all Possible Neural Trees

The results from the previous section can be proved using the discrete probability distribution function at any time step n. While these results are proven in the Methods section, we sketch here how to describe the evolution of all possible trees using a recursive function $f_n(p, q)$ as follows. For step zero we use $f_0(p, q) = 1$ to denote a single branch (see table 1, step 0). At step 1, $f_1(p, q) = p + q$, (see Table 1, step 1), indicates the existence of two possible trees. The first one has two active branches and probability of instantiation p, while the other one contains a single branch and can be instantiated with probability q. Note that since $p + q = 1$, the sum of all probabilities adds up to 1, as needed. At step 2, the existing trees can be described using:

$$f_2(p, q) = q * p + q * q + p * (p^2 + 2*p*q + q^2) = (q + p(p + q)) * (p + q).$$
These trees can be identified in the tree evolution diagram for steps 1-2 (see figure 5), and they are listed in Table 1 (step 2). Intuitively, a single-branch tree can evolve on two possible paths (extend or split) as indicated by the first two possible trees in step one (see figure 5). In the next step, the tree with two terminal branches can generate possible trees with 2, 3 and 4 terminal branches, respectively. It is now easy to show that the formula for step 2, \( f_2(p, q) \), can be obtained by substituting \( p \ast (p + q) \) for \( p \) in formula \( f_1(p, q) \) and multiplying the result by \( (p + q) \). We can now conjecture that we can generate all possible trees at step \( n \) using the following recurrence formula:

\[
\begin{align*}
    f_{n+1}(p, q) &= f_n(p \ast (p + q), q) \ast (p + q) \\
    &= f_n(p, q) \ast (p + q) \\
    &= f_1(p, q) \ast (p + q)
\end{align*}
\]

(2)

This formula is proved in the Methods section. Now, this explicit recursive description of the discrete probability distribution allows the computing of the expected number of active branches and the associated variance in the general case, extending the enumerative example showed in Table 1. As a result, when the number of steps is small enough such that all trees have total lengths below the maximum permitted value, before they run out of resources and cannot extend anymore, it can be proved that the expected value indeed follows the general formula \( E(n) = (1 + p)^n \). Furthermore, a recurrence formula for the associated variance is also derived:

\[
\begin{align*}
    Var(S_n) &= p \ast (1 - p) \ast (1 - p)^{n-1} \ast g(n) \\
    \text{where, } g(n + 1) &= p \ast g(n) + g(n) + 1, \text{ with } g(1) = 1 \\
    \text{This has the following solution: } Var_n(X) &= (1 + p)^{2(n-1)}(1 - p^2) + (1 + p)^{n-1}(p - 1) \quad n = 1, 2, ...
\end{align*}
\]

(3)
2.3 Describing Length Constraints

Obviously when the total length of the trees cannot exceed a set maximum value, some of the trees generated using formula (2) cannot be instantiated, as illustrated in Figure 6 for trees that cannot exceed a total length of 7.

![Figure 6 Diagram](image)

**Figure 6** All possible trees with lengths below a maximum value of 7. Trees that can no longer generate offspring have the terminal branches shown in red. For example the 3-terminal branches tree listed at stage 3 has a total length of 6 already. Therefore, even if all its terminal branches merely extend the resulting tree will have a length of 9, exceeding the maximal possible value of 7. Other trees will have even larger total lengths.

As a result, the expected geometrical growth is possible only near the origin, as the densest possible trees start exceeding length constraints at larger distances. In fact the growth slows down further away, reaches a peak value, and achieves a longer tail of decaying values corresponding to neural trees that seldom branch, as shown in Figure 7, again for trees that cannot exceed a total length of 7.
Figure 7 Expected number of branches as a function of distance for families of trees with maximum length 7 and branching probability $p = 0.5$. Close to the origin, the expected value of searching sites increases as a geometrical series (step 2). However, this picture changes at large distances from the origin. There is an optimal targeting region, indicated by the peak at step 3, and a long tail of decaying values for steps 4 to 7. There are no possible trees for steps 8 or larger since a tree of maximal size of 7 cannot reach beyond a total distance of 7. As a matter of fact, only one tree, the one that does not branch at all and has a probability $q^6$, will reach a distance $d = 7$ with only one active branch.

As illustrated in Figure 7, the probability distribution for all possible instantiation of neural trees can be used to determine the expected search performances of trees of maximum length $L$. The coverage area extends from $x = 1$ to $L$. The initial rise in performances follows a geometrical series trend that is determined by the exact value of the branching parameter $p$. The performance at large distances away from the origin has a long tail, corresponding to the decreasing number of trees that can make it further and further. In terms of the exact values of branching probability, trees that branch often will tend to cover the nearby area well and will have very small chances of extending far away. In contrast, trees that seldom branch will have a
much improved chance to explore farther in space, albeit concentrating in sending a small number of branches as far as possible. It is then intuitive that the optimal targeting region is determined by the exact value of the branching probability parameter \( p \), as illustrated in Figure 8 that shows the average number of dendrites at different time steps, for different values of probability of branching \( p \). This is essentially the equivalent of a Sholl plot for neural trees [19].

![Figure 8 Sholl plots for different branching probabilities \( p \). Expected number of dendrites at each time step is plotted for a tree of maximum possible length of 15, for the branching probabilities in the set \( \{0, 0.2, 0.4, 0.6, 0.8, 1\} \). The tree with \( p = 0 \), showed in blue, simply extends until it runs out of resources, and generates a flat line of 1 expected branch at each time step. In contrast, the tree with \( p = 1 \), doubles the number of branches at each time step and runs out of resources at step 3, achieving a maximum at the time it stops. In between these two extremes, families of trees shift their optimal targeting regions farther away as the branching probability decreases, at the cost of reducing the overall amplitude of the corresponding peak values (success rates).

2.4 Approximating the Location of Optimal Targeting Performances

In their work, Osan et al derived an estimate for the location of optimal targeting
performances, by examining a tree with neurites that branch periodically after growing a fixed length $L_0$, which is determined by the branching probability $p$. After computing how $L_0$ depends on $p$, the equation that determines the optimal targeting region after $n$ steps can be set up:

$$n = \frac{\ln(1+pL)}{\ln(1+p)} - 1$$  \hfill (4)

The derivation of this equation is listed in the Methods Section. This equation determines the optimal targeting region for a tree with branching probability $p$. Note that here $n$ is a continuous function of $p$, since the branching probability can take continuous values. This ‘naïve’ approximation is in very good agreement with the results from the discrete probability distribution obtained by instantiating all possible trees that do not exceed the maximal value $L$ up to time step $n$, as shown in Figure 9.
Figure 9 Comparison of naïve vs exact results for a tree of maximum length \( L = 31 \).

We use the explicit discrete probability distribution to determine the optimal targeting regions, shown as blue circles, while the red line is given by equation \( n = \frac{\ln(1+pL)}{\ln(1+p)} - 1 \). For the discrete distribution, since the total length of the tree is 31, there are enough resources for all possible tree until step 4, therefore the optimal branching probability is \( p = 1 \), corresponding to creation of most dense trees. As a result, steps 1-4 are excluded from the comparison. The blue points are computed using the explicit symbolic equation (polynomials in \( p \)) for the expected value \( s_k \) at different locations \( k < n \). Then the solution for the probability \( p \), that maximizes the expected value \( s_k(p) \) is computed and plotted at location \( k \) as a blue circle. Values for \( p \) are restricted between 0 and 1, to exclude other potential solutions for the polynomial equations. The graph indicates that while the red curve obtained from the ‘naïve’ theoretical expectation, over-estimates the analytical expected results, these curves are in agreement and exhibit the same trends.

2.5 Methods for Neuronal Culture and Analysis

To determine if the models were a good fit for experimental results, data from Firestein Lab at Rutgers University was used. The experimental procedures employed were as follows:
2.5.1 Cell Culture, Transfection, and Immunostaining:

Hippocampal neurons were isolated from embryonic rats at 18 days of gestation (E18) as previously described by Firestein [22] and Carrel [23]. Briefly, the hippocampi from Sprague Dawley rats were isolated and mechanically dissociated. Hippocampal neurons were then plated on poly-D-lysine (PDL)-coated glass coverslips (12 mm diameter) at a density of approximately 1800 cells/mm². Cells were cultured in Neurobasal medium supplemented with B27, Glutamax, and 1% penicillin/streptomycin (Life Technologies). At 5 days in vitro (DIV), neurons were transfected with cDNA encoding green fluorescent protein (GFP) in the pEGFP-C1 vector (Clontech) using Lipofectamine 2000 (Invitrogen). GFP is expressed throughout the entire neuron and ensures accurate assessment of dendrite number. At 7 DIV, immunostaining was performed to enhance the natural fluorescence of GFP. Neurons were fixed in 4% paraformaldehyde (PFA), after which they were incubated in blocking buffer. Primary antibody incubation (1:1000 dilution of rat anti-GFP from Dr. Shu-Chan Hsu of Rutgers University) occurred at 4°C overnight. Coverslips were washed 3 times with PBS and then incubated with secondary antibody (1:250 dilution of Cy2-anti rat IgG from Jackson Immunoresearch) for 1 hour at room temperature. After immunostaining was complete, coverslips were mounted onto glass microscope slides.

2.5.2 Imaging and Assessment of Dendrite Number

Transfected cells were imaged using an Olympus Optical IX50 microscope with a Cooke SensiCam charge-coupled device (CCD) cooled camera fluorescence imaging system and ImagePro software (Media Cybernetics). All images were taken at 200x magnification.

Images were processed as previously described by Kutzing et al [24] and Langhammer et al [25] using customs scripts written in Matlab (MathWorks). Briefly, cell bodies and dendrites
were traced in ImageJ using the NeuronJ plugin (NIH, Bethesda, MD). The data were exported to NeuronStudio and checked to ensure proper connectivity of dendrites. Sholl analysis was then performed at 6 μm intervals starting at 0 or 9.33 μm from the soma using the Bonfire program [24,26].

Sample neurons are provided in Figure 10A-B. We performed Sholl analysis for the family of neurons used here in order to generate the targeting profiles. We fitted our models to these profiles, by allowing changes in the following parameters: spatial distance covered in between two potential branching events (width of the distribution), number of initial branches (peak of the distribution) and branching probability/total length. We were able to produce very accurate fits for the Sholl analysis curves (Fig 11), which indicate that our model is in very good agreement with the experimental data.

![Two Samples Neurons A-B](image)

**Figure 10** Two Samples Neurons A-B
2.6 Analytical Methods

2.6.1 Derivation of the Recurrence Formula for all Possible Neural Trees

We want to prove the formula $f_{n+1}(p, q) = f_n(p \cdot (p + q), q) \cdot (p + q)$. We write the formula for step $n$ as $f_n(p, q) = \sum_{k=0}^{2^n-1} p^k g_k(q)$ where $g_k(q)$ are polynomials in $q$. Obviously, this is true for the first 2 time steps considered in Table 1. Each individual term in the $f_n(p, q)$, of the form $p^k g_k(q)$ represents the probability of generating a tree with $k + 1$ active branches. Some of the trees are isomorphs, and for simplicity that will be reflected in the coefficients from $g_k(q)$.

In order to derive the general formula for $n + 1$ step, we note that the trees that contain a $p^k$ term have $k + 1$ terminal (active) branch (see Table 1). Then, at the next step, these active branches can generate between 0 and $k + 1$ new active branches. Taking into account also the

Figure 11 Arborization profiles for two sample neurons (A and B) result in typical profile curves (blue curve) for the family of these neurons ($n = 30$). Fits of the probabilistic neural growth model are in excellent agreement (red curve).
isomorph trees, the probabilities for these new branches are described by the combinations from the \((p + q)^{k+1}\) formula. We then obtain

\[
f_{n+1}(p, q) = \sum_{k=0}^{2^n-1} p^k(p + q)^k g_k(q) = (p + q) \sum_{k=0}^{2^n-1} p^k(p + q)^k g_k(q)
\]

\[
= (p + q) \sum_{k=0}^{2^n-1} (p(p + q))^k g_k(q) = (p + q) \cdot f_n(p \cdot (p + q), q)
\]

which proves formula (2).

### 2.6.2 Derivation of the Formula for the Expected Number of Active Branches

\(E(b_n) = (1 + p)^n\)

We want to prove that the expected number of active branches at step \(n\) is \((1 + p)^n\). In order to achieve this goal, we need to prove the following intermediate steps. First, we can determine the expected number of active branches at step \(n\) as the derivative of the \(f_n \cdot p\) function: \((f_n(p, q) \cdot p)'\), where the symbol ‘\(\cdot\)’ stands for derivative with respect to \(p\). Since the terms in the sum of \(f_n(p, q)\) are all the entries in the global probability table (see Table 1), we have now the expected number of active branches at time step \(n\) (and distance \(n + 1\) away from the origin) to be:

\[
E(b_n) = \sum_{k=\text{all trees}} \text{number active branches on tree} \cdot \text{tree probability}
\]

Remember from above that:

\[
f_n(p, q) = \sum_{k=0}^{2^n-1} p^k g_k(q)
\]
We rewrite

\[ E(b_n) = \sum_{k=0}^{2^n-1} (k + 1)p^k g_k(q) = \left( \sum_{k=0}^{2^n-1} p^{k+1} g_k(q) \right)' = \left( p \cdot \sum_{k=0}^{2^n-1} p^k g_k(q) \right)' = (p \cdot f_n(p, q))' \]

Using these intermediate results, the expected number of active branches evaluates as:

\[(1 + p)^n, \text{ as proved by induction below.}\]

\[ E(b_n) = (1 + p)^n \]

\[ E(b_{n+1}) = (p \cdot f_{n+1}(p, q))' = \left( p \cdot \sum_{k=0}^{2^n-1} p^k (p + q) g_k(q) \right)' = \left( \sum_{k=0}^{2^n-1} p^k (p + q) g_k(q) \right)' = \sum_{k=0}^{2^n-1} (k + 1)(2p + q)(p \cdot (p + q))^k g_k(q) \]

\[ = \sum_{k=0}^{2^n-1} (k + 1)(1 + p)(p \cdot (p + q))^k g_k(q) \]

\[ = (1 + p) \sum_{k=0}^{2^n-1} (k + 1)(p \cdot (p + q))^k g_k(q) = (1 + p) \sum_{k=0}^{2^n-1} (k + 1)p^k g_k(q) \]

\[ = (1 + p)(p \cdot f_n(p, q))' = (1 + p)^{n+1} \]

Furthermore, we can extend this approach and compute the variance for the expected number of active branches, using \( \text{var}(X) = E(X^2) - E(x)^2 = ((f_n(p, q)*p)' * p)' - ((1 + p)^n)^2 \)

After simplifications, this can also be written as a recursion formula

\[ \text{Var}(S_n) = p \cdot (1 - p) \cdot (1 - p)^{n-1} \cdot g(n) \]
\[ g(n + 1) = p \cdot g(n) + g(n) + 1 \]

\[ g(1) = 1 \]

And we also have the general formula of the variance:

\[ \text{Var}_n(X) = (1 + p)^2(n-1) \cdot (1 - p^2) + (1 + p)^{n-1} \cdot (p - 1) \cdot n = 1, 2, .... \]

These proofs can also be derived using theorem 4.4 of Allen’s book [20] as shown below:

### 2.6.3 Derivation of the Formula for the Variance of the Expected Number of Active Branches

**Branches**

Theorem 4.4 proved by Bailey [21] gives the mean and variance of branching processes when \( x_0 = 1 \), that is, when the process begins with one individual. In the model created by Osan et al the process begins with one active branch, so it is natural to employ this theorem. The theorem uses the properties of probability generating functions together with first order difference equations (because of the necessity of the recurrence of the branching process).

\[ f(t) = P_0 + P_1 t + P_2 t^2 \]

\[ f'(t)|_{t=1} = 0 + P_1 + 2P_2 t \]

\[ m_1 = P_1 + 2P_2 \]

\[ m_2(n) = (m_1)^n = (m)^n = (P_1 + 2P_2)^n \]

Now, \( \sigma^2 = [f''(t) + f'(t) - (f'(t))^2] |_{t=1} \)

\[ f''(t) = 2P_2 \]

Then, \( \sigma^2 = 2P_2 + (P_1 + 2P_2) - (P_1 + 2P_2)^2 \)

Now, \( P_1 + 2P_2 \) where \( P_1 = q \) and \( P_2 = p \)

We have \( P_1 + 2P_2 = (1 - p) + 2p = 1 + p \)
Using this result in:

$$\sigma^2 = 2P_2 + (P_1 + 2P_2) - (P_1 + 2P_2)^2 = 2p + (1 + p) - (1 + p)^2$$

$$= 2p + (1 + p) - [(1 + p)(1 + p)] = 2p + (1 + p) - (1 + 2p + p^2)$$

$$= 2p + 1 + p - 1 - 2p - p^2 = p - p^2 = p(1 - p)$$

From Theorem 4.4 page 173 of Allen’s book [20] we have that

$$\sigma^2(n) = \frac{m^{n-1}(m^n - 1)}{m - 1} * \sigma^2 \ m \neq 1$$

Using Allen’s formula we have:

$$\sigma^2(n) = \frac{(1 + p)^{n-1}[(1 + p)^n - 1]}{(1 + p) - 1} * p(1 - p) = ((1 + p)^{n-1}[(1 + p)^n - 1]) * (1 - p)$$

$$= [(1 + p)^{2n-1} - (1 + p)^{n-1}] * (1 - p) = (1 + p)^{2n-1}(1 - p) - (1 + p)^{n-1}(1 - p)$$

$$= (1 + p)^{2n-1}(1 - p) + (1 + p)^{n-1}(p - 1)$$

And since

$$(1 + p)^{2n-1}(1 - p) = (1 + p)^{2n-2}(1 + p)(1 - p) = (1 + p)^{2(n-1)}(1 - p^2)$$

We have:

$$(1 + p)^{2n-1}(1 - p) + (1 + p)^{n-1}(p - 1) =

(1 + p)^{2(n-1)}(1 - p^2) + (1 + p)^{n-1}(p - 1)\text{ with } n = 1, 2, ...$$

2.6.4 Optimal Targeting Region for Trees with Size Restrictions

At larger distances from the origin, some trees should be eliminated from the analysis. For example, if the maximum length of the tree is equal to 7, (as shown in the Figure 6), the larger trees will not generate subtrees after time step 3. However the smaller ones will do, and the formulas for the number of searching sites at distance n, that are allowed for a tree of maximum size 7 are shown in table 3.
### Table 3 Maximum length of the tree is equal to 7

<table>
<thead>
<tr>
<th>Step 1</th>
<th>p + 1</th>
<th>enough resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>(p + 1)^2</td>
<td>enough resources</td>
</tr>
<tr>
<td>Step 3</td>
<td>(ps - 1)^2*(2<em>ps^2 + 5</em>ps + 1)</td>
<td>some trees are eliminated</td>
</tr>
<tr>
<td>Step 4</td>
<td>-(ps - 1)^3*(-2<em>ps^2 + 3</em>ps + 1)</td>
<td>some trees are eliminated</td>
</tr>
<tr>
<td>Step 5</td>
<td>(ps - 1)^4*(ps + 1)</td>
<td>the one that does not branch at all and the one that branches after 4 steps</td>
</tr>
<tr>
<td>Step 6</td>
<td>(ps - 1)^6</td>
<td>only the tree that does not branch reaches this far</td>
</tr>
</tbody>
</table>

#### 2.6.4.1 Determining the Approximation for the Peak Location

The condition, shown in Fig.12(c) from Osan et al [6], that all neurite tips reach a target distance D translates to: 

\[ D = L_0 N_{\text{max}} = L_0 \log_2(1 + L_{\text{max}}/L_0). \]

![Comparison of equal length trees under different branching scenarios](image)

**Figure 12** Comparison of equal length trees under different branching scenarios: 
(a) Undershoot, for \(L_0 = 1\) units and fixed cumulative neurite length \(L_{\text{max}}\); branches fail to reach targets in gray band; (b) Overshoot, for \(L_0 = 4\); branches reach the gray band but hit sparsely due to overextension past the target zone; (c) Optimal run, for \(L_0 = 2\); number of targets hit in the gray band is maximized. (First row of Figure 3 from Osan et al) [6]

(5)
Now we can set up the comparison: $D = L_0 N_{\text{max}} = L_0 \log_2(1 + L_{\text{max}}/L_0)$. Using the geometrical series result for the average number of active branches at time step $k$ that is $(1 + p)^k$ we obtain the total length of the tree after completion of $n$ steps:

$$L = 1 + (1 + p) + (1 + p)^2 + \cdots + (1 + p)^n = \sum_{k=0}^{n} (1 + p)^k = \frac{(1 + p)^{n+1} - 1}{1 + p - 1}$$

$$= \frac{(1 + p)^{n+1} - 1}{p}$$

Setting up the condition that the tree stops expanding when reaches $L$ we get:

$$L = \frac{(1 + p)^{n+1} - 1}{p}.$$

We can now compute $p$ from this equation: $p.L = (1 + p)^{n+1} - 1$, or equivalently $1 + p.L = (1 + p)^{n+1}$ Taking logarithms on both sides: $\ln(1 + pL) = (n + 1)\ln(1 + p)$, or equivalently $n + 1 = \frac{\ln(1 + pL)}{\ln(1 + p)}$. We finally obtain: $n = \frac{\ln(1 + pL)}{\ln(1 + p)} - 1$.

Note that when $p \to 0$, the single-branch tree expands all the way to $x = N$ (in $N - 1$ time steps) since

$$\lim_{p \to 0} n = \lim_{p \to 0} \left( \frac{\ln(1 + pL)}{\ln(1 + p)} - 1 \right) = L'Hopital's \ Rule = \lim_{p \to 0} \left( \frac{\ln'(1 + pL)}{\ln'(1 + p)} \right) - 1$$

$$= \lim_{p \to 0} \left( \frac{L}{1 + p} \right) - 1 = L - 1$$

Because the probability of this tree is $f(p) = (1 - p)^{N-1}$, function $f$ is obviously maximized at $p = 0$. Not surprisingly, the symbolic/numeric evaluation yields the same result. At the other
end of the spectrum, when $p \to 1$, the tree expands as $1 + 2 + 4 + 8 + 16 + \ldots = N$ and runs out of resources. As an example when $L = 31$, we obtain:

$$n = \frac{\ln(1 + 1L)}{\ln(1 + 1)} - 1 = \frac{\ln(1 + L)}{\ln(2)} - 1 = \ln_2(1 + L) - 1 = \ln_2(1 + 31) - 1 = 5 - 1 = 4$$

3 CONCLUSION

Similar to the computational model of Osan et al, the analysis of the discrete probability distribution corresponding to the stochastic neural growth model used here indicates that the most important component of the model, namely the branching probability $p$, has a clear impact on targeting performances. More precisely, the expected success near the origin increases as a geometrical series with a factor of $(1 + p)$. In contrast with previous work that offered a limited number of results obtained through numerical simulations, in addition to computing the values of expected number of branches as a function of distance, analytical expressions for the corresponding variance values and approximations for the optimal targeting regions were also determined. These results provide enhanced intuition on how the family of trees explores nearby areas and quantifies how changes in branches probability will impact the targeting performances at short distances.

The model also suggests that the log of active branches is a more appropriate way of constructing Sholl plots, as the average of the neural trees population is predicted to have a linear component near the origin. The use of this component can provide a better statistical determination of branching values for the linear regression used in the analysis of experimental data. It is possible that experimental data shows deviations from these trends. If this is the case, this will provide clear avenues for changing the model to account for newfound trends.

We enhanced the scope of our research by showing that the probabilistic models used in
this work were in excellent agreement with the early-stage experimental data from Firestein Lab, and indicates that the parameters used in the theoretical model captured most of the variability seen in the experimental data. The implications of the study will naturally lead to the use of these models to derive statistical test for differences between different culture types, such as normal versus the ones treated with chemical growth substances such as Cypin and BDNF; these tests are meant to replace the less informative methods such as analysis of variance (ANOVA) that are currently used.

The limitations of the study include the fact that the adequacy of the computational model was analyzed for short distances for the mean and variance of number of branches, and for intermediate distances for the formula that determines the optimal targeting region. Osan et al are currently working on the extension of this model for long-distance targeting performances.

Our model however does not account for late-stage experimental results, where intermediate-range refinements seem to take place. This suggest that our simple uniform branching model cannot account for all aspects of the neural growth in vitro and it will need to be further refined. Osan et al plan to test if the use of parameter-based branching can account for these changes. This will provide an opportunity to advance a hypothesis about the role that growth-inducing chemicals such as BDNF and Cypin have in changing the architecture of the neural tree. Furthermore, we will investigate if uniform and parameter-based pruning can lead to further improvement of these models.

Our model provides an initial effort to characterize the growth of neural trees in laboratory conditions. As such it contains a minimal number of assumptions: uniform branching probability, fixed time branching intervals and maximal length of the tree. It is surprising that our minimal models, with this small number of assumptions, can be used to obtain excellent fits
of experimental data. The simplicity of our model allows us to obtain analytical result for the expected number of branches and their variability, which are in very good agreement with experimental data and results from the more complex computational model. These results are very important because of the specific, non-linear, predictions that can validated by experimental data in the future. If that is not the case, this will provide clear directions for improvement for our models.
REFERENCES


APPENDICES

Appendix A Calculations

Appendix A.1 Obtaining the Expected Value for Step 2

In step 2 we have 6 possible trees and 15 active branches (see table 2).

So we have: \(1qq + 2qp + 2pq + 3ppq + 3pqp + 4ppp\)

Letting \(q = 1 - p\) we have:

\[
1(1 - p)(1 - p) + 2p(1 - p)(1 - p) + 3p^2(1 - p) + 3p^2(1 - p) + 4p^3
= (1 - 2p + p^2) + 2p(1 - p) + 2p(1 - 2p + p^2) + 3p^2(1 - p) + 3p^2(1 - p) + 4p^3
= 1 - 2p + p^2 + 2p - 2p^2 + 2p - 4p^3 + 2p^3 + 3p^3 - 3p^3 + 3p^2 - 3p^2 + 4p^3
= 1 + 2p + p^2 = (1 + \ p)^2
\]

Appendix A.2 Obtaining the Variance for step 2 \(n=2\)

Using: \((1 + p)^2(n-1)(1 - p^2) + (1 + p)^{(n-1)}(p - 1)\)

\[
= (1 + p)^2(2-1)(1 - p^2) + (1 + p)^{(2-1)}(p - 1) = (1 + p)(1 + p)(1 - p^2) + (1 + p)(p - 1)
= (1 + 2p + p^2)(1 - p^2) + p - 1 + p^2 - p = 1 + 2p + p^2 - p^2 - 2p^3 - p^4 + p^2 - 1
= 2p + p^2 - 2p^3 - p^4 = -p^4 - 2p^3 + p^2 + 2p
\]

Appendix A.3 Proof of Theorem 4.4

Prerequisites for the proof

Probability Generating Functions (PGF) can be used to find the mean and variance of \(X_n\), the random variable for the total population size in generation n.

The properties of a PGF:
The f(t), moment generating function (m.g.f), $M(t)=f(e^t)$, and the cumulant generating function (c.g.f), $K(t)=\ln M(t)$, satisfy the following:

\[
\begin{align*}
\frac{d}{dt}f(1) &= 1, \\
\frac{d}{dt}f'(1) &= m = E(X), \\
\frac{d}{dt}f''(1) &= E(X(X-1)), \\
M(0) &= 1, \\
M'(0) &= m, \\
M''(0) &= E(X^2)
\end{align*}
\]

\[
\begin{align*}
K(0) &= 0, \\
K'(0) &= m, \\
K''(0) &= \sigma^2 = E[(X-m)^2]
\end{align*}
\]

We denote the mean and the variance of $X_n$ as $m_n$ and $\sigma_n^2$, respectively, and the three generating functions associated with $X_n$ as $f^n(t), M_n(t), and K_n(t)$, respectively.

In the first generation, the random variable $X_1$ has mean $m_1 = m$ and variance $\sigma_1^2 = \sigma^2$.

That is,

\[
\begin{align*}
m_1 &= m = \sum_{k=1}^{\infty} kp_k \\ \sigma_1^2 &= \sigma^2 = \sum_{k=1}^{\infty} k^2 p_k - m^2
\end{align*}
\]

The three generating functions of $X_1$ are: $f' = f$, $M_1 = M$, and $K_1 = K$ respectively.

The following theorem gives the mean and variance of the following process when $X_0 = 1$ (beginning with one individual).

**Theorem 4.4**

Let $\{X_n\}_{n=0}^{\infty}$ be a branching process. Assume $X_0 = 1$ the mean of the random variable $X_n$ is: $m_n = E(X_n) = m^n$ and the variance is: $\sigma_n^2 = E[(X_n - m_n)^2] = \begin{cases} \frac{m^{n-1}(m^{n-1})}{m^{n-1}} \sigma^2, & m \neq 1 \\ n\sigma^2, & m = 1 \end{cases}$

Before the proof is given, let’s recall some properties of the three generating functions:

\[
M_n(t) = f^n(e^t) = f^{n-1}(f(e^t)) = f^{n-1}(M(t)) = f^{n-1}(e^{lnM(t)}) = M_{n-1}(lnM(t))
\]
Thus, $M_n(t) = M_{n-1}(K(t))$. Taking natural logarithms of this later identity lead to

$$Kn(t) = K_{n-1}(K(t))$$

The first and second derivatives of the preceding identity yields two relationships that are used to verify Theorem 4.4

$$K'_n(t) = K'_{n-1}[K(t)](K'(t)) \quad (4.10)$$

$$K''_n(t) = K''_{n-1}(K(t))[K'(t)]^2 + K'_{n-1}(K(t))K''(t) \quad (4.11)$$

Proof of Theorem 4.4 (The proof follows Bailey (1990) [21]).

The first identity (4.10) is evaluated at $t = 0$

$$K'_n(0) = K'_{n-1}[K(0)](K'(0))$$

$$m_n = (m_{n-1})m$$

Because

$$K_n(0) = 0, K'_n(0) = m_n \text{ and } m_1 = m$$

The Equation

$$m_n - mm_{n-1} = 0$$

is a first order, homogeneous, constant coefficient, difference equation in $m_n$.

The solution is $m_n = m^n$.

The second identity (4.11) is evaluated at $t = 0$

$$K''_n(0) = K''_{n-1}(K(0))[K'(0)]^2 + K'_{n-1}(K(0))K''(0)$$
\( \sigma_n^2 = \sigma_{n-1}^2 m^2 + m_{n-1} \sigma^2 \n\)

Substituting \( m_{n-1} = m^{n-1} \), then \( \sigma_n^2 - m^2 \sigma_{n-1}^2 = m^{n-1} \sigma^2 \) which is a first order, nonhomogeneous, constant coefficient, difference equation in \( \sigma_n^2 \). The general solution to this difference equation is a sum of the general solution to the homogeneous equation and a particular solution. The general solution to the homogeneous equation is \( cm^{2n} \). Assume the particular solution has the form \( \sigma_n^2 = Km^{n-1}, m \neq 1 \). Substituting this value into the difference equation yields:

\[ m^{n-1} [K - Km - \sigma^2] = 0 \]

Which leads to

\[ K = \frac{\sigma^2}{1 - m} \text{ provided } m \neq 1 \]

The general solution to the nonhomogeneous difference equation is:

\[ \sigma_n^2 = cm^{2n} + \frac{\sigma^2 m^{n-1}}{1 - m}, m \neq 1 \]

The constant \( c \) is found by setting \( \sigma_1^2 = \sigma^2 \) then \( c = \frac{\sigma^2}{[m(m-1)]} \). The solution is:

\[ \sigma_n^2 = \frac{m^{n-1}(m^n - 1)}{m - 1} \sigma^2, m \neq 1 \]

In the case \( m = 1 \), the particular solution has the form \( Kn \) . Substitution of this solution into the difference equation yields:

\[ Kn - K(n - 1) = \sigma^2 \text{ or } K = \sigma^2. \]
The general solution to the nonhomogeneous difference equation is \( \sigma_n^2 = c + n\sigma^2 \).

Application of \( \sigma_1^2 = \sigma^2 \) yields \( c = 0 \). Thus, the solution to the difference equation is

\[ \sigma_n^2 = n\sigma^2, \quad m = 1. \]

The proof is complete.

**Appendix A.4 Complementary steps for the proof of theorem 4.4**

\( M_n(t) = M_{n-1}(\ln M(t)) \) But \( \ln M(t) = K(t) \)

Thus, \( M_n(t) = M_{n-1}(K(t)) \). Taking natural logarithms

\[ \ln M_n(t) = \ln M_{n-1}(K(t)) \]

\[ K_n(t) = K_{n-1}(K(t)) \]

The first and second derivatives of this last identity yield:

\[ K_n(t) = K_{n-1}(K(t)) \]

Let \( K_{n-1} = f \) and \( K(t) = g \) we use \[ f(g(t))' = f'(g(t)) \cdot g'(t) \]

\[ K'_n(t) = K'_{n-1}[K(t)].(k'(t)) \quad 4.10 \]

For the second derivative:

\[ [K'_n(t)]' = [K'_{n-1}(K(t)) \cdot K'(t)]' \Rightarrow K''_{n-1}(K(t)) \cdot K'(t) \cdot K'(t) + K'_{n-1}(K(t)) \cdot K''(t) \]

\[ K''(t) \]

Thus,

\[ K''_n(t) = K''_{n-1}(K(t))[K'(t)]^2 + K'_{n-1}(K(t))K''(t) \quad 4.11 \]

Evaluating identity 4.10 at \( t = 0 \) we have:

\[ K'_n(0) = K'_{n-1}[K(0)].(k'(0)) \]

But \( K(0) = 0, K'(0) = m, K'_n(0) = m_n, \) and \( K'_{n-1}(0) = m_{n-1} \)

By substitution we get:

\[ m_n = m_{n-1}m \]

The equation \( m_n - mm_{n-1} = 0 \) is a first order, homogeneous, constant coefficient, difference equation in \( m_n \) and it is solved recursively.
\[ m_n - mn_{n-1} = 0, \text{ then:} \]
\[ m_n = m(m_{n-1}) = m(m(m_{n-2})) = m(m(m(m_{n-3})) \ldots m^{n-1}m_1 = m^n \]

(Note that: \( m^{n-1}m_{n-(n-1)} = m^{n-1}m_1 \) and \( m_1 = m = m^1 \). So, \( m^{n-1}m_1 = m^{n-1+1} = m^n \))

Evaluating identity 4.11 at \( t = 0 \) we have:
\[ K''(0) = K''_{n-1}(K(0))[K'(0)]^2 + K'_{n-1}(K(0))K''(0) \]

Since: \( K''(0) = \sigma^2, K(0) = 0, K'(0) = m, \) then \( K''_{n-1}(K(0)) = \sigma^2_{n-1}, [K'(0)]^2 = m^2 \) and 
\[ K'_{n-1}(K(0)) = m_{n-1} \]

By substitution we have:
\[ \sigma^2_n = \sigma^2_{n-1}m^2 + m_{n-1}\sigma^2 \]

Substituting \( m_{n-1} = m^{n-1} \) (By Eq. in Diff.)

We have: \( \sigma^2_n - m^2\sigma^2_{n-1} = m^{n-1}\sigma^2 \).

This is a first order, non-homogeneous, constant coefficient, difference equation in \( \sigma^2_n \).

The general solution to this equation is a sum of the general solution to the homogeneous equation and a particular solution. The general solution to the homogeneous equation is \( cm^{2n} \).

Assume the particular solution has the form \( \sigma^2_n = K m^{n-1}, m \neq 1 \) then, by substitution into the difference equation:
\[ \sigma^2_n - m^2\sigma^2_{n-1} - m^{n-1}\sigma^2 = 0 \]

We have:
\[ Km^{n-1} - m^2\sigma^2_{n-1} - m^{n-1}\sigma^2 = 0 \]

As \( \sigma^2_n = Km^{n-1} \Rightarrow \sigma^2_{n-1} = Km^{n-2} \Rightarrow m^2\sigma^2_{n-1} = m^2(Km^{n-2}) = Km^{n-2}m^2 = Km^{n-2+2} = Km^n \]
\[ Km^{n-1} - Km^n - m^{n-1}\sigma^2 = 0 \]
\[ m^{n-1}[K - Km^1] = m^{n-1}\sigma^2 \]
\[
m^{n-1} [K - Km] = \sigma^2
\]

\[
K - Km = \sigma^2
\]

\[
K(1 - m) = \sigma^2
\]

\[
K = \frac{\sigma^2}{1 - m} \text{ provided } m \neq 1
\]

The general solution to the non-homogeneous difference equation is:

\[
\sigma_n^2 = cm^{2n} + Km^{n-1}, m \neq 1
\]

Then \[
\sigma_n^2 = cm^{2n} + \frac{\sigma^2 m^{n-1}}{1 - m}
\]

The constant \( c \) is found by setting \( \sigma_1^2 = \sigma^2 \)

In particular \( n = 1 \)

\[
\sigma_1^2 = cm^{2(1)} + \frac{\sigma^2 m^{(1)-1}}{1 - m} \Rightarrow \sigma^2 = cm^2 + \frac{\sigma^2}{1 - m}
\]

\[
\sigma^2 - \frac{\sigma^2}{1 - m} = cm^2
\]

\[
\frac{\sigma^2 (1 - m) - \sigma^2}{1 - m} = cm^2
\]

\[
\frac{\sigma^2 (1 - m) - \sigma^2}{m^2 (1 - m)} = c
\]

\[
\frac{\sigma^2 (1 - m - 1)}{m^2 (1 - m)} = c
\]

\[
\frac{\sigma^2 (-m)}{m^2 (1 - m)} = c
\]

\[
\frac{-\sigma^2 m}{m^2 (1 - m)} = c
\]

\[
\frac{\sigma^2}{m(m - 1)} = c
\]

Then by substitution
\[\sigma_n^2 = \frac{\sigma^2 m^{2n}}{m(m - 1)} + \frac{\sigma^2 m^{n-1}}{1 - m}\]

\[\sigma_n^2 = \frac{\sigma^2 m^{2n} m^{-1}}{m - 1} - \frac{\sigma^2 m^{n-1}}{m - 1}\]

\[\sigma_n^2 = \frac{\sigma^2 m^{2n-1}}{m - 1} - \frac{\sigma^2 m^{n-1}}{m - 1}\]

\[\sigma_n^2 = \frac{\sigma^2 m^{2n-1} - \sigma^2 m^{n-1}}{m - 1}\]

\[\sigma_n^2 = \frac{m^{n-1}(\sigma^2 m^n - \sigma^2)}{m - 1}\]

\[\sigma_n^2 = \frac{m^{n-1}(m^n - 1)}{m - 1} \sigma^2 \quad m \neq 1\]

If \(m = 1\) and \(\sigma_n^2 = K m^{n-1}\) then \(K (\frac{1 + 1 + 1 + \ldots + 1}{n \text{ times}}) = Kn\)

Hence, by substitution:

\[\sigma_n^2 - m^2 \sigma_{n-1}^2 - m^{n-1} \sigma^2 = 0\]

\[Kn - (1)^2 K_{n-1} - \sigma^2 = 0\]

\[Kn - K(n - 1) - \sigma^2 = 0\]

\[Kn - Kn + K - \sigma^2 = 0\]

\[K - \sigma^2 = 0\]

\[K = \sigma^2\]

\[\sigma_n^2 = c m^{2n} + K m^{n-1}, \quad m = 1\]

\[\sigma_n^2 = c(1)^{2n} + K(1)^{n-1}\]

\[\sigma_n^2 = c + Kn\]

\[\sigma_n^2 = c + \sigma_n^2\]

\[\sigma_n^2 - \sigma_n^2 = c\]

\[0 = c\]

\[\sigma_n^2 = n \sigma^2\]
Appendix A.5 Obtaining the number of active branches for step 2 using the formula:

\[
\sum_{k=0}^{2^n-1} p^{k+1}
\]

\[
\sum_{k=0}^{2^2-1} p^{k+1} = \sum_{k=0}^{3} p^{k+1} = 1p^{0+1} + 2p^{1+1} + 2p^{2+1} + 1p^{3+1} \\
= 1 \text{ a.b.} + 2(2) \text{ a.b} + 2(3) \text{ a.b} + 4 \text{ a.b} = 6 \text{ trees and 15 active branches}
\]

Appendix A.6 Obtaining the total number of possible trees in step 3 using

\[
\sum_{k=0}^{n} (p + q)^{k+1} + 2[(p + q)^{n-1}] + 2[(p + q)^{n-2}]
\]

In the first step we have \((p + q)^1\)

In step 2 we have \((p + q)^1 + (p + q)^2\) and

In step 3 we have \((p + q)^1 + (p + q)^2 + (p + q)^2 + (p + q)^3 + (p + q)^4\)

The total number of possible trees until step 3 is:

\[
(p + q)^1 + (p + q)^1 + (p + q)^2 + (p + q)^1 + (p + q)^2 + (p + q)^2 + (p + q)^3 + (p + q)^4 \\
= p + q + p + q + p^2 + 2pq + q^2 + p + q + p^2 + 2pq + q^2 + p^2 + 2pq + q^2 + \\
(p + q)(p^2 + 2pq + q^2) + (p^2 + 2pq + q^2)(p^2 + 2pq + q^2) = p + q + p + q + p^2 + 2pq + q^2 + \\
p^2 + p + q + p^2 + 2pq + q^2 + p^2 + 2pq + q^2 + p^3 + 2p^2q + pq^2 + p^2q + 2pq^2 + q^3 + \\
p^4 + 2p^3q + q^2p^3q + 2p^2q^2 + 2p^3q + q^2p^2 + 2pq^3 + q^4 = 42 \text{ possible trees}
\]

The total number of possible trees is equal to the sum of the coefficients of each term.

We also have the formula for the total number of possible trees in step \(n + 1\):

\[T_{n+1} = T_n^2 + T_n\text{ with } n = 1, 2, \ldots n \in \mathbb{N}\]

Where \(T_n\) is equal to the total number of possible trees in step \(n\).

**Example:** in step 3 the total number of possible trees is: 
\[T_{2+1} = T_2^2 + T_2 = 6^2 + 6 = 42\]

In step 4 the total number of possible trees is: 
\[T_{3+1} = T_3^2 + T_3 = 42^2 + 42 = 1,806\]
Appendix B Matlab Code to Obtain the Expected Number of Active Branches using
the Derivative of the Recurrence Formula.

\[
f_{n+1}(p, q) = f_n(p \ast (p + q), q) \ast (p + q)
\]

```matlab
>> clear;close all;
>> syms p q alpha1
>> syms f0(p,q) f1(p,q)
>> syms f1_a(p,q, alpha1) f1_b(p,q, alpha1) f1_variance(p,q)
>> syms f2(p,q) f3(p,q) f4(p,q) f5(p,q)
>> syms std_f1(p,q) mean_f2(p,q) std_f2(p,q)
>> syms mean_f3(p,q) std_f3(p,q) std_f4(p,q) std_f5(p,q)
>> f0(p,q)=1;
>> f1(p,q)=f0(p*(p+q),q)*(p+q);
>> f2(p,q)=f1(p*(p+q),q)*(p+q);
>> f3(p,q)=f2(p*(p+q),q)*(p+q);
>> f4(p,q)=f3(p*(p+q),q)*(p+q);
>> f5(p,q)=f4(p*(p+q),q)*(p+q);
>> f1
f1(p, q) =p + q
>> % f1(p,q)= p+q
>> diff(f1,p)
an(p, q) =1
>> f1*p

ans(p, q) =
```
\[ p^*(p + q) \]

\[ \texttt{>> \% ans(p,q) =} \ p^*(p+q) \]

\[ \texttt{>> \% expected value of x} \]

\[ \texttt{>> \text{diff(f1*p,p)}} \]

\[ \text{ans(p, q) =} \]

\[ 2*p + q \]

\[ \texttt{>> \% ans(p,q) =} 2*p+q \]

\[ \texttt{>> \% expected value of x^2} \]

\[ \texttt{>> \text{diff(diff(f1*p,p)*p, p)}} \]

\[ \text{ans(p, q) =} \]

\[ 4*p + q \]

\[ \texttt{>> \text{subs(diff(f1*p, p),q,1-p)}} \]

\[ \text{ans(p, q) =} \]

\[ p + 1 \]

\[ \texttt{>> \% ans(p,q) =} p+1 \]

\[ \texttt{>> f2} \]

\[ \texttt{f2(p, q) =} \]

\[ (p + q)\*(q + p^*(p + q)) \]

\[ \texttt{>> \text{diff(f2*p,p)}} \]

\[ \text{ans(p, q) =} \]

\[ (p + q)\*(q + p^*(p + q)) + p^*(q + p^*(p + q)) + p^*(p + q)^*(2*p + q) \]

\[ \texttt{>> simplify(subs(diff(f2*p, p),q,1-p))} \]

\[ \text{ans(p, q) =} \]

\[ (p + 1)^2 \]

\[ \texttt{>> f3} \]

\[ \texttt{f3(p, q) =} \]
(p + q)*(q + p*(p + q))*(q + p*(p + q))
>> diff(f3*p, p)
ans(p, q) =
(p + q)*(q + p*(p + q))*(q + p*(p + q)) + p*(q + p*(p + q))*(q + p*(p + q)) + p*(p + q)*((p + q)*(q + p*(p + q)) + p*(q + p*(p + q)) + p*(p + q)*(2*p + q)) + p*(p + q)*(2*p + q)*(q + p*(p + q))*(q + p*(p + q)) + p*(p + q)*(q + p*(p + q))*(q + p*(p + q)) + p*(p + q)*(q + p*(p + q))*(q + p*(p + q))*(q + p*(p + q))
>> simplify(subs(diff(f3*p, p), q, 1 - p))
ans(p, q) =
(p + 1)^3
>> f4
f4(p, q) =
(p + q)*(q + p*(p + q))*(q + p*(p + q))*(q + p*(p + q))*(q + p*(p + q)) + p*(q + p*(p + q))*(q + p*(p + q)) + p*(p + q)*(q + p*(p + q))*((p + q)*(q + p*(p + q)) + p*(q + p*(p + q)) + p*(p + q)*(2*p + q)) + p*(p + q)*(2*p + q)*(p + q)*(q + p*(p + q))
>> diff(f4*p, p)
ans(p, q) =
(p + q)*(q + p*(p + q))*(q + p*(p + q))*(q + p*(p + q)) + p*(q + p*(p + q))*(q + p*(p + q)) + p*(p + q)*(q + p*(p + q))*((q + p*(p + q))*(q + p*(p + q)) + p*(q + p*(p + q)) + p*(p + q)*(2*p + q)) + p*(p + q)*(2*p + q)*(p + q)*(q + p*(p + q)) + p*(p + q)*(q + p*(p + q))*(q + p*(p + q))*(q + p*(p + q)) + p*(p + q)*(q + p*(p + q))*(q + p*(p + q))*(q + p*(p + q))
>> simplify(subs(diff(f4*p, p), q, 1 - p))
Despite the complexity of this formula, which accounts for all possible trees, it reduces to the simple formula below.
ans(p, q) =
(p + 1)^4