

## RESEARCH ARTICLE

# ASCENT (Automated Simulations to Characterize Electrical Nerve Thresholds): A pipeline for sample-specific computational modeling of electrical stimulation of peripheral nerves

Eric D. Musselman<sup>1</sup> , Jake E. Cariello<sup>1,2</sup> , Warren M. Grill<sup>1,3,4,5\*</sup> , Nicole A. Pelot<sup>1\*</sup> 

**1** Department of Biomedical Engineering, Duke University, Durham, North Carolina, United States of America, **2** Department of Computer Science, Duke University, Durham, North Carolina, United States of America, **3** Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina, United States of America, **4** Department of Neurobiology, Duke University, Durham, North Carolina, United States of America, **5** Department of Neurosurgery, Duke University, Durham, North Carolina, United States of America

 These authors contributed equally to this work.

\* [warren.grill@duke.edu](mailto:warren.grill@duke.edu) (WMG); [nikki.pelot@duke.edu](mailto:nikki.pelot@duke.edu) (NAP)



## OPEN ACCESS

**Citation:** Musselman ED, Cariello JE, Grill WM, Pelot NA (2021) ASCENT (Automated Simulations to Characterize Electrical Nerve Thresholds): A pipeline for sample-specific computational modeling of electrical stimulation of peripheral nerves. *PLoS Comput Biol* 17(9): e1009285. <https://doi.org/10.1371/journal.pcbi.1009285>

**Editor:** Dina Schneidman-Duhovny, Hebrew University of Jerusalem, ISRAEL

**Received:** September 18, 2020

**Accepted:** July 18, 2021

**Published:** September 7, 2021

**Copyright:** © 2021 Musselman et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** The release associated with this paper (v1.0.0), as well as other archived releases, are available through Zenodo (v1.0.0: <https://doi.org/10.5281/zenodo.5136631>; current release: <https://zenodo.org/badge/latestdoi/379064819>); the most recent commit—where there may be multiple commits per release—is available through GitHub (<https://github.com/wmglab-duke/ascent>). The code is documented in a git wiki associated with the code repository (<https://github.com/wmglab-duke/ascent/wiki>).

## Abstract

Electrical stimulation and block of peripheral nerves hold great promise for treatment of a range of disease and disorders, but promising results from preclinical studies often fail to translate to successful clinical therapies. Differences in neural anatomy across species require different electrodes and stimulation parameters to achieve equivalent nerve responses, and accounting for the consequences of these factors is difficult. We describe the implementation, validation, and application of a standardized, modular, and scalable computational modeling pipeline for biophysical simulations of electrical activation and block of nerve fibers within peripheral nerves. The ASCENT (Automated Simulations to Characterize Electrical Nerve Thresholds) pipeline provides a suite of built-in capabilities for user control over the entire workflow, including libraries for parts to assemble electrodes, electrical properties of biological materials, previously published fiber models, and common stimulation waveforms. We validated the accuracy of ASCENT calculations, verified usability in beta release, and provide several compelling examples of ASCENT-implemented models. ASCENT will enable the reproducibility of simulation data, and it will be used as a component of integrated simulations with other models (e.g., organ system models), to interpret experimental results, and to design experimental and clinical interventions for the advancement of peripheral nerve stimulation therapies.

## Author summary

Despite promising results from preclinical studies, novel therapies using electrical stimulation of peripheral nerves often fail to produce successful clinical outcomes due to

**Funding:** EM, JC, WG, and NP were supported by National Institutes of Health SPARC Program (OT2 OD025340). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors have declared that no competing interests exist.

differences in neural anatomy across species. These differences often require different electrodes to interface with the nerves and/or different stimulation parameters to achieve equivalent nerve responses. Further, differences in nerve anatomy across a population contribute to differences in nerve responses to stimulation. These inter-species and inter-individual differences can be studied using computational modeling of individual-specific peripheral nerve morphology and biophysical properties. To accelerate the process of computational modeling of individual nerve anatomy, we developed ASCENT, a software platform for simulating the responses of sample-specific nerves to electrical stimulation with custom electrodes and stimulation parameters. ASCENT automates the complex, multi-step process required to build computational models of preclinical and clinical studies and to design novel stimulation protocols using biophysically realistic simulations. The ASCENT pipeline will be used to develop technologies that increase the selectivity and efficiency of stimulation and to accelerate the translation of novel peripheral nerve stimulation therapies to the clinic.

### 3 Introduction

Targeted and reversible changes in nerve activity by electrical stimulation and block of peripheral nerves holds great promise for treatment of disease and injury. For example, vagus nerve stimulation (VNS) is an FDA-approved therapeutic device including a cuff electrode around the left cervical vagus nerve and a pulse generator implanted subcutaneously in the chest. Stimulating the nerve with low frequencies, e.g., tens of Hz, activates the underlying nerve fibers to treat pharmaco-resistant epilepsy [1–3] and depression [4–6]. Given the diverse end-organ targets of the vagus nerve, VNS is under investigation for a wide range of diseases [7] including sepsis, fibromyalgia, migraine, cardiovascular disease, ventilator-induced lung injury, stroke, diabetes, and rheumatoid arthritis [8].

Despite promising results from preclinical studies, novel applications of electrical stimulation of peripheral nerves often fail to translate to successful clinical therapies. Differences in neural anatomy across species often require different electrodes to interface with the nerves and/or different stimulation parameters to achieve equivalent nerve responses. Further, differences in nerve anatomy across individuals contribute to differences in nerve responses to stimulation. The nerve morphology, fiber types, and electrode geometry substantially impact the responses of nerve fibers to a given set of stimulation parameters, and producing equivalent responses between species or across individuals remains a challenge. The exceedingly large stimulation parameter space and nonlinear input-output relationships between applied stimulation and neural responses make it unrealistic to develop an optimized intervention using only *in vivo* experiments.

Computational modeling can be leveraged to achieve translation between preclinical and clinical studies and to optimize application-specific parameters. Prior modeling studies used generalized representations of nerve morphology [9–11] or patient-specific models of peripheral nerve stimulation [12,13], but these were bespoke implementations by experts; a standardized platform to enable the research community to implement and use such models has not been developed. The field needs a “completely customizable and modular framework . . . to interpret the results of stimulation experiments in terms of quantities of common use in neuro-prosthetics” [14].

We describe the implementation, validation, and application of a computational modeling pipeline for simulation of electrical activation and block of populations of nerve fibers within

peripheral nerves. ASCENT (Automated Simulations to Characterize Electrical Nerve Thresholds) spans from fixed nerve samples to account for individual-specific neural anatomy (i.e., species and location of intervention), through three-dimensional representations of electrodes positioned on nerves, to analysis of waveforms and electrodes that produce selective activation or block in specific nerve fiber types (S1 Text). ASCENT automates the complex, multi-step process required to model peripheral nerve stimulation and block. The resulting models provide essential tools to aid in the interpretation of experimental results, for design of experimental and therapeutic interventions, and as elements that can relate to other models (e.g., organ system models) for integrated simulations. We applied ASCENT to quantify changes in fiber thresholds for different cuff electrode positions on a multifascicular nerve; we reproduced a published model of rabbit sciatic nerve stimulation and we improved upon a published model of human VNS, demonstrating how ASCENT reduces duplication of efforts and serves as a standardized platform to enable model reproducibility and promote best practices; and we modeled dog VNS experiments, reproducing the *in vivo* activation thresholds. Thus, ASCENT provides an important contribution towards FAIR (findable, accessible, interoperable, reusable) data in the realm of computational modeling [15].

## 4 Design and implementation

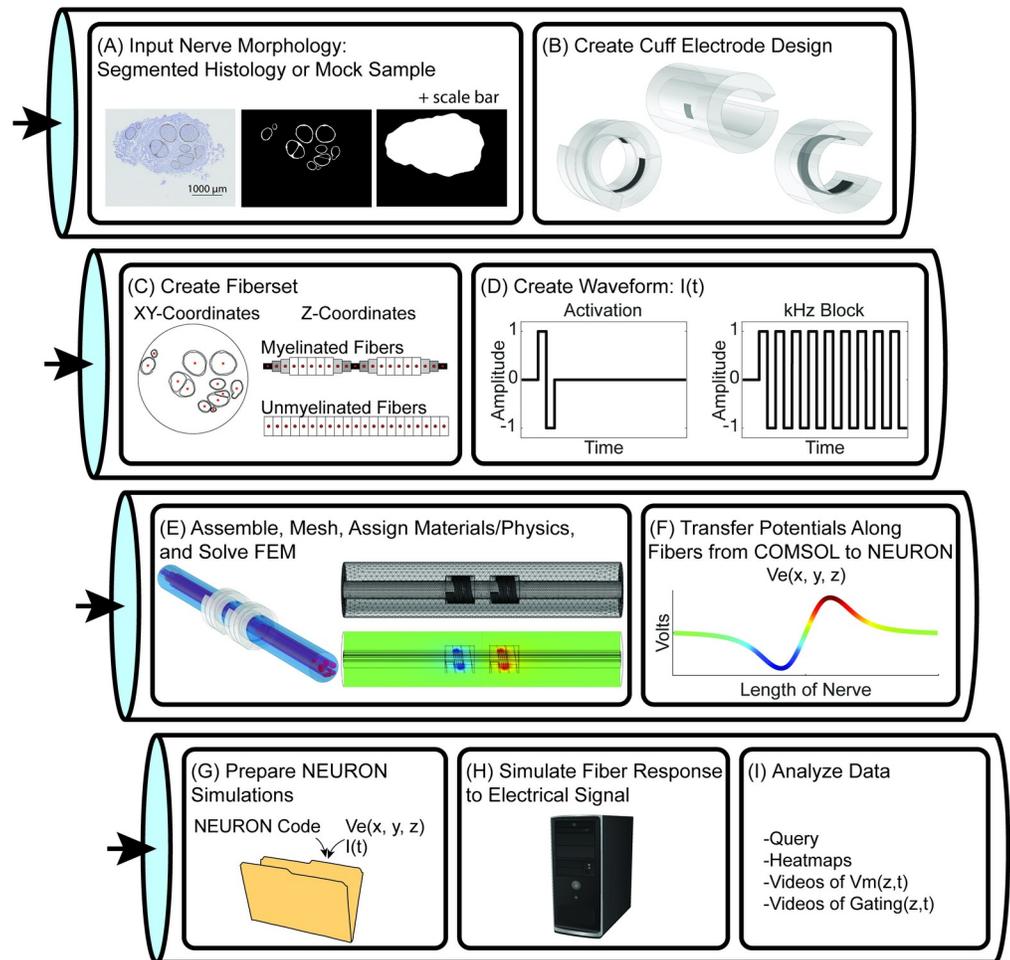
### 4.1 Overview of the ASCENT pipeline

ASCENT computes the electric fields [16] generated in inhomogeneous anisotropic nerve tissue by one or multiple current sources and applies the resulting extracellular potentials to cable models of individual fibers as a time-varying signal [17] (Fig 1). ASCENT's process minimizes duplication of computationally intensive tasks, enables scalability by preserving modularity of operations, and provides robust automation and user control. Furthermore, ASCENT enables automatic batching of model execution across all domains (e.g., nerve samples, electrode geometries and positions on the nerve, material and tissue properties, fiber models, stimulation waveforms), which is essential for model-based design of electrodes and stimulation paradigms. ASCENT has built-in control for modular runs, error handling to skip failed models, breakpoints to run only up to a specified process, and scripts for accessing and plotting intermediate data (e.g., traces of perineurium and epineurium tissue boundaries, fiber locations, and stimulation waveforms).

The pipeline uses Python 3.7 [18], Java SE Development Kit 8 (1.8) [19], COMSOL Multiphysics 5.5 (COMSOL Inc., Burlington, MA), and NEURON v7.6 [20]; see S2 Text for installation instructions, S3 Text for a description of the data hierarchy, S4 Text for instructions to complete your first run of the pipeline, and S5 Text for guidelines to report methods. After defining parameters in JSON configuration files (S6–S9 Text), the user initiates two sequential processes: the first process uses Python, Java, and COMSOL to create stimulation waveforms and extracellular potentials along the length of fibers, and the second process uses Python to prepare batched NEURON jobs for execution on a personal computer or computer cluster (S10 Text).

### 4.2 Digitized nerve morphology

The finite element model (FEM) of a nerve can be sample-specific using segmented histology (S11 Text) [21] or generalized using our mock nerve morphology generator, as described below (S12 Text). We developed a standardized object-oriented framework for representing and processing nerve morphology in Python (S13 Text). From input binary masks, Python operations prepare the nerve sample for placement in a cuff electrode, including optional correction for tissue shrinkage during histology and deformation of the nerve to conform to the



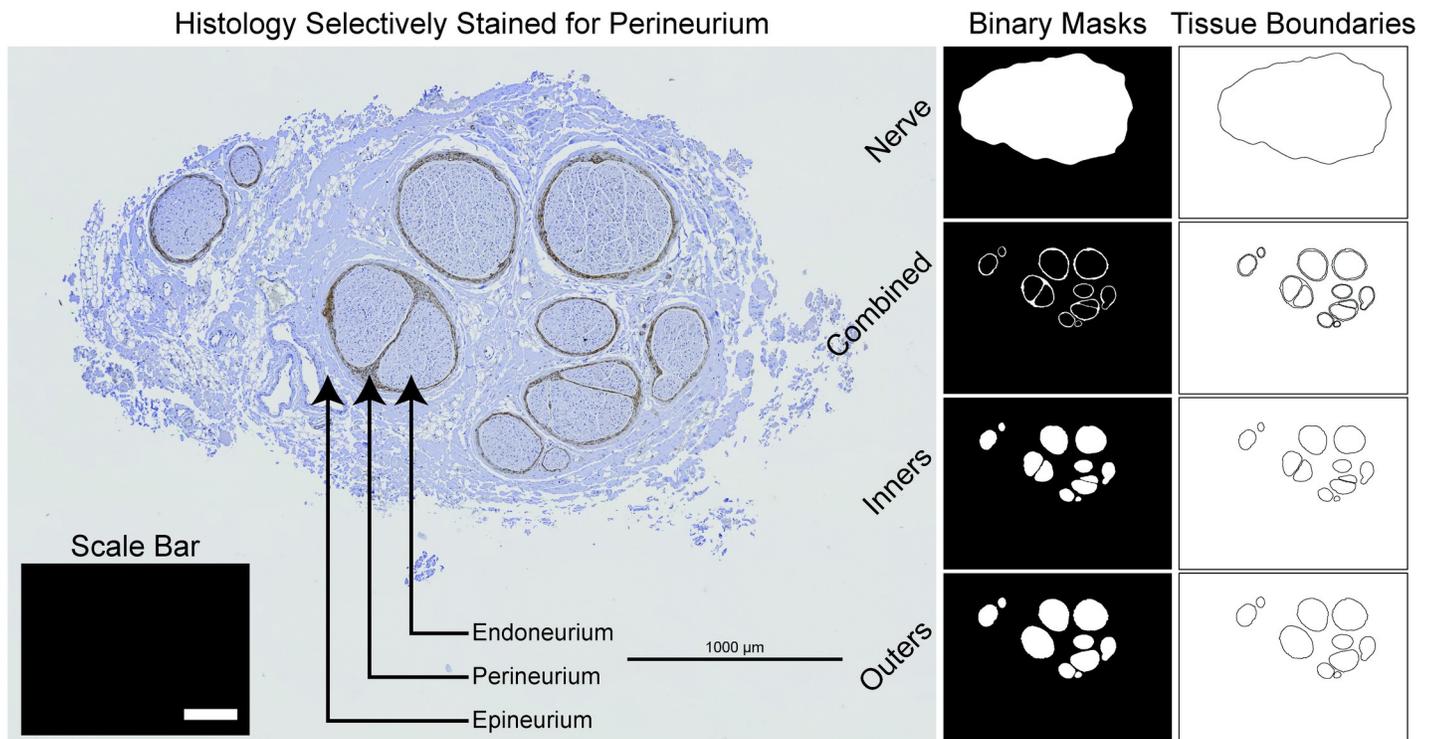
**Fig 1. Overview of the ASCENT pipeline for simulating the response of sample-specific nerves to electrical stimulation with custom cuff electrodes and stimulation waveforms. FEM: finite element model.**

<https://doi.org/10.1371/journal.pcbi.1009285.g001>

inner diameter of a cuff; Python produces two-dimensional CAD files to define nerve and fascicle tissue boundaries in COMSOL (S14 Text).

**4.2.1 Segmented histology.** A segmented cross section of a nerve sample can be used to build the FEM (Figs 1A and 2).

**4.2.2 Mock nerve morphology generator.** Alternatively, users can define a nerve sample as an elliptical nerve and a list of elliptical fascicles using our mock nerve morphology generator. The user may place fascicles using either the *explicit* option, for which the sizes (lengths of the major and minor axes), locations (centroids), and rotations of fascicles are explicitly defined by the user, or the *probabilistic* option, which randomly places a user-defined number of elliptical fascicles within the nerve: diameters and eccentricities are drawn from a statistical distribution (e.g., truncated normal, uniform), centroids are randomly chosen using disk point picking, and rotations are drawn from a uniform distribution. In randomly placing fascicles, the overlap of fascicles of fixed size and rotation is avoided by iteratively redefining the centroid of each fascicle until it is placed such that its boundaries are at least a user-defined distance from neighboring nerve and previously populated fascicle boundaries; the program will attempt to place each fascicle a user-defined number of times before skipping the fascicle and reporting the progress to the user.



**Fig 2. Define sample-specific models with segmented nerve images.** We segmented the nerve boundary (n.tif), the perineurium (either “Combined” (c.tif), or both “Inners” (i.tif) and “Outers” (o.tif), or only “Inners”), and a horizontal “Scale Bar” (s.tif) of known length from a micrograph of a human cervical vagus nerve stained selectively for perineurium. In this example, two fascicles have multiple inner traces for their single outer trace (i.e., “peanut” fascicles), which requires their perineurium to be meshed; conversely, the other fascicles have a single inner trace for each outer trace, in which case the perineurium can be either meshed or modeled with a thin layer approximation. The nerve cross section is modeled in the (x,y,0) plane and extruded into the +z direction for the length of the nerve.

<https://doi.org/10.1371/journal.pcbi.1009285.g002>

### 4.3 Cuff electrode design

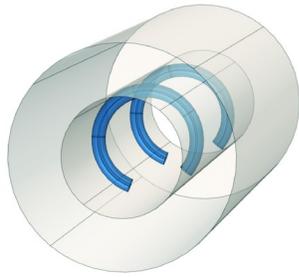
ASCENT includes tools to define and manipulate the representation of cuff electrodes and surrounding media in COMSOL (Fig 1B). The pipeline includes cuff electrodes commonly used in preclinical studies and clinical therapies (Fig 3A and S15 Text). Users can also assemble custom cuff electrodes by adding parameterized instances of “part primitives” for electrode contacts, cuff insulators, and cuff fill (between the nerve and cuff and/or in the immediate vicinity around the cuff; e.g., saline, mineral oil, or encapsulation tissue) (S16–S18 Text). The user can control the placement of the cuff with respect to the nerve, including translation and rotation (S19 Text).

### 4.4 Fiber locations and waveforms

Before interfacing with COMSOL via Java, Python defines the coordinates of “fibersets” (i.e., the coordinates where voltages are sampled in COMSOL for extracellular stimulation or block of model fibers) (Fig 3B) and defines the current amplitude versus time stimulation waveforms used in NEURON (Fig 3C).

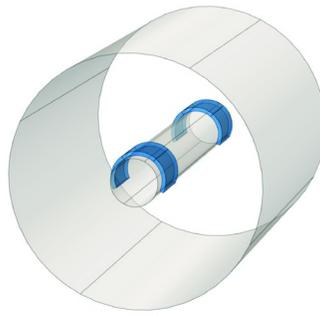
**4.4.1 Create fiberset.** ASCENT includes parameterized modes to define fiber locations (Figs 1C and 3B and S20 Text). At each fiber’s (x, y)-location in the nerve cross section, potentials are sampled along the z axis at the locations of compartments of published models of myelinated and unmyelinated fibers (S21 Text). Alternatively, users can “super-sample” the potentials along the length of the nerve in COMSOL at each (x, y)-location: the pipeline will

(A) Exposed Wire Bipolar Cuff



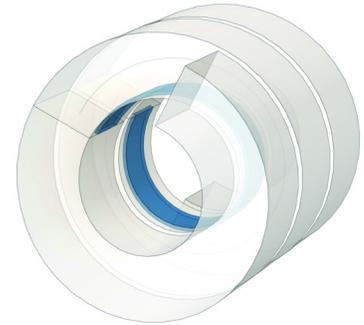
(2) Wire Contact Primitives  
(1) Tube Cuff Primitive

CorTec Bipolar Cuff



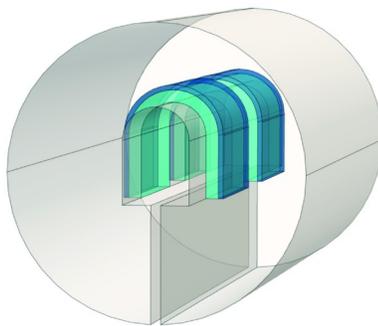
(2) Ribbon Contact Primitives  
(1) Tube Cuff Primitive

LivaNova Cuff



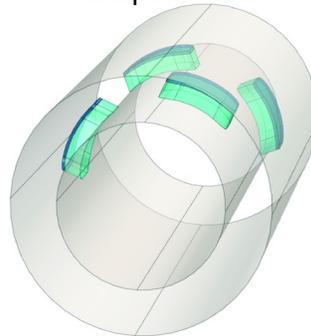
(1) LivaNova Primitive

Micro-Leads Cuff



(2) U Contact Primitives  
(1) U Cuff Primitive

Multipolar Cuff



(4) Rectangle Contact Primitives  
(1) Tube Cuff Primitive

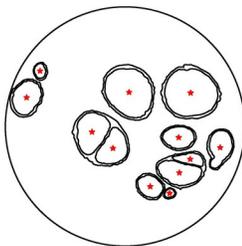
Enteromedics Cuff



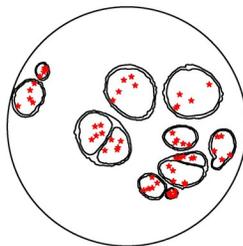
(1) Ribbon Contact Primitive  
(1) Tube Cuff Primitive

(B)

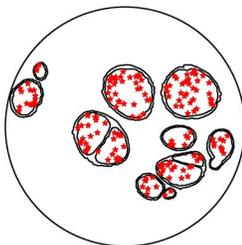
Centroid



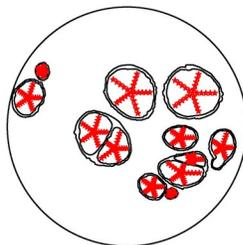
Uniform Count



Uniform Density

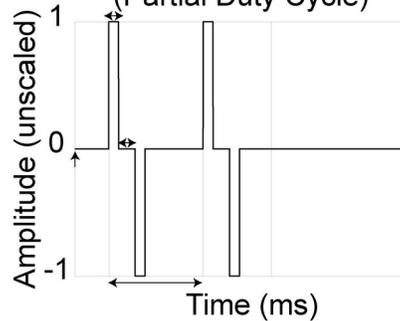


Wheel

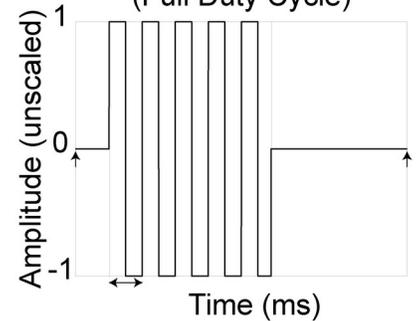


(C)

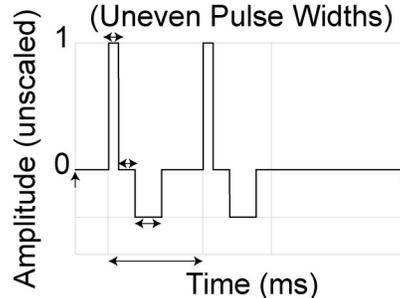
Biphasic Pulse Train  
(Partial Duty Cycle)



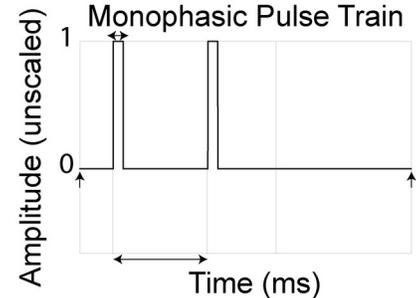
Biphasic Pulse Train  
(Full Duty Cycle)



Biphasic Pulse Train  
(Uneven Pulse Widths)



Monophasic Pulse Train



**Fig 3. Parameterized preset cuff electrodes, modes to define fiber locations in nerve cross section, and modes to define conventional stimulation waveforms.** (A) Examples of “preset” cuffs commonly used in preclinical and clinical studies assembled using part primitives from [S16 Text](#), which are provided in config/system/cuffs/. Custom cuff electrodes can be created from existing or new part primitives ([S17](#) and [S18 Text](#)). (B) Types of FiberXYMode defining the (x, y)-coordinates of the fiber locations in the nerve cross section. The parameters needed to define the (x, y)-coordinates for each “FiberXYMode” are explained in [S8 Text](#). (C) Parameterized definitions of conventional waveforms included in the ASCENT pipeline. Users can also define custom waveforms ([S8 Text](#)).

<https://doi.org/10.1371/journal.pcbi.1009285.g003>

interpolate the potentials for the spatial discretization of each fiber model to be simulated, thereby eliminating later dependency on COMSOL.

**4.4.2 Create waveforms.** The pipeline provides parameterized definitions of conventional waveforms (Figs [1D](#) and [3C](#)), defined using the same timestep and duration as the NEURON simulation. We use the SciPy Signal package [[22](#)] and the NumPy library [[23](#)] to generate the conventional waveforms. Users can also define custom arbitrary waveforms. The stimulation waveform input to NEURON is “unscaled”, meaning that the maximum magnitude is one; the waveform delivered to the model fibers in NEURON is then scaled by the stimulation amplitude ([S22 Text](#)).

## 4.5 Finite element modeling

After creating nerve morphology CAD files, fibersets, and waveforms in Python, a Java subprocess interfaces with COMSOL to build, mesh, solve, and extract data from FEMs ([Fig 1E](#) and [S23](#) and [S24 Text](#)).

**4.5.1 Assemble and mesh the FEM.** The medium surrounding the nerve and cuff electrode is represented by a cylindrical domain ([S25 Text](#)). The user can define different meshing parameters for the “proximal” domains (i.e., a cylinder containing the full length of the nerve and the cuff electrode) and the remaining “distal” domains; thus, the spatial discretization may be defined more coarsely outside of the target nerve area. The pipeline builds an unstructured mesh with tetrahedral elements and reports the meshing statistics (quality, number of elements, meshing time). Since files of FEM meshes can be quite large, we provide the option to save them; the pipeline will reuse a previously saved FEM mesh to reduce computation time if the current model has the identical geometry and mesh parameters, particularly in sweeps of material properties and boundary conditions ([S26 Text](#)).

**4.5.2 Assign conductivities and sources to the FEM.** After the geometry of the model is discretized with a mesh, the pipeline applies the conductivities of the materials (cuff “insulator”, contact “conductor”, contact “recess”, cuff “fill”) and tissues (surrounding medium, endoneurium, epineurium, perineurium) to their corresponding domains. Relevant material and tissue conductivities from literature are built into ASCENT, and users may opt to define their own materials [[24–31](#)] ([S27 Text](#)). Perineurium can be modeled as a meshed domain or with a thin layer approximation to reduce mesh complexity (except with “peanut” fascicles; see an example in [Fig 2](#) where the latter is termed “contact impedance” in COMSOL [[27](#)] ([S28 Text](#))).

A point source of current is placed at the center of each contact. A grounded boundary condition ( $V = 0$ ) is applied to the outer-most surfaces of the surrounding medium, which includes the ends of the nerve, to simulate a distant ground (e.g., implanted pulse generator for chronic studies, percutaneous needle for acute experiments). However, if stimulating with more than one contact where the sum of the currents across the contacts is zero (e.g., bipolar cuff with +1 mA at one contact and -1 mA at the other), the current sunk to or sourced from the ground will be effectively zero after the difference is taken between the potentials resulting from each contact’s FEM solution. Alternatively, users may use an insulating boundary condition on the most distant surfaces.

**4.5.3 Solve the FEM.** The pipeline solves the FEM once for each contact, where each contact is assigned 1 mA and the other contacts are floating [32]; this defines the solution “bases”. The potentials are obtained using conjugate gradients to solve Laplace’s equation using second order solution and geometry shape functions:

$$\nabla \cdot (\sigma \cdot \nabla \phi) = 0 \quad (1)$$

Since files of FEM solutions can be quite large, we provide the option to save them; subsequent simulations of fibers within the same FEM but with new waveforms, a different ratio of currents across the contacts, or different fiber models can use either previously saved COMSOL files (\*.mph files in bases/) or previously super-sampled potentials (\*.dat files in ss\_bases/).

**4.5.4 Transfer potentials from COMSOL to NEURON.** The pipeline samples the potentials at each coordinate (x, y, z) of each fiber within fibersets/ or ss\_coords/ (i.e., “super-sampled” coordinates) using the COMSOL Java API for each of the solution bases (Fig 1F and S29 Text). The linearity of scaling from Ohm’s law indicates that we can compute the potentials generated by any weighting of active contacts (i.e., different current amplitudes delivered by different contacts) by the dot product of our “bases” with a vector of contact weights.

## 4.6 Simulating fiber responses to electrical signals

When all potentials from COMSOL are saved to file, the pipeline returns to Python to build NEURON simulation directories (i.e., n\_sims/) (Fig 1G and S30 Text) to be submitted to a computer or computer cluster using the submit.py or submit.sh scripts (Fig 1H and S10 Text). Users can simulate activation thresholds, block thresholds, or responses to set stimulation amplitudes (S22, S31 and S32 Text) for the MRG (McIntyre-Richardson-Grill) myelinated fiber models [33,34], the Rattay and Aberham, Sundt et al., and Tigerholm et al. unmyelinated [35–37] fiber models, or user-defined fiber models.

## 4.7 Data analysis

We provide tools for selecting and loading data created by the pipeline for analysis and visualization. We provide example scripts to plot inputs (e.g., nerve morphologies, fiber locations and discretization, waveforms) and outputs (e.g., heatmaps of thresholds) (Fig 1I and S33 Text). We also provide code to generate videos of transmembrane potential or ion channel state variables (i.e., gating parameters) as functions of space and time.

## 4.8 User oversight

Although ASCENT has built-in exception checking for many invalid parameter cases, the user is ultimately responsible for confirming that the chosen parameters are appropriate. As with all computational modeling, it is possible to obtain results (i.e., without the pipeline throwing exceptions) but for the results to be incorrect. In addition to checks around trends in the expected outputs, the user can leverage ASCENT tools to verify inputs and intermediate data.

We recommend that users plot their waveforms and fiber locations using the provided scripts (S33 Text). Additionally, we recommend that users verify that the FEM geometry is as expected in the COMSOL GUI with materials, boundary conditions, and physics assigned to the correct components; modular runs (e.g., only building the nerve) and partial runs (i.e., adding breakpoints to run only up to a specified step) (S8 Text) are useful for checking the simulations at multiple levels. Users must also conduct convergence studies to ensure that parameters are appropriately defined for sufficient numerical accuracy of the target outcomes (e.g., thresholds), including the size (i.e., length and radius) of the surrounding medium, the

length of the fibers, the resolution of the mesh, and the time step (S34 Text) [38]. Users must also verify that the location of action potential detection is sufficiently far from the electrode cuff such that propagating activity is detected rather than only an ohmic rise in transmembrane potential, and thus, they must also use sufficient simulation time for any action potentials to propagate from the initiation site to the recording site.

## 5 Results

We applied ASCENT's extensive set of features to quantify the change in fiber responses across cuff rotations for a multifascicular nerve, to implement previously published models of rabbit sciatic [11] and human vagus [39] nerve stimulation, and to reproduce findings from an *in vivo* study of dog VNS [40].

First, we designed test simulations to verify the accuracy of ASCENT's activation thresholds. The verifications were conducted in collaboration with The Foundation for Research Technologies in Society (IT<sup>2</sup>IS) with the Sim4Life simulation platform (<https://zmt.swiss/sim4life/>) (Zurich, Switzerland) (S35 Text). We found strong agreement in thresholds between models generated with ASCENT and Sim4Life. Cross-validation of thresholds was performed for a monofascicular rat nerve with a bipolar cuff (<4.2% difference in thresholds), an idealized multifascicular nerve with a bipolar cuff (<3% difference), and a multifascicular human nerve with a bipolar LivaNova helical coil cuff (<2.5% difference). We verified ASCENT's usability by asking beta testers to perform a task aided by supporting documentation (S4 Text).

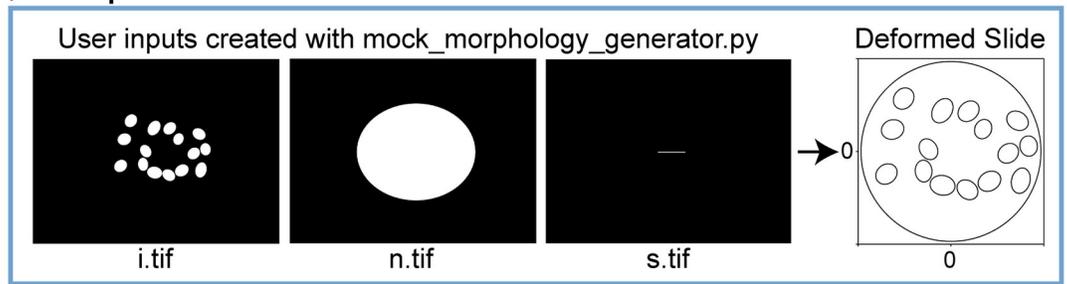
### 5.1 Investigating impact of cuff rotation on multifascicular nerve

ASCENT can be used to design cuff geometries prior to fabrication and to investigate the sensitivity of activation patterns to cuff geometry and placement. As an example, we simulated the effects of the rotation of a monopolar cuff electrode on thresholds for activation and block of 10  $\mu\text{m}$  diameter myelinated nerve fibers in a multifascicular nerve. Using the `mock_morphology_generator.py` script, we created binary image masks of the inners (`i.tif`) and nerve (`n.tif`) (Fig 4A). The **Model** configuration file contains parameters that define the geometry, materials, and physics of the FEM, as well as the settings for model meshing and solving (Fig 4B and S8 Text). The "preset" cuff parameter in **Model** for this example is the monopolar Enteromedics cuff. To simulate thresholds for four different cuff orientations on the nerve, we created four **Model** configurations with distinct "add\_ang" parameter values (0°, 90°, 180°, and 270°) (provided in `examples/results/cuff_rotation/` as `model_<angle>.json`).

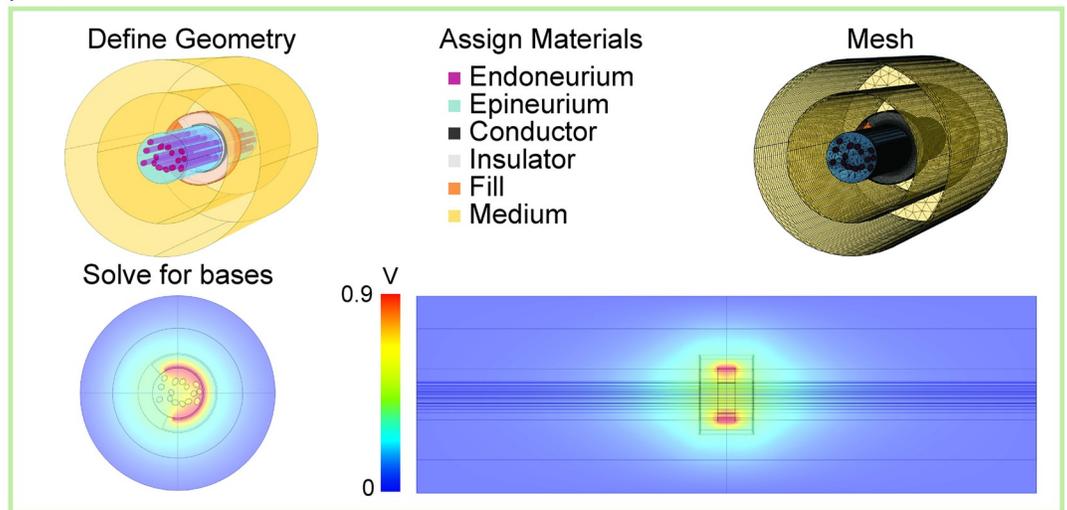
We extracted potentials from the four FEM solutions (i.e., one **Model** for each cuff rotation) at the coordinates defined by `fibersets/` to create potentials/ (Fig 4C). The **Sim** configuration files (provided in `examples/results/cuff_rotation/` as `sim_activation.json` and `sim_block.json`) contain parameters to define `fibersets/` for the (x,y)-coordinates (UNIFORM\_DENSITY), z-coordinates (compartment spacing for the 10  $\mu\text{m}$  diameter MRG fiber), stimulation waveforms (activation: BIPHASIC\_PULSE\_TRAIN; block: BIPHASIC\_FULL\_DUTY), and NEURON simulation control parameters for determining thresholds.

The binary searches to determine activation and block thresholds are conducted in NEURON for each of the four **Model** (i.e., four cuff rotations) and two **Sim** (i.e., activation and block) configurations. The resulting thresholds are shown as heatmaps in Fig 4D. The results demonstrate that, for this cuff design, cuff rotation on the nerve can substantially impact the distribution of thresholds for activation and block. Thresholds were lowest for fibers closest to the contact. Thus, the cuff rotation changed thresholds for a fiber by up to ~4x, which, depending on the targeted physiological response, could be perceived as a feature or flaw of the cuff's design.

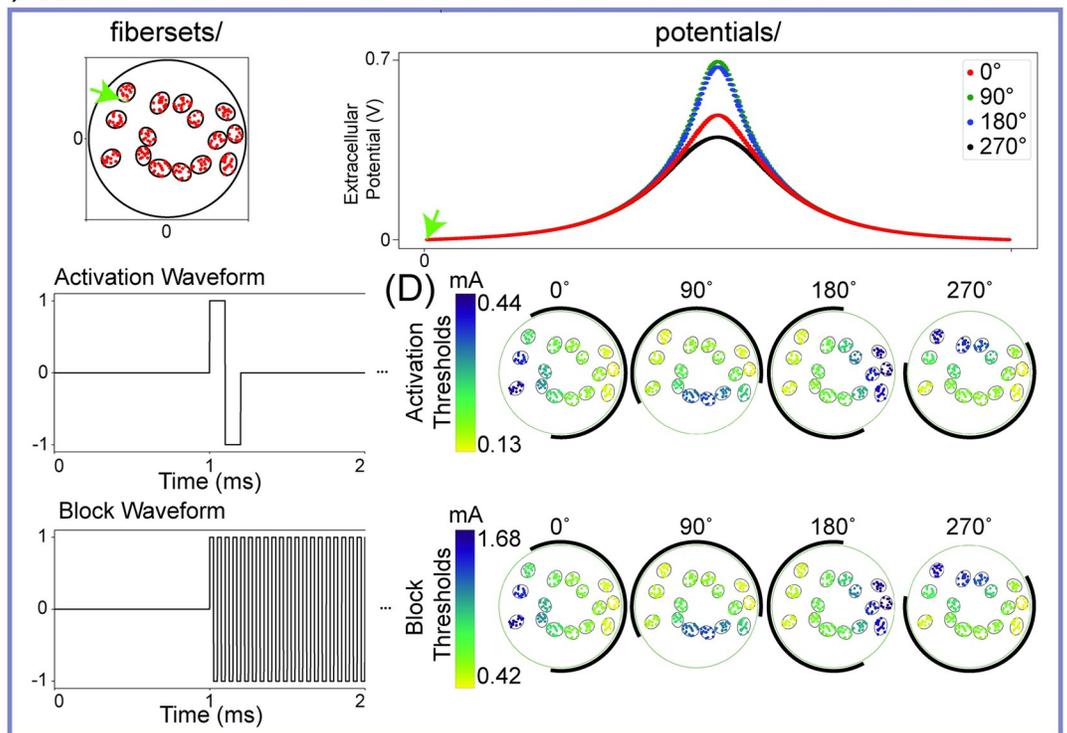
(A) Sample



(B) Model



(C) Sim



**Fig 4. Modeling a multifascicular nerve with different cuff rotations to illustrate key ASCENT processes in context of configuration data hierarchy.** (A) Binary images of the fascicle boundaries (“inner” perineurium: i.tif) and nerve boundary (n.tif) to serve as inputs to the pipeline (left) and the resulting deformed traces for placement in a circular cuff electrode (right) following physics-based Python operations. The sample JSON configuration used is provided in examples/results/cuff\_rotation/mock.json along with the binary images shown. (B) COMSOL FEM solved for the potentials generated by 1 mA delivered through the monopolar Enteromedics cuff. A contact impedance describes the perineurium ensheathing each fascicle (not shown). Given that it is a monopolar cuff, there is only one solution basis for a given cuff placement; a given *Model* will result in a basis solution for each contact, i.e., a basis \*.mph file for each contact delivering 1 mA. (C) Inputs to NEURON simulations for activation and block: extracellular potentials (i.e., potentials/ sampled from bases/ corresponding to coordinates defined by fibersets/) and waveforms. The green arrow in the fibersets/ panel points to the (x,y,0)-location for the fiber potentials plotted along the length of the nerve in the potentials/ panel. (D) Heatmaps of activation and block thresholds for 10  $\mu\text{m}$  diameter MRG fibers for four different cuff rotations on a multifascicular nerve. The black arc around the nerve represents the exposed contact length delivering current on the inside surface of the Enteromedics cuff. The heatmaps were generated using Query’s heatmaps() method.

<https://doi.org/10.1371/journal.pcbi.1009285.g004>

## 5.2 Reproducing a computational model

ASCENT will reduce duplication of efforts in the computational modeling research community and serve as a standardized platform to enhance model reproducibility. As an example, we implemented models from a published study that used computational modeling and *in vivo* electrophysiology to compare vagus nerve and sciatic nerve responses across cuff electrode geometries, with the objective of improving efficiency of fiber recruitment [11].

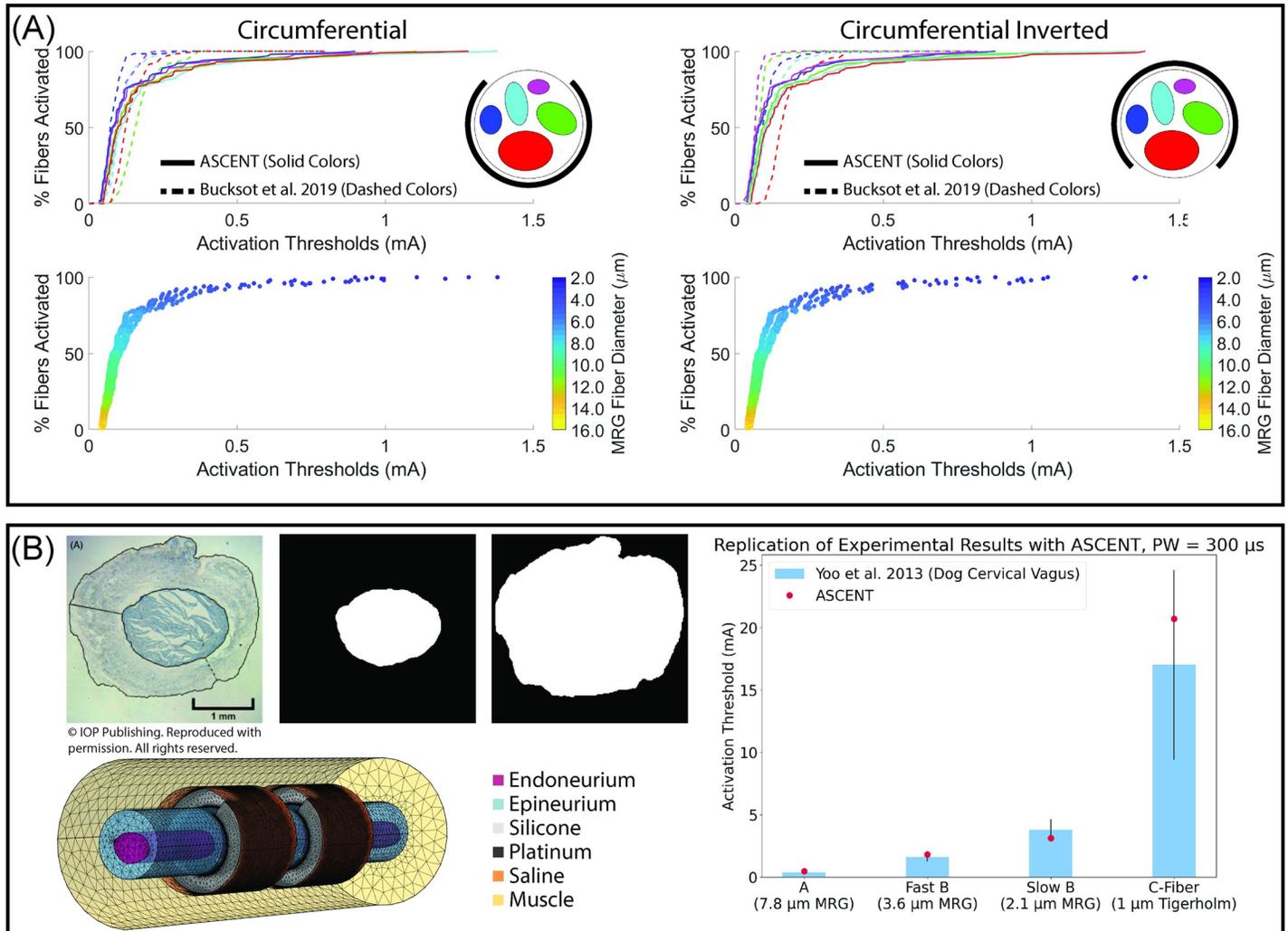
We used ImageJ’s “Analyze Particles” tool [41] to determine the best-fit ellipses of the rabbit sciatic nerve modeled in Bucksot et al. 2019 [11], which we used to define binary images of tissue boundaries using our mock\_morphology\_generator.py script. We reproduced the electrode geometry, material conductivities, nerve fiber models, and stimulation waveform described in the publication. ASCENT produced fiber recruitment curves that demonstrate preferential recruitment of larger diameter fibers in fascicles that are smallest and closest to the electrode contact (Fig 5A). We obtained similar activation thresholds to the data published by Bucksot et al. 2019, but they were not an exact match due to the incomplete and inconsistent description of model parameters in the publication [11]. Furthermore, ASCENT’s interpolation of the MRG fiber model differs from that in the original publication, resulting in higher thresholds for small diameter myelinated fibers, given our interpolation’s shorter internodal and paranodal lengths (S36 Text).

## 5.3 Modeling *in vivo* vagus nerve stimulation experiment

ASCENT can build biophysically realistic simulations of preclinical animal studies. We built a computational model to complement a previously published study of cervical VNS in dogs [40] (Fig 5B). The simulated activation thresholds correspond closely to the *in vivo* thresholds across four fiber types. These data suggest that the therapeutic mechanisms of VNS in large mammals are driven by activation of A- and/or B-fibers (i.e., large and small myelinated fibers, respectively), since the C-fiber (i.e., unmyelinated fiber) activation thresholds are substantially higher than clinical stimulation amplitudes. The accuracy of ASCENT in reproducing this experimental setup demonstrates that it can be used to predict inter-individual variability in fiber response across animals to inform experimental designs. Additionally, a user could leverage ASCENT to design next generation electrodes and stimulation waveforms to activate preferentially fibers of different types or locations in the nerve cross section or to minimize inter-individual variability in fiber response to stimulation.

## 5.4 Predicting nerve responses where experiments are not yet feasible

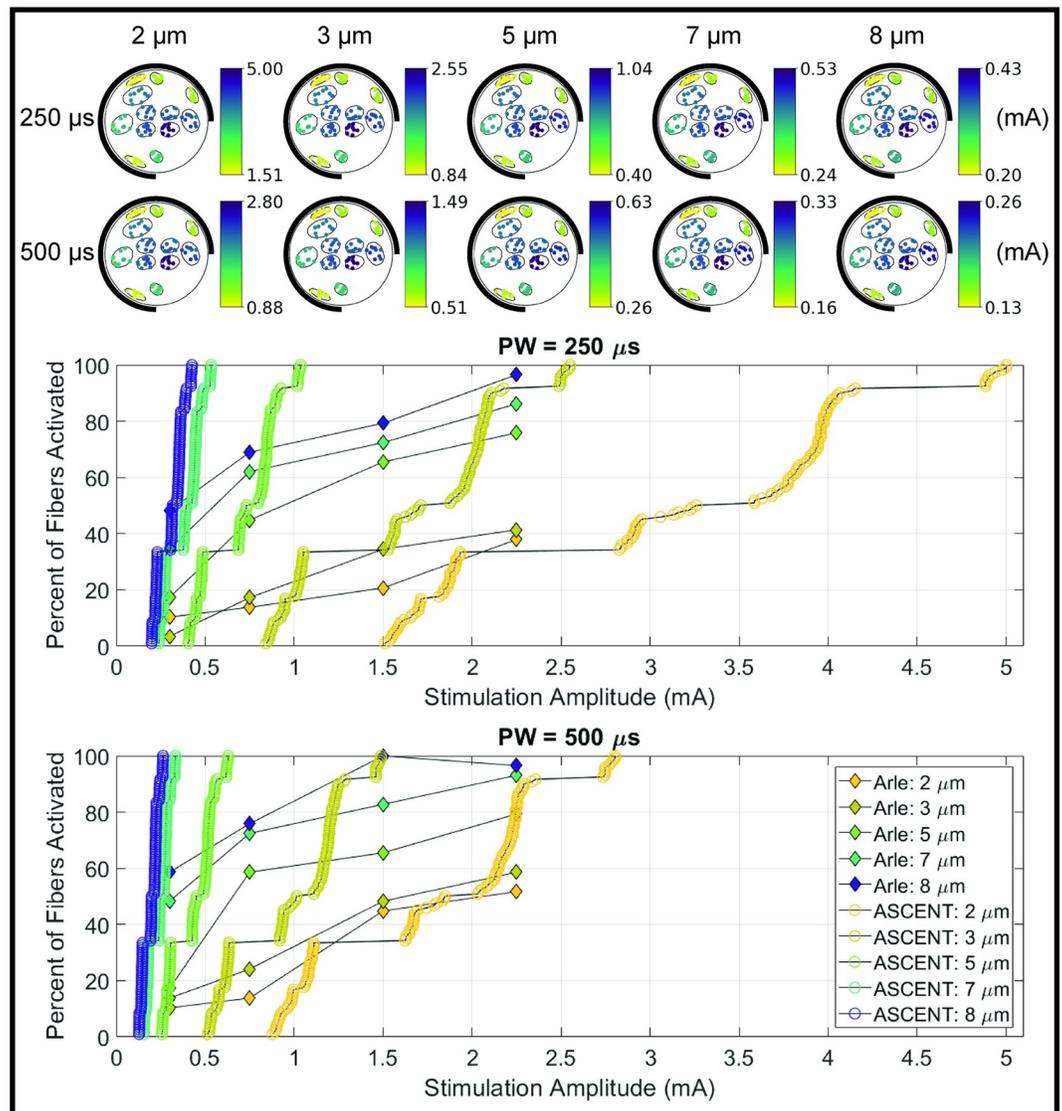
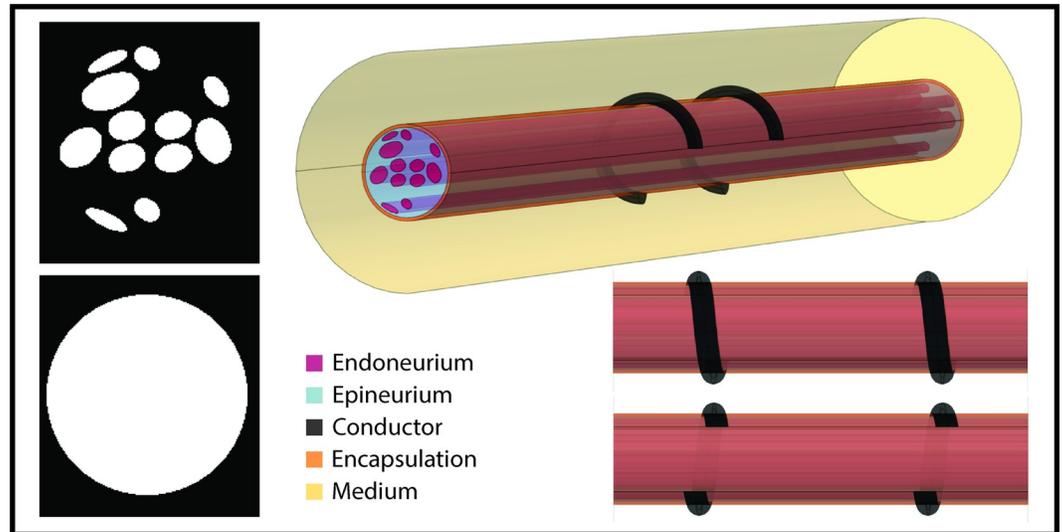
ASCENT can predict fiber responses to electrical stimulation where experimental neural recordings are difficult or impossible to obtain. We used ASCENT to simulate activation



**Fig 5. Using ASCENT to replicate previously published computational models and to model an *in vivo* experiment.** (A) Recruitment curve for each fascicle in a model of the rabbit sciatic nerve for two rotations of a bipolar cuff electrode [11]. Fiber thresholds are in response to a 100 μs per phase biphasic rectangular pulse (i.e., BIPHASIC\_PULSE\_TRAIN). We simulated mammalian myelinated fibers at 0.5 μm diameter increments from 2 to 16 μm (i.e., MRG\_INTERPOLATION) with 10 fibers of each diameter at consistent locations in each fascicle (i.e., UNIFORM\_COUNT). We generated fiber recruitment curves by defining a population of fiber diameters from a normal distribution ( $8.85 \pm 3.1 \mu\text{m}$ , 100 fibers per fascicle) and rounded each fiber diameter to the nearest 0.5 μm to draw from our modeled activation thresholds randomly. Configuration and input files used to replicate the results are provided in examples/results/bucksot\_2019/. (B) Using ASCENT to model electrophysiology study of dog vagus stimulation [40]. We used Photoshop (Adobe Inc., San Jose, CA) to segment a micrograph from one of the animals in the original publication. We placed the nerve in a 2 mm diameter LivaNova helical cuff electrode delivering a 300 μs monophasic rectangular pulse (i.e., MONOPHASIC\_PULSE\_TRAIN), as used experimentally. We computed activation thresholds for myelinated (i.e., MRG\_INTERPOLATION: 2.1, 3.6, and 7.8 μm diameter) and unmyelinated (i.e., TIGERHOLM: 1.0 μm diameter) fiber models, corresponding to the fiber types reported in the original study. We report the modeled threshold of a single fiber of each type at the centroid of the fascicle (i.e., CENTROID) (red dot) overlaid on the *in vivo* ENG thresholds (blue bars with black error bars: mean ± SD) of both myelinated (n = 5 dogs) and unmyelinated (n = 4 dogs) vagus nerve fibers. Our prior modeling studies show that fibers of a consistent diameter placed in the same fascicle have nearly identical thresholds [10]. Configuration and input files used to model the experiment are provided in examples/results/yoo\_2013/.

<https://doi.org/10.1371/journal.pcbi.1009285.g005>

thresholds of human VNS for the nerve and cuff electrode modeled in Arle et al. 2016 (Fig 6) [39]. ASCENT improves upon Arle’s model implementation in multiple ways. First, their fiber models only contained passive membrane mechanisms, whereas we used biophysically realistic models of myelinated fibers [34] (S21 Text). Second, many of their fibers seemed to be placed outside of the fascicles, in the epineurium; we instead seeded 10 fiber locations in each fascicle



**Fig 6. Implementation of a human VNS model as studied in Arle et al. 2016 [39].** Top panel: segmented nerve geometry modeled in the original publication and image of FEM produced with ASCENT. As modeled in Arle et al. 2016 [39], the electrode lacks insulation. Bottom panel: heatmaps and recruitment curves for fiber activation thresholds across fiber diameters and pulse widths. We generated the heatmaps using Query's heatmaps() method. Consistent with our prior modeling studies [10], the heatmaps show that fibers in the same fascicle with the same diameter have nearly identical thresholds. Configuration and input files used to replicate the results are provided in examples/results/arle\_2016/.

<https://doi.org/10.1371/journal.pcbi.1009285.g006>

**Table 1. Material conductivities used in Arle et al. 2016 [39] and ASCENT.**

Material	Arle et al. 2016 [39]	ASCENT
Encapsulation	0.03 [S/m]	0.159 [S/m]
Epineurium	0.054 [S/m]	0.159 [S/m]
Perineurium	0.005668 [S/m]	0.0008703 [S/m]
Endoneurium	{0.08, 0.08, 0.05} [S/m]	{0.167, 0.167, 0.571} [S/m]
Medium	Fat: 0.04 [S/m]	Muscle: {0.086, 0.086, 0.35} [S/m]

<https://doi.org/10.1371/journal.pcbi.1009285.t001>

(i.e., “UNIFORM\_COUNT”). Third, we used more accurate parameter values for material conductivities and perineurium thickness (S27 and S28 Text and Table 1). We obtained activation thresholds over a similar range of amplitudes; however, our data show much steeper recruitment curves, with a smaller amplitude range between onset and saturation thresholds for each fiber diameter.

## 6 Availability and future directions

The ASCENT pipeline is publicly available for download to any Linux, Mac or Windows machine: the release associated with this paper (v1.0.0), as well as other archived releases, are available through Zenodo (v1.0.0: <https://doi.org/10.5281/zenodo.5136631>; current release: <https://zenodo.org/badge/latestdoi/379064819>); the most recent commit—where there may be multiple commits per release—is available through GitHub (<https://github.com/wmglab-duke/ascent>). The code is documented in a git wiki associated with the code repository (<https://github.com/wmglab-duke/ascent/wiki>). The project is licensed under the MIT License for non-commercial use (see the LICENSE file in the root of the repository for details).

ASCENT will be used to design experimental and clinical interventions, to interpret experimental results, and as a component of integrated simulations with other models (e.g., organ system models). It will aid in analysis and design of peripheral nerve stimulation therapies including quantification of the effects of intra- and inter-species differences on nerve activation and block, as well as approaches to increase the selectivity and efficiency of activation and block with novel stimulation paradigms and electrode designs.

Future developments of the pipeline will address limitations, including relaxing the assumption of quasi-static conditions for approximating electric fields [16], accounting for the impedance of the electrode-tissue interface [42,43], incorporating variations in nerve cross section that occur along its length, and eliminating dependency on COMSOL. However, the difficulty of building, meshing, and solving such complex geometries should not be underestimated.

We intend for the ASCENT pipeline to become the standard resource for modeling peripheral nerve stimulation and thereby reduce duplication of efforts, improve accessibility of model-based analysis and design of nerve stimulation, encourage best-practices, and enable reproducibility of simulation data [15,44,45].

## Supporting information

**S1 Text. Appendix.** Metadata required to model an *in vivo* experiment using the ASCENT pipeline.

(PDF)

**S2 Text. Appendix.** Installation.

(PDF)

**S3 Text. Appendix.** ASCENT data hierarchy.

(PDF)

**S4 Text. Appendix.** Your first run.

(PDF)

**S5 Text. Appendix.** Template for methods reporting.

(PDF)

**S6 Text. Appendix.** Enums.

(PDF)

**S7 Text. Appendix.** JSON configuration files.

(PDF)

**S8 Text. Appendix.** JSON file parameter guide.

(PDF)

**S9 Text. Appendix.** Python utility classes.

(PDF)

**S10 Text. Appendix.** Submitting NEURON jobs.

(PDF)

**S11 Text. Appendix.** Morphology files.

(PDF)

**S12 Text. Appendix.** Python MockSample class for creating binary masks of nerve morphology.

(PDF)

**S13 Text. Appendix.** Python classes for representing nerve morphology (Sample).

(PDF)

**S14 Text. Appendix.** Creating sample-specific nerve morphologies in COMSOL.

(PDF)

**S15 Text. Appendix.** Micro-Leads cuff measurements.

(PDF)

**S16 Text. Appendix.** Library of part primitives for electrode contacts and cuffs.

(PDF)

**S17 Text. Appendix.** Creating custom preset cuffs from instances of part primitives.

(PDF)

**S18 Text. Appendix.** Creating new part primitives.

(PDF)

**S19 Text. Appendix.** Cuff placement on nerve.

(PDF)

**S20 Text. Appendix.** Fiberset.

(PDF)

**S21 Text. Appendix.** Implementation of NEURON fiber models.

(PDF)

**S22 Text. Appendix.** Simulation protocols.

(PDF)

**S23 Text. Appendix.** ModelWrapper class.

(PDF)

**S24 Text. Appendix.** Making geometries in COMSOL (Part class).

(PDF)

**S25 Text. Appendix.** Control of medium surrounding nerve and cuff electrode.

(PDF)

**S26 Text. Appendix.** Java utility classes.

(PDF)

**S27 Text. Appendix.** Defining and assigning materials in COMSOL.

(PDF)

**S28 Text. Appendix.** Definition of perineurium.

(PDF)

**S29 Text. Appendix.** Data interchange between COMSOL and NEURON.

(PDF)

**S30 Text. Appendix.** Python Simulation class.

(PDF)

**S31 Text. Appendix.** NEURON launch.hoc.

(PDF)

**S32 Text. Appendix.** NEURON Wrapper.hoc.

(PDF)

**S33 Text. Appendix.** Data analysis tools.

(PDF)

**S34 Text. Appendix.** Convergence analysis example\*.

(PDF)

**S35 Text. Appendix.** Sim4Life validation.

(PDF)

**S36 Text. Appendix.** Comparison of MRG fit to Bucksot et al. 2019.

(PDF)

## Acknowledgments

Thanks to our beta-testers (Calvin Eiber, Chance Fleeting, John Gilbert, Minhaj Hussain, Karthik Kumaravelu, Shreya Kurdukar, Edgar Peña, Matthew Ward) for very generously testing and providing feedback on the pipeline.

Thanks to Edgar Peña and Aman Aberra for contributing to the interpolated fits of the original MRG myelinated fiber models.

Thanks to Christopher Davis and Liane Wong for providing measurements of the Micro-Leads cuffs.

Thanks to Brandon Thio for helping to implement an earlier version of the pipeline code.

Thanks to Antonino Cassara and the IT<sup>2</sup>IS team for helping to verify pipeline thresholds with Sim4Life's software.

Thanks to Christopher Davis for setting up the GitHub wiki.

## Author Contributions

**Conceptualization:** Eric D. Musselman, Jake E. Cariello, Warren M. Grill, Nicole A. Pelot.

**Data curation:** Eric D. Musselman, Jake E. Cariello.

**Formal analysis:** Eric D. Musselman, Jake E. Cariello.

**Funding acquisition:** Warren M. Grill, Nicole A. Pelot.

**Investigation:** Eric D. Musselman, Jake E. Cariello.

**Methodology:** Eric D. Musselman, Jake E. Cariello, Warren M. Grill, Nicole A. Pelot.

**Project administration:** Warren M. Grill, Nicole A. Pelot.

**Resources:** Warren M. Grill, Nicole A. Pelot.

**Software:** Eric D. Musselman, Jake E. Cariello, Nicole A. Pelot.

**Supervision:** Warren M. Grill, Nicole A. Pelot.

**Validation:** Eric D. Musselman, Jake E. Cariello.

**Visualization:** Eric D. Musselman, Jake E. Cariello.

**Writing – original draft:** Eric D. Musselman, Jake E. Cariello.

**Writing – review & editing:** Eric D. Musselman, Jake E. Cariello, Warren M. Grill, Nicole A. Pelot.

## References

1. Handforth A, DeGiorgio CM, Schachter SC, Uthman BM, Naritoku DK, Tecoma ES, et al. Vagus nerve stimulation therapy for partial-onset seizures: a randomized active-control trial. *Neurology*. 1998 Jul; 51(1):48–55. Available from: <https://doi.org/10.1212/wnl.51.1.48> PMID: 9674777
2. Ben-Menachem E, Manon-Espaillat R, Ristanovic R, Wilder BJ, Stefan H, Mirza W, et al. Vagus nerve stimulation for treatment of partial seizures: 1. A controlled study of effect on seizures. *First International Vagus Nerve Stimulation Study Group. Epilepsia*. 1994; 35(3):616–26. Available from: <https://doi.org/10.1111/j.1528-1157.1994.tb02482.x> PMID: 8026408
3. A randomized controlled trial of chronic vagus nerve stimulation for treatment of medically intractable seizures. *The Vagus Nerve Stimulation Study Group. Neurology*. 1995 Feb; 45(2):224–30. Available from: <https://doi.org/10.1212/wnl.45.2.224> PMID: 7854516
4. Marangell LB, Rush AJ, George MS, Sackeim HA, Johnson CR, Husain MM, et al. Vagus nerve stimulation (VNS) for major depressive episodes: one year outcomes. *Biol Psychiatry [Internet]*. 2002 Feb 15; 51(4):280–7. Available from: [https://doi.org/10.1016/s0006-3223\(01\)01343-9](https://doi.org/10.1016/s0006-3223(01)01343-9) PMID: 11958778
5. Nierenberg AA, Alpert JE, Gardner-Schuster EE, Seay S, Mischoulon D. Vagus Nerve Stimulation: 2-Year Outcomes for Bipolar Versus Unipolar Treatment-Resistant Depression. *Biol Psychiatry [Internet]*. 2008 Sep 15; 64(6):455–60. Available from: <https://doi.org/10.1016/j.biopsych.2008.04.036> PMID: 18571625
6. Rush AJ, George MS, Sackeim HA, Marangell LB, Husain MM, Giller C, et al. Vagus nerve stimulation (VNS) for treatment-resistant depressions: a multicenter study. *Biol Psychiatry*. 2000 Feb; 47(4):276–86. Available from: [https://doi.org/10.1016/s0006-3223\(99\)00304-2](https://doi.org/10.1016/s0006-3223(99)00304-2) PMID: 10686262

7. Birmingham K, Gradinaru V, Anikeeva P, Grill WM, Pikov V, McLaughlin B, et al. Bioelectronic medicines: a research roadmap. *Nat Rev Drug Discov* [Internet]. 2014; 13(6):399–400. Available from: <https://doi.org/10.1038/nrd4351> PMID: 24875080
8. Johnson RL, Wilson CG. A review of vagus nerve stimulation as a therapeutic intervention. *J Inflamm Res* [Internet]. 2018 May 16; 11:203–13. Available from: <https://pubmed.ncbi.nlm.nih.gov/29844694> <https://doi.org/10.2147/JIR.S163248> PMID: 29844694
9. Lubba CH, Le Guen Y, Jarvis S, Jones NS, Cork SC, Eftekhari A, et al. PyPNS: Multiscale Simulation of a Peripheral Nerve in Python. *Neuroinformatics* [Internet]. 2019; 17(1):63–81. Available from: <https://doi.org/10.1007/s12021-018-9383-z> PMID: 29948844
10. Pelot NA, Behrend CE, Grill WM. Modeling the response of small myelinated axons in a compound nerve to kilohertz frequency signals. *J Neural Eng*. 2017 Aug; 14(4):46022. Available from: <https://iopscience.iop.org/article/10.1088/1741-2552/aa6a5f> PMID: 28361793
11. Bucksot JE, Wells AJ, Rahebi KC, Sivaji V, Romero-Ortega M, Kilgard MP, et al. Flat electrode contacts for vagus nerve stimulation. *PLoS One* [Internet]. 2019; 14(11):1–22. Available from: <https://doi.org/10.1371/journal.pone.0215191> PMID: 31738766
12. Helmers SL, Begnaud J, Cowley A, Corwin HM, Edwards JC, Holder DL, et al. Application of a computational model of vagus nerve stimulation. *Acta Neurol Scand* [Internet]. 2012; 126(5):336–43. Available from: <https://doi.org/10.1111/j.1600-0404.2012.01656.x> PMID: 22360378
13. Zelechowski M, Valle G, Raspopovic S. A computational model to design neural interfaces for lower-limb sensory neuroprostheses. *J Neuroeng Rehabil*. 2020 Feb; 17(1):24. Available from: <https://doi.org/10.1186/s12984-020-00657-7> PMID: 32075654
14. Romeni S, Valle G, Mazzoni A, Micera S. Tutorial: a computational framework for the design and optimization of peripheral neural interfaces. *Nat Protoc* [Internet]. 2020; 15(10):3129–53. Available from: <https://doi.org/10.1038/s41596-020-0377-6> PMID: 32989306
15. Wilkinson MD, Dumontier M, Aalbersberg IJJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci data*. 2016 Mar; 3:160018. Available from: <https://doi.org/10.1038/sdata.2016.18> PMID: 26978244
16. Bossetti CA, Birdno MJ, Grill WM. Analysis of the quasi-static approximation for calculating potentials generated by neural stimulation. *J Neural Eng*. 2008 Mar; 5(1):44–53. Available from: <https://doi.org/10.1088/1741-2560/5/1/005> PMID: 18310810
17. McNeal DR. Analysis of a Model for Excitation of Myelinated Nerve. *IEEE Trans Biomed Eng*. 1976; BME-23(4):329–37. Available from: <https://doi.org/10.1109/TBME.1976.324593> PMID: 1278925
18. Van Rossum G, Drake FL. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
19. Arnold K, Gosling J, Holmes D. *Java(TM) Programming Language, The (4th Edition)*. Addison-Wesley Professional; 2005.
20. Hines ML, Carnevale NT. The NEURON simulation environment. *Neural Comput*. 1997 Aug; 9(6):1179–209. Available from: <https://doi.org/10.1162/neco.1997.9.6.1179> PMID: 9248061
21. Pelot NA, Goldhagen GB, Cariello JE, Musselman ED, Clissold KA, Ezzell JA, et al. Quantified Morphology of the Cervical and Subdiaphragmatic Vagus Nerves of Human, Pig, and Rat. *Front Neurosci* [Internet]. 2020; 14:1148. Available from: <https://doi.org/10.3389/fnins.2020.601479> PMID: 33250710
22. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods* [Internet]. 2020; 17(3):261–72. Available from: <https://doi.org/10.1038/s41592-019-0686-2> PMID: 32015543
23. Oliphant TE. *A Guide to NumPy* [Internet]. Trelgol Publishing; 2006. Available from: <https://books.google.com/books?id=fKulSgAACAAJ>
24. Callister WD, Rethwisch DG. *Fundamentals of Material Science and Engineering An Integrated Approach*. In: *Fundamentals Of Material Science and Engineering An Intergrated Approach*. 2012.
25. de Podesta M, Laboratory NP, UK. *Understanding the Properties of Matter*. Understanding the Properties of Matter. 1996.
26. Ranck JB, BeMent SL. The specific impedance of the dorsal columns of cat: An anisotropic medium. *Exp Neurol* [Internet]. 1965 Apr 1 [cited 2020 Apr 20]; 11(4):451–63. Available from: [https://doi.org/10.1016/0014-4886\(65\)90059-2](https://doi.org/10.1016/0014-4886(65)90059-2) PMID: 14278100
27. Pelot NA, Behrend CE, Grill WM. On the parameters used in finite element modeling of compound peripheral nerves. *J Neural Eng* [Internet]. 2019; 16(1):16007. Available from: <https://doi.org/10.1088/1741-2552/aaeb0c> PMID: 30507555
28. Grill WM, Mortimer TJ. Electrical properties of implant encapsulation tissue. *Ann Biomed Eng* [Internet]. 1994; 22(1):23–33. Available from: <https://doi.org/10.1007/BF02368219> PMID: 8060024

29. Gielen FLH, Wallinga-de Jonge W, Boon KL. Electrical conductivity of skeletal muscle tissue: Experimental results from different muscles in vivo. *Med Biol Eng Comput* [Internet]. 1984; 22(6):569–77. Available from: <https://doi.org/10.1007/BF02443872> PMID: 6503387
30. Geddes LA, Baker LE. The specific resistance of biological material—A compendium of data for the biomedical engineer and physiologist. *Med Biol Eng* [Internet]. 1967; 5(3):271–93. Available from: <https://doi.org/10.1007/BF02474537> PMID: 6068939
31. Horch K. *Neuroprosthetics: Theory and practice: Second edition*. Neuroprosthetics: Theory and Practice: Second Edition. 2017. 1–925 p.
32. Pelot NA, Thio BJ, Grill WM. Modeling Current Sources for Neural Stimulation in COMSOL. *Front Comput Neurosci* [Internet]. 2018; 12:40. Available from: <https://doi.org/10.3389/fncom.2018.00040> PMID: 29937722
33. McIntyre CC, Grill WM, Sherman DL, Thakor N V. Cellular effects of deep brain stimulation: model-based analysis of activation and inhibition. *J Neurophysiol*. 2004 Apr; 91(4):1457–69. Available from: <https://doi.org/10.1152/jn.00989.2003> PMID: 14668299
34. McIntyre CC, Richardson AG, Grill WM. Modeling the excitability of mammalian nerve fibers: influence of afterpotentials on the recovery cycle. *J Neurophysiol*. 2002 Feb; 87(2):995–1006. Available from: <https://doi.org/10.1152/jn.00353.2001> PMID: 11826063
35. Sundt D, Gamper N, Jaffe DB. Spike propagation through the dorsal root ganglia in an unmyelinated sensory neuron: a modeling study. *J Neurophysiol*. 2015 Dec; 114(6):3140–53. Available from: <https://doi.org/10.1152/jn.00226.2015> PMID: 26334005
36. Tigerholm J, Petersson ME, Obreja O, Lampert A, Carr R, Schmelz M, et al. Modeling activity-dependent changes of axonal spike conduction in primary afferent C-nociceptors. *J Neurophysiol*. 2014 May; 111(9):1721–35. Available from: <https://doi.org/10.1152/jn.00777.2012> PMID: 24371290
37. Rattay F, Aberham M. Modeling axon membranes for functional electrical stimulation. *IEEE Trans Biomed Eng*. 1993 Dec; 40(12):1201–9. Available from: <https://doi.org/10.1109/10.250575> PMID: 8125496
38. Howell B, Grill WM. Evaluation of high-perimeter electrode designs for deep brain stimulation. *J Neural Eng* [Internet]. 2014/07/16. 2014 Aug; 11(4):46026. Available from: <https://doi.org/10.1088/1741-2560/11/4/046026> PMID: 25029124
39. Arle JE, Carlson KW, Mei L. Investigation of mechanisms of vagus nerve stimulation for seizure using finite element modeling. *Epilepsy Res* [Internet]. 2016; 126:109–18. Available from: <https://doi.org/10.1016/j.eplepsyres.2016.07.009> PMID: 27484491
40. Yoo PB, Lubock NB, Hincapie JG, Ruble SB, Hamann JJ, Grill WM. High-resolution measurement of electrically-evoked vagus nerve activity in the anesthetized dog. *J Neural Eng*. 2013 Apr; 10(2):26003. Available from: <https://doi.org/10.1088/1741-2560/10/2/026003> PMID: 23370017
41. Schneider CA, Rasband WS, Eliceiri KW. NIH ImageJ: 25 years of image analysis. *Nat Methods* [Internet]. 2012; 9(7):671–5. Available from: <https://doi.org/10.1038/nmeth.2089> PMID: 22930834
42. Newman J. Current Distribution on a Rotating Disk below the Limiting Current. *J Electrochem Soc*. 1966; 113:1235–1241. Available from: <https://doi.org/10.1149/1.2423795>
43. Cantrell DR, Inayat S, Taflove A, Ruoff RS, Troy JB. Incorporation of the electrode—electrolyte interface into finite-element models of metal microelectrodes. *J Neural Eng* [Internet]. 2007 Dec; 5(1):54–67. Available from: <https://doi.org/10.1088/1741-2560/5/1/006> PMID: 18310811
44. Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, et al. Best Practices for Scientific Computing. *PLOS Biol* [Internet]. 2014; 12(1):1–7. Available from: <https://doi.org/10.1371/journal.pbio.1001745> PMID: 24415924
45. Wilson G, Bryan J, Cranston K, Kitzes J, Nederbragt L, Teal TK. Good enough practices in scientific computing. *PLoS Comput Biol*. 2017 Jun; 13(6):e1005510. Available from: <https://doi.org/10.1371/journal.pcbi.1005510> PMID: 28640806