Structure of the Afferent Terminals in Terminal Ganglion of a Cricket and Persistent Homology

Jacob Brown, Tomáš Gedeon*

Department of Mathematical Sciences, Center for Computational Biology, Montana State University, Bozeman, Montana, United States of America

Abstract

We use topological data analysis to investigate the three dimensional spatial structure of the locus of afferent neuron terminals in crickets *Acheta domesticus*. Each afferent neuron innervates a filiform hair positioned on a cercus: a protruding appendage at the rear of the animal. The hairs transduce air motion to the neuron signal that is used by a cricket to respond to the environment. We stratify the hairs (and the corresponding afferent terminals) into classes depending on hair length, along with position. Our analysis uncovers significant structure in the relative position of these terminal classes and suggests the functional relevance of this structure. Our method is very robust to the presence of significant experimental and developmental noise. It can be used to analyze a wide range of other point cloud data sets.

Citation: Brown J, Gedeon T (2012) Structure of the Afferent Terminals in Terminal Ganglion of a Cricket and Persistent Homology. PLoS ONE 7(5): e37278. doi:10.1371/journal.pone.0037278

Editor: Manuel S. Malmierca, University of Salamanca-Institute for Neuroscience of Castille and Leon and Medical School, Spain

Received January 9, 2012; **Accepted** April 15, 2012; **Published** May 23, 2012

Funding: This work has been partially supported by NSF (National Science Foundation) grants DMS-081878 and CMMI-0849433, and a Defense Advanced Research Project Agency (DARPA) grant 00001741/HR0011-09-1-0055. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. No additional external funding received for this study.

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

* E-mail: gedeon@math.montana.edu

Introduction

One of the most pressing issues in biology in general, and particularly in neuroscience, is the development of computational methods that can extract relevant information from noisy data. We are currently in a situation where there is still a lack of high precision quantitative data, but often a relative abundance of noisy and imprecise data. In the present work, we show how one can use ideas from persistent homology [1,2,3] to analyze the three dimensional spatial structure of the afferent neurons locus in the cercal system of a cricket. This method is applicable to any point cloud data set that exhibits an underlying topological structure corrupted by noise.

The cercal system in a house cricket *Acheta domesticus* (Figure 1) is a near-field flow sensor that senses fluid particle motion via an array of very thin mechanosensory hairs called filiform hairs. These hairs are distributed along two antenna-like, shallowly-tapering appendages called cerci at the rear of the abdomen. Different hairs have different directional and frequency sensitivities that are determined by the biomechanical properties of the hair and its socket. Each hair is innervated by a single neuron, whose axon projects into the terminal ganglion located near the rear end of the animal. Each axon forks into a tree-like structure. Individual branches are covered with synaptic boutons that make synaptic contact with one of the ganglion's interneurons. By staining the afferent neuron and imaging with a scanning microscope, one can locate the three dimensional coordinates of the set of synaptic boutons for a particular hair. Hairs are functionally characterized by the orientation of their socket, which determines the orientation of air motion that this hair mechanically responds to; by the length of the hair, which determines the sensitivity to frequency of air motion; and by the position on the cercus, which determines the latency of arrival of the signal to the terminal ganglion. By superimposing the sets of synaptic terminals of all hairs in all categories, we obtain the overall afferent synaptic locus. The goal of this paper is to study the topological structure of this locus and its stratification by the hair directional sensitivity and length.

1.1. Structure of the afferent locus

The initial experiments and analysis of the structure of the afferent locus was done in a series of papers by Jacobs and collaborators [4,5,6]. The first paper constructed an anatomical database consisting of twelve different identified sensory afferents for five specimens, which spanned the entire range of directional tunings seen in the system. All twelve afferents were associated with the longest mechanoreceptors ($>900\mu$m) and, as a result of experimental accessibility, were selected from the proximal part of the cercus (i.e the initial 15% of the length from the base of the cercus). Jacobs and Theunissen [4] have shown that the locus of the afferent terminals changes continuously with the directional tuning of the corresponding hairs. In the second paper [5] the database was extended to include medium hairs ($500-900\mu$m) from the proximal part of the cercus. Their research indicates that there is no significant statistical difference between the positions of the terminals of medium hairs and those of long hairs. Finally, in an elegant paper, Jacobs and Theunissen [6] showed how the directional tuning curves of four interneurons that are downstream from the afferents arise from the overlap of the dendritic trees of the interneurons with the afferent terminal locus. In particular, the interneuron sensitivity to the motion from direction θ arises from the connectivity of its dendrite primarily with the terminals of afferents that innervate hairs that mechanically respond to the direction θ.

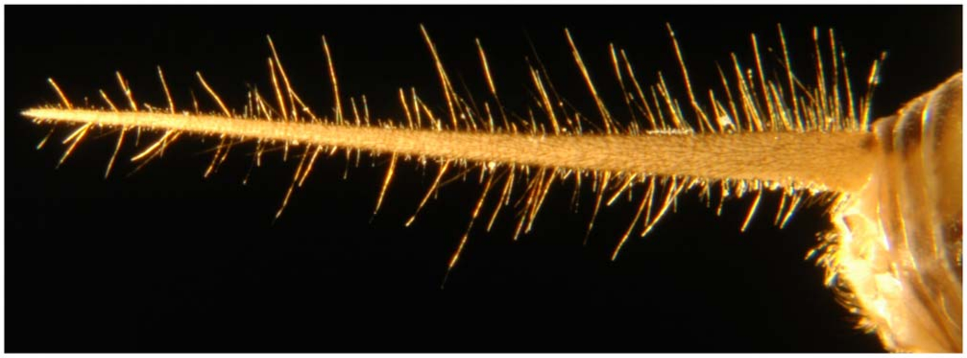

Figure 1. The cercus of *Acheta Domestica* with filiform hairs clearly visible. The length of the cercus is approximately 1 cm.
doi:10.1371/journal.pone.0037278.g001

1.2. Statement of the problem

The locus of afferent terminals in the terminal ganglion of a cricket is a well-defined object with complex structure. Afferent terminals of hairs that respond to air movement from different directions $\theta_1 \neq \theta_2$ map to different positions. As θ sweeps all angles, this position continuously changes [4]. Each cercus hair is distinguished by its response angle, by its position on the cercus and by its length. Our goal is to describe how the structure of the ganglion is stratified by hair position and hair length. This structure, superimposed on the dendrite structure of the downstream interneurons, constrains the response characteristics of these interneurons [6]. This result uses only the directional response characterization of the hairs and relies on the fact that the afferent terminals corresponding to similarly oriented hairs clearly cluster in the same location (illustrated with the color coding in Figure 2). Is the position of the hair along the cercus and the length of the hair similarly encoded in the structure of the afferent terminals locus? If so, then the interneurons connected to afferents in a particular location will receive input from a class of hairs of a certain length or a certain position on the cercus, and thus this information is available to the interneurons. We note that even if such a structure does not exist, it would still be possible for interneurons to connect preferentially to specific classes of afferents, but the developmental control of the connection process would have to be very complex. A stratification by hair length shows that the structure is more complex, and harder to interpret, than the stratification by response direction. Therefore, more sophisticated methods are required to reveal the stratification structure with respect to the length and position of the hairs.

A casual look at the data set (see Figure S2 for a full three dimensional data set, which can be rotated using an appropriate software) reveals that the defining characteristics of the set are the number and size of the voids, or tunnels, in the point cloud. The data sets (short, medium and long proximal hair terminals; long distal hair terminals), which we will present in this paper, all have a vaguely similar overall shape and structure. However, as we put together different combinations of these data sets, we seem to get varying numbers and sizes of tunnels. How do we quantify the number and size of tunnels in a data set consisting entirely of points? The answer to this question is complicated by the uncertainty in the data, which had been collected from multiple animals and multiple hairs. How do we decide whether the medium and short hair collections map to the same area or not, especially if both the medium and short hairs came from different animals and were themselves all of different lengths? Therefore the need for a robust method with respect to noise is balanced by the need for a sensitive method that is capable of detecting small changes on a local scale, which may have a large effect on a scale of the entire locus. If the terminals of the medium hairs are slightly shifted with respect to terminals of the short hairs, then this small local change may have a large effect on significantly diminishing the size of the tunnel in the terminals of the short data set.

The last challenge stems from the fact that some of the holes and tunnels in the terminal clouds are a result of the functional constraints of the terminal ganglion. Since dendrites of the downstream interneurons must access the terminals of the afferents to make synaptic contacts, some, or probably most, of the tunnels are access points of these dendrites. Therefore, we have to be able to distinguish between the non-essential tunnels, which serve as access points to the afferent terminals, and the essential tunnels, which are a consequence of removing a particular class of terminals.

In view of these constraints and challenges, we identify the essential tunnels in the point cloud data sets as those that correspond to the persistent homology of the data sets. As is explained next, each essential tunnel is in one-to-one correspondence with a circle lying within the data cloud that surrounds the tunnel.

1.3. Persistent Homology

In this subsection we outline in lay terms our basic approach to data analysis of a point cloud. We will leave a more formal definition of the concepts of homology and persistence for Section 4.2. For the purpose of an introduction, it is sufficient to note that the homology calculation for each set S in R^3 will efficiently compute three groups of embedded objects known as generators. The generators in each of these groups are topologically distinct, which means that one cannot deform any given generator onto any other generator within the set S. The generators in the first group (the 0th homology group) represent connected pieces of the set S. We note that each component of S will have one generator of the first kind embedded in it. The generators of the second group (1st homology) are circles that circumscribe tunnels in the set S. Finally, the generators of the third group (2nd homology) are represented by spheres that surround voids in S. The number of generators in each of these three groups are referred to as *Betti* numbers, and denoted β_0, β_1 and β_2, respectively.

It is not a priori clear for how one can apply the concepts of homology to a data set in the form of a point cloud in a nontrivial way. The homology of a point cloud, when considered as a topological space, is entirely straightforward: the number of 0^{th} homology generators is equal to the number of points, and there are clearly no higher order generators (circles, or spheres). Imagine, however, that if the points of a point cloud were drawn

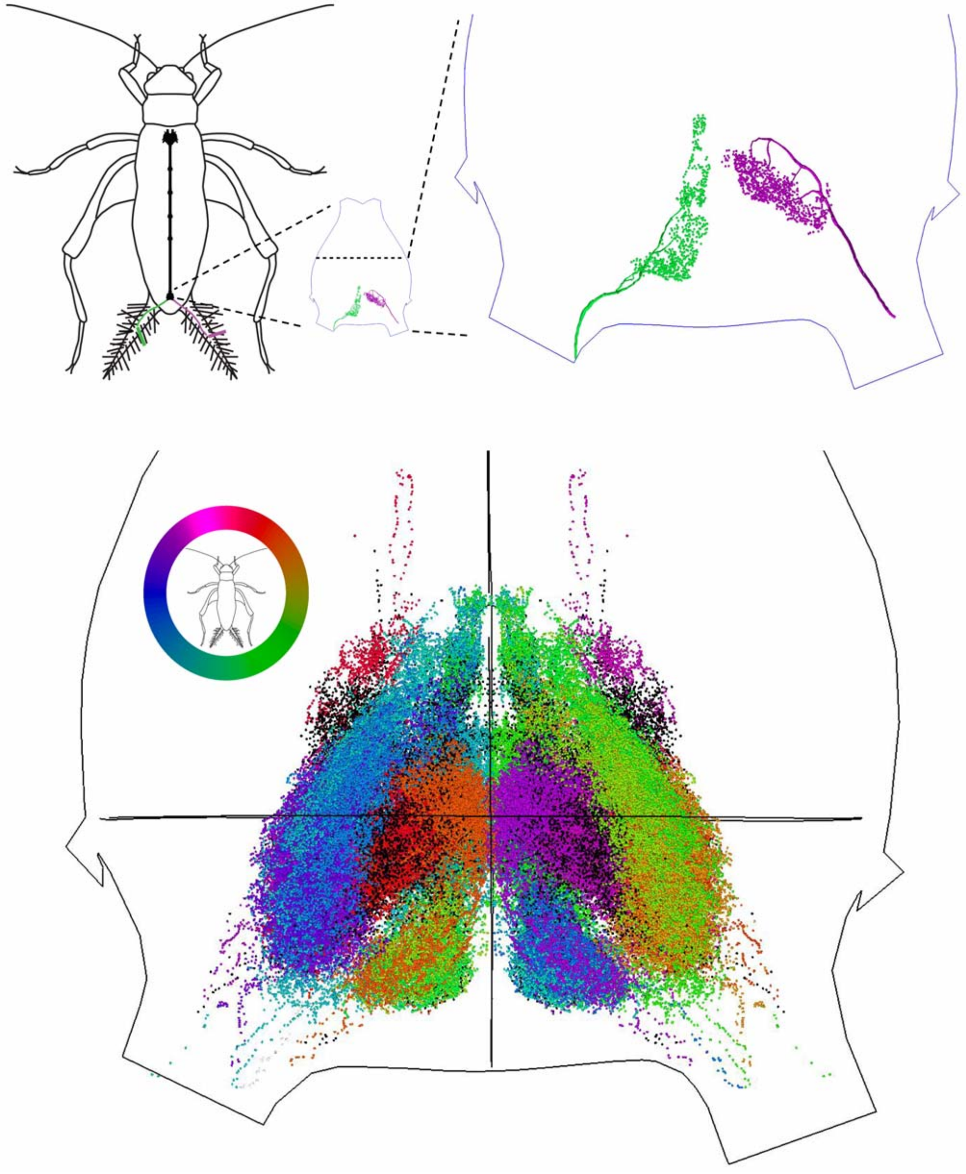

Figure 2. The afferent terminals locus of *Acheta Domestica*. (Upper left) Axons of afferent neurons, which are attached to filiform hairs, project into the terminal ganglion. (Upper middle and upper right) Each axon branches into a tree-like structure in which individual branches are covered by synaptic boutons that provide connections with interneurons in the terminal ganglion. (Bottom) The entire collection of synaptic boutons over all experimentally examined filiform hairs. The color wheel represents the preferred direction of motion of a hair, which correspondingly represents the preferred response direction of the afferent neurons. (Data generously shared by G. Jacobs and collaborators).
doi:10.1371/journal.pone.0037278.g002

from a distribution that is centered around a large circle with a standard deviation that is less than, but comparable to, the radius of that circle. Then, can we use the ideas from homology to "discover" this circle from the collection of this noisy data set? The idea of topological persistence [2,3] elegantly addresses this issue. We introduce a parameter ϵ and perform the following construction for a sequence $\epsilon_0 < \epsilon_1 < \ldots\ldots < \epsilon_{n-1} < \epsilon_n$ of increasing values of ϵ. We center a box of the dimension of the ambient space and side ϵ_i over each data point. We call the union of these boxes a *complex* K_{ϵ_i} and compute its homology. We note that if $\epsilon_i < \epsilon_j$ then $K_{\epsilon_i} \subset K_{\epsilon_j}$. While we refer the reader for detailed definitions of persistent homology and persistent generators to the original literature ([1,3], see also section 4.2), the main idea is very intuitive and is captured in Figure 3. For a very small ϵ_i the topology, and hence homology, of the complex K_{ϵ_i} will be identical to that of the underlying point cloud. But as we increase ϵ_i, some of the boxes will start to intersect, and as a consequence, β_0 will start to decrease. At some value of ϵ_i, the complex K_{ϵ_i} will include a generator of the first homology group that corresponds to the underlying circle of the sample distribution. As ϵ_i changes, we can track the behavior of generators as a function of ϵ_i. In particular, under very general conditions (see section 4.2) it is possible to pair births and deaths of generators in such a way that each generator born at some value ϵ_i disappears at some larger value ϵ_j. We call the difference $j-i$ a *lifespan* of that particular generator and $[i,j]$ the *persistence interval*. The information about lifespans can be encoded as a *barcode* of a particular point cloud and it can be used to distinguish essential topological features of the cloud from spurious features introduced by experimental and measurement noise. In particular, generators that capture essential topological features will have distinguishably longer lifespans than those that characterize spurious features. These are called *persistent generators*. We will use persistent first homology generators to understand the robust structure of the locus of afferent terminals.

It is worth noting that there is no data-independent threshold that one can use to separate persistent and non-persistent generators. It is only in the context of a given data set that one can define some generators persistent, and this choice is necessarily arbitrary. Our choice led to a conclusion that the combined set of all afferents does not have any persistent homology (i.e. any essential tunnels). This is an attractive conclusion, which is consistent with all our other computations, and which has interesting consequences that we elaborate upon in the discussion. However, this conclusion is ultimately the result of an arbitrary cut-off that separates persistent and non-persistent generators.

Results

Before presenting our results in terms of persistent first homology generators, we summarize the results in terms of essential tunnels in the point cloud data set. An essential tunnel is encircled by a persistent generator of the first homology group.

Our computations reveal that the afferent terminals of both short and medium hairs have three essential tunnels. However, when we combine the short and medium data sets, the resulting set contains only two essential tunnels. Therefore, the three tunnels in the medium set are not the same as the three tunnels in the short set. This means that at least one tunnel in the short set is filled (or rendered non-essential) by the medium set; and at least one tunnel in the medium set is filled by the short set.

The long proximal set has four essential tunnels; whereas, the combined set of long proximal and long distal afferent terminals contains only two essential tunnels. Therefore, the long distal set fills two tunnels in the long proximal set. Our long distal data set contains substantially less points than the long proximal set. Consequently, it is entirely possible that if we had more data for the long distal set, one or both of the remaining tunnels in the long proximal set would be filled. To address this issue we construct a Gaussian Mixture Model (GMM) for both long proximal and long distal hairs. We then sample these models to create an enhanced set of terminals. The analysis with the sampled data yields the same results as analysis of the experimental data: the combined sampled set contains two essential tunnels.

A combined proximal set, consisting of afferent terminals of all hair lengths (short, medium and long) has only one essential tunnel. This means that the addition of the long proximal class to the combined, short and medium, class fills one essential tunnel.

Finally, the set of all terminals has no essential tunnels remaining. This means that the terminals of the distal hairs, while sparse, fill in the last essential tunnel in the combined set of proximal afferent terminals. We obtain the same result by using either the sparse experimental distal data or substantially larger distal data sampled from a GMM model. As we have noted before, there are still multiple tunnels left in the set of terminals, but these are not essential and likely serve as the interneuron dendrite access points.

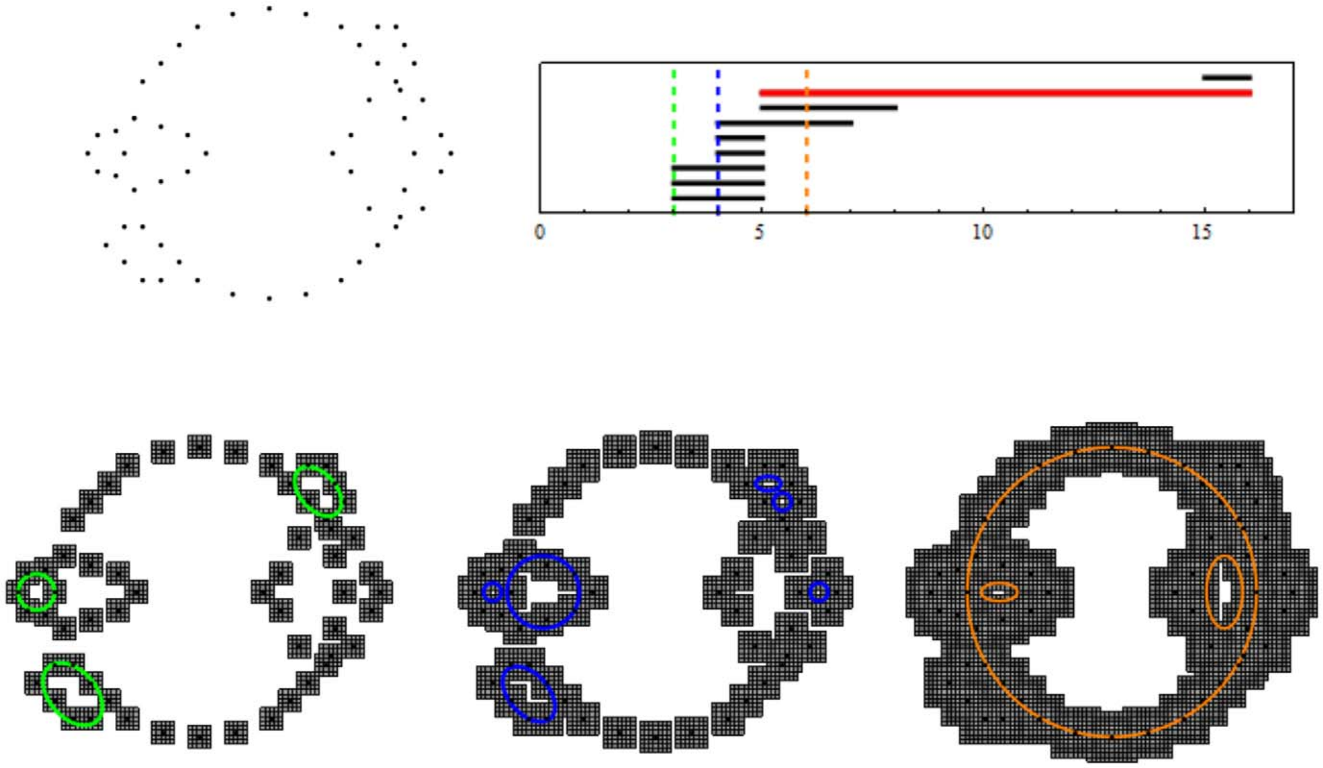

Figure 3. Persistent cubical homology. (Top, left) An example of a point cloud. (Bottom, left to right) As the size of the squares (in arbitrary units) around each point in the point cloud increases from 3 (left), through 4 (center) to 6 (right), different holes open and close in the gray set of squares. The colored circles around the holes represent generators of the first homology group, which are graphed as a function of the size in the barcode (top, right). The dashed lines are color coded and correspond to the figures in the bottom row. The red line in the barcode represents a *persistent* generator which indicates a hole that is present for a substantial range of sizes of the squares.
doi:10.1371/journal.pone.0037278.g003

2.1. Data modeling and filtering

We analyze the database of afferent terminals that is based on the work of Jacobs and collaborators [4,5,6]. Recently more data has been added to the database, and some of the data has been refined and corrected. We obtained permission from G. Jacobs and J. P. Miller to use this database.

The total number of data points in the terminal locus is divided into two large categories: terminals of afferents from proximal hairs [4,5] and, more recently added, terminals of afferents from distal hairs. The proximal hairs are located within the nearest 15% of the length of the cercus to its base, whereas the distal hairs are located beyond the nearest 30% of the length [7]. We will label the corresponding data sets *proximal* and *distal*. The proximal data set is further divided by the length of the contributing hairs into three subclasses, which we will label as long ($>900\mu$m), medium ($500-900\mu$m) and short ($\sim 50-450\mu$m) [4,5,6]. The proximal data set totals to 99167 data points divided among the three length subclasses. There are 42428 data points representing the terminals of afferents attached to long hairs, 27442 points for those attached to medium hairs and 29297 points for those attached to short hairs. In addition to length, each hair responds preferentially to a particular direction of air motion. We split these directions into 13 classes and assign to each data point the class of the corresponding hair.

Distal hairs are all long and are sparser than any other data set. Therefore, we split the preferred angles of the distal hairs into only 4 directional classes. Since the taper of the cercus makes access to afferents in the distal part of the cercus difficult, the database only contains 6194 distal hair data points. In order to address the discrepancy between the number of proximal and distal points, we have developed a Gaussian Mixture Model (GMM) that uses one GMM per each of 13 directional classes for long proximal hairs, one GMM per each of 13 directional classes for medium proximal hairs and one GMM per each of 12 directional classes for small proximal hairs (one class has no data points). We used 4 GMM's for the distal hairs: one per each direction class. This modeling has allowed us to sample additional distal points as to supplement the sparse experimental distal data.

2.2. Data filtering

We apply preliminary filtering to the data sets in order to accomplish two goals. The first stems from our main goal of trying to describe topological features of the point clouds given by the afferents of different classes of hairs. These features are defined by the location of the higher density region of the point cloud. Therefore, we aim to reduce the number of outliers that are caused by experimental and/or developmental noise. The presence of such outliers, in what would otherwise be void areas of space, can decrease the lifespans of persistent generators. The second goal of our filtering is to remove redundant points from the dense regions of our point cloud. These points, generally, do not affect the overall shape and topological features of the set; however, their inclusion can significantly slow down the computations.

Our filter $F_{k,j}$ has two parameters k and j. When applied to a point set, it will keep those points that have at least j neighboring points within a distance of k μm. This filter is described in more detail in section 4.1. We experimented with many different choices of k and j. Our goal was to find a combination of k and j, that would delete $10\%-30\%$ of the least dense points in each of the long proximal, medium proximal and short proximal categories, while keeping a percentage of data points that was similar for each. We chose the filtering function $F_{6,6}$, which keeps points having at least $j=6$ neighboring points within a distance of $k=6\mu$m. This choice led to a reduced long proximal set with 33950 points (which is 80% of the original), a medium proximal set with 21039 points (76%) and a short proximal set with 22753 points (77.7%). For all data sets, unless otherwise noted, we have applied the $F_{6,6}$. Thus, the long, medium and short data labels, will refer to these filtered data sets for the remainder of this paper.

Our second justification of the use of filter $F_{6,6}$ was achieved through a comparison of the resulting persistent generators of the long proximal data set with those of the reduced long proximal data set. This comparison is given in Table 1. The table has the following layout, shared by all subsequent tables in the paper: *data* refers to the input data set, with *length* referring to the length of the lifespan of generators throughout the filtration. The columns, labeled with a number n, record the number of generators in the filtration of each corresponding data set that had a lifespan of exactly n. We observe that the proximal long data set has 4 generators, which have distinguishably longer lifespans (three of length 11 and one of length 14). The filtered proximal long data set retains all four persistent generators. Furthermore, the gap in the lifespan length between these four generators and that of any other generator has been widened. This indicates that the filtering works as desired: enhancing the features already present in the data.

2.3. Topological structure of afferent terminals

We describe the topological structure of the terminals of afferents by computing persistent generators for various subsets. These subsets are formed in four steps.

(1) We form reduced (i.e. filtered) sets consisting of long proximal, medium proximal and short proximal hairs separately.

(2) We combine the sets of short and medium hairs.

(3) We combine together long proximal and (long) distal hairs. However, since the number of data points is much smaller for distal hairs, we will use both direct comparison of the reduced data sets, as well as a comparison using data sampled from a Gaussian Mixture Model.

(4) We put together experimental data sets from (2) and (3) to reconstruct the entire afferent terminus.

There are two natural ways to combine any two sets of data. The first approach is to combine the original data sets, then reduce this combined data set using the $F_{6,6}$ reduction. The second approach is to combine the reduced data sets; that is, we apply the $F_{6,6}$ reduction to each individual set, then combine the reduced data sets. While we will not display results for both approaches, we have found that there are minimal differences between the results of either reduction approach. In particular, the number of persistent generators is the same for each approach. Therefore, throughout the rest of the paper we will combine the data sets using the second approach.

2.3.1. Topology of short, medium and long proximal data sets. We compute persistent generators of the reduced set of long proximal, medium proximal and short proximal hairs. The results are given in the following two forms (Figure 4): A table collecting the number of generators of a given length and a β_1-barcode that displays lifespans of each generator. In the β_1-barcode we highlight in red the persistent generators.

We interpret the existence of a persistent generator as evidence for the presence of a significant tunnel-like void in the data. The data presented in Figure 4 shows that

(1) The reduced long proximal set has 4 persistent generators with lifespans of 11, 13, 19 and 23.

Table 1. Comparison of filtered and unfiltered data.

Data\Length	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
UnFiltered Long	78	13	8	4	3	2	2	0	0	3	0	0	1	0	0	0	0	0	0	0	0	0
Filterd Long	15	11	4	3	1	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	1

The first column describes the data set, the top row is the length of the lifespan of generators. An integer entry k in the column labeled n indicates that there is k generators with a lifespan of exactly n.

doi:10.1371/journal.pone.0037278.t001

(2) The reduced medium proximal set has 3 persistent generators with lifespans of 11, 13 and 14.

(3) The reduced short proximal set has 3 persistent generators with lifespans of 12, 13 and 14.

Therefore, our results show that there are four significant tunnels in the long data set and three, each, in the short and medium data sets.

A natural question is whether the generators for the short and medium data sets are the same, and if, in addition, those generators are a subset of the generators of the long proximal set. To address this question, we combine the medium and short hairs into a single set, and then compute persistence. If the three generators of the short data set are equivalent (homologous) to the three generators of the medium data set, then the number of generators of the combined set will remain three; however, if some of these generators are homologous to zero in the combined set (i.e. the points in one set "fill the hole" in the other set), the combined data set may have a different number of persistent generators.

The persistence results of the combined medium+small data set, as displayed in Figure 5, show that two persistent generators remain after the combination of the two proximal sets. To illustrate what this information means for the data, we display the short, medium and combined point clouds with generators in Figure 6. The combined cloud is displayed in a different orientation in the Figure S1. The respective point clouds of the short and medium data sets seem to occupy approximately the same space, although appear to be slightly offset.

In conclusion, one of the persistent generators for the medium set is filled by the terminals from the short hairs; along with one of the persistent generators of the short data set being filled by the terminals of the medium hairs. This filling appears to be attributed to the two point clouds being slightly offset. The remaining tunnels, characterized by the orange and purple generators in each set, line up enough to allow for the corresponding two generators to remain persistent in the combined point cloud.

2.3.2. Topology of long proximal and long distal sets. Next we look at combining the reduced long proximal data with the distal data. The main issue we need to address is the large discrepancy between the number of points in the long proximal and long distal sets. Since there are more long proximal hairs than long distal hairs, some, but not all, of this difference can be attributed to experimental accessibility. Therefore, in addition to investigating the topology of the union of the two experimental data sets, we will also create a Gaussian Mixture Model (GMM) for both data sets. We then sample a large number of points from each desired data set, process this data set in the same way as the experimental proximal data and compute the persistence for each set separately, as well as for the combined set.

2.3.3. Comparison of experimental data sets of long hairs. Since the distal point cloud is made up of only 6194 data points, we do not perform any additional reductions on it. Instead, we combine the entire distal point cloud with the data set of the reduced long proximal hairs, then compute the persistence of the resulting set. As we see from the results in Figure 7, the combined point cloud has 2 persistent generators with lifespans of length 12 and 16. Recall that the proximal long point cloud has 4 persistent generators: two with longer lifespans (19 and 23) and two with shorter lifespans (11 and 13). The addition of the distal data destroys the two smaller persistent generators, while reducing the lifespans of the other two generators from 19 and 23 to 12 and 16.

Data\Length	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Long	15	11	4	3	1	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	1
Medium	41	12	6	1	3	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Short	48	9	3	2	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0

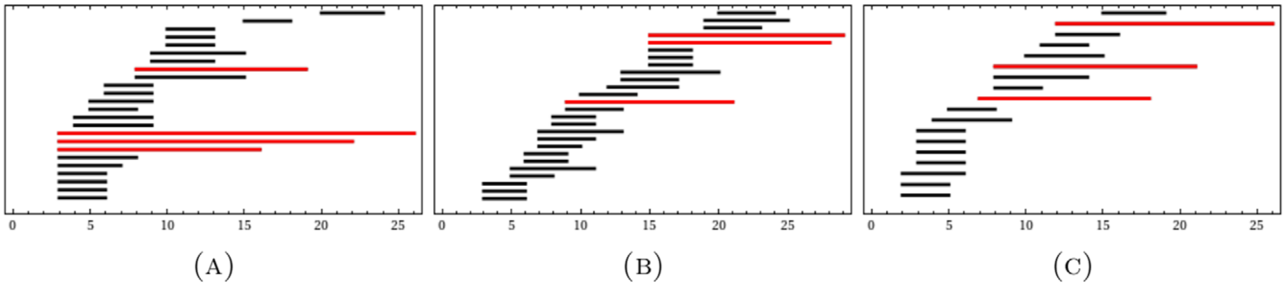

Figure 4. β_1-persistence intervals (only lifespans >2 are shown) for the reduced proximal (A) long; (B) medium; and (C) short data sets.

doi:10.1371/journal.pone.0037278.g004

Data\Length	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Medium+Short	52	10	2	5	1	2	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

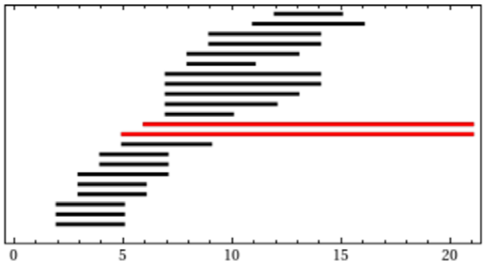

Figure 5. β_1-persistence intervals (only lifespans >2 are displayed) for the reduced proximal combined medium+short data set.
doi:10.1371/journal.pone.0037278.g005

In our interpretation this means that the distal data fills the tunnels in the proximal long data set that support the two shorter persistent generators, and is lining the tunnels in the proximal long data set that contribute to the two larger persistent generators, thereby reducing their lifespan. However, it is also possible that had we had access to more experimental data for the distal hairs, the additional data would fill the voids corresponding to the two remaining persistent generators of the long proximal data set as well. In an attempt to address this question, we construct a GMM for each hair category: distal and proximal long hairs. We sample from each model to obtain a more numerous distal set as well as a sampled proximal long set for which to compare with. The persistence results of the computations using the sampled data are compared to the persistence results of the experimental data sets and displayed in Figure 7 and Figure 8.

2.3.4. Sampling from the GMM model. As mentioned in the introduction, we constructed 42 Gaussian Mixture Models (GMM): one for each of 13 directional categories corresponding to both the medium and long proximal hairs, one for each of the 4 directional categories of the distal hairs, and one for each of the occupied 12 directional categories of the short proximal hairs. Each GMM has an associated weight w_i, $\sum_{i=1}^{42} w_i = 1$, reflecting the relative size of the corresponding class of hairs. Furthermore, each model is itself made up of n_i Gaussians with corresponding mean $\mu_{i,j}$ and covariance matrix $\Sigma_{i,j}$. Each Gaussian has an individual weight $w_{i,j}$ with $\sum_{j=1}^{n_i} w_{i,j} = 1$. Given an overall desired number of points N to be sampled, we sample $N * w_i * w_{i,j}$ (rounded to the nearest integer) data points from each Gaussian.

2.3.5. GMM for distal hairs. In the last step we sample from the complete GMM to create a GMM distal data set and a GMM long proximal set. These samples are then filtered, removing the densest and the least dense parts. Lastly, we compute persistence on each individual data set as well as on the combined set.

We note that sampling any Gaussian will result in a dense sample set close to the mean. On the other hand, since the support of a Gaussian is the entire space $\mathbb{R}^3$, samples will eventually fill arbitrary compact regions around the mean. Therefore, we expect that the oversampling of any particular set would lead to the annihilation of all topological features of the combined long proximal and long distal point-clouds. In order to avoid this oversampling artifact, we combine the initial sampling with two reduction algorithms. The first, designed to eliminate the densest parts of the sampled point cloud, uses *co-density* [1], see section 4.1. The second filter $F_{6,6}$, used on all experimental data sets, will eliminate the least dense parts of the point cloud.

We calibrated the co-density filter on the long proximal data set using the results previously displayed for the experimental data and selected the $X[1000;20,0]$ co-density filter (see section 4.1). We will refer to the GMM long proximal data set that was filtered

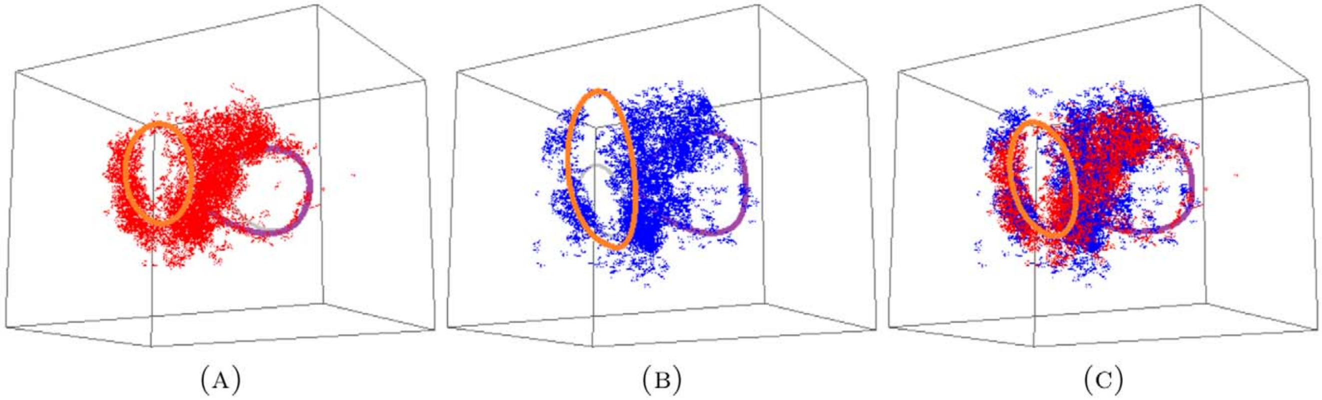

Figure 6. The experimental data for (A) small; (B) medium; and the combined (C) medium+small sets. There are three persistent generators in both the small and medium data sets, but only two persistent generators in the combined set. These two are consistent with two (orange and purple) of the three generators from both the small and medium sets. The voids corresponding to the third generator (grey) in both the small and medium sets are filled in the combined set. In this perspective only the orange persistent generator clearly encircles a void in the data. For a different perspective, showing clearly the purple generator, see Figure S1 and Figure S2.
doi:10.1371/journal.pone.0037278.g006

Data\Length	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Long+Distal	60	16	7	5	3	2	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0

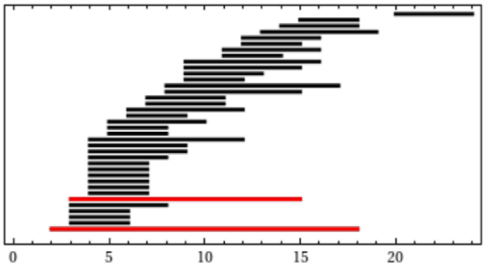

Figure 7. β_1-persistence intervals (only lifespans >2 are displayed) for the combined long+distal data set.
doi:10.1371/journal.pone.0037278.g007

by $X[1000;20,0]$ and $F_{6,6}$ as the GMM long proximal data for the remainder of the paper.

We do not know the appropriate ratio between the physiological number of terminals of the distal afferents and the number of terminals of the proximal afferents of the long hairs. Since distal hairs are sparser, we assume the number of hairs in each of the directional categories is comparable to the minimum population of any proximal directional class. Therefore, we assume that $w_k^* \approx \min_i(w_i)$ for any class k in the four distal classes and i ranging over all proximal directional classes. Due to the uncertainty in the actual numbers, we will create the GMM distal data set by sampling in two different ways.

(1) Distal(min) will refer to the GMM distal data set with the weight w_k^* for each distal class chosen to be that of the minimum proximal class weight as described above.

(2) Distal(max) will refer to the GMM distal data set with the weight w_k^* for each distal class chosen at the maximum proximal class weight; that is, $w_k^* = \max_i(w_i)$, where the k's and i's are as described above.

The first approach leads to a sampling of 12587 initial data points. The low number of sampled points leads us to use the filtering $X[1000;20,20]$, which removes the dense areas as well as the outliers, rather than the filter $F_{6,6}$ which had been used on all other data sets. Therefore, in removing the most dense 20% and least dense 20% of the sampled points, we obtain the set distal(min), which contains 7553 data points.

The second approach generates 55960 initial data points. After removing the densest 20% of the points using the $X[1000;20,0]$ filter, we arrive at 44768 data points. With subsequent filtering by the $F_{6,6}$ reduction, we obtain the data set distal(max), which has a cardinality of 39039 points (when $X[15;20,0]$ was used in the first step the set contained 39050 points). The results of the computation of persistence for these sets, as well as each's combination with the GMM proximal long set, are in Table 2.

Observe that distal(max) has two persistent generators with lifespans of 10 and 19; whereas the Distal(min) data set has a single persistent generator with lifespan 12. However, when we add these sets to the GMM long proximal data set, the resulting combinations both yield 2 persistent generators with lifespans of 17 and 21 for the combination using distal(max) (17 and 24 when using distal(min)). These are the same results as we obtained for the combined experimental data sets in Figure 7 and Figure 8.

We conclude that

- The experimental long proximal set and GMM long proximal set each have 4 persistent generators, with lifespans of 11, 13, 19, 23 and 11, 13, 23, 27, respectively.
- The addition of either the small sample (7553 points) or large sample (39039 points) of the GMM distal cloud to the GMM long proximal data fills the voids corresponding to the two

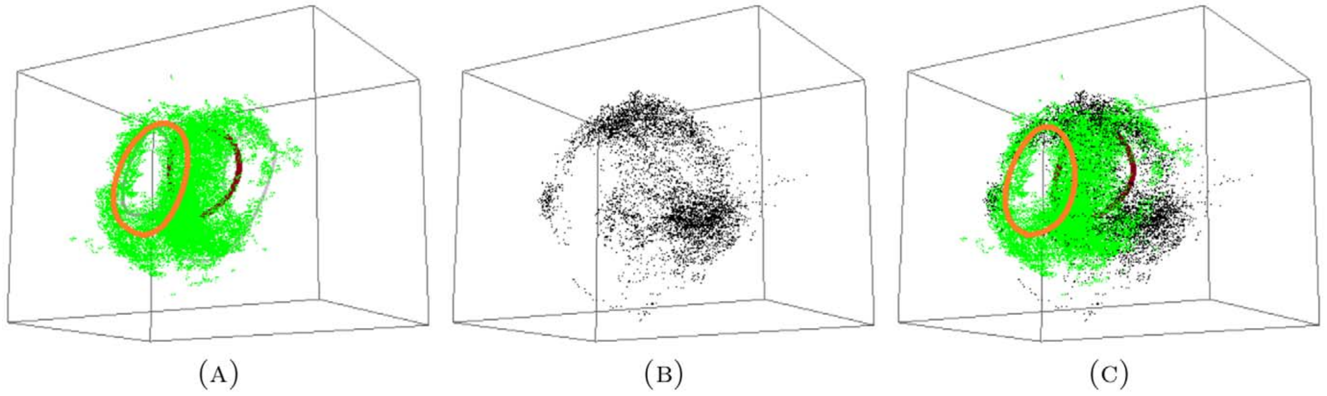

Figure 8. The experimental data for (A) long; (B) distal; and the combined (C) long+distal set. There are four persistent generators for the long set, two of which are consistent between long and long+distal sets (orange and crimson). Grey generators in the long set are not persistent in the combined set. We did not compute the generators for distal set since the data is much sparser than that of the long set.
doi:10.1371/journal.pone.0037278.g008

Table 2. Persistent Generators for GMM Data Sets.

Data\Length	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Distal(max)	20	8	2	2	0	2	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Distal(min)	19	4	3	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Long	19	7	4	2	2	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1
Long+Distal(max)	35	16	2	1	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
Long+Distal(min)	40	9	7	1	2	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Number of generators with prescribed length for sampled distal hairs(at different weights), sampled long proximal hairs and the combined sampled long proximal and distal sets.

doi:10.1371/journal.pone.0037278.t002

smaller persistent generators, while reducing the lifespans of the other 2 larger persistent generators. Therefore, we obtain the same result as with the experimental distal data: the (GMM) distal data is "lining" the tunnels of the (GMM) long proximal set, corresponding to the 2 more significantly distinguished persistent generators, see Figure 9.

2.4. Combined proximal set

We create the proximal data cloud that includes afferent terminals from proximal hairs of all lengths and all orientations by combining the experimental data for proximal long, medium and short hairs. The results of this combined experimental proximal data cloud are in Figure 10.

We note that there is only one persistent generator. Therefore the addition of the long proximal data set to the combined short and medium data set has annihilated one of the persistent generators in that latter set and three of the persistent generators in the former set.

2.5. Entire data set

Finally, we compute persistent generators for the entire set of terminals. In doing so, we analyze three data sets. The first is the complete experimental data set that combines the combined data sets of long+distal and medium+small. Then, because the experimental distal point cloud is so sparse, we also replace the experimental distal data with the two sampled GMM distal data sets. Recall, the sampled data sets are filtered in the way described in section 2.3.5. We add the distal(min) and distal(max) data sets to the experimental long+medium+short data set to form the combined(min) and combined(max) data sets, respectively.

The persistent homology results for the experimental combined set are displayed in Figure 11 and Figure 12. There are no longer any persistent generators. Recall that the combined proximal long+medium+short set does have a single persistent generator. Therefore, the computations reveal that the sparse experimental distal afferent terminals make the distinguishable tunnel in the proximal data smaller. A close inspection of (B) of Figure 11 shows multiple terminals within the central tunnel (black dots) in the data cloud. These are distal terminals within the void in the proximal set. This observation puts in question the robustness of our method and result. Recall we did not filter the outliers from the experimental distal set. To that end, if these few terminals are the main cause of the loss of persistence of the proximal generator (orange in (A) of Figure 11), our result, and method, could not be considered robust. To address this issue, we analyze the combined(max) and combined(min) point cloud data sets.

The results of these persistence computations are displayed in Figure 13 and Figure 14. The comparison between combined(min) and the experimental combined set (Figure 12) reveals almost an exact match. The more interesting comparison is between combined(max), which may represent the true physiological abundance of the distal terminals, and the experimental combined set. We note that the lifespans of all the generators are shorter for the combined(max) set than are those for the experimental combined set, in spite of the fact that the there are no longer any

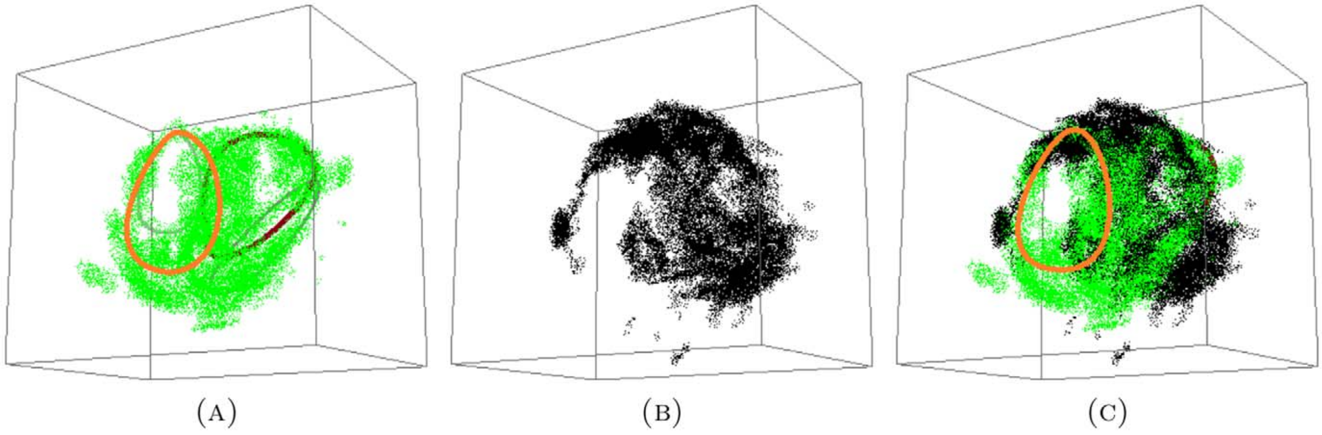

Figure 9. The GMM data for (A) long proximal; (B) distal(max); and (C) the combined long+distal(max) set. There are four persistent generators for the GMM long proximal set, two of which are consistent between the GMM long and GMM long+distal sets (orange and crimson). Grey generators in the GMM long proximal set are not persistent in the combined set.

doi:10.1371/journal.pone.0037278.g009

Data\Length	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
L+M+S	26	7	2	3	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0

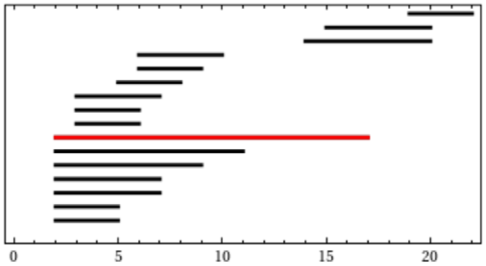

Figure 10. β_1-persistence intervals (only lifespans >2 are shown) for the reduced experimental proximal combined long+medium+short data set.
doi:10.1371/journal.pone.0037278.g010

visible distal terminals within the central tunnel. The terminals visible in the tunnel in Figure 11 were filtered out as outliers. This shows that these points were not the main cause of the loss of persistence of the proximal generator (orange in (A) of Figure 11). Rather, the distal terminals are lining the inside walls of this tunnel, and thus, reducing its size. This result confirms that the loss of persistence of the proximal generator from the addition of the distal data points is a robust phenomena.

We want to emphasize that there are still multiple homology generators in the combined data sets (see Figure 12 and Figure 14) that correspond to different tunnels in the point cloud; however, there is no persistent generator that is robustly present in the data. These remaining tunnels very likely contain the dendrites of the interneurons in the terminal ganglion, which are downstream from the afferents, and which make synaptic contact with the afferent terminal cloud.

Discussion

The lack of persistent generators in the combined proximal and distal set illustrates that the set of terminals of all hairs has no significant tunnels or voids. This result is not surprising from an efficiency standpoint: the space in the terminal ganglion is limited and the terminals of afferent neurons are filling the entire available space. The potential consequences for neural processing are more intriguing. Given that the directionally sensitive interneurons (IN) from the ganglion are sampling the terminals of afferents with the same directional sensitivity, and since it is likely that the three dimensional position of both dendrites of IN's and afferent terminals are developmentally determined only up to some finite precision, the spatial separation of afferent terminals puts the limit on the precision with which the downstream processes can distinguish direction of the air motion. Since the angles form a circle (S^1), the afferent terminals must be embedded in the ganglion as an approximate image of a full torus $S^1 \times B^2$. Here B^2 is a disc that approximates the spatial extend of the afferent terminals with the same response angle. The need of separating the terminals of hairs with the nearby response angle to enhance acuity of the system, is balanced by the need to pack the terminals in the smallest possible volume: a ball B^3. These two needs are not compatible. There is no embedding of the full torus to a 3-disc B^3; the space $S^1 \times B^2$ has a different homology than the space B^3. As a consequence, there must be at least one point in B^3 where the directional tuning is not well defined. This is similar to topological singularities (pinwheels, vortices) in a primate cortical striate cortex, which result from the fact that $S^1 \times B^1$ cannot be embedded into a two dimensional disc D^2. The only difference is that the pinwheel in the terminal ganglion is three dimensional, while spatial orientations of columns in a striate cortex is a two dimensional phenomenon.

It is interesting to note that the distal hairs are filling the last essential tunnel in the combined proximal set. Since distal hairs are more sparse, not all directions are represented in the set of their preferred directional responses. One can speculate that the hairs further along the cercus fill more and more central positions along the tunnel in the proximal set. If, in addition, there was a hair at the very tip of the cercus, this would be the hair that would map into the pinwheel position in the terminal ganglion. This

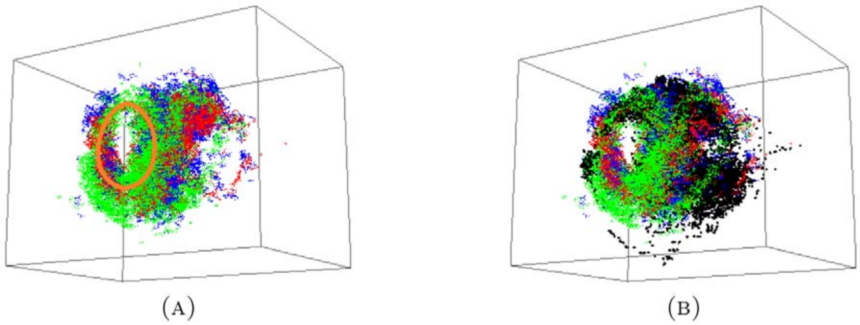

Figure 11. The experimental data for (A) long+medium+short; and (B) long+medium+short+distal. The combined set of all proximal hair afferent terminals has only one persistent generator ((A), orange), while the entire set does not have a persistent generator.
doi:10.1371/journal.pone.0037278.g011

Data\Length	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
L+M+S+D	29	8	1	3	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

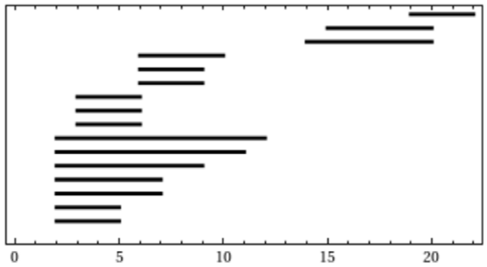

Figure 12. β_1-persistence intervals (only lifespans >2 are shown) for the combined experimental set.
doi:10.1371/journal.pone.0037278.g012

hypothetical arrangement of terminals (more distal hairs have afferent terminals closer to the pinwheel) would also make sense from another perspective. Our collaborators have recently shown [8] that the cercus acts as a delay line; a signal from greater distal hairs travels to the ganglion significantly ($>5ms$) longer than the signal from the proximal hairs. If the system works as a delay line, it makes sense to separate the terminals of the distal hairs from those of the proximal hairs, but still align the directional sensitivities. Our current data does not have the spatial resolution to confirm this conjecture, but our analysis, which shows that the distal hairs line the tunnel in the proximal cloud, is compatible with this theory.

We also comment on our results about the proximal hair terminals. We have shown that while short and medium hairs have three relatively weak persistent generators each, the combined medium-short set has two robust persistent generators. This suggests that our separation of the medium-short set into two groups is artificial and only the combined set has spatial structure that suggests functional relevance. The addition of the long proximal hairs to the medium-short set annihilates one of these generators. The resulting set of terminals has only one generator that encircles the 3-D pinwheel where the angular response is not well defined.

Our goal in this paper was to show the applicability of sophisticated approaches from computational homology to the analysis of noisy neuroscience data. The source of noise is both developmental, encompassing animal to animal variability, and experimental. Our methods provide robust results that give fresh insight into the structure, as well as suggest functional relevance, of the spatial organization of the terminals of afferent neurons.

Methods

4.1. Data filtering

Density estimation is a highly developed area within statistics [9] and, following the lead of Lee *et. al.* [10] (see also the review by Carlsson [1]), we will employ a *codensity function* as well as an outlier-reduction function that we call the *DN-density function.*

The *codensity function* is defined as follows: For any fixed positive integer k and the point cloud X, we define the *k-codensity function* δ_k for $x \in X$ by

$$\delta_k(x) = d(x, k(x)).$$

where $d(\cdot,\cdot)$ denotes the distance function in X, and $k(x)$ denotes the k^{th} nearest neighbor of $x \in X$. The function $\delta_k(\cdot)$ is inversely related with density, since a dense region will have smaller distances to the k^{th} nearest neighbor. Considering that we are interested in dense regions, we will study subcollections of points for which $\delta_k(\cdot)$ is bounded from above and/or below by given threshold percentages. We also note that each δ_k yields a different density estimator, since for large values of k, δ_k computes density using points in large neighborhoods of x; whereas, for small values of k, small neighborhoods are used. Therefore, for large k, δ_k corresponds to a smoothed out notion of density and for small k, δ_k corresponds to a version that carries more of the detailed structure of the data set. Following similar notation as was used by Carlsson [1], we denote a subset $X[k;u,l] \subset X$, where k is a positive integer and u (l) is the upper (lower) percentage bound on the points to keep. More precisely,

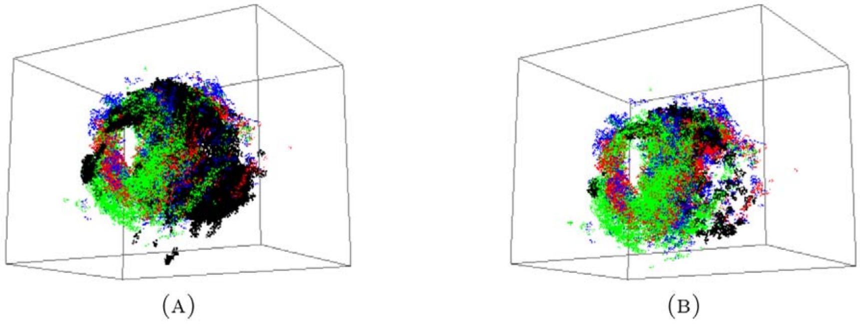

Figure 13. The data sets for (A) Combined(max); and (B) Combined(min). There are no persistent generators in either combined data set.
doi:10.1371/journal.pone.0037278.g013

Data\Length	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
L+M+S+D(max)	19	8	2	0	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L+M+S+D(min)	31	11	3	2	2	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

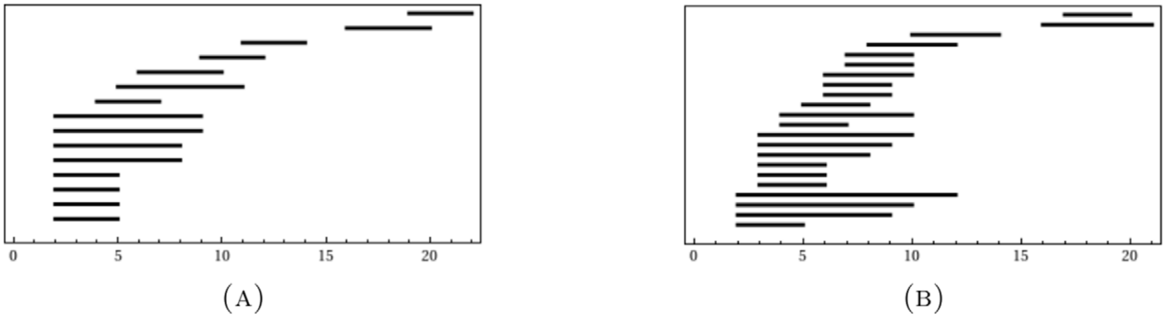

Figure 14. β_1-persistence intervals (only lifespans >2 are shown) for the combined data sets: (A) Combined(max) and (B) Combined(min).

doi:10.1371/journal.pone.0037278.g014

$$X[k;u,l] = \{x \in X \mid \delta_k(x) \text{ lies between the } (100-u)\,\% \text{ and } l\,\% \text{ lowest values of } \delta_k(x) \text{ in } X\}$$

This approach to data reduction will be used exclusively for the data sampled from the Gaussian Mixture Model to achieve a reduction in the densest part of the sampled point cloud. The *DN-density function*, described below, will be used for both the experimental and sampled data in order to eliminate outliers in the least dense regions.

The *DN-density function* is defined as follows: For any fixed positive integer k and the point cloud X, we define the *k-DN density* function DN_k of $x \in X$ by

$$DN_k(x) = \{\#\ of\ y \in X\ such\ that\ d(x,y) \le k\}$$

where $d(\cdot,\cdot)$ denotes the distance function on X and the integer k is a distance threshold to the neighbors of $x \in X$. Then, for a fixed positive integer threshold j, we define the *thresholding* function $F_{k,j}$ on a set X by

$$F_{k,j}(X) := \{x \in X \mid DN_k(x) \ge j\}.$$

Finally, we let

$$X_{k,j} := \bigcap_{l=1}^{\infty} F^{l}_{k,j}(X)$$

be the intersection of the iterates of the data reducing function $F_{k,j}$. Note that these sets are finite and nested, $\bigcap_{l=1}^{s+1} F^{l}_{k,j}(X) \subset \bigcap_{l=1}^{s} F^{l}_{k,j}(X)$, so that the procedure terminates at a finite iteration. In addition, note that for poorly chosen values of j and/or k, the set $X_{k,j}$ can be empty; for instance, if we select j or k to be extremely large. In practice, the sets stabilize after a few iterations.

4.1.1. Calibration of the GMM filtering. We calibrate our method on the long proximal data set using the results previously displayed for the experimental data. The GMM long proximal data set is created by sampling a total of 150000 points from the complete GMM as formerly described. We eliminate 20% of the points from the most dense regions of the sampled set by applying either the filter $X[15;20,0]$ or the filter $X[1000;20,0]$. These two different filtering mechanisms, which use significantly different sized neighborhoods, allow us to compare the effect of the smoothness of the co-density function on the selection. The final filtering of the GMM long proximal data set is the same as that of the experimental long proximal data set; that is, we filter the GMM long proximal set using the $F_{6,6}$ reduction.

Following this procedure, the GMM long proximal data set equates to 52974 points of the total 150000. Removing the most dense 20%, using either of the $X[\cdot;20,0]$ filters, leaves a reduced GMM long proximal data set with 42380 data points. The cardinality of the set compares favorably to that of the non-reduced experimental long proximal data set, which contains 42428 points. The application of the $F_{6,6}$ filter further reduces the GMM long proximal data set to a set of 30716 points, if the density filter $X[15;20,0]$ was previously applied, or 30656 points, if the filter $X[1000;20,0]$ had been applied. The cardinality of each of these reduced GMM sets compares favorably to the reduced experimental proximal long data set, which contains 33950 points. The final step of the calibration is to compute cubical persistence on each reduced GMM data set, with results displayed in Table 3, and compare these results to those of the experimental long proximal data as displayed in Figure 4.

We observe that the data sets created by both the $X[15;20,0]$ and $X[1000;20,0]$ density filters yield the same results: in either case, the GMM long proximal data has 4 persistent generators at lifespans of $11, 13, 23$ and 27. Recall that the experimental long proximal data had 4 persistent generators at lifespans of $11, 13, 19$ and 23. Thus, seeing that we have an equal number of persistent generators for the GMM and experimental data and, in addition, the lifespan length for each is nearly identical, we conclude that the Gaussian Mixture Model is a good model for the proximal long data.

4.2. Persistence theory

In this subsection we outline the main concepts of topological persistence, which combines the idea of homology with that of filtration. The homology groups and Betti numbers associated with these groups are computable invariants that have been developed

Table 3. Comparison of GMM long proximal models with different filters.

Data\Length	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**
Long-1000	19	7	4	2	2	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1
Long-15	12	4	2	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Long-1000 refers to GMM sampled long proximal data with filter $X[1000;20,0]$.
Long-15 refers to GMM sampled long proximal data with filter $X[15;20,0]$.
doi:10.1371/journal.pone.0037278.t003

over the last century (see [11] for a relatively accessible introduction). Homology groups form a graded abelian group; that is, there is one such group for every non-negative integer, although for reasonable topological spaces such as compact manifolds, these groups are trivial for integers larger than some finite n. For our purposes it is sufficient to know that the dimension of the 0^{th} homology group (and thus the 0^{th} Betti number: β_0) corresponds to the number of connected components of the set, while the dimension of the 1^{st} homology group (which in our case will be always equal to the 1^{st} Betti number: β_1) measures the number of one-dimensional holes in the set. A one-dimensional hole is characterized, roughly speaking, by a loop that can be embedded into the set around a hole for which the loop cannot be smoothly (i.e. without tearing it apart) shrunk to a point within the set. We will call such a loop a *generator* of the first homology group.

We now define more precisely the idea of topological persistence in the context of cubical complexes used in this paper. For more details the reader is referred to [2,3]. Recall that for a sequence $\epsilon_0 < \epsilon_1 < \ldots\ldots < \epsilon_{n-1} < \epsilon_n$ of increasing values of ϵ, we center a box with a side of length ϵ_i over each data point. We call the union of these boxes a *complex* K_{ϵ_i} and note that if $\epsilon_i < \epsilon_j$ then $K_{\epsilon_i} \subset K_{\epsilon_j}$. These complexes form a filtration of the complex $K := K_{\epsilon_n}$. A *filtration* of a complex K is a nested sequence of subcomplexes that starts with an empty complex and ends with the complete complex K

$$\varnothing = K^0 \subset K^1 \subset \ldots \subset K^n = K.$$

Define Z_k^s and B_k^s to be the k-th cycle and k-th boundary group, respectively, of the s complex K^s in the filtration. To capture persistent cycles in K^s, we factor its k-th cycle group by the k-th boundary group of K^{s+p}, where K^{s+p} is p complexes further along in the filtration. Therefore, the *p-th persistent k-th homology group* of K^s is

$$H_k^{s,p} = Z_k^s / (B_k^{s+p} \cap Z_k^s).$$

This is well defined since both B_k^{s+p} and Z_k^s are subgroups of the chain complex C^{s+p} of K^{s+p}, and thus, a group. The *p-th persistent k-th Betti number $\beta_k^{s,p}$* of K^s is the rank of $H_k^{s,p}$.

One can also define the p-th persistence group using inclusion induced injective homomorphisms of ordinary homology groups. The main observation is that if two cycles are homologous in K^s then they still exist and are homologous in K^{s+p}. Therefore an inclusion induced map

$$\eta_k^{s,p} : H_k^s \to H_k^{s+p}$$

is well-defined and maps a homology class into one that contains it. The image of this homomorphism is isomorphic to the p-th persistent homology group of K^s,

$$Im\ \eta_k^{s,p} \cong H_k^{s,p}.$$

4.3. Data analysis

In this subsection, we present the preprocessing and processing used in our computation of persistent homology on each data set. The overall size of each individual data set, the computational memory needed to store the filtration of each data set and the memory needed for the persistence computations has forced us to use cubical homology for our computations (see [12] for relatively accessible introduction). Cubical persistent homology allows for the filtration K to be created using non-overlapping 3-dimensional basis cubes, which then can be stored as a bitmap, greatly reducing the computational cost [13,14]. The construction is based on a cubical grid of the part of the space that contains all the data points of the point cloud. In this way, the ϵ_j's that parameterize the filtration will all be multiples of the size of the elementary grid element (i.e. *elementary cube* [12]).

We first translate the data into the positive quadrant of $\mathbb{R}^3$. The critical input for this process is the size u of the basis cube (in μm). An appropriate choice is important for two reasons: if one chooses a basis cube too large, the growing of the filtration is too fast, resulting in a lack of persistent generators; if the cube is too small, the size of the filtration is too large and computationally costly, not providing any further information beyond that obtained from a filtration with a larger basis cube size. The appropriate choice for the vast majority of data throughout this paper was $u = 0.9\mu$m. However, we had to choose $u = 1.1\mu$m for the largest of our data sets (those that combined data from all hairs) due to the large computational cost that we incurred at input sizes of $u = 0.9\mu$m and $u = 1.0\mu$m. Given the entire translated data set, we define $R \subset \mathbb{R}^{3+}$ to be the smallest box

$$R := [0,x_{max}] \times [0,y_{max}] \times [0,z_{max}]$$

that contains the point cloud data set, for which $x_{max} = n_x u$, $y_{max} = n_y u$, and $z_{max} = n_z u$ are integer multiples of the basic length u. We divide R into $n_x \times n_y \times n_z$ elementary cubes. Each cube

$$c(i,j,k) := [(i-1)u, iu] \times [(j-1)u, ju] \times [(k-1)u, ku]$$

is uniquely determined by its coordinates (i,j,k).

Next, we build our filtration K. The initial complex K_1 consists of all elementary cubes in R that contain at least one point from the data set X. The next step is to increase the size of the elementary box and construct a complex from all larger boxes that contain at least one point from X. A naive way to increase the size of the elementary box is to simply double its size. The resulting

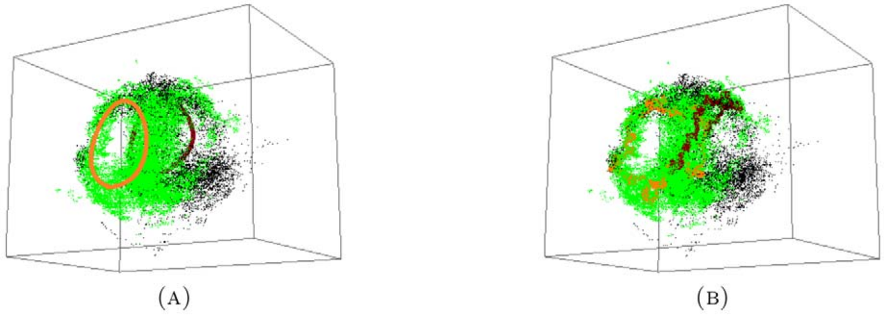

Figure 15. The persistant generators of the experimental data set long+distal are displayed. (A) long+distal with smooth represenative generators; and (B) long+distal with actual generators produced in computation.
doi:10.1371/journal.pone.0037278.g015

exponential growth of the elementary cubes would yield only a few complexes K in the filtration. An alternative way, which we describe next, creates nested complexes and results in a higher resolution persistence computation. We associate to each elementary cube a positive integer value b, which will be referred to as a *birth time*. Initially, we assign a birth time of zero, $b(c(i,j,k)) := 0$, to all cubes. We will then inductively define a nonzero birth time value of b as we construct our filtration.

We start by assigning the birth time value 1 to all cubes that contain at least one point of the set X

$$b(c(i,j,k)) := 1 \Leftrightarrow X \cap c(i,j,k) \neq \emptyset.$$

In the inductive step, for each elementary cube with $b(c(i,j,k)) = 0$, we record all bordering elementary cubes' birth times in the set $B := \{b_q \mid q = 1,2,...,26\}$. Then, if $max(B) \neq 0$, we increment the birth time $b(c(i,j,k)) := 1 + min\{b_q \neq 0 \mid q = 1,2,...,26\}$. However, if $max(B) = 0$, then b remains unchanged. This process continues until every elementary cube in R has a nonzero birth time assigned to it. There is a finite number of cubes and, since we only change to a non-zero value of b once, this procedure is finite. Let b^* be the maximum birth time across all cubes in R. We define the complexes K_b, $b \in \{2, \ldots, b^*\}$ of our filtration K by

$$K_b = \{c(i,j,k) \mid b(c(i,j,k)) \leq b\}.$$

We compute cubical persistence of the filtration complex K using code developed by Mrozek, Batko and Wanner [13,14,15] called *cubPersistenceMD*. While the final computation of homology is based on the classic Smith Normal Form algorithm, the significant computational improvements are found in its preprocessing coreduction algorithm ([13,14], also, See Section 4.4). This preprocessing algorithm reduces the overall size of the input data to the Smith Normal Form while preserving the homology of the complex. The (co-)reduction runs in linear time, whereas the Smith diagonalization algorithm has a complexity of $O(n^{3.376...})$ [13,14,16].

In the computations described above, the output of the process is a set of lists of birth/death times of generators in the filtration K. The β_1-Persistence Intervals, as displayed throughout the paper, are constructed as a result. These barcodes provide detail as to what is happening in the point cloud data set, but in these situations it is often the "where" that is just as important as the "what". Therefore, we have employed subprograms from the package CHomP [12], specifically *Homcubes*, to aid in acquiring a means to visualize the persistent generators of each data set. The input of *Homcubes* is the complex $\bigcup_{b=1}^{i} K_b$. The choice of i is based on the birth/death times of the persistent generators of each specific point cloud data set; the only requirement is that birth time $\leq i \leq$ death time. The resulting set of generators is constructed using basis cubes from the complex $\bigcup_{b=1}^{i} K_b$, many of which are not present in the initial complex K_1. However, we display these generators with the cubes from complex K_1, since these cubes closely approximate the actual position of afferent terminals. This often results in generators that seemingly protrude into free space, which is just a consequence of a mismatch between the displayed set K_1 and the set $\bigcup_{b=1}^{i} K_b$ where generators are computed. Generators are also very rarely smooth. Therefore, throughout the paper, we have displayed figures with smooth circular representations of the computed generators. We illustrate the difference in the Figure 15.

4.4. Co-reduction algorithm

Given a simplicial (or cubical) complex K and a free chain complex $\mathcal{C} = \{C_q(K), \partial_q\}$ with basis S, we say

(1) A pair (a,b) of elements of S is said to be an ***elementary reduction pair*** if

$$cbd_S\ a = \{b\}$$

(2) A pair (a,b) of elements of S is said to be an ***elementary coreduction pair*** if

$$bd_S\ b = \{a\}$$

Theorem [13,14,15]: If (a,b) is an elementary reduction or coreduction pair in S, then

$$H_*(K) \cong H_*(K\{a,b\})$$

Function Coreduction (Homology Complex S, a vertex s) [13,14,15]

```
begin
$Q :=$ empty queue of generators;
enqueue($Q$,$s$)
while $Q \neq \emptyset$ do begin
$s :=$ dequeue($Q$);
```

```
    if $bd_S\ s$ contains exactly one element $t$ then begin
        $S:=S\backslash\{s\};$
        for each $u \in cbd_S\ s$ do
            if $u \notin Q$ then enqueue($Q$,$u$);
        $S:=S\backslash\{t\};$
    end
    else if $bd_S\ s=\emptyset$ then
        for each $u \in cbd_S\ s$ do
            if $u \notin Q$ then enqueue($Q$,$u$);
end;
return $S$;
```

When implemented as a bitmap, for a cubical complex K the coreduction algorithm runs in $O(2d^2n)$, where d denotes the embedding dimension of the cubical set [13,14,15].

Supporting Information

Figure S1 The experimental data set medium+small is displayed in two perspectives. (A) medium+small data in the perspective that was displayed throughout the paper; (B) medium+small data in a second perspective providing a clear view of the second (purple) persistent generator.
(TIF)

Figure S2 Each figure is a Mathematica *.nb* file containing a 3-D figure. These figures depict the medium+small point cloud; the medium+small point cloud with the computed generators; and the medium+small point cloud with computed generators and containing the black marked-double-circle, whose arclength can is parameterized by the response angle of the afferent hairs.
(NB)

Acknowledgments

We would like to thank Gwen Jacobs and John P. Miller for their generosity in sharing their data, John P. Miller for help with a figure preparation, S. Kruger and G. Cummins for help with data analysis and M. Mrozek and K. Mischaikow for sharing the persistence software.

Author Contributions

Conceived and designed the experiments: TG JB. Performed the experiments: JB. Analyzed the data: JB. Wrote the paper: TG JB.

References

1. Carlsson G (2009) Topology and data. Bulletin of the American Mathematical Society 46: 255–308.
2. Edelsbrunner H, Letscher D, Zomorodian A (2000) Topological persistence and simplification. IEEE Symposium on Foundation of Computer Science. pp 454–463.
3. Goodman JE, Pach J, Pollack R, eds (2008) Surveys on Discrete and Computational Geometry. Twenty Years Later, Providence, RI: Amer. Math. Soc., chapter Persistent Homology- a Survey. Contemporary Mathematics 453. pp 257–282.
4. Jacobs G, Theunissen F (1996) Functional organization of a neural map in the cricket cercal sensory system. The Journal of Neuroscience 16: 769–784.
5. Paydar S, Doan C, Jacobs G (1999) Neural mapping of direction and frequency in the cricket cercal sensory system. The Journal of Neuroscience 19: 1771–1781.
6. Jacobs G, Theunissen F (2000) Extraction of sensory parameters froma neural map by primary sensory interneurons. The Journal of Neuroscience 20: 2934–2943.
7. Miller JP Personal communication.
8. Mulder-Rosi J, Cummins G, Miller JP (2010) The cricket cercal system implements delay line processing. J Neurophysiol 103: 1823–1832.
9. Silverman BW (1986) Density Estimation for Statistics and Data Analysis. Monographs on Statistics and Applied Probability. Chapman & Hall, London.
10. Lee AB, Pedersen KS, Mumford D (2003) The nonlinear statistics of high-contrast patches in natural images. International Journal of Computer Vision 54: 83–103.
11. Munkres JR (1984) Elements of Algebraic Topology. Addison-Wesley.
12. Kaczynski T, Mischaikow K, Mrozek M (2004) Computational Homology. Applied Mathematical Sciences. Springer.
13. Mrozek M, Batko B (2009) Coreduction homology algorithm. Discrete and Computational Geometry 41: 96–118.
14. Mrozek M, Wanner T (2010) Coreduction homology algorithm for inclusions and persistent homology. Computers and Mathematics with Applications 60.
15. Mrozek M (2006) Homology software website. Available: http://www.ii.uj.edu.pl/~mrozek/software/homology.html Accessed 2012 April 23.
16. Storjohann A (1996) Near optimal algorithms for computing smith normal forms of integer matrices. In: Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computations, ISAAC 1996 ACM Press. pp 267–274.