PLOS ONE

RESEARCH ARTICLE

Puzzle Imaging: Using Large-Scale Dimensionality Reduction Algorithms for Localization

Joshua I. Glaser[1]*, Bradley M. Zamft[2], George M. Church[2], Konrad P. Kording[1,3,4]

1 Department of Physical Medicine and Rehabilitation, Northwestern University and Rehabilitation Institute of Chicago, Chicago, Illinois, United States of America, **2** Department of Genetics, Harvard Medical School, Boston, Massachusetts, United States of America, **3** Department of Physiology, Northwestern University, Chicago, Illinois, United States of America, **4** Department of Applied Mathematics, Northwestern University, Chicago, Illinois, United States of America

* j-glaser@u.northwestern.edu

Citation: Glaser JI, Zamft BM, Church GM, Kording KP (2015) Puzzle Imaging: Using Large-Scale Dimensionality Reduction Algorithms for Localization. PLoS ONE 10(7): e0131593. doi:10.1371/journal.pone.0131593

Editor: Joseph Najbauer, University of Pécs Medical School, HUNGARY

Received: March 18, 2015

Accepted: June 2, 2015

Published: July 20, 2015

Data Availability Statement: Code for analysis and simulations is available on Github: https://github.com/jglaser2/Puzzle_Imaging.

Funding: Joshua Glaser was supported by NIH grant 5R01MH103910 and NIH grant T32 HD057845. Bradley Zamft was supported by NIH grant 5R01MH103910. George Church acknowledges support from the Office of Naval Research and the NIH Centers of Excellence in Genomic Science. Konrad Kording is funded in part by the Chicago Biomedical Consortium with support from the Searle Funds at The Chicago Community Trust, and is also supported by NIH grants 5R01NS063399,

Abstract

Current high-resolution imaging techniques require an intact sample that preserves spatial relationships. We here present a novel approach, "puzzle imaging," that allows imaging a spatially scrambled sample. This technique takes many spatially disordered samples, and then pieces them back together using local properties embedded within the sample. We show that puzzle imaging can efficiently produce high-resolution images using dimensionality reduction algorithms. We demonstrate the theoretical capabilities of puzzle imaging in three biological scenarios, showing that (1) relatively precise 3-dimensional brain imaging is possible; (2) the physical structure of a neural network can often be recovered based only on the neural connectivity matrix; and (3) a chemical map could be reproduced using bacteria with chemosensitive DNA and conjugative transfer. The ability to reconstruct scrambled images promises to enable imaging based on DNA sequencing of homogenized tissue samples.

Introduction

Many biological assays require the loss/destruction of spatial information of samples, making it difficult to create a high-resolution image of cellular properties. As a prime example, determining genetic properties of a biological sample usually requires breaking apart that sample for DNA sequencing (but see [1]). This limits the resolution of an image of genetic content to the precision of tissue sectioning prior to sequencing. Along with determining gene expression, researchers are attempting to use genetic information to determine neural connectivity [2], neural activity [3], other cellular properties [4], and chemical concentrations [5]. Being able to image these types of properties at high resolution and large scale could therefore lead to expanded measurement and recording applications. This would be possible if we could recover spatial information post hoc.

P01NS044393, and 1R01NS074044. 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.

In order to recover a sample's spatial information, information about its relative spatial location could be embedded and utilized. For example, imagine that each piece of genetic information was attached to a puzzle piece (the embedded relative spatial information; Fig 1A). While the puzzle pieces by themselves don't provide spatial information, fitting the pieces together would lead to a spatially correct image of the genetic information. Thus, the use of relative spatial information (how the puzzle pieces' locations relate to one another) could allow for higher-resolution imaging.

Using relative spatial information to reconstruct an image can be thought of as a dimensionality reduction problem. If there are N puzzle pieces, one can construct an $N \times N$ similarity matrix $\mathbf{S}$, where $\mathbf{S}_{ij}$ determines how close puzzle piece i and j are (higher similarity means shorter distance; Fig 1B). The goal is to map this high dimensional similarity matrix to accurate 2- or 3-dimensional locations of each piece (Fig 1C). Importantly, there is a whole class of dimensionality reduction methods that aims to preserve high dimensional distances in the reduced dimension (e.g. [6–8]). These types of techniques would allow a "piecing of the puzzle back together."

Here, we propose "puzzle imaging," and develop two dimensionality reduction algorithms that would allow large-scale puzzle imaging. We describe three concrete examples in which puzzle imaging would be beneficial: (1) "Neural Voxel Puzzling," in which a relatively high-resolution 3-dimensional brain map is reproduced by giving DNA barcodes to neurons; (2) "Neural Connectomics Puzzling," in which neural connections are used to recover neural locations; and (3) "Chemical Puzzling," in which a chemical map could be reproduced using bacteria with chemosensitive DNA and conjugative transfer. Each of these examples leverage the faster-than-Moore's law advances in both the cost and speed of genetic sequencing [9], and therefore are likely to become more relevant as this "genetic revolution" proceeds. In each example, we use our algorithms on simulated data, and provide a preliminary demonstration of the capabilities of puzzle imaging.

Results

Neural Voxel Puzzling

Overview. The purpose of neural voxel puzzling is to create a 3-dimensional image of the brain at high resolution. This image could provide useful neuroanatomical information by itself, or could be used in conjunction with other technologies to determine the locations of genetically encoded neural properties in the brain [10].

The first step in voxel puzzling is to label each neuron in the brain with a unique DNA or RNA barcode throughout its entire length (Fig 2A). Ongoing research aims to tackle this challenge [2, 11, 12]. Recently, researchers have succeeded in having bacteria generate a large diversity of barcodes *in vivo* [11]. Next, the brain is shattered into many voxels (Fig 2B; note that voxels are only squares for simplification), and the DNA in each voxel is sequenced, yielding a record of which neurons were in which voxels. This provides us with relative spatial information about voxel placement: voxels that share more neurons will likely be closer to each other. We can use this relative spatial information to puzzle the voxels into their correct locations.

Dimensionality Reduction. More formally, puzzling together the brain is a dimensionality reduction problem, with each voxel represented by an N-dimensional object, where N is the number of neurons. Dimension k corresponds to neuron k being in that voxel (Fig 2C). These N-dimensional voxels must be mapped to their correct 3-dimensional coordinates (or 2 dimensions in Fig 2's simplified example). This can be done because voxels that are closer in 3-dimensional space will have more neurons in common, and thus will be more similar in N-dimensional space. With the knowledge of the similarity between all pairs of voxels, we can

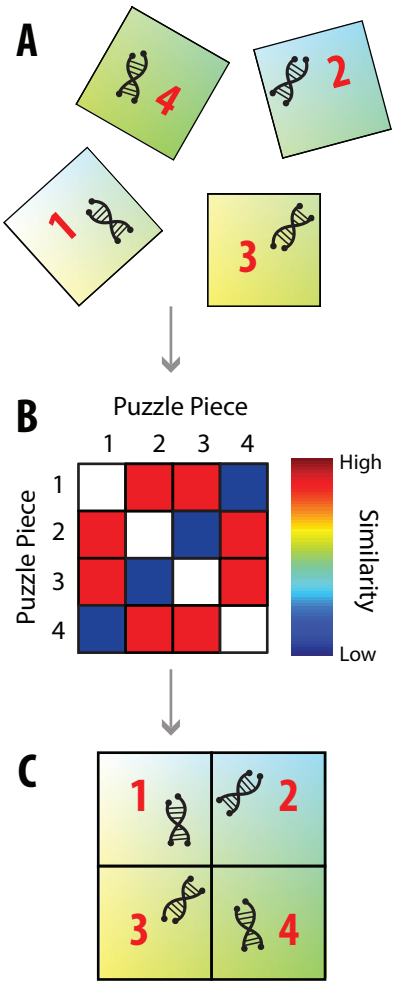

Fig 1. Puzzle Imaging. There are many properties, such as genetic information, that are easier to determine when the original spatial information about the sample is lost. However, it may be possible to still image these properties using relative spatial information. **(A)** As an example, let us say that each piece of genetic information is attached to a puzzle piece. While the puzzle pieces don't provide absolute spatial information, they provide relative spatial information: we know that nearby pieces should have similar colors, so we can use color similarity to determine how close puzzle pieces should be to one another. **(B)** We can make a similarity matrix of the puzzle pieces, which states how similar the puzzle pieces' colors are to each other, and thus how close the pieces should be to one another. **(C)** Through dimensionality reduction techniques, this similarity matrix can be used to map each puzzle piece to its correct relative location.

doi:10.1371/journal.pone.0131593.g001

create a similarity matrix (Fig 2D) and then puzzle the voxels back together (Fig 2E). Without any additional information, it is impossible to know exact locations; only the relative locations of voxels can be known. Thus, the output could be a flipped or rotated version of the input (Fig 2E). That being said, additional information could be used to correctly orient the reconstruction. For example, the overall shape could be matched for a sample that is asymmetric (like the brain), or a few samples with known locations could act as landmarks.

In order to convert the knowledge of which neurons are in which voxels (Fig 2C) to a reconstructed puzzle (Fig 2E), a dimensionality reduction algorithm is needed. We demonstrate the performance of two algorithms that are promising for large-scale problems. The first is a

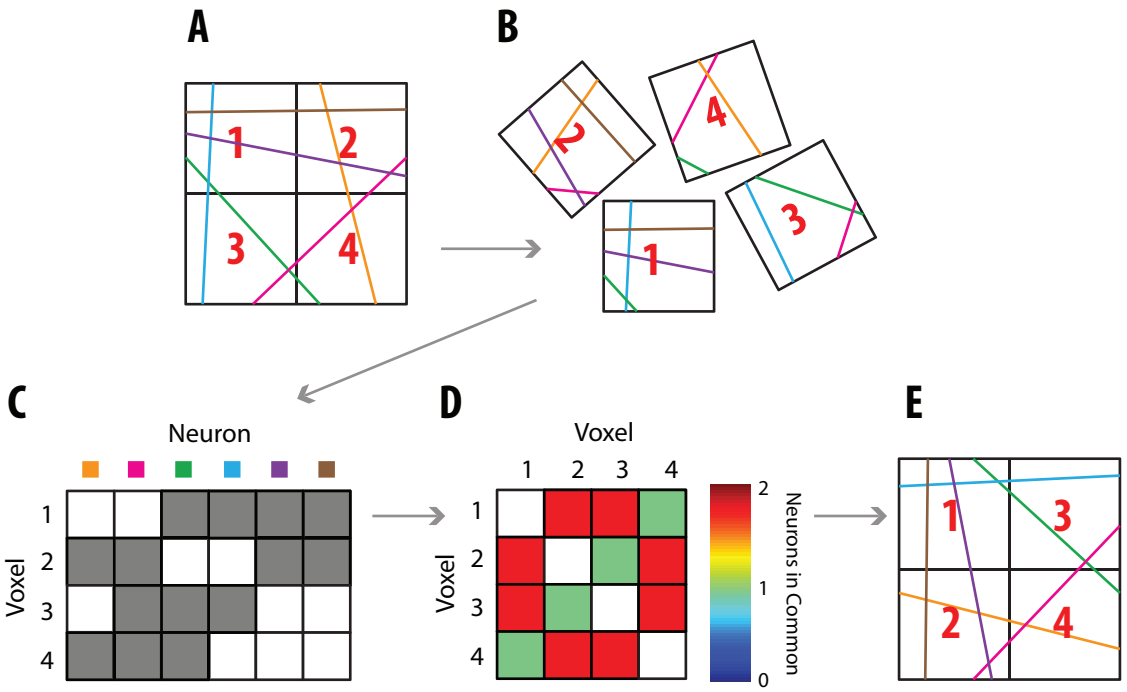

Fig 2. Neural Voxel Puzzling Overview. (A) An example of 6 "neurons" (lines) going through 4 voxels. Each neuron has a unique DNA barcode (here color). **(B)** These voxels are broken apart. **(C)** A coincidence matrix, **X**, is constructed describing which neurons are in which voxels. Gray signifies that a neuron is in a particular voxel. **(D)** A similarity matrix is constructed describing how many neurons a pair of voxels has in common. This matrix can be calculated as $\mathbf{X}\mathbf{X}^T$. **(E)** The voxels are puzzled back together. The reconstruction may be rotated or flipped, as shown here.

doi:10.1371/journal.pone.0131593.g002

variant of Diffusion Maps [8], which uses a sparse similarity (affinity) matrix instead of the standard calculation. We will refer to this method as Sparse Diffusion Maps, or SDM. The second is a variant of Landmark Isomap [13, 14], which is faster for unweighted graphs (binary similarity matrices). We will refer to this method as Unweighted Landmark Isomap, or ULI. In the below simulations, we use 10 landmark points. For the full algorithms, see Methods.

Performance. We tested the ability of both dimensionality reduction algorithms to determine the locations of 8000 simulated voxels of varying dimensions. Voxels were not confined to be cubes; they could be any shape. In our simplistic simulations, our "neurons" were long rods with cross-sectional areas of 1 μm^2 (about the size of an axon [15]) and were assumed to fully go through each voxel they entered. We set the total number of neurons in our simulations so that they would fill voxels of the chosen size. Neurons were oriented within the volume at randomly determined angles. See Methods for simulation details.

In our simulations, we used two metrics to determine the quality of the reconstruction. The first metric was the mean error in distance across all voxels. Because the final reconstruction will be a scaled, rotated, and reflected (flipped) version of the initial voxel locations, we first transformed (scaled, rotated, and reflected) the final reconstruction to match the original voxel locations. We used the transformation that minimized the mean error.

Another metric we used is the correlation coefficient (R value) of the plot of reconstructed distances between all points against their true distances. This metric determines how well relative distances between points are preserved, and does not depend on any transformations following the reconstruction. A perfect reconstruction (ignoring differences in scaling, rotation,

and reflection) would have a linear relationship between the true and reconstructed distances, and thus an R value of 1.

We first tested both methods on simulations with voxels with average sides of 5 μm (Fig 3A). The SDM method led to a faithful reconstruction, with the exception that the reconstructed voxels tended to be overrepresented around the outside of the cube and underrepresented in the middle. The ULI method also leads to a faithful reconstruction (Fig 3B).

We next tested both methods of reconstruction while varying the voxel size (Fig 3C, 3D, 3E). While voxels could be any shape, for ease of understanding, we report voxel sizes as the edge length of the cube corresponding to the average voxel size. For most voxel sizes, ULI leads to a slightly more accurate reconstruction than SDM. In general, when looking at the error in terms of absolute distance, using smaller voxels increases possible resolution. As an example of the excellent resolution that can be achieved, both methods achieve mean errors below 6 μm using an average voxel size of 3 μm (Fig 3E).

Finally, we tested the performance of puzzle imaging when removing a fraction of the voxels. We did this because in real neurons, there are cell bodies that would fill multiple voxels. These voxels would not add any information to puzzle reconstruction, so we exclude them to simulate the existence of cell bodies. Moreover, this generally simulates the robustness of neural voxel puzzling to missing data. In these simulations, we used a voxel size of 5 μm (Fig 3F, 3G). For both methods, when no voxels were removed, the average mean error rate was below 2 voxels. The average mean error rate was still below 2.5 voxels when 60% of the voxels were removed. This demonstrates that these methods are robust to missing voxels (cell bodies).

Neural Connectomics Puzzling

Overview. The purpose of neural connectomics puzzling is to estimate the locations of neurons based on their neural connections. There is ongoing work to provide a unique DNA barcode to every neuron and link those barcodes when neurons are synaptically connected [2]. This DNA could later be sequenced to determine all neural connections. For this proposed technology, it would be useful to know the locations of connected neurons, rather than simply knowing that two neurons are connected.

The first steps in connectomics puzzling are to label each neuron in the brain with a unique DNA barcode and label neural connections via the pairing of barcodes (Fig 4A). One proposed method would be to have viruses shuttle the barcodes across synapses, where they can be integrated [2]. Other techniques for creating, pairing, and transporting barcodes have been proposed [10, 16]. Next, the brain is homogenized, and the DNA barcode pairs are sequenced, yielding a record of which neurons are connected (Fig 4B).

Dimensionality Reduction. Each neuron can be treated as an *N*-dimensional object, where *N* is the number of neurons. Each dimension corresponds to a connection with a given neuron (Fig 4C). These *N*-dimensional neurons must be mapped to their correct 3-dimensional coordinates (or 2 dimensions in Fig 4's simplified example). This can be done because neurons are more likely to be connected to nearby neurons (e.g. [17, 18]), and thus we can treat the connectivity matrix as a similarity matrix. Using this similarity matrix, we can then puzzle the neurons back into place (Fig 4D). As with voxel puzzling, it is impossible to know exact locations without any additional information; only the relative locations of neurons can be known (Fig 4D).

In order to convert the knowledge of which neurons are connected (Fig 4C) to a reconstructed puzzle (Fig 4D), we used SDM, which we previously used for Neural Voxel Puzzling. See *Algorithm Comparison*: for an explanation on why ULI will not work for Neural Connectomics Puzzling.

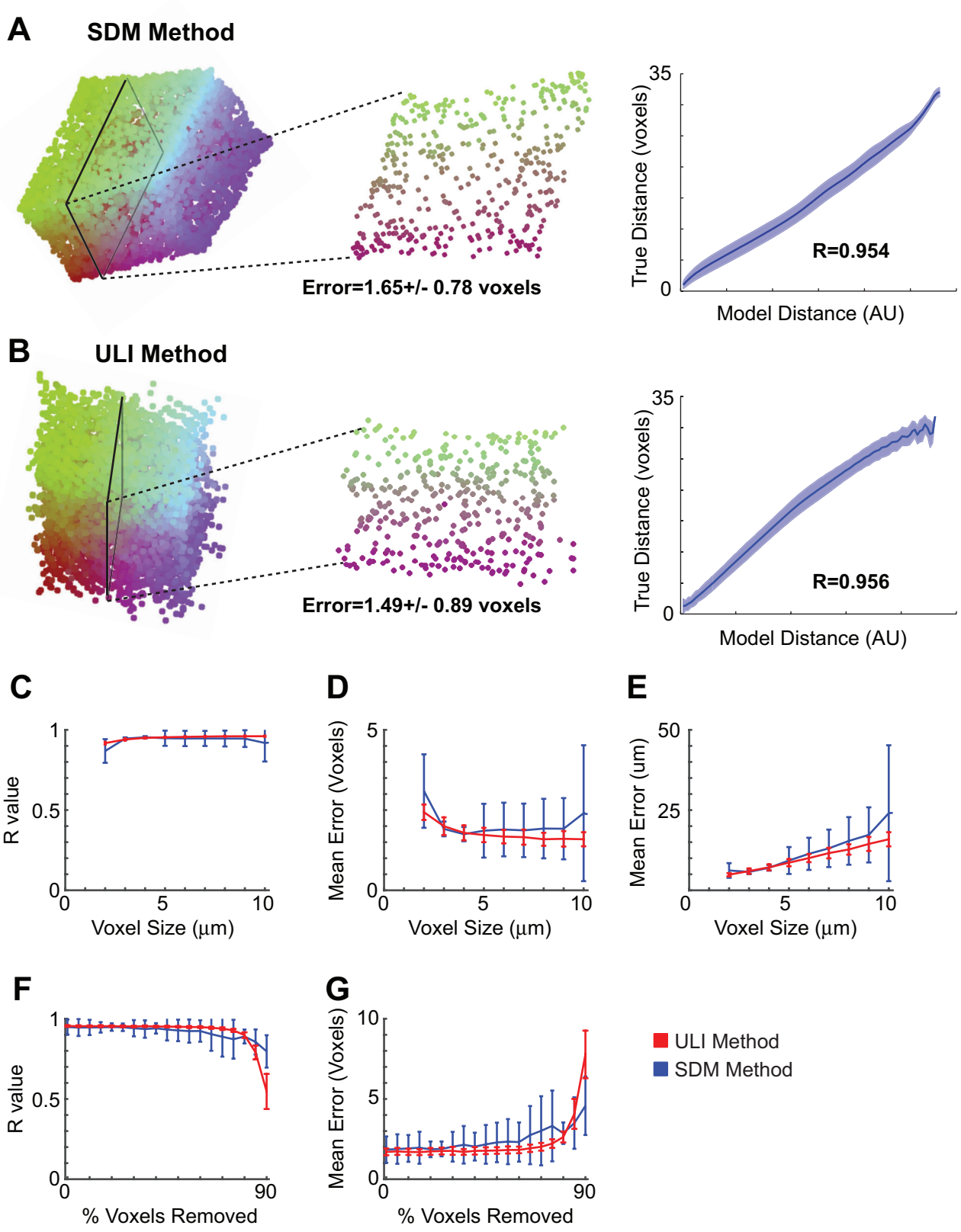

Fig 3. Neural Voxel Puzzling Performance. (A) On the **left**, an example reconstruction of voxel locations using the SDM method. Colors are based on initial locations: those with a larger initial *x* location are redder, while those with a larger initial *y* location are bluer. In the **middle**, a 2-dimensional slice through

reconstructed volume. The distance errors are calculated following scaling and rotating the reconstructed volume to match the original volume. On the **right**, one metric for the accuracy of reconstruction is shown by plotting the reconstructed distances between all points against their true distances for the reconstruction in this panel. The mean plus/minus the standard deviation (shaded) is shown. A perfect reconstruction would be a straight line, corresponding to an R value of 1. **(B)** Same as panel A, except an example reconstruction using the ULI method. **(C)** R values for simulations using the SDM method (blue) and ULI method (red), as a function of the voxel size. While voxels were not confined to be cubes, for ease of understanding, we report voxel sizes as the edge length of the cube corresponding to the average voxel size. Error bars represent the standard deviation across simulations in each panel. **(D)** Mean distance errors in voxels for both methods as a function of the voxel size. **(E)** Mean distance errors in microns for both methods as a function of the voxel size. **(F)** Voxels are removed to represent voxels that do not contain location information (such as voxels that contain a single cell body). R values for simulations using both methods are plotted as a function of the percentage of voxels removed. **(G)** Mean distance errors are plotted as a function of the percentage of voxels removed.

doi:10.1371/journal.pone.0131593.g003

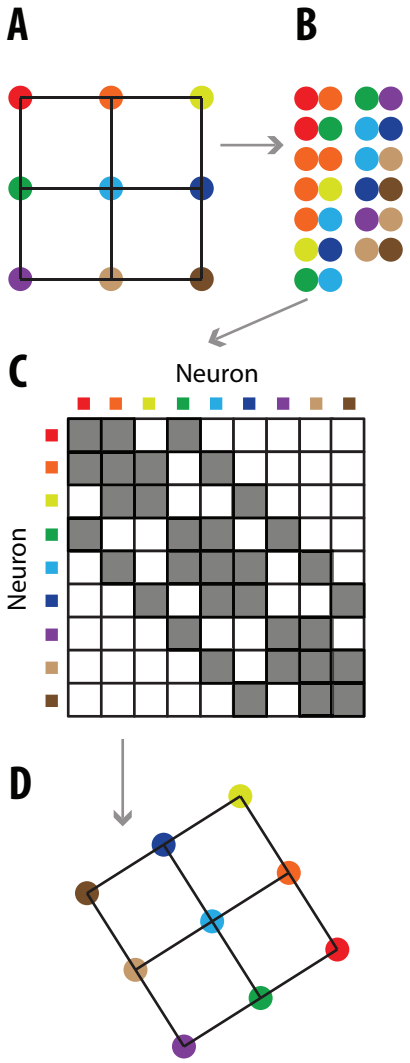

Fig 4. Neural Connectomics Puzzling Overview. (A) An example of 9 connected neurons (circles). Lines signify connections. **(B)** After the brain is homogenized, the only remaining information is a record of the connections. Connections are shown here as adjacent circles. **(C)** A connectivity matrix is constructed describing the connections between neurons. Gray signifies a connection. Since connections are correlated with how close neurons are to one another, this connectivity matrix can be treated as the similarity matrix. **(D)** The neurons are puzzled back together. The formation may be rotated or flipped, as shown here.

doi:10.1371/journal.pone.0131593.g004

Performance. We tested the ability of connectomics puzzling to determine the locations of a simulated network of neurons. Hellwig [17] described the probability of connections as a function of distance between pyramidal cells in layers 2 and 3 of the rat visual cortex. We simulated 8000 neurons in a 400 μm edge-length cube (so they were on average spaced $\sim$ 20 μm apart from each other in a given direction). Connections between neurons were randomly determined based on their distance using the previously mentioned probability function (Fig 5E).

We first simulated only connections within layer 3. As our example simulation shows (Fig 5A), the locations of neurons were able to be estimated very accurately. As with voxel puzzling, we use mean error and R values to describe the quality of reconstruction, although now the distance is between neurons rather than between voxels. Next we simulated connections between layers 2 and 3; in our simulation, half of the neurons were in layer 2 and half were in layer 3. In an example simulation (Fig 5B), it's clear that the locations of neurons could still be estimated accurately. In fact, the reconstruction clearly separated the neurons from the two layers.

Looking quantitatively at the differences between layer 3 reconstructions and layer 2/3 reconstructions, R values are slightly higher, and mean errors are slightly lower for layer 3 reconstructions. Median R values across simulations are 0.97 vs. 0.91 (layer 3 vs. 2/3), and median mean errors across simulations are 19 μm vs. 42 μm (Fig 5C, 5D). This disparity is largely due to the gap between layers in the reconstructions (Fig 5B); reconstructions within each layer are as accurate (in terms of R values) as the layer 3 reconstruction.

Next, in order to test when connectomics puzzling would be useful, we tested how reconstruction is dependent on the parameters of the connection probability distribution. For these simulations, we assumed the layer 3 connection probability distribution, and then changed either the baseline connection probability (the connection probability at a distance of 0) or the standard deviation of the connection probability distribution. There was a great improvement in reconstruction accuracy when the baseline probability increased from 0.10 to 0.15, and accuracy continued to increase until a baseline probability of 0.35, where it plateaued (Fig 5F). In general, there are high accuracy reconstructions for a wide range of baseline probabilities, with the exception of low probabilities.

When looking at the effect of the standard deviation of the probability distribution, there was a general trend that reconstruction accuracy decreased as the standard deviation increased (Fig 5G). This is because connections become less closely related to distances when the standard deviation increases. For example, if the standard deviation was infinite so that the probability of connection was the same for all distances, then knowing the connections would no longer allow us to infer the distances between neurons.

One notable exception from the above trend is that there is a low reconstruction accuracy for the smallest standard deviations considered (50–100 μm). We have found that the algorithm does not work well in connectomics puzzling when the connection matrix is far too sparse. When we use a cube with 200 μm edges and 1000 neurons instead of 400 μm edges and 8000 neurons (the same neuronal density), simulations with a 50 μm standard deviation perform very well. Thus, it's important to note that the quality of reconstruction will depend on the size of the cube being used.

Chemical Puzzling

Overview. Puzzle imaging has a number of potential applications besides neuroscience, including some that are potentially more tractable and therefore may have a higher probability of being implemented in the near future. One example we develop here is that of reconstructing spatial chemical environments from populations of recording cells that have been mixed, a process we call "chemical puzzling."

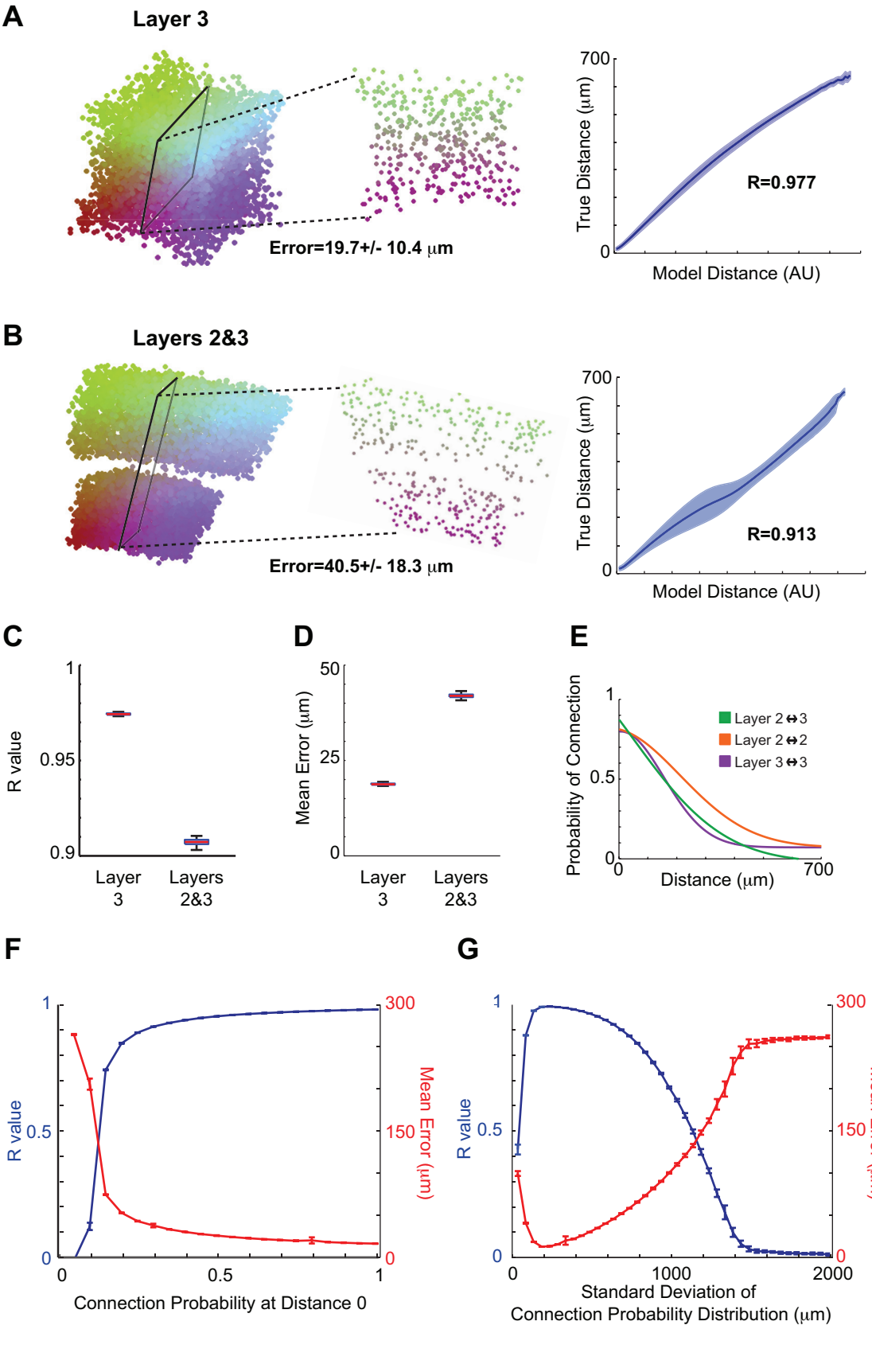
A
Layer 3
Error=19.7+/- 10.4 μm
700
True Distance (μm)
0
R=0.977
Model Distance (AU)
B
Layers 2&3
Error=40.5+/- 18.3 μm
700
True Distance (μm)
0
R=0.913
Model Distance (AU)
C
1
0.95
0.9
R value
Layer 3
Layers 2&3
D
50
25
0
Mean Error (μm)
Layer 3
Layers 2&3
E
1
0.5
0
Probability of Connection
Layer 2↔3
Layer 2↔2
Layer 3↔3
0
700
Distance (μm)
F
1
0.5
0
R value
300
150
0
Mean Error (μm)
0
0.5
1
Connection Probability at Distance 0
G
1
0.5
0
R value
300
150
0
Mean Error (μm)
0
1000
2000
Standard Deviation of
Connection Probability Distribution (μm)

Fig 5. Neural Connectomics Puzzling Performance. (A) On the **left**, an example reconstruction of neural locations based on a simulation of connections of pyramidal cells in layer 3 of cortex. Colors are based on initial locations: those with a larger initial x location are more red, while those with a larger initial y location are more blue. The distance errors are calculated following scaling and rotating the reconstructed volume to match the original volume. In the **middle**, a 2-dimensional slice through reconstructed volume. On the **right**, one metric for the accuracy of reconstruction is shown by plotting the reconstructed distances between all points against their true distances for the reconstruction in this panel. The mean plus/minus the standard deviation (shaded) is shown. A perfect reconstruction would be a straight line, corresponding to an R value of 1. **(B)** Same as panel A, except based on an example simulation of pyramidal cells in both layers 2 and 3 of cortex. **(C)** Boxplots of R values for layer 3 simulations, and layer 2/3 simulations. The 5 lines (from bottom to top) correspond to 5%, 25%, 50%, 75%, and 95% quantiles of R values across simulations. **(D)** Boxplots (as in panel C) of mean errors across simulations. **(E)** The probability of connection as a function of distance between pyramidal cells, which is used in the simulations of the other panels [17]. **(F)** Using the parameters of the connectivity probability distribution of layer 3, the baseline connection probability (the probability of connection at a distance of 0) of the connectivity distribution is changed. R values and mean errors are shown as a function of this baseline probability. Error bars represent the standard deviation across simulations in this and the next panel. **(G)** Using the parameters of the connectivity probability distribution of layer 3, the standard deviation of the connectivity distribution is changed. R values and mean errors are shown as a function of the standard deviation.

doi:10.1371/journal.pone.0131593.g005

One example of a chemical puzzling assay would consist of the initial spreading of many "pioneer" cells across an environment containing a heterogeneous distribution of a particular chemical (Fig 6A, 6B). These pioneer cells would be endowed with the ability to detect the presence of that chemical and to record its concentration into a nucleotide sequence. This could be done through molecular ticker-tape methods using DNA polymerases [3, 19, 20], similar strategies using RNA polymerases, or other mechanisms [4, 5, 21, 22], for example involving chemically-induced methylation or recombination.

Some pioneer cells would also be given the ability to share genetic information with other cells—in the case of prokaryotes, this could be by the introduction of barcoded F-like plasmids encoding the components essential to conjugative transfer of the plasmid [23]. The pioneer cells would be placed at random places within the chemical environment (Fig 6B), and would then begin to grow outwards (Fig 6C). When the colonies become large enough to contact neighboring colonies, those that contain the F-plasmid (or equivalent), denoted F^+, will copy it and transfer it to those without it (F^-) (Fig 6D). We can think about the F-plasmid transfer between the colonies descended from two pioneer cells as these two pioneer cells becoming "connected" (just like a neural connection in the previous section). This sharing of genetic information provides information about which pioneer cells are close to each other, which can then be used to reconstruct where each cell was spatially.

Again, note that a benefit resulting from this strategy is that the spatial information can be destroyed and reconstructed *post hoc*. In terms of this example, this would be equivalent to wiping the surface that the bacteria are growing on and reconstructing its chemical spatial information *after* sequencing, in "one pot." Preparation of the DNA for sequencing could also be done in one pot, i.e. all of the cells could be lysed and the DNA extracted at one time, if the genetic material carrying the connection information and the chemical information were physically linked (e.g. if a barcode on the F-plasmid were inserted into the chemical record region of the recipient cell's genetic information). Otherwise, amplification of the connection information and the chemical record could be done on each cell using emulsion-based methods [24].

Dimensionality Reduction. Each pioneer cell can be thought of as an N-dimensional object, where N is the number of pioneer cells. Each dimension corresponds to a connection with another pioneer cell (whether the pioneer cell's descendants are involved in a conjugative transfer with another pioneer cell's descendants) (Fig 6E). These N-dimensional cells must be mapped to their correct 2-dimensional coordinates. This can be done because pioneer cells will be more likely to be connected to nearby pioneer cells. We construct a similarity matrix by determining all cells two connections away. This is because the matrix of connections doesn't directly provide accurate information about how close the pioneer cells are to each other, as F^-'s can't be connected to F^-'s (and same for F^+'s). The matrix of mutual connections allows

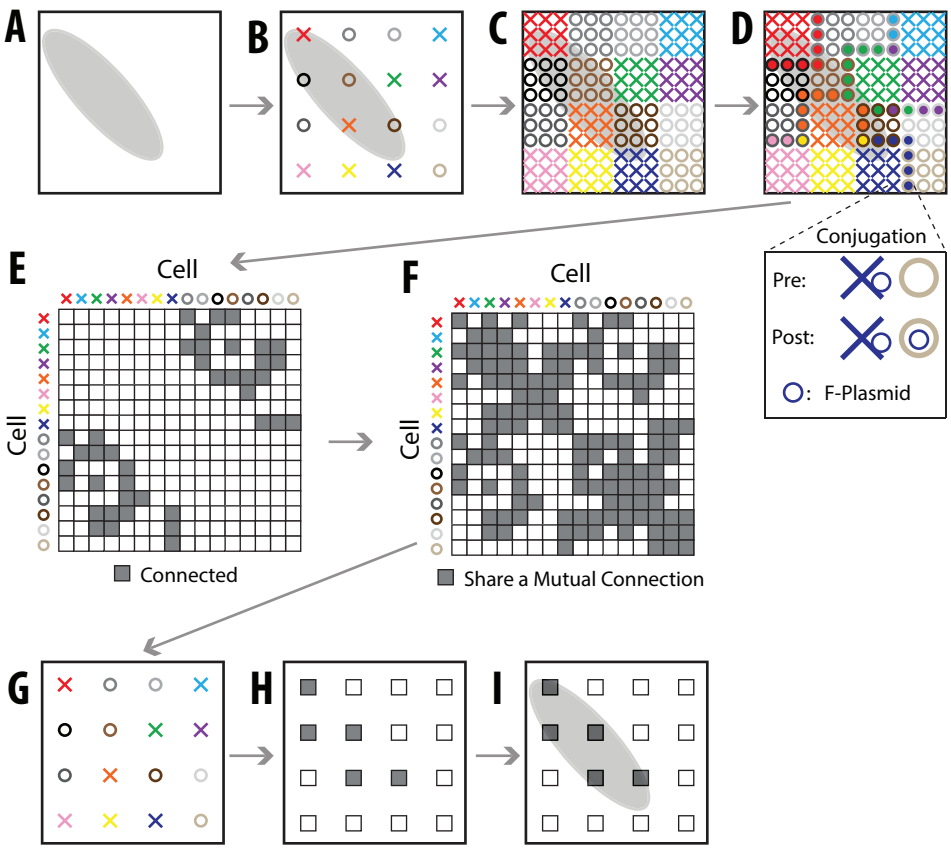

Fig 6. Chemical Puzzling Overview. (A) An example area contains a chemical whose concentration is represented by a grayscale value. **(B)** Pioneer cells (X's and O's, each with a different color) are put on the plate. X's represent F^+ cells that can transfer an F-plasmid into an F^- cell (O's). **(C)** The pioneer cells replicate and spread. Descendants of a pioneer are shown in the same color. **(D)** When the cell colonies become large enough to contact neighboring colonies, the F^+ cells (X's) will copy the F-plasmid and transfer it to the F^- cells (O's). This is shown as the X's color filling in the center of the O's. In the inset (below), the F-plasmid transfer (conjugation) is shown. **(E)** The DNA is sequenced to determine which pioneer cells are "connected" (which had a conjugative transfer occur between their colonies). A connectivity matrix is made from this data. **(F)** The matrix of connections doesn't directly provide accurate information about how close the original cells are to each other because O's can't be connected to O's (and same for X's). As our similarity matrix, we thus use the matrix of mutual connections, which allows O's to be connected to O's. **(G)** The location of the original cells is estimated from the matrix in panel F. **(H)** The chemical concentrations at each of the original cells locations is known as the cells' DNA acts as a chemical sensor. **(I)** The chemical concentration everywhere is extrapolated based on the chemical concentrations at the known pioneer cells.

doi:10.1371/journal.pone.0131593.g006

F^-'s to be connected to F^-'s. With the knowledge of the similarity between all pairs of pioneer cells, we can puzzle the pioneers back into place (Fig 6G). The chemical recording functionality of the cells then allows the chemical environment to be determined at the locations of the pioneer cells (Fig 6H), and extrapolated beyond (Fig 6I).

In order to convert the knowledge of the similarity between cells (Fig 6F) to a reconstructed puzzle (Fig 6G), a dimensionality reduction algorithm is needed. Here we use Unweighted Landmark Isomap, as we used in Neural Voxel Puzzling, with only 5 landmark points. See *Algorithm Comparison*: for an explanation on why SDM will not work well for Chemical Puzzling.

Performance. To further demonstrate the potential for chemical puzzling, we performed a simulation of the chemical puzzling problem. We used a complex chemical concentration described by the letter "P" (for Puzzle Imaging), with the concentration also decreasing when moving outwards from the center, and a constant background concentration (Fig 7A). The image size is 1000 x 1000 pixels, and each pixel is 1 μm^2 (about the size of a cell [25]). The corresponding size of the letter, then, is about 600 μm x 800 μm. Each pioneer cell is randomly placed on a single pixel.

We simulated the placement and growth of thousands of pioneer cells in that environment (Fig 7B, 7C). Each cell synthesized a DNA element in which the percentage GC composition of the synthesized DNA was proportional to the chemical concentration at that spot. We were able to successfully reconstruct the letter "P" in the image (Fig 7B, 7C), though this was dependent on the number of pioneer cells used and the length of the incorporated DNA element. Fewer pioneer cells resulted in a decrease in the spatial resolution, whereas the dynamic range greatly increased with longer DNA incorporations (Fig 7B, 7C). When only two base pairs are used, the background concentration and the concentration decrease away from the center were unable to be accurately detected. With 50 base pairs, the chemical concentrations were reconstructed very accurately.

Lastly, the above simulations assumed that when a F^+ and F^- cell are in contact, transfer of genetic information occurs 100% of the time, which would likely not occur [26]. Still, relatively faithful chemical reconstruction was accomplished with conjugation efficiencies as low as 30% (see S1 Fig and S1 Text). Overall, this technique holds the potential to determine chemical concentrations at very high resolution.

Discussion

Here we proposed the concept of puzzle imaging. We developed two possible large-scale nonlinear dimensionality reduction algorithms for use in puzzle imaging, and demonstrated some of puzzle imaging's abilities and weaknesses in three possible applications. Using simplistic simulations, we showed that voxel puzzling may allow locating neural structures within about 10 μm. In regards to connectomics puzzling, knowing only the connections between neurons within a layer of cortex could be sufficient to localize neurons within about 20 μm. We also showed that chemical puzzling could be used to accurately determine chemical concentrations at a resolution well below 1 mm. Lastly, we describe how Sparse Diffusion Maps is faster than Diffusion Maps, and Unweighted Landmark Isomap is faster than Isomap (see Methods).

Potential Uses of Puzzle Imaging

For neuroscience, puzzle imaging could be a scalable way to localize important molecular information within the brain. Neuronal RNA/DNA barcodes could be annotated with additional molecular information [10, 16]. For example, molecular ticker tapes have been proposed that would record the activity of a neuron into DNA [3, 19, 20]. It would be very valuable to know the location of the neurons that are being recorded from. Additionally, RNA that is being expressed could be annotated to the barcodes. This could provide information about what genes are being expressed in different locations of the brain, or about the distribution of cell types throughout the brain. If engineered cells capable of recording and/or stimulating adjacent neurons can circulate in and out of the brain, the concepts outlined here might help achieve input/output to many/all neurons without surgery.

The applications of chemical puzzling go beyond that of determining the chemical composition of bacterial cells growing on a surface. Indeed, biology often has the ability to survive and thrive in the harshest of environments, including spaces too small or desolate for even robotic

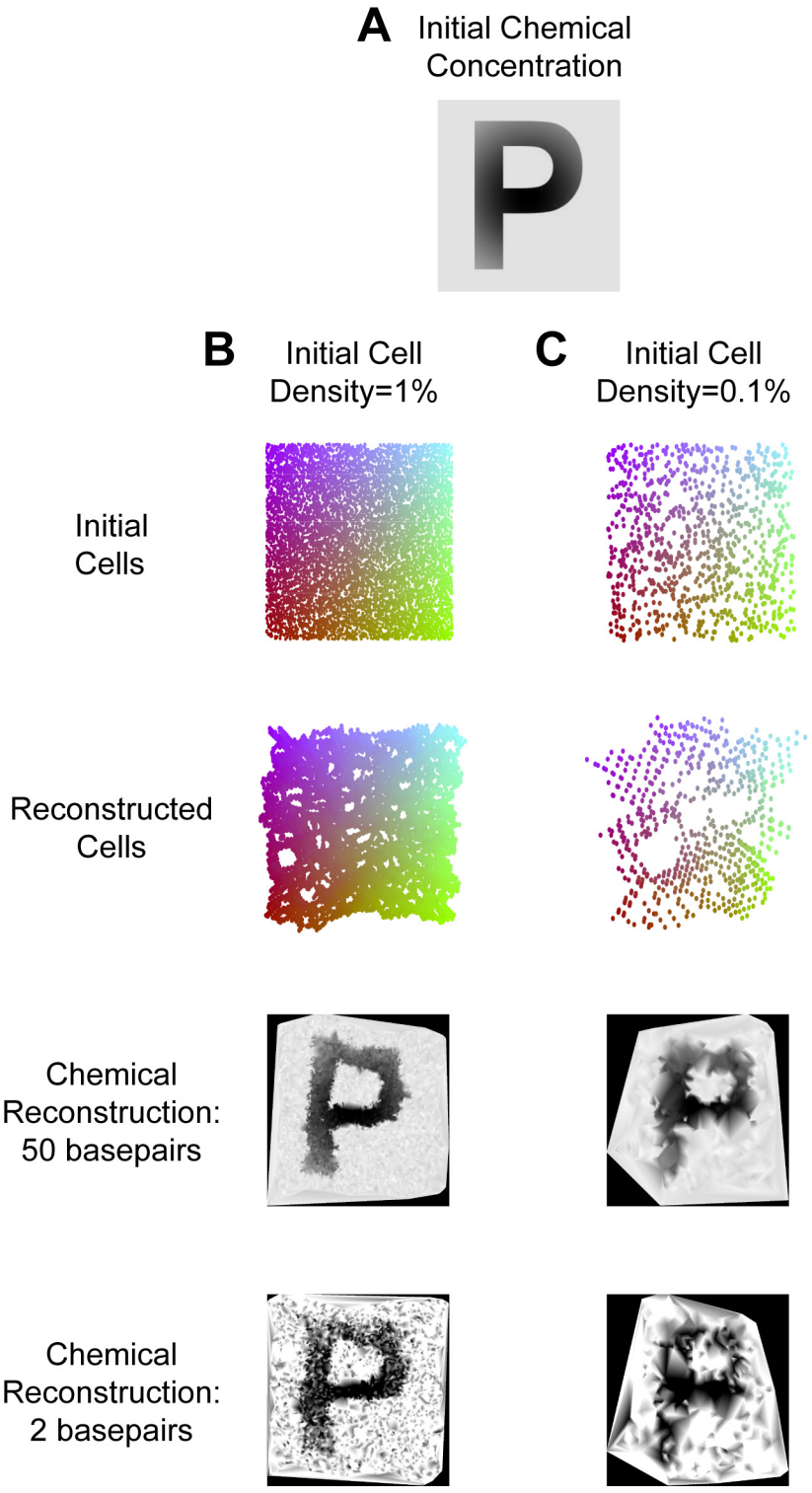

Fig 7. Chemical Puzzling Performance. (A) The chemical concentration across the plate. It is described by the letter "P," with the concentration decreasing moving outwards from the center, and a constant background concentration. **(B)** A simulation is done with an initial cell density of 1%. **(C)** A simulation is done with an initial cell density of 0.1%. For panels B and C, the **top row** shows the initial locations of the pioneer cells. They are color-coded by location. The **second row** shows example reconstructed locations of the pioneer cells. The

third row shows the reconstructed chemical concentrations when 50 base pairs are used to detect the concentration. The **bottom row** shows the reconstructed chemical concentrations when 2 base pairs are used to detect the concentration. Note that the black border represents regions of unknown concentration.

doi:10.1371/journal.pone.0131593.g007

access. Biological sensing and recording can open these areas to characterization and perhaps utilization. Applications could include those in the fields of geology (in which the composition of fractured geologic strata, which contain millions of microscopic cracks and pores, can be assayed) and medicine (in which the composition of the complex biological environments of, for example, the gastrointestinal tract, can be assayed).

Simulation Limitations

In all our simulations, we made simplifications in order to provide a preliminary demonstration of the feasibility of puzzle imaging. In neural voxel puzzling, our simulations assumed that there were equal neuronal densities across the volume and that neurons were oriented at random angles. In neural connectomics puzzling, our simulations used a maximum of two neuronal layers, and assumed that connection probability distributions did not differ within a layer (although in reality there are different cell types [27]). For both of these neuroscience examples, reconstruction errors due to these simplifying assumptions could likely be remedied by using previously known neuroanatomical information (e.g. the orientation of axons in a brain region) or molecular annotations (e.g. about whether a barcode is in an axon or dendrite, or about cell type [28]). In chemical puzzling, our simulations assumed that cells were stationary, and we used a simple model of outward cellular growth. For locations with viscous flow or rapidly moving cells, complex cell growth and movement models would be necessary to achieve accurate reconstructions.

For a more in-depth discussion of limitations for all simulations, see S2 Text.

Algorithms

In this paper, we demonstrated two algorithms that could be used for puzzle imaging. Refinements of the presented algorithms would be beneficial for puzzle imaging (to overcome the limitations in *Results*). For instance, a different metric could be used to create the similarity matrix in both methods. In addition, the number of landmark points in Unweighted Landmark Isomap can be optimally tuned for the specific use. Moreover, novel algorithms for large-scale nonlinear dimensionality reduction would be beneficial for more accurate puzzle imaging reconstruction.

While the algorithms presented here were designed for puzzle imaging, they could be generally used as faster versions of Diffusion Maps and Landmark Isomap for large problems. Both methods can preserve relative locations when reconstructing a swiss roll or helix, classical tests of nonlinear dimensionality reduction techniques. Further research needs to be done to see how these methods compare to traditional methods for other applications.

Conclusion

Here we have provided a preliminary demonstration that puzzle imaging may be possible. In order to make puzzle imaging a reality, significant biological work still needs to be done. For example, having location or neuron specific barcodes would be needed to make the

Table 1. Sparse Diffusion Maps Algorithm.

Sparse Diffusion Maps Algorithm	
Step 1	Efficiently create a sparse, symmetric, nonnegative, similarity matrix, **S** that is a connected graph. Methods used to create **S** in our applications follow.
Step 2	Create matrix **M** by normalizing **S** so that each row sums to 1. That is, $\mathbf{M} = \mathbf{D}^{-1}\,\mathbf{S}$, where **D** is a diagonal matrix with $\mathbf{D}_{i,j} = \sum_j \mathbf{S}_{i,j}$
Step 3	Find the k+1 largest eigenvalues $e_1, \ldots, e_{k+1}$ of **M** and their corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}$. Each eigenvector is N-dimensional.
Step 4	The final k-dimensional positions of the N points are the N rows in $[e_2\,\mathbf{v}_2, \ldots, e_{k+1}\,\mathbf{v}_{k+1}]$.

doi:10.1371/journal.pone.0131593.t001

neuroscience puzzle imaging approaches possible. We hope that this paper will inspire experimentalists and theoreticians to collaborate to help make puzzle imaging a reality.

Methods

Sparse Diffusion Maps Algorithm

Sparse Diffusion Maps is the same as the standard Diffusion maps algorithm [8] when using 1 timestep, except we here use a sparse similarity matrix. Thus, in the below algorithm, only step 1 differs from standard Diffusion maps. The algorithms take N high-dimensional points, and reduce each point to k dimensions (where k is 2 or 3 in our applications). The algorithm follows in Table 1.

Unweighted Landmark Isomap Algorithm

Unweighted Landmark Isomap is based on Landmark Isomap [13, 14]. The most important change is that we compute geodesic distances more efficiently due to our graph being unweighted (Step 3). Additionally, we create the similarity matrices uniquely for each application (Step 1). All other steps are identical to Landmark Isomap. The algorithm follows in Table 2.

Note that the method of constructing the similarity matrix (Step 1) and the number of landmark points (Step 2) are user options within this algorithm. We discuss our choices for these steps for our applications below.

Neural Voxel Puzzling

Similarity Matrix. Let **X** be an $n \times p$ (voxels × neurons) coincidence matrix that states which neurons are in which voxel. To construct the similarity matrix, we first compute $\mathbf{A} = \mathbf{X}\,\mathbf{X}^T$, an $n \times n$ matrix that gives the similarity between voxels. We then threshold this matrix (an approximate nearest neighbors calculation). Thresholding makes the matrix more sparse (beneficial for SDM), and makes points not connected to far-away points (important for increasing resolution in ULI). We use different thresholding methods prior to ULI and SDM. The specific thresholding methods below are what we used for our simulations; the method of thresholding is a tunable parameter.

For ULI, we threshold each row independently. The threshold for a row is the maximum of that row divided by 2. That is, for row i, $T_i = \frac{1}{2}\max_j \mathbf{A}_{ij}$, where $\mathbf{A}_{ij}$ is the element of **A** in row i

Table 2. Unweighted Landmark Isomap Algorithm.

Unweighted Landmark Isomap Algorithm	
Step 1	Efficiently create a sparse, binary, symmetric, similarity matrix, **S** that is a connected graph. Methods used to create **S** in our applications follow.
Steps 2&3	These steps select a landmark point (Step 2) and then calculate the distance from all points to that landmark point (Step 3). In total, we will select ℓ landmark points, so these steps will be repeated in alternation ℓ times.
Step 2	Select a landmark point. We do this in the following way (MaxMin algorithm from [13]):The first landmark point is selected randomly. Each additional landmark point is chosen in succession in order to maximize the minimum distance to the previously selected landmark points. In other words, for every point i, we calculate the minimum distance m_i to all the previously selected landmark points (indexed by j): $m_i = \min_j \gamma_{i,j}$, where $\gamma_{i,j}$ is the calculated geodesic distance between the points in Step 3. We then choose the point that has the largest minimum distance: $landmark = \max_i m_i$.
Step 3	Calculate the geodesic distance from all N points to the previously selected landmark point. The geodesic distance $\gamma_{i,j}$ from point i to landmark point j is the number of steps it would to get from point i to landmark point j in the graph of **S**. This can be calculated efficiently using a breadth first search (as opposed to using Dijkstra's algorithm) because **S** is an unweighted graph.
Step 4	Use classical multidimensional scaling (MDS) [6] to place the landmark points. This means that the points are placed in order to minimize the sum squared error between the Euclidean distances of the placed landmark points and their known geodesic distances. This is done in the following way [13, 14]:Let $\mathbf{\Delta}^{\ell}$ be the matrix of squared geodesic distances between all the landmark points. Let μ_i be the mean of row i of $\mathbf{\Delta}^{\ell}$ and μ be the mean of all entries of $\mathbf{\Delta}^{\ell}$. Create the matrix **B** with entries $\mathbf{B}_{j,k} = [\Delta^{\ell}]_{j,k} - \mu_j - \mu_k + \mu$. Find the first (largest) k eigenvalues and eigenvectors of **B**. Let $\mathbf{V}_{i,j}$ be the i^{th} component of the j^{th} eigenvector. Let λ_j be the j^{th} eigenvalue. The matrix **L** has the coordinates of the landmark points in its columns. **L** has the entries $\mathbf{L}_{i,j} = \mathbf{V}_{j,j} \cdot \sqrt{\lambda_i}$.
Step 5	For each point, triangulate its location using the known geodesic distances to each of the landmark points. This is done in the following way [13, 14]:Let $\mathbf{L}^{\#}{}_{i,j} = \frac{\mathbf{V}_{j,j}}{\sqrt{\lambda_i}}$, and let $\mathbf{\Delta}$ be the matrix of squared geodesic distances between the landmark points and all points. The position of point a, $\mathbf{x}_a$, is calculated by $\mathbf{x}_a = -\frac{1}{2}\mathbf{L}^{\#}\left(\left(\Delta_a - \bar{\Delta}^{\ell}\right)\right)$, where $\mathbf{\Delta}_a$ is row a of the matrix $\mathbf{\Delta}$ (the squared geodesic distances from point a to all the landmark points), and $\bar{\Delta}^{\ell}$ is the column mean of the matrix $\mathbf{\Delta}^{\ell}$.

doi:10.1371/journal.pone.0131593.t002

and column j. After thresholding, the entries of the non-symmetric similarity matrix are $\mathbf{N}_{ij} = \mathbf{A}_{ij} \geq T_i$. We create a symmetric unweighted similarity matrix: $\mathbf{S} = (\mathbf{N} + \mathbf{N}^T) > 0$.

For SDM, the entire matrix is thresholded by the same value. We first create a vector **w** that contains the maximum of each row of **A**. Its entries are $w_i = \max_j \mathbf{A}_{ij}$. The threshold is the minimum of **w**. That is, $T = \min \mathbf{w}$. Thresholding all entries, we get $\mathbf{S} = \mathbf{A} \geq T$.

Simulations. For our simulations, we initially chose the average voxel size. If the average voxel had a volume of x^3 μm^3, we reported the voxel size as x μ (the length of an edge of a cube of comparable size). We next created a cube with edges of length $20x$ μm (so it had a volume of $8000x^3$ μm^3). We then randomly placed 8000 voxel centers in the cube. We next added in long rectangular prisms (neurons) so that the entire volume would be filled. We assumed the neurons fully went through each voxel they entered. As the cross-sectional area of a voxel was on average x^2 μm^2, and we assumed neurons with cross-sectional areas of 1 μm^2 (about the size of an axon), we added enough neurons so that each voxel would contain x^2 neurons. Neurons were oriented within the volume at randomly determined angles.

When doing simulations with removed voxels, we placed voxels and neurons in the regular way. Then, we discarded voxels and aimed to reconstruct the locations of the other voxels. The

R values and mean errors listed are for the voxels that were reconstructed (removed voxels were not included in the analysis).

Additionally, in our simulations, voxels that did not contain any neurons, or that did not share any neurons with another voxel, were excluded from analysis. When voxels were 2 μm, 3 μm, 4 μm, and 5 μm, 2.8%, 0.14%, 0.0046%, and 0.0003% of voxels were respectively excluded. Also, when removing voxels (Fig 2F, 2G), we excluded the remaining voxels that did not make up the main connected component of the similarity graph. When using ULI, when 80%, 85%, and 90% of voxels were removed, 0.36%, 1.7%, and 34.7% of remaining voxels were excluded from analysis. When using SDM, when 80%, 85%, and 90% of voxels were removed, 0.05%, 0.09%, and 0.37% of remaining voxels were excluded from analysis.

In our simulations, when using ULI, we used 10 landmark points (see *Unweighted Landmark Isomap Algorithm* above).

Neural Connectomics Puzzling

Similarity Matrix. Let $\mathbf{C}$ be an $n \times n$ (neurons × neurons) connectivity matrix that states which neurons are connected to each other. As mentioned in *Results*, this connectivity matrix can be directly used as the similarity matrix.

Simulations. We randomly placed 8000 points (neurons) in a 400 μm edge-length cube. Thus, the neurons were on average $\sim$ 20 μm apart from each other in a single direction. We initially assumed all neurons were pyramidal cells in layer 3 of rat visual cortex and simulated connections according to the known connection probability distribution as a function of distance (Fig 5E) [17]. Every pair of neurons was randomly assigned a connection (or not) based on this probability distribution. Next, we simulated neurons in layers 2 and 3. The top half of the cube was assumed to be layer 2, and the bottom half was assumed to be layer 3. Again, every pair of neurons was randomly assigned a connection (or not) based on the relevant probability distribution (either between layer 2 and 2, layer 2 and 3, or layer 3 and 3; Fig 5E) [17].

Chemical Puzzling

Similarity Matrix. Let $\mathbf{C}$ be an $n \times n$ (pioneer cells × pioneer cells) connectivity matrix that states which pioneer cells are "connected" to each other. The similarity matrix is calculated as $\mathbf{C}^T\mathbf{C}$.

Simulations. We randomly placed pioneer cells on a pixel in the 1000 × 1000 pixel image. The number of pioneer cells was 1 million (the number of pixels) times the initial density, so 10000 in Fig 3B, and 1000 in Fig 3C. Each cell was randomly assigned to be F^+ or F^-. We assumed the cells grew out circularly over time with approximately the same rate of growth. For every pixel, we determined which colony (progeny from which pioneer cell) would reach the pixel first. This produced a final map of all pixels assigned to different pioneer cells. On the borders of colonies, cells could conjugate with one another if one colony was F^+ and the other was F^-. For each pixel on the border, a conjugation occurred according to the probability of conjugation (100% probability for Fig 7; varying for S1 Fig). More specifically, any time different colonies occupied pixels horizontal or vertical of each other, a conjugation could occur for the cells in the bordering pixels.

We reconstructed the pioneer cells' locations using ULI with 5 landmark points (see *Unweighted Landmark Isomap Algorithm* above). We assumed that 3 cells were placed on the plate so that their locations were known. We scaled, rotated, and reflected the reconstructed locations of the pioneer cells so that the locations of the 3 known cells were matched (the average distance error was minimized).

The chemical concentration on the plate was between 0 and 1. If a pioneer cell was at a location with a chemical concentration of ρ, each base pair has a probability of being a G or C with probability ρ. The reconstructed chemical concentration was the total number of G's and C's divided by the number of base pairs. Thus, if there were 2 base pairs, the reconstructed chemical concentration could be 0, 0.5, or 1. We used linear interpolation to determine the chemical concentration of areas on the plate that did not have a reconstructed concentration.

We also note that along with our approximate simulation method of outward circular growth, we also ran a smaller, more realistic, simulation. In this stochastic growth simulation, during each round, cells randomly divided into any single adjacent unoccupied pixel. When an F^+ colony grew next to an F^- colony, or vice versa, cells could conjugate (according to the conjugation probability). F^- cells turned F^+ upon receipt of the F-plasmid from an F^+ cell. This growth simulation continued until the plate was full. On smaller plates (when it was less time consuming to run the realistic simulation), both simulation types produced nearly equivalent results.

Computational Complexity

Sparse Diffusion Maps. We list the complexity of Step 1 for the individual methods below. The run-time of this algorithm is dominated by Step 3. For the below complexity explanation, $\mathbf{S}$ is an $n \times n$ sparse matrix with m non-zero elements. The complexity of this algorithm is listed in Table 3.

Unweighted Landmark Isomap. We list the complexity of Step 1 for the individual methods below. The complexity of this algorithm is listed in Table 4.

Neural Voxel Puzzling. To compute the similarity matrix, we can take a shortcut instead of performing the sparse matrix multiplication $\mathbf{X}\,\mathbf{X}^T$ (which will be $\mathcal{O}(n^2)$ in the best case scenario [31]). The entries of $\mathbf{X}\,\mathbf{X}^T$ tell us how many neurons are shared between particular voxels. This can also be calculated by drawing the bipartite graph that $\mathbf{X}$ describes, with voxels on one side and neurons on the other (and edges when a neuron goes through a voxel). For a given voxel, to determine which voxels it shares a neuron with, we only need to trace all the paths to its connected neurons, and then the paths from those neurons back to voxels. This has a complexity of $\mathcal{O}(\alpha_{in} \cdot \alpha_{out})$, where α_{in} is the largest number of neurons in a voxel, and α_{out} is the largest number of voxels a neuron goes through. Thus, for all voxels, the complexity is $\mathcal{O}(n \cdot \alpha_{in} \cdot \alpha_{out})$. Determining the similarity matrix can take a comparable amount of time to the steps in ULI or SDM.

Table 3. Sparse Diffusion Maps Computational Complexity.

Sparse Diffusion Maps Computational Complexity		
Step	**Worst-case complexity**	**Further Explanation**
Step 3 (and Total)	$\mathcal{O}(m)$	This is the complexity for the power iteration algorithm [29], which is very efficient for solving for a small number of eigenvalues/eigenvectors. In the power iteration method, the matrix is continually multiplied by a vector, which has complexity $\mathcal{O}(m)$, until convergence. The number of steps to calculate eigenvalue e_i depends on $\lvert e_i/e_{i+1}\rvert$: a larger ratio means quicker convergence. If this was not a sparse matrix, as is the case for standard Diffusion Maps, then this step would be $\mathcal{O}(n^2)$.

doi:10.1371/journal.pone.0131593.t003

Table 4. Unweighted Landmark Isomap Computational Complexity.

Unweighted Landmark Isomap Computational Complexity		
Step	**Worst-case complexity**	**Further Explanation**
Step 2	$\mathcal{O}(\ell \cdot n)$	[13]
Step 3	$\mathcal{O}(\ell \cdot m)$	A breadth first search has a complexity of $\mathcal{O}(m)$. This is done for each of ℓ landmark points. If the graph had been weighted (as assumed in standard Landmark Isomap), then we would need to use Dijkstra's algorithm. The fastest general implementation is $\mathcal{O}(m + n \cdot \log n)$ [30] for each landmark point, yielding $\mathcal{O}(\ell \cdot m + \ell \cdot n \cdot \log n)$.
Step 4	$\mathcal{O}(\ell^3)$	[13]
Step 5	$\mathcal{O}(k \cdot \ell \cdot n)$	[13]
Total	$\mathcal{O}(\ell \cdot m + k \cdot \ell \cdot n + \ell^3)$	

doi:10.1371/journal.pone.0131593.t004

Note that this step is faster than the standard method of computing a similarity matrix in Diffusion Maps: $\mathbf{S}_{i,j} = \exp\left(\frac{-\|x_i - x_j\|_2{}^2}{2\sigma^2}\right)$, where x_i and x_j are columns within $\mathbf{X}$. For our sparse matrices, this would take $\mathcal{O}(q \cdot n)$, where q is the number of nonzero entries in $\mathbf{X}$, due to the pairwise vector subtraction. The Landmark Isomap algorithm does not give a method for computing the similarity matrix.

Neural Connectomics Puzzling. As no computation is required to construct the similarity matrix, the overall complexity of Neural Connectomics Puzzling is the complexity of the SDM algorithm.

Chemical Puzzling. As in neural voxel puzzling, we can take a shortcut to calculate the similarity matrix instead of performing the sparse matrix multiplication $\mathbf{C}^T\mathbf{C}$ (which would take $\mathcal{O}(n^2)$ at best for very sparse matrices [32]). Again, we can construct a bipartite graph representing $\mathbf{C}$, where the pioneer cells are now on both sides of the graph (and there's an edge between connected cells). We can determine if a pioneer cell is connected within 2 steps of another pioneer cell by counting all the ways to get to the other side of the graph (via connections) and back to cells on the same side. For a given cell, this has the complexity $\mathcal{O}(\beta^2)$, where β is the largest number of connections a pioneer cell has (how many different cell colonies that pioneer cell's colony has conjugated with). Thus, the total complexity of determining the similarity matrix is $\mathcal{O}(n \cdot \beta^2)$. Determining the similarity matrix can take a comparable amount of time to the steps in ULI.

Additionally, chemical puzzling has the extra step of reconstructing the image of chemical concentrations. The chemical concentrations are known at the reconstructed locations of the pioneer cells, but interpolation needs to be used to estimate the concentrations at other locations. This interpolation step can dominate the total time, depending on the desired resolution and interpolation method.

Algorithm Comparison: ULI vs SDM

There are pros and cons of both Sparse Diffusion Maps (SDM) and Unweighted Landmark Isomap (ULI), which make them suitable for different applications (Table 5).

Table 5. Algorithm Comparison Summary: ULI vs. SDM.

Pros/Cons Summary	
Sparse Diffusion Maps	**Unweighted Landmark Isomap**
Con: Only accurately reconstructs cubes	*Pro: Can accurately reconstruct non-cubes*
Con: Biased towards exterior points	*Pro: Not biased towards exterior points*
Pro: Is robust to problems that have high similarity between some far-away points.	Con: Is generally not robust to problems that have high similarity between some far-away points.

doi:10.1371/journal.pone.0131593.t005

Domain. To run ULI, the graph entered into the algorithm must contain only short-range connections. Practically, ULI will not work when there is high similarity (a large value in the similarity matrix) between points that are far away from one another. This is because having high similarity between far away points would make those far away points near each other in the reconstruction, a problem known as "short circuiting" [33]. This limitation means that neural connectomics puzzling, which contains long-range connections that are indistinguishable for short-range connections, is not compatible with ULI. SDM, on the other hand, is robust to high similarity values between far-away points, as long as similarity values are generally higher between nearby points. Thus, SDM does work for neural connectomics puzzling.

Reconstruction Accuracy. Reconstructions using SDM are generally biased towards having points around the perimeter, and therefore don't faithfully reconstruct the center of the volume. This perimeter bias makes the SDM method unsuitable for use in Chemical puzzling. When used with Chemical puzzling, SDM leads to faulty chemical reconstructions near the center of the image. For neural voxel puzzling, as seen in our simulations (Fig 3), ULI was slightly more accurate than SDM, except for when at least 85% of the voxels had been removed.

Another important note is that the SDM method will only accurately reconstruct the volume when the volume is a cube (S2 Fig). Thus, for the SDM method to be used in practice with neural voxel or connectomics puzzling, the brain would need to first be sectioned into cubes, and then the cubes would need to be pieced together. The ULI method, on the other hand has the benefit of accurately reconstructing volumes that are not cubes (S2 Fig).

Speed. Both SDM and ULI are designed to be fast dimensionality reduction methods for use with large datasets. In practice, when directly comparing their speed in the neural voxel puzzling simulations using 8000 5 μm voxels, SDM took about 1.5 seconds, while ULI (with 10 landmark points) took about 0.9 seconds on a 2.3 GHz processor running Matlab. The previous times did not include constructing the similarity matrix. See *Computational Complexity* above, and *Algorithm Time Improvements* below, for the computational complexity of larger problems.

Algorithm Time Improvements

Our algorithms took previous algorithms and adapted them to be faster for large-scale puzzle imaging.

Sparse Diffusion Maps vs. Diffusion Maps. The difference between Sparse Diffusion Maps and Diffusion Maps is that we construct a sparse similarity matrix. The most time consuming step in these algorithms (besides constructing the similarity matrix) is finding the largest k eigenvalues and eigenvectors, where k is the dimension size you're projecting into (2 or 3 for puzzle imaging). This computation will be significantly faster when the similarity matrix is

sparse. For instance, one fast algorithm, "power iteration," [29] has a complexity of $\mathcal{O}(m)$, where m is the number of non-zero elements in the similarity matrix $\mathbf{S}$ (see *Computational Complexity*).

Computing the similarity matrix is the other time consuming step in Diffusion Maps. Let's say $\mathbf{X}$ is an $n \times p$ matrix, and we want to compute the $n \times n$ similarity matrix $\mathbf{S}$ that gives the similarity between the columns of $\mathbf{X}$. This is generally calculated as $\mathbf{S}_{i,j} = \exp\left(\frac{-\|x_i - x_j\|_2^2}{2\sigma^2}\right)$, where x_i and x_j are columns within $\mathbf{X}$. Computing $\mathbf{S}$ would have a complexity of $\mathcal{O}(q \cdot n)$, where q is the number of nonzero entries in $\mathbf{X}$ (see *Computational Complexity*).

In our scenarios where we use SDM, we compute a sparse similarity matrix more efficiently. For neural connectomics puzzling, computing the similarity matrix takes no time, as it simply describes the connections. For neural voxel puzzling, $\mathbf{X}\,\mathbf{X}^T$ (the main step of computing the similarity matrix) can be calculated with a complexity of $\mathcal{O}(n \cdot \alpha_{in} \cdot \alpha_{out})$, where α_{in} is the largest number of neurons in a voxel, and α_{out} is the largest number of voxels a neuron goes through (see *Computational Complexity*). This approach is thus significantly faster than Diffusion Maps for our problem.

Unweighted Landmark Isomap vs. Landmark Isomap. The main difference between ULI and Landmark Isomap is that ULI uses an unweighted similarity matrix. One of the time-consuming steps in Landmark Isomap is computing the geodesic distance between all points and each of the ℓ landmark points. For weighted graphs, this can be solved fastest with Dijkstra's algorithm in $\mathcal{O}(\ell \cdot m + \ell \cdot n \cdot \log n)$ [30]. For unweighted graphs, we can simply use a breadth first search for each landmark point, which has a complexity of $\mathcal{O}(\ell \cdot m)$. When we include the other steps of the algorithm (other than constructing the similarity matrix), Landmark Isomap has a complexity of $\mathcal{O}(\ell \cdot m + \ell \cdot n \cdot \log n + k \cdot \ell \cdot n + \ell^3)$, while ULI is faster, with a complexity of $\mathcal{O}(\ell \cdot m + k \cdot \ell \cdot n + \ell^3)$ (see *Computational Complexity*).

It is important to note that computing the similarity matrices in neural voxel puzzling and chemical puzzling may take a comparable amount of time as ULI or Landmark Isomap. As mentioned above, computing the similarity matrix for neural voxel puzzling has a complexity of $\mathcal{O}(n \cdot \alpha_{in} \cdot \alpha_{out})$. For chemical puzzling, computing the similarity matrix has a complexity of $\mathcal{O}(n \cdot \beta^2)$, where β is the largest number of connections a pioneer cell has (how many different cell colonies that pioneer cell's colony has conjugated with). Additionally, for chemical puzzling, using interpolation to estimate the chemical concentration can be a very time-intensive step depending on the resolution desired.

Supporting Information

S1 Fig. Chemical Puzzling with Low Conjugation Efficiency. We do a simulation of chemical puzzling, as in Fig 7B, except now with varying conjugation efficiencies. **(A)** The original cell locations **(left)**, and initial chemical concentration **(right)**. **(B-E)** On the **left**, the reconstructed cells, and on the **right**, the reconstructed chemical concentration, using **(B)** 100% conjugation probability, **(C)** 40% conjugation probability, **(D)**, 30% conjugation probability, and **(E)** 20% conjugation probability. The reconstructed chemical concentrations assumed the pioneer cells had 50 base pairs to encode the concentration. The black area on the outside is a border, not a chemical concentration.
(TIF)

S2 Fig. Reconstruction of Non-cubes. We do voxel puzzling as in Fig 3A, 3B, with 8000 5 μm voxels. However, now the overall shape is a rectangular prism (height is half of the length and

width) rather than a cube. **(A)** The original voxel locations. **(B)** Reconstruction using SDM. **(C)** Reconstruction using ULI.
(EPS)

S1 Text. Chemical Puzzling Conjugation Efficiency. A discussion about our assumptions regarding conjugation efficiency in our Chemical Puzzling simulations, and results about what happens when those assumptions are relaxed.
(PDF)

S2 Text. Simulation Limitations. Further discussion on the limitations of our simulations.
(PDF)

Acknowledgments

We would like to thank Ted Cybulski and Mohammad Gheshlaghi Azar for very helpful comments on previous versions of the manuscript. We would like to thank Josh Vogelstein for helpful initial discussions.

Author Contributions

Conceived and designed the experiments: JG BZ GC KK. Performed the experiments: JG BZ. Analyzed the data: JG BZ KK. Wrote the paper: JG BZ GC KK.

References

1. Lee JH, Daugharthy ER, Scheiman J, Kalhor R, Yang JL, Ferrante TC, et al. Highly multiplexed subcellular RNA sequencing in situ. Science. 2014; 343(6177):1360–1363. doi: 10.1126/science.1250212 PMID: 24578530
2. Zador AM, Dubnau J, Oyibo HK, Zhan H, Cao G, Peikon ID. Sequencing the connectome. PLoS Biol. 2012; 10(10):e1001411. doi: 10.1371/journal.pbio.1001411 PMID: 23109909
3. Kording KP. Of toasters and molecular ticker tapes. PLoS Comput Biol. 2011; 7(12):e1002291. doi: 10.1371/journal.pcbi.1002291 PMID: 22219716
4. Farzadfard F, Lu TK. Genomically encoded analog memory with precise in vivo DNA writing in living cell populations. Science. 2014; 346(6211):1256272. doi: 10.1126/science.1256272 PMID: 25395541
5. Moser F, Horwitz A, Chen J, Lim WA, Voigt CA. Genetic Sensor for Strong Methylating Compounds. ACS synthetic biology. 2013; 2(10):614–624. doi: 10.1021/sb400086p PMID: 24032656
6. Kruskal JB. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika. 1964; 29(1):1–27. doi: 10.1007/BF02289565
7. Tenenbaum JB, De Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. Science. 2000; 290(5500):2319–2323. doi: 10.1126/science.290.5500.2319 PMID: 11125149
8. Coifman RR, Lafon S. Diffusion maps. Applied and computational harmonic analysis. 2006; 21(1):5–30. doi: 10.1016/j.acha.2006.04.006
9. Carr PA, Church GM. Genome engineering. Nat Biotechnol. 2009; 27(12):1151–62. doi: 10.1038/nbt.1590 PMID: 20010598
10. Marblestone AH, Daugharthy ER, Kalhor R, Peikon ID, Kebschull JM, Shipman SL, et al. Rosetta Brains: A Strategy for Molecularly-Annotated Connectomics. arXiv preprint arXiv:14045103. 2014;.
11. Peikon ID, Gizatullina DI, Zador AM. In vivo generation of DNA sequence diversity for cellular barcoding. Nucleic acids research. 2014; 42(16):e127–e127. doi: 10.1093/nar/gku604 PMID: 25013177
12. Gerlach C, Rohr JC, PeriÃ© L, van Rooij N, van Heijst JW, Velds A, et al. Heterogeneous differentiation patterns of individual CD8+ T cells. Science. 2013; 340(6132):635–639. doi: 10.1126/science.1235487 PMID: 23493421
13. De Silva V, Tenenbaum JB. Sparse multidimensional scaling using landmark points. Technical report, Stanford University; 2004.
14. Silva VD, Tenenbaum JB. Global versus local methods in nonlinear dimensionality reduction. In: Advances in neural information processing systems; 2002. p. 705–712.

15. Perge JA, Niven JE, Mugnaini E, Balasubramanian V, Sterling P. Why do axons differ in caliber?. The Journal of Neuroscience. 2012; 32(2):626–638. doi: 10.1523/JNEUROSCI.4254-11.2012 PMID: 22238098

16. Marblestone AH, Daugharthy ER, Kalhor R, Peikon I, Kebschull J, Shipman SL, et al. Conneconomics: The Economics of Large-Scale Neural Connectomics. bioRxiv. 2013;.

17. Hellwig B. A quantitative analysis of the local connectivity between pyramidal neurons in layers 2/3 of the rat visual cortex. Biological cybernetics. 2000; 82(2):111–121. doi: 10.1007/PL00007964 PMID: 10664098

18. Chen BL, Hall DH, Chklovskii DB. Wiring optimization can relate neuronal structure and function. Proceedings of the National Academy of Sciences of the United States of America. 2006; 103(12):4723–4728. doi: 10.1073/pnas.0506806103 PMID: 16537428

19. Zamft BM, Marblestone AH, Kording K, Schmidt D, Martin-Alarcon D, Tyo K, et al. Measuring Cation Dependent DNA Polymerase Fidelity Landscapes by Deep Sequencing. PLoS One. 2012; 7(8): e43876. doi: 10.1371/journal.pone.0043876 PMID: 22928047

20. Glaser JI, Zamft BM, Marblestone AH, Moffitt JR, Tyo K, Boyden ES, et al. Statistical Analysis of Molecular Signal Recording. PLoS Comput Biol. 2013; 9(7):e1003145. doi: 10.1371/journal.pcbi.1003145 PMID: 23874187

21. Friedland AE, Lu TK, Wang X, Shi D, Church G, Collins JJ. Synthetic gene networks that count. Science. 2009; 324(5931):1199–1202. doi: 10.1126/science.1172005 PMID: 19478183

22. Church GM, Shendure J. Nucleic acid memory device [Generic]. US Patent 20,030,228,611; 2003.

23. Willetts N, Skurray R. The conjugation system of F-like plasmids. Annual review of genetics. 1980; 14 (1):41–76. doi: 10.1146/annurev.ge.14.120180.000353 PMID: 6111288

24. Nakano M, Komatsu J, Matsuura Si, Takashima K, Katsura S, Mizuno A. Single-molecule PCR using water-in-oil emulsion. Journal of biotechnology. 2003; 102(2):117–124. doi: 10.1016/S0168-1656(03)00023-3 PMID: 12697388

25. Grossman N, Ron E, Woldringh C. Changes in cell dimensions during amino acid starvation of Escherichia coli. Journal of bacteriology. 1982; 152(1):35–41. PMID: 6749809

26. Ponciano JM, De Gelder L, Top EM, Joyce P. The population biology of bacterial plasmids: a hidden Markov model approach. Genetics. 2007; 176(2):957–968. doi: 10.1534/genetics.106.061937 PMID: 17151258

27. Liley D, Wright J. Intracortical connectivity of pyramidal and stellate cells: estimates of synaptic densities and coupling symmetry. Network: Computation in Neural Systems. 1994; 5(2):175–189. doi: 10.1088/0954-898X/5/2/004

28. Usoskin D, Furlan A, Islam S, Abdo H, Lönnerberg P, Lou D, et al. Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing. Nature neuroscience. 2014. doi: 10.1038/nn.3881 PMID: 25420068

29. Golub GH, Van Loan CF. Matrix computations. vol. 3. JHU Press; 2012.

30. Fredman ML, Tarjan RE. Fibonacci heaps and their uses in improved network optimization algorithms. Journal of the ACM (JACM). 1987; 34(3):596–615. doi: 10.1145/28869.28874

31. Kaplan H, Sharir M, Verbin E. Colored intersection searching via sparse rectangular matrix multiplication. In: Proceedings of the twenty-second annual symposium on Computational geometry. ACM; 2006. p. 52–60.

32. Yuster R, Zwick U. Fast sparse matrix multiplication. ACM Transactions on Algorithms (TALG). 2005; 1 (1):2–13. doi: 10.1145/1077464.1077466

33. van der Maaten LJ, Postma EO, van den Herik HJ. Dimensionality reduction: A comparative review. Journal of Machine Learning Research. 2009; 10(1–41):66–71.