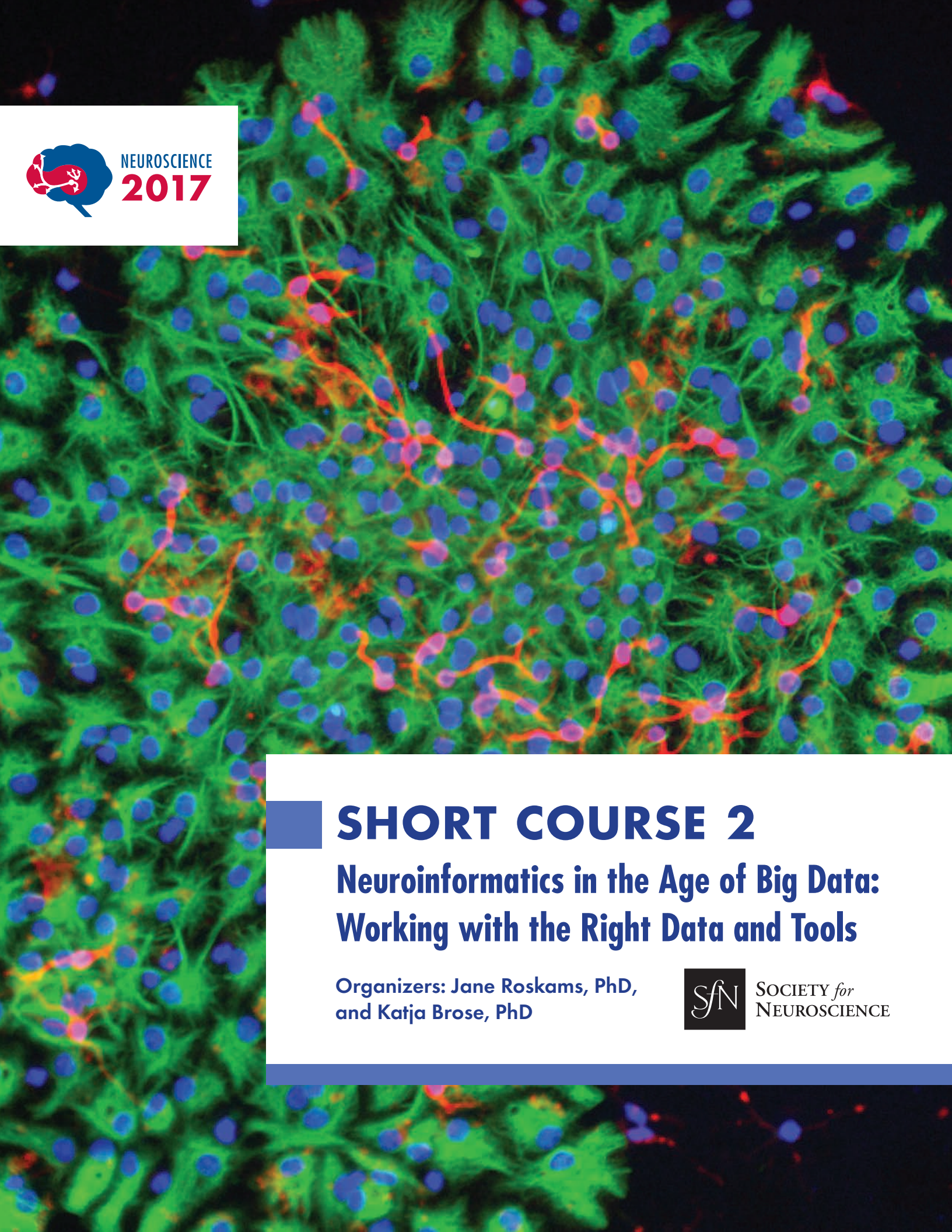

NEUROSCIENCE 2017

# SHORT COURSE 2

## Neuroinformatics in the Age of Big Data: Working with the Right Data and Tools

Organizers: Jane Roskams, PhD, and Katja Brose, PhD

SfN Society for Neuroscience

# Short Course 2

## Neuroinformatics in the Age of Big Data: Working with the Right Data and Tools

Organized by Jane Roskams, PhD, and Katja Brose, PhD

SfN Society for Neuroscience

Please cite articles using the model:
[AUTHOR'S LAST NAME, AUTHOR'S FIRST & MIDDLE INITIALS] (2017)
[CHAPTER TITLE] In: Neuroinformatics in the Age of Big Data: Working with the Right Data and Tools.
(Roskams J, Brose K, eds) pp. [xx-xx]. Washington, DC: Society for Neuroscience.

NEUROSCIENCE 2017

# SHORT COURSE 2

## Neuroinformatics in the Age of Big Data: Working with the Right Data and Tools

Organized by Jane Roskams, PhD and Katja Brose, PhD
Friday, November 10, 2017
8 a.m.–6 p.m.
Location: Washington, DC Convention Center • Room: Ballroom B

| TIME | TOPIC | SPEAKER |
|---|---|---|
| 7:30 – 8 a.m. | CHECK-IN | |
| 8 – 8:10 a.m. | Opening Remarks | Jane Roskams, PhD • University of British Columbia |
| 8:10 – 8:55 a.m. | Probing Transcriptomic Diversity of Human Cortical Cell Types | Trygve Bakken, MD, PhD • Allen Institute for Brain Science |
| 8:55 – 9:40 a.m. | Harnessing Publicly Accessible Transcriptomics Data in Neuroscience | Paul Pavlidis, PhD • University of British Columbia |
| 9:40 – 10 a.m. | MORNING BREAK | |
| 10 – 10:45 a.m. | Connectome Coding | Joshua Vogelstein, PhD • Johns Hopkins University |
| 10:45 a.m. – 11:30 a.m. | Neuroinformatics and Simulation Neuroscience | Sean Hill, PhD • École Polytechnique Fédérale de Lausanne |
| 11:30 a.m. – 12:15 p.m. | Big Neurophysiology | Kenneth Harris, PhD • University College, London |
| 12:15 – 1:15 p.m. | LUNCH — ROOM 145 AB | |
| 1:15 – 2 p.m. | Mining Long-Range Neuronal Projections with the Open Data and Tools from the Allen Mouse Brain Connectivity Atlas. | Jennifer Whitesell, PhD • Allen Institute for Brain Science |
| 2 – 2:45 p.m. | Global Neuroscience Data-Sharing: Current Issues and Solutions | Jean-Baptiste Poline, PhD • McGill University* |
| 2:45 – 3:30 p.m. | The HBP Human Brain Atlas — Sharing Data Across Different Levels of Brain Organisation | Katrin Amunts, MD, PhD • University of Düsseldorf |
| 3:30 – 3:45 p.m. | AFTERNOON BREAK | |

**AFTERNOON BREAKOUT SESSIONS • PARTICIPANTS SELECT DISCUSSION GROUPS AT 3:45 AND 5:00 P.M.**

| TIME | BREAKOUT SESSIONS | SPEAKERS | ROOM |
|---|---|---|---|
| 3:45 – 4:45 p.m. | Analysis and Applications of Single and Purified Cell Transcriptomes | Trygve Bakken & Paul Pavlidis | 150A |
| | Data-Driven Neurophysiology and Neuronal Modelling | Joshua Vogelstein, Sean Hill, & Kenneth Harris | 144BC |
| | HBP — Canadian Brain Atlas Tools | Katrin Amunts & Alan Evans Lab Group | Ballroom B |
| 4:45 – 5 p.m. | AFTERNOON BREAK | | |
| 5 – 6 p.m. | Repeat sessions above. Select a second breakout group. | | |

*Content created in collaboration with Alan Evans, PhD.

# Table of Contents

# Introduction

Neuroscience is a fertile landscape of discovery surrounded by growing mountains of data that are often hard to navigate. Much of these data are open and accessible, but it is becoming hard to know where to find data we can trust and use, what they represent, and how to use them to better understand and accelerate our research. This course has been designed to put more usable data in the hands of neuroscientists who are not expert neuroinformaticians. Here we bring together leaders in the neuroscience/informatics field to guide attendees (armed with a laptop) through a hands-on course highlighting some of the most broadly accessible open datasets and to guide their independent scientific voyage of discovery. We do not expect you to be experts in informatics or data science—though we hope that some of you will feel that way by the time you complete the workshop.

The objectives for this short course are threefold. First, we will walk you through online portals to find data that you might be able to use; explain their source, generation, and organization; and outline what they may be able to tell you. Second, we will demonstrate some of the established and more recent open-source tools that have been generated to help interpret different subsets of neuroscience data. Finally, we will provide sessions where you can ask some of your own questions (or ones we will provide for you) hands on and be guided by our experts and TAs.

The hands-on sessions will be based on material covered in the lectures, and you will get the most out of them if you have some basic programming skills, and at least a couple of areas of interest. The focus is on participants learning how to discover more about their areas of interest using openly available data, aided by leading experts from around the world.

The course will cover a broad range of topics, including single-cell transcriptomics, large-scale gene expression analysis, physiology of identifiable neurons (electrophysiology and optogenetics), mouse and human connectomics, human and mouse circuit function and modeling, and interpretation and analysis of human imaging. It is the course I wish I had been able to take when I was a grad student or postdoc! The course should instill a solid understanding of the basic techniques you will need to open a browser and generate new insights and hypotheses to further any research question you might want to ask.

# Defining Cell Types by Single-Cell RNA Sequencing

Trygve E. Bakken, MD, PhD

Allen Institute for Brain Science
Seattle, Washington

NOTES

## Introduction

Cells are a fundamental functional unit of organisms and come in many varieties. One major goal in biology is to characterize the cell types found in different tissues and species. For example, the human body is composed of approximately 30 trillion cells, of which less than 1% are neural cells (Sender et al., 2016), and neurons show striking morphological, electrophysiological, and molecular diversity. Many of these characteristics have direct functional consequences for a neuron, including its connectivity and responsiveness to different neurotransmitters.

Neurons can be grouped into types based on shared features, and these cell types simplify the description of neural circuits and facilitate probing circuit function. While many neuronal types have already been described, a comprehensive survey will require high-throughput assays. Recent technological development of RNA sequencing (RNA-seq) of individual cells has enabled profiling gene expression in thousands of neurons, and this has led to a refined census of neuron types in mouse cortex (Tasic et al., 2016) and a coarser census in human cortex (Lake et al., 2016). These cell types have selective expression of one or more genes, and molecular tools can be created to target these genes to further characterize the function of these neurons.

In this chapter, we will walk through the analysis steps required to define transcriptomic cell types from single-cell RNA-seq data. We will consider cell sampling strategies and the expected power to detect cell types based on their frequency and distinctiveness. Then we will examine the key steps of clustering: expression normalization, variable gene selection, dimensionality reduction, and clustering algorithms.

## Cell Sampling

Cell types can be difficult to identify owing to their low frequency or similarity to other cell types (Fig. 1*a*). Monte Carlo simulations can be used to estimate the number of cells that must be sampled to be 95% confident of capturing at least *N* cells with frequency *X* in the population (Fig. 1*b*). The number of cells required to discriminate two cell types varies as a function of the number of genes that

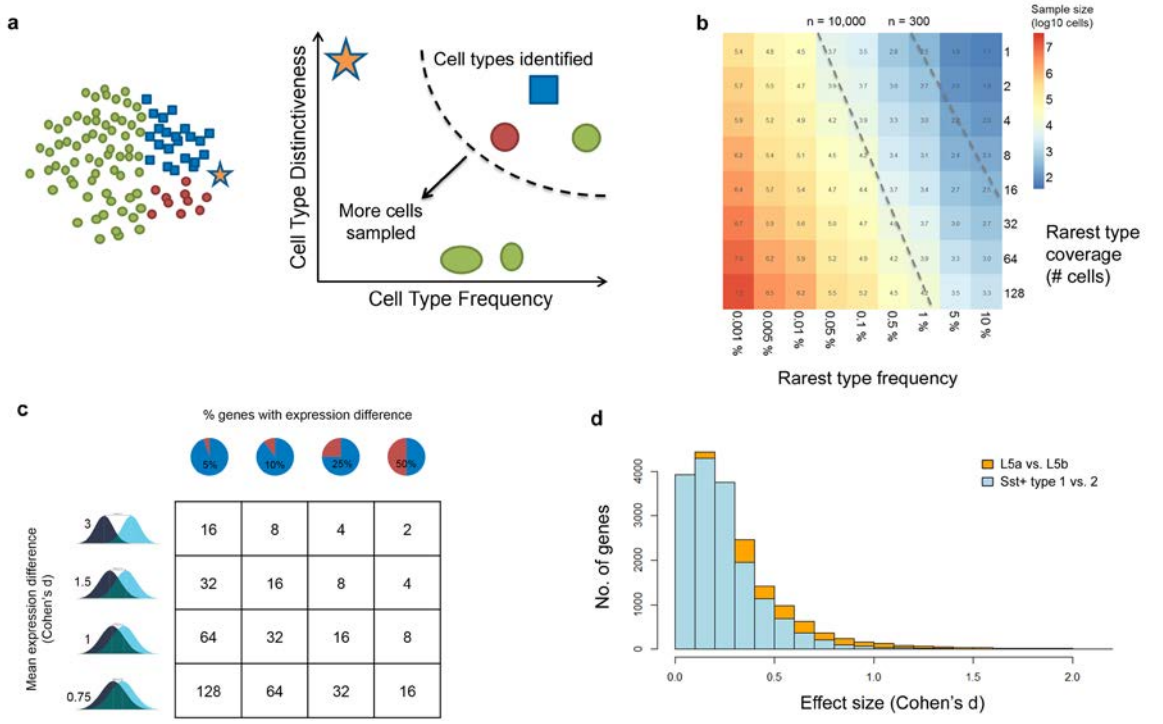


**Figure 1.** A quantitative sampling strategy to target cell types. ***a,*** Example of a population of cells that includes cell types with different frequencies and distinctiveness. Greater sampling depth captures more cell types. ***b***, Cell sample sizes required to capture 95% confidence for rare cell types with a given depth. Dashed lines show the rarest cell types captured at various depths when sampling 300 versus 10,000 cells. ***c***, Simulation demonstrating that fewer cells are required to differentiate a pair of cell types with more differentially expressed genes or larger expression differences (Cohen's *d*). ***d***, Two pairs of excitatory (layer Va vs Vb) neuron types and inhibitory (two Sst+ subtypes) neuron types in mouse primary visual cortex (Tasic et al., 2016) are distinct based on small expression differences of thousands of genes and large differences of a few genes.

NOTES

are differentially expressed between those types and the magnitude of the expression differences (Fig. 1*c*). For example, approximately eight cells are needed to differentiate layer Va from layer Vb mouse cortical neurons (Tasic et al., 2016), and this requirement is the result of small expression differences of thousands of genes and larger differences of a few genes (Fig. 1*d*).

One can improve detection of low-frequency cell types by enriching for cells, for example, by labeling and sorting cells using known molecular markers or by dissecting a region enriched with cells of that type. Some cells will rarely be captured if they are vulnerable to tissue dissociation, such as adult human neurons, and profiling single nuclei rather than whole cells has provided a less biased survey of human cortical neuron types (Lake et al., 2016). Finally, one can simply sample more cells with high-throughput (e.g., droplet-based) RNA-seq methods (Macosko et al., 2015) that cost less per cell but detect fewer genes.

## Gene Detection

One can improve detection of similar cell types by increasing the sensitivity and reducing the noise of gene expression measurements. Single-cell RNA amplification methods vary in their rates of gene detection, dropouts, noise, and cost (Ziegenhain et al., 2017), and the appropriate method will vary by experiment. For example, methods that amplify the full length of the transcript increase sensitivity to low-expression transcripts that may be important markers of cell types. Methods that quantify absolute transcript levels with unique molecular identifiers (UMIs) reduce amplification noise that may obscure subtle expression differences. Some droplet-based RNA-seq methods have significantly lower gene detection and higher dropout rates (Ziegenhain et al., 2017), and this should be considered when balancing the number of cells profiled versus the resolution to discriminate among closely related cell types.

Gene expression dropouts in single cells result from missed detection and biological variability, such as transcriptional bursting, and can obscure relationships between cells. The number of dropouts varies across cells because of mRNA quality and across genes due to expression levels, and these effects can be modeled and accounted for as a weighting factor when calculating differential expression and similarities among cells (Kharchenko et al., 2014). Another approach to mitigate the effect of dropouts is to impute expression values by pooling information across many similar cells and correlated sets of genes (van Dijk et al., 2017).

## Cell Clustering

After performing RNA-seq of single cells or nuclei, aligning reads to a reference transcriptome, and quantifying gene expression, one goal is to group cells based on shared transcriptomic signatures. Many clustering approaches share four steps:

1. Expression normalization
2. Variable gene selection
3. Dimensionality reduction
4. Clustering

The following presents examples of how to approach each of these steps.

### Expression normalization

Ideally, one could measure the absolute number of every transcript in each cell. If UMIs are not available, then the relative number of transcripts must be inferred from the number of reads that map to each gene. If full-length transcripts are sequenced, then longer genes and more highly expressed genes will have more reads, so it is common to normalize read counts by transcript length. However, normalization requires an accurate reference transcriptome that may not be available. For example, many nuclear transcripts include intronic sequence (Lake et al., 2016), and gene lengths will be greatly underestimated by a reference that includes only spliced transcripts. Therefore, for single-nuclei RNA-seq data, it is important to construct a new reference transcriptome with better estimates of gene length or else normalize only by total read depth, for example, counts per million reads. Various methods have been developed to remove unwanted technical variation from single-cell expression data. These include using ERCC spike-in control RNAs (developed by the External RNA Controls Consortium and manufactured by ThermoFisher Scientific) with known levels (Vallejos et al., 2017). It is also common to log-transform and *z*-score the normalized expression values to reduce the influence of outliers and high-expression genes on clustering.

### Variable gene selection

The goal of variable gene selection is to choose genes with variable expression due to real biological effects and not technical noise. Gene dropouts inflate expression variance and should be accounted for by incorporating an appropriate noise model, as described earlier. Technical noise increases with average expression, and this relationship can be estimated either using ERCCs (Brennecke et al., 2013) or directly from the gene expression data (Fan et al., 2016), enabling identification of genes with

NOTES

significant biological variation in expression. Note that genes with expression that is restricted to rare cell types may have relatively low variability across all cells and may be missed in this step. One way to mitigate this omission is to iterate clustering on clusters identified in the first round.

## Dimensionality reduction

Groups of genes that act in shared biological pathways are often coordinately regulated and have correlated expression. Therefore, one can represent the expression of thousands of genes in a much lower-dimensional space that captures most of the variation in expression and reduce noise by pooling information across correlated genes. In other words, dimensionality reduction aims to learn the low-dimensional manifold in which cells reside within gene expression space. A classical technique is principal component analysis, which defines an orthogonal set of principal components (PCs) that are a linear combination of genes ordered by the amount of variance explained (Fig. 2*a*). One can then select PCs that explain more variance than is expected by chance. Weighted gene coexpression network analysis (WGCNA) is another intuitive dimensionality reduction technique that groups genes into modules that share correlated neighbors (Langfelder and Horvath, 2008). Modules and

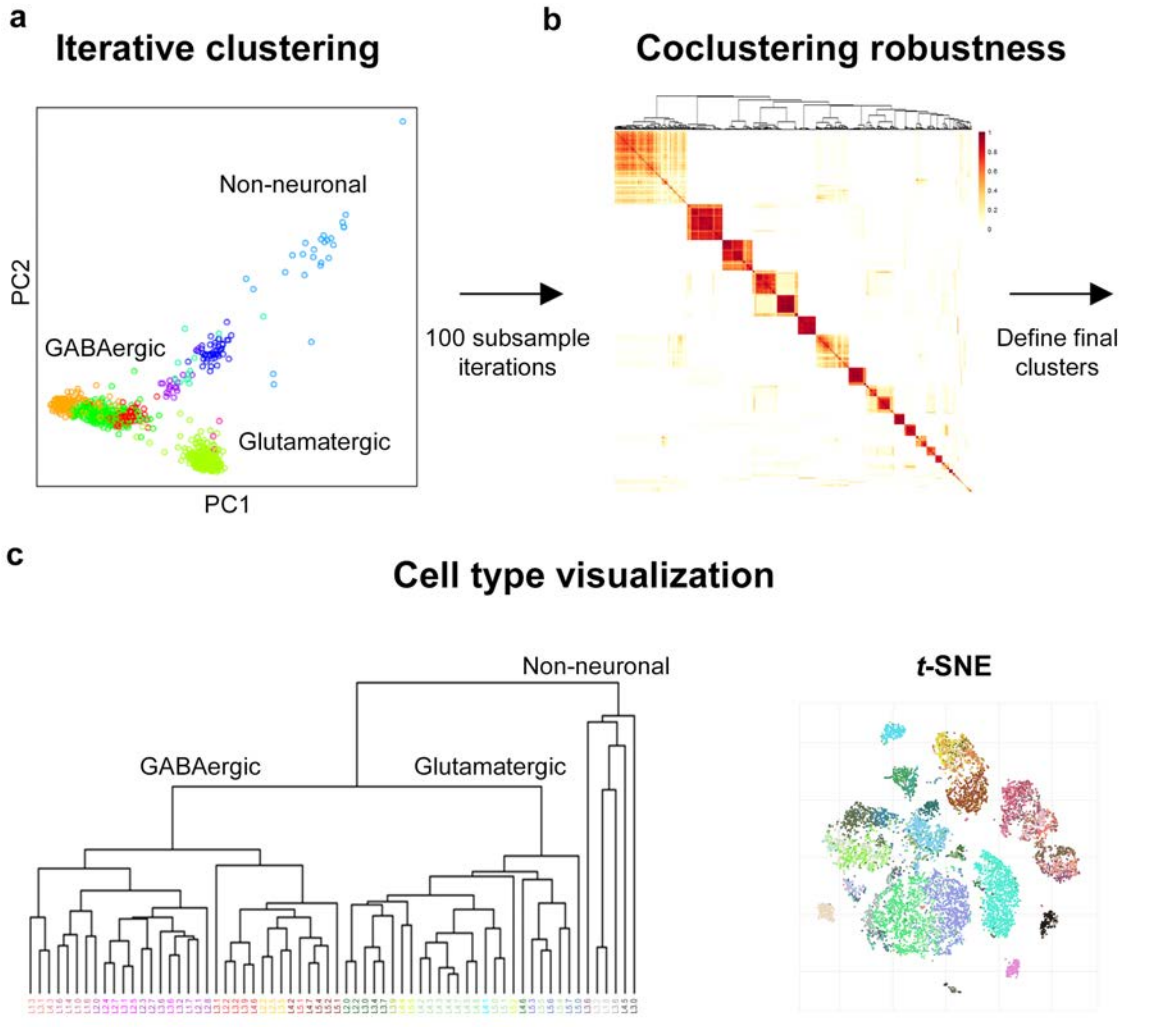


**Figure 2.** Example of steps to cluster and visualize cell types. ***a***, Cells were projected onto the first two PCs of a PCA of significantly variable genes. Cells were clustered based on their location using all significant PCs and colored based on cluster membership. ***b***, Coclustering matrix showing the proportion of 100 clustering iterations (each using a random 80% subsample of cells) that each pair of cells was placed in the same cluster. Final clusters can be defined as sets of cells that consistently cocluster, as represented by the red boxes along the diagonal. ***c***, Left, Cell-type dendrogram based on hierarchical clustering of median expression of marker genes. Right, Visualization of transcriptional heterogeneity of cells within clusters (corresponding to colors) using *t*-SNE (van der Maaten and Hinton, 2008).

NOTES

PCs often consist of genes with shared biological functions and can be annotated with gene ontology enrichment analysis. This can help identify technical sources of expression variance, and these dimensions can be removed from further analysis. Finally, it can be helpful to apply a second round of nonlinear dimensionality reduction to better separate groups of similar cells using *t*-distributed stochastic neighbor embedding (*t*-SNE) (van der Maaten and Hinton, 2008), which preserves local but not global distance relationships.

## Clustering

Unsupervised clustering techniques aim to group cells that are more like one another than they are to other cells. Density clustering, such as DBSCAN (density-based clustering of applications with noise) (Ester et al., 1996), identifies cells that are neighbors in feature space and is most effective when closely related cells are tightly packed and well separated from other cells. Therefore, it is helpful to first apply *t*-SNE to reduce features to two dimensions before density clustering. PhenoGraph (Levine et al., 2015) takes a related approach and represents cells as a nearest-neighbor graph that transforms the problem of finding densely packed cells in high-dimensional expression space to a problem of finding sets of highly interconnected cells. Because efficient community detection algorithms exist for large networks, this technique is scalable to hundreds of thousands of cells. BackSPIN (Zeisel et al., 2015) also avoids dimensionality reduction by performing hierarchical biclustering and simultaneously identifies sets of correlated genes and cells with similar expression patterns.

One can attempt to further split clusters by repeating the previous steps (including variable gene selection) independently on each cluster, and this may identify more subtle or rare cell types that were missed on the first round. One must define stopping criteria for clustering, such as a minimum cluster size and lack of differentially expressed genes. Cluster robustness can be quantified by repeating iterative clustering on subsamples of cells and counting how often each pair of cells clusters together (Fig. 2*a*). This coclustering matrix can be used to define a final set of clusters by identifying groups of cells that consistently cocluster (Fig. 2*b*).

Cluster relatedness can be quantified by coclustering between clusters and by similarity of gene expression. Two visualizations can aid in the biological interpretation of cell types. A dendrogram tree of cell types can be constructed by applying hierarchical clustering to a correlation-based distance matrix of median expression across clusters. This tree can be annotated with marker genes that label branches of similar cell types (e.g., broad classes of GABAergic interneurons) and compared with similarities in developmental lineage, electrophysiology, and morphology among types. A *t*-SNE plot of all cells can be used to visualize cluster properties, such as the expression of specific genes or technical covariates, and this can aid cluster curation (Fig. 2*c*).

## Example clustering pipeline

1. Iteratively cluster cells
   - 1.1. Select significantly variable genes among cells.
   - 1.2. Reduce dimensionality of gene expression.
   - 1.3. Cluster cells based on proximity in reduced space.
   - 1.4. For each cluster, repeat steps 1.1–1.3.
   - 1.5. Stop when there are no significantly variable genes, PCs, or clusters.

2. Assess cluster robustness
   - 2.1. Subsample 80% of cells.
   - 2.2. Perform iterative clustering on each subsample (steps 1.1–1.5).
   - 2.3. Calculate proportion of clustering iterations that each pair of cells is coclustered.

3. Define and visualize clusters
   - 3.1. Cluster coclustering matrix to identify cells that consistently cluster together and not with other cells.
   - 3.2. Exclude "outlier" clusters based on significantly lower quality control metrics.
   - 3.3. Merge clusters that do not meet criteria for being distinct "cell types" (e.g., those that lack distinct marker genes).
   - 3.4. Construct a cluster dendrogram based on the median expression of marker genes.
   - 3.5. Compare transcriptional heterogeneity of clusters using dimensionality reduction (e.g., *t*-SNE) (van der Maaten and Hinton, 2008).

# Conclusion

High-throughput RNA-seq of single cells and nuclei will reveal a broad diversity of cell types across tissues, species, development, and disease. Fully characterizing these transcriptomic cell types, including their shape, functional properties, local environment, and connections, will require the advancement of new techniques. For example, multiplex fluorescence *in situ* hybridization methods are rapidly improving that can map distributions of cell types *in situ* based on marker gene expression and localize transcripts to different subcellular compartments. Finally, new gene editing methods,

such as those based on clustered regularly interspaced short palindromic repeats (CRISPR), will provide a means to develop mechanistic models of gene function that should shed light on cell function in health and disease.

## Resources

- RNA-seq datasets
  - Allen Brain Atlas cell types database: http://celltypes.brain-map.org/rnaseq
  - Single Cell Portal Beta (The Broad Institute): https://portals.broadinstitute.org/single_cell
  - Single Cell Analysis Program—Transcriptome Project (SCAP-T): https://www.scap-t.org/content/data-portal
  - National Center for Biotechnology Information GEO DataSets: https://www.ncbi.nlm.nih.gov/gds
- Analysis tools
  - Cell sampling (Satija Lab): http://satijalab.org/howmanycells
  - BASiCS (Vallejos C): https://github.com/catavallejos/BASiCS
  - RUVSeq: http://bioconductor.org/packages/release/bioc/html/RUVSeq.html
  - DESeq2: https://bioconductor.org/packages/release/bioc/html/DESeq2.html
  - scde http://hms-dbmi.github.io/scde/
  - WGCNA: an R package for weighted correlation network analysis (Langfelder P, Horvath S): https://labs.genetics.ucla.edu/horvath/CoexpressionNetwork/Rpackages/WGCNA/
  - *t*-SNE (van der Maaten LJP): https://lvdmaaten.github.io/tsne/
  - ToppGene Suite GO enrichment: https://toppgene.cchmc.org/enrichment.jsp
- Clustering
  - DBSCAN (Hahsler M, Piekenbrock M, Arya S, Mount D): https://cran.r-project.org/web/packages/dbscan/
  - Pagoda (Harvard Medical School Department of Bioinformatics): https://github.com/hms-dbmi/pagoda2
  - Seurat (Satija Lab): http://satijalab.org/seurat/
  - BackSPIN (Linnarsson Lab): R toolkit for single cell genomics: https://github.com/linnarsson-lab/BackSPIN
  - PhenoGraph (Dana Pe'er Lab of Computational Systems Biology): https://www.c2b2.columbia.edu/danapeerlab/html/phenograph.html
  - SIMLR (Batzoglou Lab): https://github.com/BatzoglouLabSU/SIMLR

## References


Brennecke P, Anders S, Kim JK, Kołodziejczyk AA, Zhang X, Proserpio V, Baying B, Benes V, Teichmann SA, Marioni JC, Heisler MG (2013) Accounting for technical noise in single-cell RNA-seq experiments. Nat Methods 10:1093–1095.

Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), pp 226–231.

Fan J, Salathia N, Liu R, Kaeser GE, Yung YC, Herman JL, Kaper F, Fan J-B, Zhang K, Chun J, Kharchenko PV (2016) Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. Nat Methods 13:241–244.

Kharchenko P V, Silberstein L, Scadden DT (2014) Bayesian approach to single-cell differential expression analysis. Nat Methods 11:740–742.

Lake BB, Ai R, Kaeser GE, Salathia NS, Yung YC, Liu R, Wildberg A, Gao D, Fung HL, Chen S, Vijayaraghavan R, Wong J, Chen A, Sheng X, Kaper F, Shen R, Ronaghi M, Fan JB, Wang W, Chun J, et al. (2016) Neuronal subtypes and diversity revealed by single-nucleus RNA sequencing of the human brain. Science 352:1586–1590.

Langfelder P, Horvath S (2008) WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics 9:559.

Levine JH, Simonds EF, Bendall SC, Davis KL, Amir el-AD, Tadmor MD, Litvin O, Fienberg HG, Jager A, Zunder ER, Finck R, Gedman AL, Radtke I, Downing JR, Pe'er D, Nolan GP (2015) Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. Cell 162:184–197.

Macosko EZ, Basu A, Satija R, Nemesh J, Shekhar K, Goldman M, Tirosh I, Bialas AR, Kamitaki N, Martersteck EM, Trombetta JJ, Weitz DA, Sanes JR, Shalek AK, Regev A, McCarroll SA (2015) Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell 161:1202–1214.

Sender R, Fuchs S, Milo R (2016) Revised estimates for the number of human and bacteria cells in the body. PLoS Biol 14:e1002533.


NOTES

Tasic B, Menon V, Nguyen TN, Kim TK, Jarsky T, Yao Z, Levi B, Gray LT, Sorensen SA, Dolbeare T, Bertagnolli D, Goldy J, Shapovalova N, Parry S, Lee C, Smith K, Bernard A, Madisen L, Sunkin SM, Hawrylycz M, et al. (2016) Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. Nat Neurosci 19:335–346.

Vallejos CA, Risso D, Scialdone A, Dudoit S, Marioni JC (2017) Normalizing single-cell RNA sequencing data: challenges and opportunities. Nat Methods 14:565–571.

van der Maaten LJP, Hinton GE (2008) Visualizing high-dimensional data using *t*-SNE. J Mach Learn Res 9:2579–2605.

van Dijk D, Nainys J, Sharma R, Kathail P, Carr AJ, Moon KR, Mazutis L, Wolf G, Krishnaswamy S, Pe'er D (2017) MAGIC: A diffusion-based imputation method reveals gene–gene interactions in single-cell RNA-sequencing data. bioRxiv:111591.

Zeisel A, Muñoz-Manchado AB, Codeluppi S, Lönnerberg P, La Manno G, Juréus A, Marques S, Munguba H, He L, Betsholtz C, Rolny C, Castelo-Branco G, Hjerling-Leffler J, Linnarsson S (2015) Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. Science 347:1138–1142.

Ziegenhain C, Vieth B, Parekh S, Reinius B, Guillaumet-Adkins A, Smets M, Leonhardt H, Heyn H, Hellmann I, Enard W (2017) Comparative analysis of single-cell RNA sequencing methods. Mol Cell 65:631–643.

NOTES

# Interpreting Cell-Type-Specific Changes in Bulk Tissue Transcriptomics Data

Ogan Mancarci, Lilah Toker, PhD,
Shreejoy Tripathy, PhD, and Paul Pavlidis, PhD

Department of Psychiatry and Michael Smith Laboratories
University of British Columbia
Vancouver, Canada

NOTES

## Introduction

RNA expression profiling is a powerful means of assaying the state of a biological sample but has many interpretational difficulties. One challenge is "cellular composition," which refers to sample-to-sample variability in the types and proportions of cells present. Dissected "bulk" tissue is the source of the vast majority of RNA used in transcriptome studies, in which RNA from multiple cell types is intermingled. While it has long been a concern (especially in studies of the nervous system) that analysis of complex bulk tissue might result in the dilution of effects occurring in a subset of cells, it is also known that sample-to-sample differences in cellular composition occur. These differences can be caused by the random statistical effects of sampling from small pieces of tissue, but could also reflect biological differences among individuals. Further, while some conditions such as neurodegenerative diseases are well known to result in changes in cellular composition, there is a growing recognition that such effects should be considered in all analyses of brain tissue transcriptomes. At one extreme, a change in measured gene expression could be entirely the result of changes in cellular composition without any change in gene regulation within the cells (Fig. 1).

It is clear from these examples that taking cellular composition into account is important for interpreting expression analyses of bulk tissue data. That is, we should question whether a measured change in a gene transcript level results from a regulatory event within cells, or whether it represents an alteration in the number of cells expressing the gene, or some combination of such effects. The difficulty is that measuring cellular proportions directly (e.g., by stereology) is generally incompatible with destructive bulk tissue RNA sampling. Although there is hope that single-cell methodologies can resolve this problem, bulk tissue sampling remains the norm.

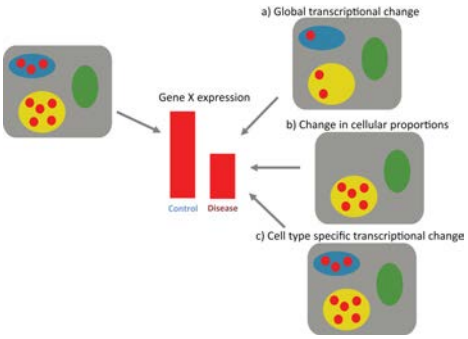


**Figure 1.** Observed expression levels are the combination of multiple sources of variability. In this toy example, the expression of gene *X* as measured from bulk tissue is lower in a sample from the Disease group than in a sample from the Control group. However, the bulk tissue is composed of a mixture of cells (yellow, green, and blue ovals), only a subset of which expresses gene *X* (red circles). Thus, the observed change can be induced by ***a***, a regulatory event affecting the gene expression level in all cells; ***b***, a decrease in the number of cells expressing the gene; or ***c***, a regulatory event affecting expression in only a subset of cells.

With this challenge in mind, methods for estimating cellular proportions based entirely on transcriptome measurements have been developed. A variety of approaches are available, but in general, they rely on the use of marker genes that distinguish one cell type from another. Simplistically stated, a change in the expression of marker genes is used as a surrogate for a change in the abundance of the relevant cell type. Although it is impossible to definitively assign such changes in expression to changes in cell-type proportions (direct cell counting remains the gold standard), the benefits of applying these methods outweigh the caveats. Thus, cell-type-specific (or enriched) marker genes have been used to gain cell-type-specific information from brain bulk tissue data, and have generally been interpreted as indicating changes in cell-type proportion (Sibille et al., 2008; Tan et al., 2013; Skene and Grant, 2016).

In this chapter, we demonstrate how cell-type-specific gene expression profiles can assist the interpretation of transcriptomics data derived from bulk tissue samples. The approach we cover is based on using multiple cell-type-specific markers identified using purified cell-type or single-cell transcriptomes, in this case, as captured in the NeuroExpresso.org database. These markers are used together to measure a marker gene profile (MGP) for each cell type, which can then be used either directly as a proxy cell-type proportion measure, or as a covariate in statistical models to "normalize" for cell-type proportion changes, allowing gene regulation effects to be estimated more accurately. Throughout, we highlight potential pitfalls and troubleshooting steps to assist practitioners in being informed users of the MGP approach in their own research.

## Marker Gene Profile Estimation Overview

In this tutorial, we reproduce an analysis presented in Mancarci et al. (2016) to infer cell-type proportion changes in the human midbrain in Parkinson's disease (PD).

MGPs are summarized expression levels of transcripts enriched in specific cell types. Because the simplest explanation for concordant change in the expression level of a large number of genes enriched in a specific cell type is change in the abundance of this cell type,

NOTES

MGPs can be carefully used as surrogates for relative cellular abundance.

Two components are needed to estimate MGP. First, one needs a set of marker genes, defined as genes specifically expressed in, or highly enriched in, a particular cell type in the context of a region or tissue. Second, one needs a transcriptomic dataset from relevant bulk tissue, representing gene expression profiles across a number of samples (Fig. 2).

In the example illustrated in this tutorial, we use the marker genes derived from mouse brain cell types described by Mancarci and colleagues (2017) and gene expression data from substantia nigra samples from healthy subjects and PD patients collected by Lesnick and colleagues (2007).

## Package Installation and Data Download

In this tutorial, we make use of R code and data provided in the markerGeneProfile package available at GitHub: https://github.com/oganm/markerGeneProfile/. This package can be installed from GitHub directly within R using the devtools package (version 1.12.0). This code also makes use of several third-party packages: ggplot2, dplyr, gplots, and viridis. We assume the reader is either familiar with R and these packages or can follow along with the assistance of the built-in R help system. (Note that in the following example, R console output is shown in a contrasting color, but not all output is shown.)

```
install.packages('devtools')
devtools::install_github('oganm/homologene')
devtools::install_github('oganm/ogbox')
devtools::install_github('oganm/
markerGeneProfile')
install.packages('ggplot2')
install.packages('gplots')
install.packages('viridis')
install.packages('dplyr')
library(markerGeneProfile)
```

### Mouse marker genes

For this tutorial, we make use of lists of marker genes derived from gene expression datasets corresponding to specific mouse brain cell types. Specifically, these gene expression datasets were collected from published datasets reflecting purified, pooled brain cell types and single cells. We have made these data accessible within the NeuroExpresso database and resource at http://www.neuroexpresso.org.

After assembling these mouse cell-type-specific data, we computationally identified marker gene sets for each cell type. An individual marker gene set is composed of genes highly enriched in a cell type in the context of a brain region. That is, the expression level of genes in a cell type in the specified region is evaluated in comparison with all cell types available in NeuroExpresso for the same region. This means,

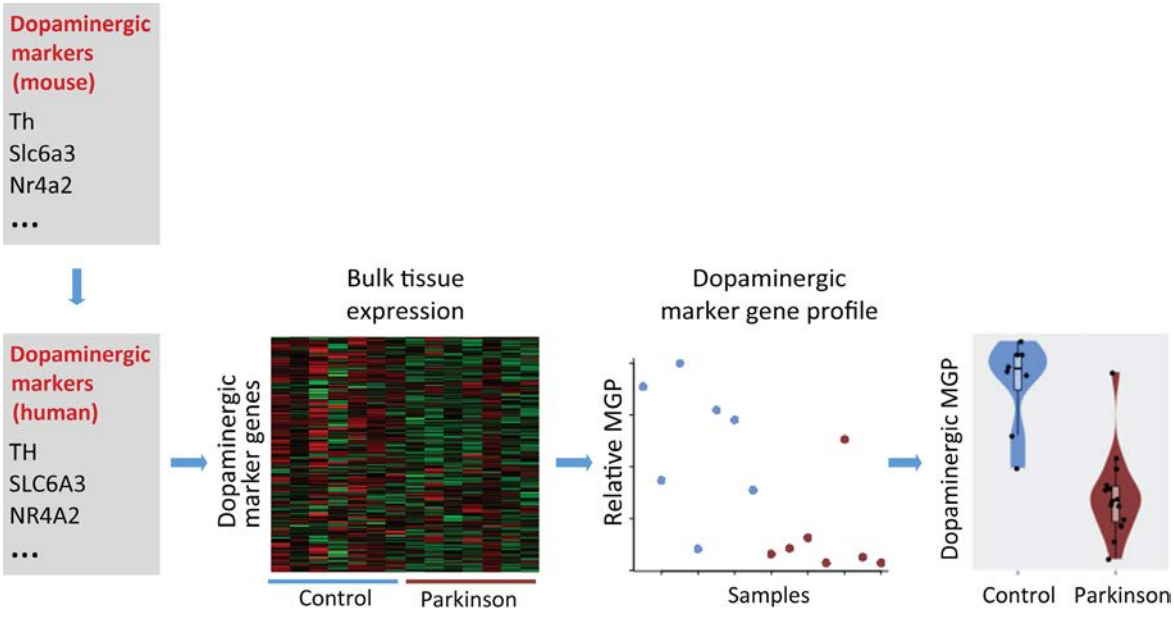


**Figure 2.** Schematic representation of MGP analysis. The input for the analysis comprises mouse marker genes for the cell type of interest, derived from the NeuroExpresso database (http://www.neuroexpresso.org). For the analysis of human bulk tissue, mouse genes are first converted to human orthologues, and then their expression values are extracted from the human bulk tissue data. In the next step, the expression signals of the marker genes are summarized into a single value for each sample based on PCA. The summarized values indicate the MGP across the sample. After obtaining the MGPs, they can be statistically analyzed to obtain the group differences.

NOTES

for example, that the marker genes identified for astrocytes in the cortex can differ from the marker genes identified for the astrocytes in hippocampus. We selected marker genes based on (1) fold of change relative to other cell types in the brain region and (2) a lack of overlap of expression levels in other cell types. A complete description of the methodology used to define marker genes for mouse brain cell types is available in Mancarci et al. (2016).

Within the markerGeneProfile R package, we have made the marker gene lists for each cell type available in the `mouseMarkerGenes` object as a nested list. Each nesting shows the marker genes for each cell type in the context of a specific brain region, e.g., Midbrain (including the substantia nigra), Cortex, or Cerebellum, shown below. "All" indicates selection of marker genes in the context of the whole brain (i.e., compared with all cells available in NeuroExpresso).

```
data(mouseMarkerGenes)
names(mouseMarkerGenes)
## [1] "All" "Amygdala" "BasalForebrain"
## [4] "Brainstem" "Cerebellum" "Cerebrum"
## [7] "Cortex" "Hippocampus" "LocusCoeruleus"
## [10] "Midbrain" "SpinalCord" "Striatum"
## [13] "Subependymal" "SubstantiaNigra"
"Thalamus"
```

Below, we list the first three cell types available in NeuroExpresso for the Midbrain region as well as the first few marker genes associated with each cell type. Note that the genes are identified here by mouse gene symbols.

```
lapply(mouseMarkerGenes$Midbrain[1:3],head, 14)
```

```
## $Astrocyte
## [1] "Aass" "Acsbg1" "Acsl6" "Acss1" "Add3"
"Adhfe1"
## [7] "AI464131" "Aldh1l1" "Aldh6a1" "Antxr1"
"Aox1" "Apoe"
## [13] "Aqp4" "Axl"
##
## $BrainstemCholin
## [1] "2310030G06Rik" "Anxa2" "Cabp1" "Calca"
## [5] "Calcb" "Cd24a" "Cd55" "Cda"
## [9] "Chodl" "Ecel1" "Fxyd7" "Hebp2"
## [13] "Hspb1" "Hspb8"
##
## $Dopaminergic
## [1] "Cacna2d2" "Cadps2" "Chrna6" "Mapk8ip2"
"Nr4a2" "Ntn1"
## [7] "Prkcg" "Rian" "Scn2a" "Slc6a3" "Snhg11"
"Tenm1"
## [13] "Th" "Zim3"
```

As described above, these markers are genes that have high expression levels in one cell type but not other cell types within the same brain region. In Figure 3, we have plotted a heat map of the gene

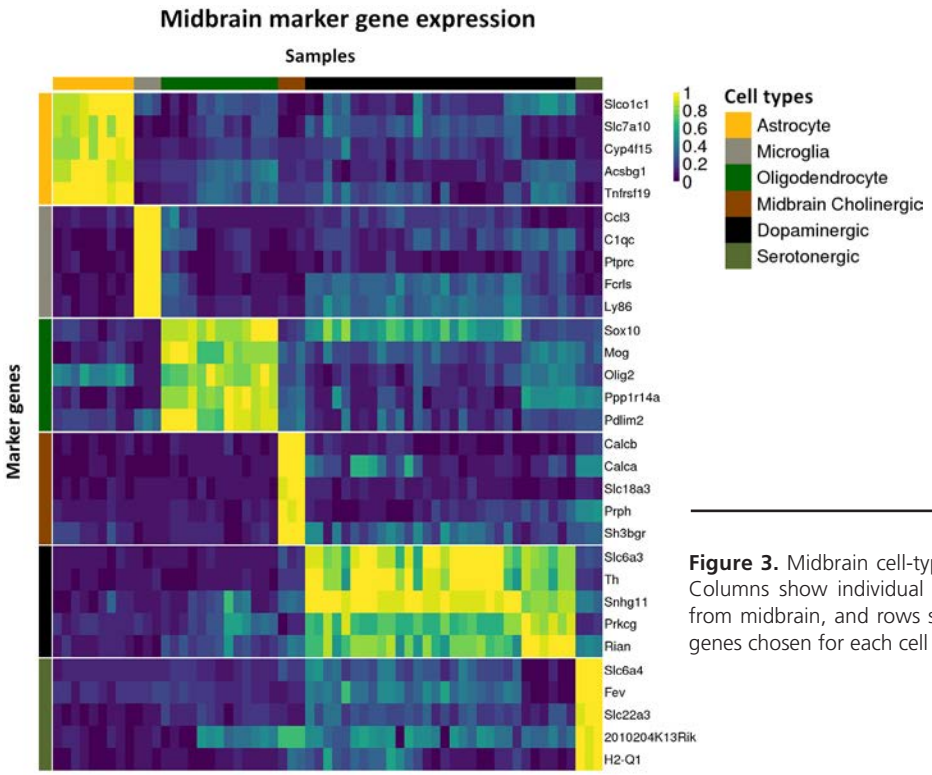


**Figure 3.** Midbrain cell-type-specific marker genes. Columns show individual cell-type-specific samples from midbrain, and rows show the top five marker genes chosen for each cell type.

expression levels for the top five marker genes per cell type annotated to the region Midbrain. As expected, known dopaminergic marker genes, including tyrosine hydroxylase (gene symbol *Th*), are selected as marker genes for midbrain dopaminergic cells.

We note that these cell-type-specific mouse gene expression data are not part of the markerGeneProfile package but can be accessed freely online through Neuroexpresso.org or github.com/oganm/neuroExpressoAnalysis.

### Bulk tissue transcriptomic data

As an example of a bulk tissue brain-region-specific gene expression dataset amenable for MGP analysis, we selected a dataset collected by Lesnick et al. (2007) in which postmortem gene expression profiles from the midbrains of human controls and PD patients were assayed using microarrays.

We have made the preprocessed Lesnick dataset available as part of the markerGeneProfile package. This matrix (the object mgp_LesnickParkinsonsExp) is organized into a data frame with unique samples on columns and genes/microarray probes as rows. The first few rows of the gene expression data matrix are shown below. The metadata are provided in the object `mgp_LesnickParkinsonsMeta`, which lists which group (Control or Disease) each sample belongs to.

```
data(mgp_LesnickParkinsonsExp)
mgp_LesnickParkinsonsExp %>%
    dplyr::select(-GeneNames) %>%
    head %>% {.[,1:6]}
```

```
## Probe Gene.Symbol NCBIids GSM184354.cel
GSM184355.cel
## 43955 1007_s_at DDR1 780 10.236880 9.891552
## 2278 1053_at RFC2 5982 5.421790 5.280541
## 45312 117_at HSPA6|HSPA7 3310|3311 5.164445
4.651754
## 43710 121_at PAX8 7849 7.076004 7.035090
## 13573 1255_g_at GUCA1A 2978 3.107388
3.418976## 21022 1294_at UBA7 7318 6.644858
6.182664
## GSM184356.cel
## 43955 10.498371
## 2278 5.852467
## 45312 4.729189
## 43710 6.698765
## 13573 3.491832
## 21022 5.982642
data(mgp_LesnickParkinsonsMeta)
mgp_LesnickParkinsonsMeta %>% head
## GSM disease
## 1 GSM184354 Control
## 2 GSM184355 Control
## 3 GSM184356 Control
## 4 GSM184357 Control
## 5 GSM184358 Control
## 6 GSM184359 Control
```

### Preprocessing bulk tissue expression data

One of the most important preprocessing steps before estimating MGPs is to filter out genes with low expression signals. This step is important because a low expression signal often indicates that the gene is not expressed (i.e., the source for the signal is noise rather than biological signal). This is especially relevant for microarray data in which all genes have non-zero background signals regardless of the biological expression. Lowly expressed genes will only interfere with later analysis steps, so we remove them here.

Another preprocessing step required for MGP estimation is to summarize multiple probesets (for microarray) or splice isoforms so that each gene is represented only once in the postprocessed bulk tissue expression dataset. Although there are many probeset summarization methods, for this tutorial, we remove all probesets with a maximum expression below the median and select the most variable probeset per gene. We perform both of these steps using the single function mostVariable, below.

```
unfilteredParkinsonsExp = mgp_
LesnickParkinsonsExp # keep this for later
medExp = mgp_LesnickParkinsonsExp %>%
    ogbox::sepExpr() %>% {.[[2]]} %>%
    unlist %>% median

# mostVariable function is part of this package
that does
# probe selection and filtering for you
mgp_LesnickParkinsonsExp = mostVariable(mgp_
LesnickParkinsonsExp,
threshold = medExp,
threshFun= median)
```

## MGP Estimation Overview

The approach we cover makes use of multiple cell-type marker genes summarized as a single measure using principal component analysis (PCA). Specifically, we summarize the expression profiles of marker genes as the first principal component (PC) of their expression levels (Xu et al., 2014; Chikina et al., 2015; Westra et al., 2015). We refer to these summaries as MGPs. One MGP is estimated for each cell type being considered. The intuition we follow is that we are interested in the common signal change across the marker genes as best reflecting changes in cell proportions. Although there are multiple potential sources of variability in marker gene expression levels, including biological factors (e.g., regulation), technical factors (e.g., RNA quality), and sampling noise, the major source of common variance in their expression (captured in

NOTES

the first PC) most likely represents changes in cell-type abundance. Various validations of this approach are presented in the papers cited earlier as well as in Mancarci et al. (2016).

Despite the power of the PC approach, we implement additional quality checks rather than treating the MGP analysis as a black box (described in more detail in later sections). For example, for good quality results, the first PC should explain 40–70% of the variance in marker expression. Values lower than this mean that the marker genes are not strongly correlated, suggesting that factors other than cell-type proportions are dominating the signals. Another consideration is expression level: Including marker genes that are not robustly detected as expressed (as well as genes that are not sufficiently specific to the cell type) will have a strong adverse effect on the analysis. This is especially important when analyzing human tissue because cell-type-specific markers are often inferred from rodent studies (as we do here). We anticipate that some of the marker genes in mouse cell types will not be equivalently expressed in the corresponding human cell types, or that their expression in bulk tissue might be too low to reliably detect. For all these reasons, it is probable that for some datasets and/or cell types, MGPs cannot be confidently estimated. We discuss various additional caveats in a later section.

## MGP calculation exercise

The death of dopaminergic cells within the substantia nigra is a known hallmark of PD. In the exercise that follows, we apply the list of marker genes derived from midbrain mouse cell types (e.g., dopaminergic cells, cholinergic cells, astrocytes) to the bulk tissue expression data from human control and Parkinsonian subjects.

Although estimating MGPs requires a simple calculation of PC scores per sample using only the marker gene sets for a particular cell type, we have provided a convenience function within the markerGeneProfile R package for estimating MGP. This function, mgpEstimate, takes as input bulk tissue expression data (**exprData** below), marker genes (**genes**), and experimental groups (**groups**) and returns as outputs the calculated MGPs as well as a number of quality control (QC) metrics. The function outputs a variable, **estimations**, containing the MGP estimates per cell type.

Because the marker genes are defined as mouse gene symbols, whereas genes in the bulk tissue expression data are defined using human gene symbols, the function also transforms the mouse gene names into human gene names (using the homologene R package). Please see the documentation for more information on optional inputs to the function for further customization and extra information on the outputs under **estimations**.

```
estimations =  mgpEstimate(exprData=mgp_
LesnickParkinsonsExp,
  genes=mouseMarkerGenes$Midbrain,
  geneColName='Gene.Symbol',
  geneTransform =
       function(x){homologene::mouse2human(x)$
       humanGene},
  groups=mgp_LesnickParkinsonsMeta$disease)
```

The values for the MGP estimations are stored within the estimates object for each cell type:

```
ls(estimations$estimates)
## [1] "Astrocyte" "BrainstemCholin"
## [3] "Dopaminergic" "Microglia"
## [5] "Microglia_activation" "Microglia_
deactivation"
## [7] "Oligo" "Serotonergic"
```

Below, we create a data frame to store the sample-by-sample MGP estimation results corresponding to the dopaminergic cell type. The groups are indicated by the **state** variable.

```
dopaminergicFrame =
   data.frame(`Dopaminergic MGP` = estimations$
   estimates$Dopaminergic,
      state = estimations$groups$Dopaminergic,
      check.names=FALSE)
```

We plot the sample-by-sample MGP estimates below (Fig. 4). Each human Midbrain gene expression sample is shown by a single dot, and the groups have been separated by disease state.

```
library(ggplot2)
ggplot2::ggplot(dopaminergicFrame,
      aes(x = state, y = `Dopaminergic MGP`)) +
   ogbox::geom_ogboxvio() + geom_jitter(width
   = .05)
```

As a contrast to the MGP estimates, we can plot the gene expression values of the dopaminergic marker genes from the Midbrain bulk tissue samples (Fig. 5). Note that most of the dopaminergic marker genes, including *TH* and *SLC6A3*, have higher expression values in the Control samples than in the PD samples.

NOTES

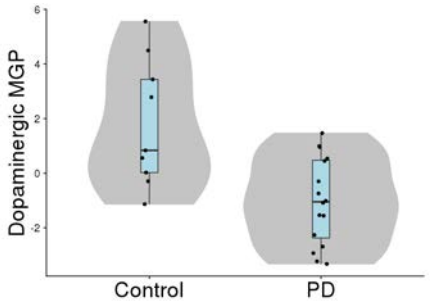


**Figure 4.** Sample-by-sample MGP estimate plots for dopaminergic cells. Each human Midbrain bulk tissue gene expression sample is shown by a single dot, and the groups have been separated by disease state: Control and PD.

```
estimations$usedMarkerExpression$Dopaminergic%>%
    as.matrix %>%
    gplots::heatmap.2(trace = 'none',
         scale='row',Rowv = FALSE,Colv = FALSE,
         dendrogram = 'none',
         col= viridis::viridis(10),cexRow = 1,
         cexCol = 0.5,
         ColSideColors =
         estimations$groups$Dopaminergic %>%
           ogbox::toColor(palette = c('Control'
           = 'blue',
           "PD" = "red")) %$% cols ,
         margins = c(5,5))
```

Indeed, in general, most marker genes have a high expression in control samples (indicated by the blue bar above the heat map) compared with the PD samples (red bar).

## Testing for group differences in MGPs

Once we have MGP estimates for each sample, a natural question arises as to whether these values differ in distribution among experimental groups. There are myriad ways to perform statistical tests for group differences (e.g., Student's *t*-test of the mean). Here, we apply the nonparametric Wilcoxon rank-sum test (Mann–Whitney *U* test):

```
group1 = estimations$estimates$Dopaminergic
[estimations$groups$Dopaminergic %in% "Control"]
group2 = estimations$estimates$Dopaminergic
[estimations$groups$Dopaminergic %in% "PD"]
wilcox.test(group1,group2)
```

```
##
## Wilcoxon rank sum test
##
## data: group1 and group2
## W = 119, p-value = 0.006547
## alternative hypothesis: true location shift
is not equal to 0
```

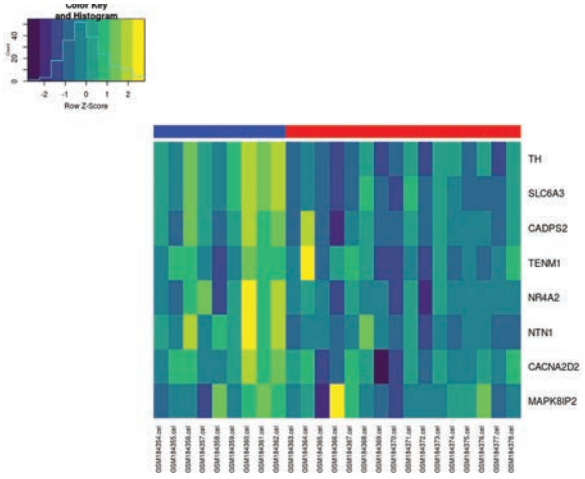


**Figure 5.** Heat map showing gene expression values of dopaminergic marker genes from Midbrain bulk tissue samples. Most of the dopaminergic marker genes (right) have higher expression values in the Control samples (blue bar) than in the PD samples (red bar).

Based on these results, we can say that there is a significant difference ($p = 0.0065$) in the dopaminergic MGPs between Control and PD patients.

Thus far, we have applied MGP estimation in a fairly bare-bones manner. However, we encourage a closer inspection of the data and the use of QC metrics (discussed in the next section) to ensure that the method is giving meaningful results.

It is important to note that not all of the mouse cell-type-specific marker genes (which NeuroExpresso provides) can be used to estimate MGPs in human bulk tissue samples. There are three main reasons why genes are not used in the estimation of MGPs: (1) there is no matching gene orthologue between mouse to human; (2) the gene is not represented in the bulk tissue dataset (e.g., not sampled on microarray platform) or is filtered out due to low expression level; and (3) the gene has low correlation between its expression in the bulk tissue data compared with the majority of marker genes of the same cell type. We detail these aspects in the next demonstrations.

NOTES

We start by listing all genes that are identified as dopaminergic cell markers having human orthologues:

```
mouseHumanGeneTable = mouseMarkerGenes$Midbrain
$Dopaminergic %>% homologene::mouse2human()
allHumanDopaGenes = mouseHumanGeneTable %$%
humanGene
mouseHumanGeneTable

## mouseGene humanGene
## 1 Cacna2d2 CACNA2D2
## 2 Cadps2 CADPS2
## 3 Chrna6 CHRNA6
## 4 Mapk8ip2 MAPK8IP2
## 5 Nr4a2 NR4A2
## 6 Ntn1 NTN1
## 7 Prkcg PRKCG
## 8 Slc6a3 SLC6A3
## 9 Tenm1 TENM1
## 10 Th TH
```

Thus, of the 14 mouse dopaminergic marker genes, 10 have human orthologues. However, *CHRNA6*, as shown by the next line of code, is not present in the dataset (because of expression-level based filtering), so we remove it from consideration:

```
allHumanDopaGenes[!allHumanDopaGenes %in% mgp_
LesnickParkinsonsExp$Gene.Symbol]
```

```
## [1] "CHRNA6"
```

```
allGenesInDataset = allHumanDopaGenes[allHuman
DopaGenes %in% mgp_LesnickParkinsonsExp$Gene.
Symbol]
```

Next, because the MGP estimation algorithm uses PCA to summarize the expression of multiple marker genes specific to a cell type, the algorithm (by default) removes or drops individual marker genes if they have poor correlation (across samples) with other marker genes corresponding to a cell type (as is the case for *PRKCG*, below). (Details of this process can be found in Mancarci et al., 2016.) We show genes that were dropped during estimation as follows:

```
allGenesInDataset[!allGenesInDataset %in%
rownames(estimations$usedMarkerExpression$
Dopaminergic)]
## [1] "PRKCG"
```

To better help visualize these gene-dropping steps, we plot the raw expression levels for all the human bulk tissue samples for the 10 dopaminergic marker genes (Fig. 6). Here, we can see that *CHRNA6* and *PRKCG* have very low expression levels. *CHRNA6* was removed at our earlier expression-level filtering step. *PRKCG* just barely passed that threshold but was removed later owing to its low overall correlation (across samples) with the other dopaminergic marker genes.

```
genesUsed =  rownames(estimations$usedMarker
Expression$Dopaminergic)



toPlot =
    unfilteredParkinsonsExp[unfilteredParkinsons
Exp$Gene.Symbol %in%
   homologene::mouse2human(mouseMarkerGenes$
Midbrain$Dopaminergic)$humanGene,] %>%
    mostVariable(threshold = 0)




toPlot %>%
    mostVariable(threshold = 0) %>%
    ogbox::sepExpr() %>%
    {.[[2]]} %>% as.matrix()%>%
    {rownames(.) =
        toPlot$Gene.Symbol[toPlot$Gene.Symbol
%in% homologene::mouse2human(mouseMarkerGenes$M
idbrain$Dopaminergic)$humanGene];.} %>%
    reshape2::melt() %>% {colnames(.) = c('Gene
','Sample','Expression');.} %>%
    dplyr::mutate(`Is used?` =
rep('used',length(Gene)) %>%
                       {.[Gene %in% 'CHRNA6'] =
'CHRNA6 - not expressed';.[Gene %in% 'PRKCG'] =
'PRKCG - not correlated';.}) %>%
    ggplot(aes(y = Expression, x = Sample,
group = Gene, color = `Is used?`)) +
    geom_line() +
    cowplot::theme_cowplot() +
    theme( axis.text.x= element_blank()) +
    ggtitle('Nonscaled expression of markers')
```

## QC metrics in MGP estimation

As a more formal metric for controlling the quality of MGP estimates, we suggest using the fraction of removed genes. The rationale is that if a significant portion of the genes do not correlate well with each other, this might indicate that the variance explained by the first PC cannot be explained by changes in cellular abundance. For example, this can happen if some of the genes are highly regulated in a subset of the samples or if a substantial proportion of the genes is either not sufficiently expressed or not specific to the cell type. For all cell types, this ratio is calculated and output as the `removedMarkerRatios`. The function outputs a warning if this ratio exceeds 0.4:1.0 for any cell type (i.e., > 40% of marker genes for a cell type are removed). Of note, for some cell types, the expression level of the markers in bulk tissue is relatively low, and thus, the ratio of removed genes is normally relatively high but without affecting the reliability of the results. In general, the ratio of the removed genes should always be evaluated in the context of variance explained by the first PC, as described below:

NOTES

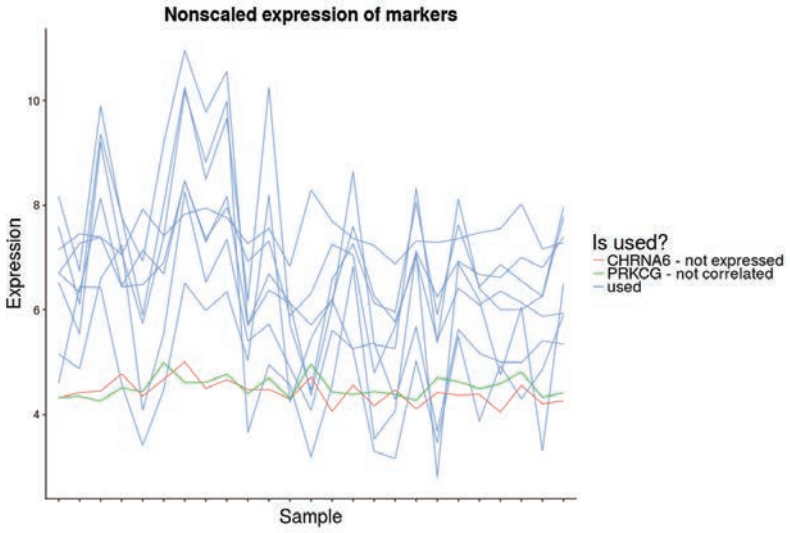


**Figure 6.** Plot of raw expression levels for all human bulk tissue samples for 10 dopaminergic marker genes. Of these, *CHRNA6* and *PRKCG* have very low expression levels.

```
estimations$removedMarkerRatios
## Astrocyte BrainstemCholin Dopaminergic
## 0.12962963 0.37500000 0.11111111
## Microglia Microglia_activation Microglia_
deactivation
## 0.09649123 0.08219178 0.10909091
## Oligo Serotonergic
## 0.05882353 0.00000000
```

Note that the proportion of removed marker genes for Midbrain dopaminergic cells is fairly low: 1 of 9 total marker genes pass the expression-level-based filtering. Unlike dopaminergic cells, cholinergic cells seem to have a higher proportion of their marker genes removed (~0.375).

As a complementary QC metric, we suggest inspecting the amount of variance explained in the first PC of marker gene expression. For the dopaminergic cell-type marker genes here, it is 66%. If this value is low, it is very likely that the first PC does not correspond to variance explained by changes in cellular abundance.

```
estimations$trimmedPCAs$Dopaminergic %>%
summary()
## Importance of components%s:
## PC1 PC2 PC3 PC4 PC5 PC6
## Standard deviation 2.3056 1.0829 0.87149
0.53287 0.45796 0.38075
## Proportion of Variance 0.6645 0.1466 0.09494
0.03549 0.02622 0.01812
## Cumulative Proportion 0.6645 0.8110 0.90597
0.94147 0.96768 0.98580
## PC7 PC8
## Standard deviation 0.28376 0.18177
## Proportion of Variance 0.01006 0.00413
## Cumulative Proportion 0.99587 1.00000
```

Looking at the variation explained by the first PC of cholinergic cells reveals that the first PC explains only 29% of all variance. Thus, both the proportion of genes removed and the variance explained by the first PC indicate that the MGP estimations for the cholinergic cell type are more suspect:

```
estimations$trimmedPCAs$BrainstemCholin %>%
summary()
## Importance of components%s:
## PC1 PC2 PC3 PC4 PC5 PC6 PC7
## Standard deviation 1.7002 1.3215 1.2445
1.1559 0.8752 0.7804 0.68942
## Proportion of Variance 0.2891 0.1746 0.1549
0.1336 0.0766 0.0609 0.04753
## Cumulative Proportion 0.2891 0.4637 0.6186
0.7522 0.8288 0.8897 0.93725
## PC8 PC9 PC10
## Standard deviation 0.55879 0.44616 0.34094
## Proportion of Variance 0.03122 0.01991
0.01162
## Cumulative Proportion 0.96847 0.98838
1.00000
```

## Limitations and Pitfalls of MGP Interpretation

Although the data inspection and QC steps described above help ensure that users understand the source and meaning of the outputs of MGP analysis, users should pay heed to a number of other caveats and guidelines. Here we outline a few of the most pertinent; for additional discussion of limitations and caveats, see Mancarci et al. (2016).

NOTES

### Interpretation of MGPs as cellular proportion rather than regulatory changes

The most likely explanation for concordant change in a large proportion of marker genes is change in cellular abundance, but this does not always have to be so. It is important to remember that many marker genes encode for proteins involved in the biological function of a specific cell type (e.g., many of the oligodendrocyte markers encode proteins involved in synthesis and maintenance of myelin). This implies that under specific conditions, the genes can be coregulated (e.g., if myelin synthesis is blocked or if the condition affects the maturation state of the cells), resulting in transcriptomic regulatory changes in multiple marker genes inside individual cells. Thus, changes in MGPs should be interpreted carefully while considering alternative explanations for the observed change.

It is also important to remember that we treat the first PC as capturing the variance that correlates with cellular abundance. However, this might not be true if larger sources of variation exist in the data. For example, sample pH and mRNA quality are known to affect the measured expression signal. Thus, in datasets with large pH differences or datasets of low mRNA quality, the main source of variance in the expression of marker genes might not be related to cellular abundance but rather correspond to other biological or technical effects. Although the quality metrics described above can help to identify such cases, it is advisable to make sure that the data are of good quality before applying the algorithm.

### Necessary sample size

As with any differential expression analysis, we recommend applying the method to datasets with a large number of samples (> 10). This is important because PCA is sensitive to outliers, and it follows that the existence of outliers would have greater effect in datasets with small sample size.

### Necessary numbers of marker genes per cell type

Because individual genes can be regulated by different conditions, if MGPs are calculated based on a small number of starting marker genes, the impact of any regulation is more likely to be problematic. We thus suggest that MGPs based on fewer than three marker genes not be trusted as indicators of cellular abundance.

### Which cell types to use

Not every cell type is present in every brain region, so it makes sense to estimate MGPs only for cell types expected to be present. In addition, the markers provided by NeuroExpresso are, where possible, chosen in a brain-region-specific manner. This means that markers for cells types might overlap across regions. A good example is Purkinje cells of the cerebellum, which share markers with inhibitory interneurons found in the neocortex. Although estimation of a Purkinje cell MGP from neocortical data will "work," the results are obviously not interpretable as intended.

### MGPs as relative measures

In an ideal world, the MGP approach would yield the fraction of the sample made up from each cell type, and those fractions would add up to 1.0. Unfortunately, this is not the case, and for many reasons would be difficult to achieve with any method. The MGP estimates are effectively on an arbitrary unit-less scale and can be compared only with the MGP for the same cell type across samples in the same dataset. It is imperative that MGPs not be referred to as proportions, but rather as being correlated with proportions.

## Conclusions

In this tutorial, we have outlined some of the applications of the MGP approach, focusing on the simplest use of estimating MGPs for cell types across samples and comparing them between experimental groups. We have also highlighted the need for caution in interpreting MGPs and the importance of high-quality data. Another common application, not demonstrated here, is to use the MGP values as covariates in statistical models in order to account for cell-type proportion differences as a source of variability that might obscure differences of more primary interest (e.g., disease).

In our work, we have found that the MGPs themselves provide useful insight into the data, to the extent that inferred cell-type-specific effects likely caused by compositional changes are the major signal in the data (Mancarci et al., 2016). We suggest that such effects not be treated as a nuisance. Instead, compositional changes may commonly be present in conditions besides neurodegenerative disorders, and gene expression profiling provides an indirect but efficient means of assessing such effects. Ongoing work on the MGP approach aims at improving the diagnostic and interpretation aids as well as revising and extending the marker gene sets available as more data become available.

## References

Chikina M, Zaslavsky E, Sealfon SC (2015) CellCODE: a robust latent variable approach to differential expression analysis for heterogeneous cell populations. Bioinformatics 31:1584–1591.

Lesnick TG, Papapetropoulos S, Mash DC, Ffrench-Mullen J, Shehadeh L, de Andrade M, Henley JR, Rocca WA, Ahlskog JE, Maraganore DM (2007) A genomic pathway approach to a complex disease: axon guidance and Parkinson disease. PLoS Genet 3:e98.

Mancarci BO, Toker L, Tripathy S, Li B, Rocco B, Sibille E, Pavlidis P (2016) NeuroExpresso: a cross-laboratory database of brain cell-type expression profiles with applications to marker gene identification and bulk brain tissue transcriptome interpretation. bioRxiv 89219: 1–39.

Sibille E, Arango V, Joeyen-Waldorf J, Wang Y, Leman S, Surget A, Belzung C, Mann JJ, Lewis DA (2008) Large-scale estimates of cellular origins of mRNAs: enhancing the yield of transcriptome analyses. J Neurosci Methods 167:198–206.

Skene NG, Grant SGN (2016) Identification of vulnerable cell types in major brain disorders using single cell transcriptomes and expression weighted cell type enrichment. Neurogenomics 10:16.

Tan PPC, French L, Pavlidis P (2013) Neuron-enriched gene expression patterns are regionally anti-correlated with oligodendrocyte-enriched patterns in the adult mouse and human brain. Front Neurosci 7:5.

Westra H-J, Arends D, Esko T, Peters MJ, Schurmann C, Schramm K, Kettunen J, Yaghootkar H, Fairfax BP, Andiappan AK, Li Y, Fu J, Karjalainen J, Platteel M, Visschedijk M, Weersma RK, Kasela S, Milani L, Tserel L, Peterson P, et al. (2015) Cell specific eQTL analysis without sorting cells. PLoS Genet 11:e1005223.

Xu X, Wells AB, O'Brien DR, Nehorai A, Dougherty JD (2014). Cell type-specific expression analysis to identify putative cellular mechanisms for neurogenetic disorders. J Neurosci 34:1420–1431.

NOTES

# Graph Classification Using Signal-Subgraphs: Applications in Statistical Connectomics

Joshua T. Vogelstein, PhD,[1] William R. Gray, PhD,[2] R. Jacob Vogelstein, PhD,[2] and Carey E. Priebe, PhD[1]

[1]Department of Applied Mathematics and Statistics
Johns Hopkins University
Baltimore, Maryland

[2]Johns Hopkins University Applied Physics Laboratory
Laurel, Maryland

NOTES

## Introduction

This chapter considers the following "graph classification" question: Given a collection of graphs and associated classes, how can one predict the class of a newly observed graph? To address this question, we propose a statistical model for graph/class pairs. This model naturally leads to a set of estimators to identify the class-conditional signal, or "signal-subgraph," defined as the collection of edges that are probabilistically different between the classes. The estimators admit classifiers that are asymptotically optimal and efficient but differ by their assumption about the "coherency" of the signal-subgraph (coherency is the extent to which the signal-edges "stick together" around a common subset of vertices). Via simulation, the best estimator is shown to be a function of not just the coherency of the model but also the number of training samples. These estimators are employed to address a contemporary neuroscience question: Can we classify "connectomes" (brain graphs) according to sex? The answer is yes, and significantly better than for all benchmark algorithms considered. Synthetic data analysis demonstrates that even when the model is correct, given the relatively small number of training samples, the estimated signal-subgraph should be taken with a grain of salt. We conclude by discussing several possible extensions. Graphs are emerging as a prevalent form of data representation in fields ranging from optical character recognition and chemistry (Bunke and Riesen, 2011) to neuroscience (Hagmann et al., 2010). Whereas statistical inference techniques for vector-valued data are widespread, statistical tools for the analysis of graph-valued data are relatively rare (Bunke and Riesen, 2011). In this chapter, we consider the task of "labeled graph classification": Given a collection of labeled graphs and their corresponding classes, can we accurately infer the class for a new graph? Note that we assume throughout that each vertex has a unique label, and that all graphs have the same number of vertices with the same vertex labels. The methods developed herein, however, can straightforwardly be relaxed to use them in more general settings.

We propose and analyze a joint graph/class model—sufficiently simple to characterize its asymptotic properties but sufficiently rich to afford useful empirical applications. This model admits a class-conditional signal encoded in a subset of edges: the signal-subgraph. Finding the signal-subgraph amounts to providing an understanding of the differences between the two graph classes. Moreover, borrowing a term from the compressive sensing literature (Donoho et al., 2006; Candès and Wakin, 2008), we are interested in learning to what extent this signal is coherent—that is, to what extent the signal-subgraph edges are incident to a relatively small set of vertices. In other words, if the signal is sparse in the edges, then the signal-subgraph is incoherent; if it is also sparse in the vertices, then the signal-subgraph is coherent (we formally define these notions below).

This graph model–based approach is qualitatively different from most previous approaches, which utilize only unique vertex labels or graph structure. In the former case, simply representing the adjacency matrix with a vector and applying standard machine learning techniques ignores graph structure (for instance, it is not clear how to implement a coherent signal-subgraph estimator in this representation). In the latter case, computing a set of graph invariants (such as clustering coefficient) and then classifying using only these invariants ignores vertex labels (Kudo et al., 2005; Ketkar et al., 2009; Bunke and Riesen, 2011).

Although some of the above approaches consider attributed vertices or edges, we are unable to find any that utilize both unique vertex labels and graph structure. The field of connectomics (the study of brain graphs), however, is ripe with many examples of brain graphs with vertex labels. In invertebrate brain graphs, for example, often each neuron is named such that one can compare neurons across individuals of the same species (North and Greenspan, 2007). In vertebrate neurobiology, even though neurons are rarely named, "neuron types" (Shepherd and Huganir, 2007) and neuroanatomical regions (Nolte, 2002) are named. Moreover, a widely held view is that many psychiatric issues are fundamentally "connectopathies" (Lichtman et al., 2008; Bassett and Bullmore, 2009). For prognostic and diagnostic purposes, merely being able to differentiate groups of brain graphs from one another is sufficient. However, for treatment, it is desirable to know which vertices and/or edges are malfunctioning so that therapy can be targeted to those locations. This is the motivating application for our work.

We demonstrate via theory, simulation, analysis of a neurobiological dataset (magnetic resonance [MR]–based connectome sex classification), and synthetic data analysis that utilizing graph structure can significantly enhance classification accuracy. However, the best approach for any particular dataset is not just a function of the model, but also the amount of data. Moreover, even when the model is true, given a relatively small sample size,

NOTES

the estimated signal-subgraph will often overlap with the truth but not fully capture it. Nonetheless, the classifiers described below still significantly outperform the benchmarks.

# Methods

## Setting

Let $\mathbb{G} : \Omega \to \mathscr{G}$ be a graph-valued random variable with samples $G_i$. Each graph $G = (\mathscr{V}, E)$ is defined by a set of $V$ vertices, $\mathscr{V} = \{v_i\}_{i \in [V]}$, where $[V] = \{1, \ldots, V\}$, and a set of edges between pairs of vertices $E \subseteq V \times V$. Let $A : \Omega \to \mathscr{A}$ be an adjacency matrix-valued random variable taking values $a \in \mathscr{A} \subseteq \mathbb{R}^{V \times V}$, identifying which vertices share an edge. Let $Y : \Omega \to \mathscr{Y}$ be a discrete-valued random variable with samples $y_i$. Assume the existence of a collection of $n$ exchangeable samples of graphs and their corresponding classes from some true but unknown joint distribution: $\{(\mathbb{G}_i, Y_i)\}_{i \in [n]} \overset{exch.}{\sim} F_{\mathbb{G},Y}$. Our aim (exploitation task) is to build a graph classifier that can take a new graph, $\mathbb{G}$, and correctly estimate its class, $y$, assuming that they are jointly sampled from some distribution, $F_{\mathbb{G},Y}$. Moreover, we are interested solely in graph classifiers that are *interpretable* with respect to the vertices and edges of the graph. In other words, nonlinear manifold learning, feature extraction, and related approaches are unacceptable.

We adopt the common practice of identifying graphs with their adjacency matrices. We note, however, that operations available on the latter (addition, multiplication) are not intrinsic to the former.

## Model

Consider the model, $\mathscr{F}_{\mathbb{G},Y}$, which includes all joint distributions over graphs and classes under consideration: $\mathscr{F}_{\mathbb{G},Y} = \{F_{\mathbb{G},Y}(\cdot;\theta) : \theta \in \Theta\}$, where $\theta \in \Theta$ indexes the distributions. We proceed via a hybrid generative–discriminative approach (Lasserre et al., 2006) whereby we describe a generative model and place constraints on the discriminant boundary.

First, assume that each graph has the same set of uniquely labeled vertices so that all the variability in the graphs is in the adjacency matrix, which implies that $F_{\mathbb{G},Y} = F_{A,Y}$. Second, assume edges are independent, that is, $F_{A,Y} = \prod_{u,v \in \mathscr{E}} F_{A_{uv},Y}$, where $\mathscr{E} \subseteq V \times V$ is the set of all possible edges. Now, consider the generative decomposition $F_{A,Y} = F_{A|Y} F_Y$, and let $F_{uv|y} = F_{A_{uv}|Y=y}$ and $\pi_y = F_{Y=y}$. Third, assume the existence of a class-conditional difference, that is, $F_{uv|0} \neq F_{uv|1}$ for some $(u, v) \in \mathscr{E}$, and denote the edges satisfying this condition as the *signal-subgraph*, $S = \{(u, v) \in \mathscr{E} : F_{uv|0} \neq F_{uv|1}\}$. Fourth, although the following theory and algorithms are valid for both directed and undirected graphs, for concreteness, assume that the graphs are simple graphs, that is, undirected, with binary edges, and lacking (self-) loops (so $\mathscr{E} = \binom{V}{2}$). Thus, the likelihood of an edge between vertex $u$ and $v$ is given by a Bernoulli random variable with a scalar probability parameter: $F_{uv|y}(A_{uv}) = \text{Bern}(A_{uv}; p_{uv|y})$. Together, these four assumptions imply the following model:

$$\mathscr{F}_{\mathbb{G},Y} = \{F_{A,Y}(a,y;\theta) \;\; \forall a \in \mathscr{A}, y \in \mathscr{Y} : \theta \in \Theta\}, \qquad (1)$$

$$\text{where} \quad F_{A,Y}(a,y;\theta) = \prod_{uv \in \mathscr{S}} \text{Bern}(a_{uv}; p_{uv|y}) \pi_y \times \prod_{uv \in \mathscr{E} \setminus \mathscr{S}} \text{Bern}(a_{uv}; p_{uv}), \qquad (2)$$

and $\theta = \{p, \pi, \mathscr{S}\}$. The likelihood parameter is constrained such that each element must be between zero and one: $p \in (0, 1)^{\binom{V}{2} \times |\mathscr{Y}|}$. The prior parameter, $\pi = (\pi_1, \ldots, \pi_{|\mathscr{Y}|})$, must have elements greater than or equal to zero and sum to one: $\pi_y \geq 0$, $\sum_y \pi_y = 1$. The signal-subgraph parameter is a nonempty subset of the set of possible edges, $\mathscr{S} \subseteq \mathscr{E}$ and $\mathscr{S} \neq \emptyset$.

We consider up to two additional constraints on $\mathscr{S}$. First, the size of the signal-subgraph may be constrained such that $|\mathscr{S}| \leq s$. Second, the minimum number of vertices onto which the collection of edges is incident to is constrained such that $\mathscr{S} = \{(u, v) : u \cup v \in \mathscr{U}\}$, where $\mathscr{U}$ is a set of *signal-vertices* with $|\mathscr{U}| \leq m$. Edges in the signal-subgraph are called *signal-edges*. Note that given a collection of signal-edges, the signal-vertex set may not be unique. Although it may be natural to treat $S$ as a prior, we treat it as a parameter of the model; the constraints, $s$ and $m$, are considered to be hyperparameters.

Note that given a specification of the class-conditional likelihood of each edge and class-prior, one completely defines a joint distribution over graphs and classes; the signal-subgraph is implicit in that parameterization. However, the likelihood parameters for all edges not in the signal-subgraph, $p_{uv|y} = p_{uv} \; \forall y \in \mathscr{Y}$, $(u, v) \notin \mathscr{S}$, are "nuisance parameters"; that is, they contain no class-conditional signal. When computing a relative posterior class estimate, these nuisance parameters cancel in the ratio.

## Classifier

A graph classifier, $h \in \mathscr{H}$, is any function satisfying $h : \mathscr{G} \to \mathscr{Y}$. We desire the "best" possible classifier, $h_*$. To define best, we first choose a loss function, $\ell_h : \mathscr{G} \times \mathscr{Y} \to \mathbb{R}_+$, specifically the $0-1$ loss function:

$$\ell_h(G,y) \triangleq \mathbb{I}\{h(G) \neq y\}, \qquad (3)$$

where $\mathbb{I}\{\cdot\}$ is the indicator function, equaling one whenever its argument is true and zero otherwise.

Further, let risk, $\boldsymbol{R} : \boldsymbol{\mathcal{F}} \times \boldsymbol{\mathcal{H}} \to \mathbb{R}_+$, be the expected loss under the true distribution:

$$\boldsymbol{R}(F,h) \triangleq \mathbb{E}_F[\ell_h(\mathbb{G},Y)]. \quad (4)$$

The Bayes optimal (best) classifier for a given distribution $F$ minimizes risk. It can be shown that the classifier that maximizes the class-conditional posterior $F_{Y|\mathbb{G}}$ is optimal (Bickel and Doksum, 2000):

$$\begin{aligned} h_* &= \operatorname*{argmin}_{h\in\mathcal{H}} \mathbb{E}_F[\ell_h(\mathbb{G}, Y)] \quad (5)\\ &= \operatorname*{argmax}_{y\in\mathcal{Y}} F_{\mathbb{G}|Y=y} F_{Y=y}. \end{aligned}$$

Given the proposed model, Equation 5 can be further factorized using the above four assumptions:

$$h_*(G) = \operatorname*{argmax}_{y\in\mathcal{Y}} \prod_{u,v\in\mathcal{S}} \text{Bern}(A_{uv}; p_{uv|y})\pi_y. \quad (6)$$

Unfortunately, Bayes optimal classifiers are typically unavailable. In such settings, it is therefore desirable to induce a classifier estimate from a set of *training data*. Formally, let $\mathcal{T}_n = \{(\mathbb{G}_i, Y_i)\}_{i\in[n]}$ denote the training corpus, where each graph-class pair is sampled exchangeably from the true but unknown distribution: $(\mathbb{G}_i, Y_i) \overset{exch.}{\sim} F_{\mathbb{G},Y}$. Given such a training corpus and an unclassified graph G, an induced classifier predicts the true (but unknown) class of G, $\hat{h} : \boldsymbol{\mathcal{G}} \times (\boldsymbol{\mathcal{G}} \times \boldsymbol{\mathcal{Y}})_n \to \boldsymbol{\mathcal{Y}}$. When a model $\boldsymbol{\mathcal{F}}_{\mathbb{G},Y}$ is specified, a beloved approach is a *Bayes plug-in classifier*. Because of the above simplifying assumptions, the Bayes plug-in classifier for this model is defined as follows: First, obtain parameter estimates $\theta = \{\mathcal{S}, p, \pi\}$. Second, plug those estimates into the above equation. The result is a Bayes plug-in graph classifier:

$$\hat{h}(G;\mathcal{T}_n) \triangleq \operatorname*{argmax}_{y\in\mathcal{Y}} \prod_{u,v\in\hat{\mathcal{S}}} \hat{p}_{uv|y}^{a_{uv}}(1-\hat{p}_{uv|y})^{(1-a_{uv})}\hat{\pi}_y \quad (7)$$

where the Bernoulli probability is explicit. To implement such a classifier estimate, we specify estimators for $\boldsymbol{\mathcal{S}}$, $\pi$, and $\boldsymbol{p}$.

## Estimators

### *Desiderata*

We desire a sequence of estimators, $\hat{\theta}_1, \hat{\theta}_2, \ldots$, that satisfy the following five desiderata, listed in no particular order:

1. **Consistent:** An estimator is consistent (in some specified sense) if its sequence converges in the limit to the true value: $\lim_{n\to\infty} \hat{\theta}_n = \theta$.
2. **Robust:** An estimator is robust if the resulting estimate is relatively insensitive to small model misspecifications. Because the space of models is massive (uncountably infinite), it is intractable to consider all misspecifications, so we consider only a few of them, as described below.
3. **Quadratic complexity:** Computational time complexity should be no more than quadratic in the number of vertices.
4. **Interpretable:** We desire that the parameters be interpretable with respect to a subset of vertices and/or edges.
5. **Finite sample/empirical performance:** At the end of the day, we are concerned with having a classifier that works to solve our applied problems.

### *Signal-Subgraph Estimators*

Naively, one might consider a search over all possible signal-subgraphs by plugging each one into the classifier and selecting the best performing option. This strategy is intractable because the number of signal-subgraphs scales super-exponentially with the number of vertices (Fig. 1, left panel). Specifically, the number of possible edges in a simple graph with $V$ vertices is $d_V = \binom{V}{2}$, so the number of unique possible signal-subgraphs is $2^{\binom{V}{2}}$. Searching over all of them is sufficiently computationally taxing so as to motivate the search for other alternatives.

Before proceeding, recall that we assume each edge is independent; thus, one can evaluate each edge separately (although treating edges independently is not necessarily advisable, considering the Stein estimator [Stein, 1956]). Formally, consider a hypothesis test for each edge. The simple null hypothesis is that the class-conditional edge distributions are the same, so $H_0 : F_{uv|0} = F_{uv|1}$. The composite alternative hypothesis is that they differ, so $H_A : F_{uv|0} \neq F_{uv|1}$. Given such hypothesis tests, one can construct test statistics $T_{uv}^{(n)} : \boldsymbol{\mathcal{T}}_n \to \mathbb{R}_+$. We reject the null in favor of the alternative whenever the value of the test statistic is greater than some critical value: $T_{uv}^{(n)}(\boldsymbol{\mathcal{T}}_n) > c$. We can therefore construct a significance matrix $\boldsymbol{T} \triangleq T_{uv}^{(n)}$, which is the sufficient statistic for the signal-subgraph estimators. Example test statistics include Fisher's and chi-squared, which will be discussed further below. Whichever test statistic one uses, the sufficient statistics are captured in a $2 \times |\boldsymbol{\mathcal{Y}}|$ contingency table, indicating the number of times edge $u, v$ was observed in each class. For example, the two-class contingency table for each edge is given by:

| | Class 0 | Class 1 | Total |
|---|---|---|---|
| Edge | $n_{uv\,\mid 0}$ | $n_{uv\,\mid 1}$ | $n_{uv}$ |
| No Edge | $n_0 - n_{uv\,\mid 0}$ | $n_1 - n_{uv\,\mid 1}$ | $n - n_{uv}$ |
| Total | $n_0$ | $n_1$ | $n$. |

For simplicity, we will assume that $|\boldsymbol{\mathcal{Y}}| = 2$ for the remainder, though the general case is relatively straightforward.

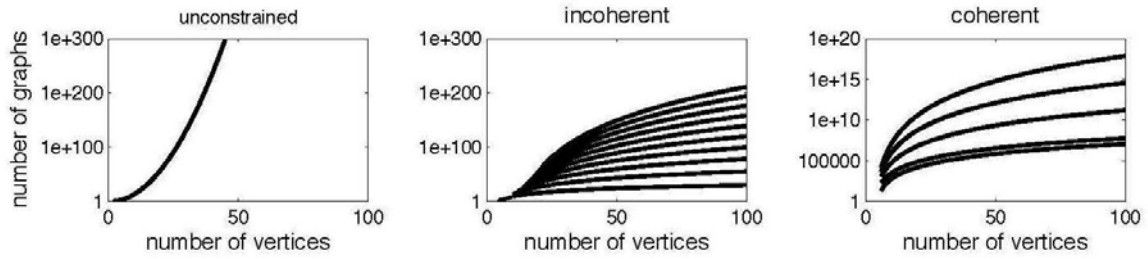


**Figure 1.** Exhaustive searches for the signal-subgraph, even given severe constraints, are computationally intractable even for small graphs. The three panels illustrate the number of unique simple subgraphs as a function of the number of vertices $V$ for the three different constraint types considered: unconstrained, edge constrained (incoherent), and both edge and vertex constrained (coherent). Note the ordinates are all log scale. On the left is the unconstrained scenario, that is, all possible subgraphs for a given number of vertices. In the middle panel, each line shows the number of subgraphs with a fixed number of signal-edges, $s$, ranging from 10 to 100, incrementing by 10 with each line. The right panel shows the number of subgraphs for various fixed $s$ and only a single signal-vertex; that is, all edges are incident to one vertex. Reprinted with permission from Vogelstein JT et al. (2013), Figure 1. Copyright 2013, Institute of Electrical and Electronics Engineers.

***Incoherent signal-subgraph estimators.*** Assume the size of the signal-subgraph, $|\mathcal{E}| = s$, is known. The number of subgraphs with $s$ edges on $V$ vertices is given by $\binom{dV}{s}$, which is also super-exponential (Fig. 1, middle panel). Thus, searching them all is currently computationally intractable. When $s$ is given under the independent-edge assumption, one can choose the critical value a posteriori to ensure that only $s$ edges are rejected under the null (that is, have significant class-conditional differences):

$$\text{minimize } c$$

$$\text{subject to } \sum_{(u,v)\in\mathcal{E}} \mathbb{I}\{T_{uv}^{(n)} < c\} \geq s. \qquad (8)$$

Therefore, an estimate of the signal-subgraph is the collection of $s$ edges with minimal test statistics. Let $T_{(1)} < T_{(2)} < \cdots < T_{(dV)}$ indicate the *ordered* test statistics (dropping the superscript indicating the number of samples for brevity). Then, the *incoherent signal-subgraph estimator* is given by $\hat{\mathcal{S}}_n(s) = \{e_{(1)}, \ldots, e_{(s)}\}$, where $e_{(u)}$ indicates the $u^{th}$ edge ordered by significance of its test statistic, $T_{(u)}$.

Note that the number of distinct test-statistic values is typically much smaller than the number of possible settings of $s$; specifically, the number of unique test statistic values will be $t \leq \min(|\mathcal{E}|, (n_0 + 1)(n_1 + 1))$. In practice, $t$ is often far less than either of the upper bounds, because not every edge has a unique contingency table. In such scenarios, certain settings of the hyperparameters will lead to "ties," that is, edges that are equally valid under the assumptions. In such settings, we simply randomly choose edges satisfying the criterion.

Pseudocode for implementing the incoherent signal-subgraph estimator is provided in Algorithm 1, and MATLAB code is available from http://jovo.me.

**Algorithm 1.** Pseudocode for estimating an incoherent signal-subgraph.

**Input:** $\mathcal{T}_n$ and $s$

**Output:** $\hat{\mathcal{S}}_n(s)$

1. Compute test statistics $T_{uv}^{(n)}$ for all $(u, v) \in \mathcal{E}$
2. Sort each edge according to its test-statistic rank, $T_{(1)} < T_{(2)} < \cdots < T_{(dV)}$
3. Let $\hat{\mathcal{S}}_n(s) = \{e_{(1)}, \ldots, e_{(s)}\}$, arbitrarily breaking ties as necessary.

***Coherent signal-subgraph estimators.*** In addition to the size of the signal-subgraph, also assume that each of the edges in the signal-subgraph is incident to one of $m$ special vertices called signal-vertices. Although this assumption further constrains the candidate sets of edges, the number of feasible sets still scales super-exponentially (Fig. 1, right panel). Therefore, we again take a greedy approach.

First, compute the significance of each edge as above, yielding ordered test statistics. Second, rank edges by significance with respect to each vertex, $e_{k,(1)} \leq e_{k,(2)} \leq \cdots \leq e_{k,(n-1)}$ for all $k \in \mathcal{V}$. Third, initialize the critical value at zero, $c = 0$. Fourth, assign each vertex a score equal to the number of edges incident to that vertex more significant than the critical value, $w_{v;c} = \sum_{u\in[V]} \mathbb{I}\{T_{v,u} > c\}$. Fifth, sort the vertex significance scores, $w_{(1);c} \geq w_{(2);c} \geq \cdots \geq w_{(V);c}$. Sixth,

check if there exist $m$ vertices whose scores sum to greater than or equal the size of the signal-subgraph, $s$. That is, check whether the following optimization problem is satisfied:

$$\text{minimize } c$$
$$\text{subject to } \sum_{v\in[m]} w(v);c \geq s. \quad (9)$$

If so, call the collection of $s$ most significant edges from within that subset the *coherent signal-subgraph estimate*, $\hat{\mathcal{S}}_n(s, m)$. If not, increase $c$ and go back to step four. As above, we break ties arbitrarily. Pseudocode for implementing the coherent signal-subgraph estimator is provided in Algorithm 2, and MATLAB code is available from http://jovo.me.

**Algorithm 2.** Pseudocode for estimating a coherent signal-subgraph.

**Input:** $\mathcal{T}_n$ and $(s, m)$

**Output:** $\hat{\mathcal{S}}_n(s, m)$

1. Compute test statistics $T_{uv}^{(n)}$ for all $(u, v) \in \mathcal{E}$
2. Sort each edge according to its vertex-conditional test-statistic rank, $T_{(1),k} < T_{(2),k} < \cdots < T_{(d_V),k}$ for all $k \in \mathcal{V}$
3. Let $c = 0$
4. Let $w_{v;c}=\sum_{u\in\mathcal{V}} \mathbb{I}\{T_{v,u}>c\}$ for all $v \in \mathcal{V}$
5. Let $w_c = \sum_{v\in[m]} {}^{w(v);c}$
6. **while** $w_c < s$ **do**
7. Let $c \leftarrow c + 1$
8. Update $w_c$
9. **end while**

10: Let $\hat{\mathcal{S}}_n(s, m)$ be the collection of $s$ edges from among those that satisfy Equation 9 for the final value of $c$, arbitrarily breaking ties as necessary.

***Coherograms:*** In the process of estimating the incoherent signal-subgraph, one builds a "coherogram." Each column of the coherogram corresponds to a different critical value $c$, and each row corresponds to a different vertex $v$. The $(c, v)^{th}$ element of the coherogram $w_{v;c}$ is the number of edges incident to vertex $v$ with test statistic larger than c. Thus, the coherogram gives a visual depiction of the coherence of the signal-subgraph (see, e.g., Fig. 2, right column).

### *Likelihood estimators*

The class-conditional likelihood parameters $p_{uv|y}$ are relatively simple. In particular, because the graphs are assumed to be simple, $p_{uv|y}$ is just a Bernoulli parameter for each edge in each class. The maximum likelihood estimator (MLE), which is simply the average value of each edge per class, is a principled choice:

$$\hat{p}_{uv|y}^{MLE} = \frac{1}{n_y} \sum_{i|y_i=y} a_{uv}^{(i)}, \quad (10)$$

where $\sum_{i|y_i=y}$ indicates the sum is over all training samples from class $y$. Unfortunately, the MLE has an undesirable property; specifically, if the data contain no examples of an edge in a particular class, then the MLE will be zero. If the unclassified graph exhibits that edge, then the estimated probability of it being from that class is zero, which is undesirable. We therefore consider a smoothed estimator

$$\hat{p}_{uv|y} = \begin{cases} \eta_n & \text{if } \max_i a_{uv}^{(i)} = 0 \\ 1-\eta_n & \text{if } \min_i a_{uv}^{(i)} = 1 \\ \hat{p}_{uv|y}^{MLE} & \text{otherwise} \end{cases} \quad (11)$$

where we let $\eta_n = 1/(10n)$.

### *Prior estimators*

The priors are the simplest. The prior probabilities are Bernoulli, and we are concerned only with the case where $|\mathcal{Y}| \ll n$, so the maximum likelihood estimators suffice:

$$\hat{\pi}_y = \frac{n_y}{n}, \quad (12)$$

where $n_y = \sum_{i\in[n]} \mathbb{I}\{y_i = y\}$.

### *Hyperparameter selection*

The signal-subgraph estimators require specifying the number of signal-edges $s$, as well as the number of signal-vertices $m$ for the coherent classifier. In both cases, the number of possible values is finite. In particular, $s \in [d_V]$ and $m \in [V]$. Thus, to select the best hyperparameters we implement cross-validation procedures (see discussion of classifiers, below, for details), iterating over $(s, m) \in \vec{s} \times \vec{m} \subseteq [d_V] \times [V]$. Note that when $m = V$, the coherent signal-subgraph estimator reduces to the incoherent signal-subgraph estimator. For all simulated data, we compared hyperparameter performance via a training and held-out set. For the real data application, we decided to use a leave-one-out cross-validation procedure owing to the small sample size.

### *All together*

Putting the above pieces together, Algorithm 3 provides pseudo-code for implementing our signal-subgraph classifiers. MATLAB code is available from the first author's website, http://jovo.me.

NOTES

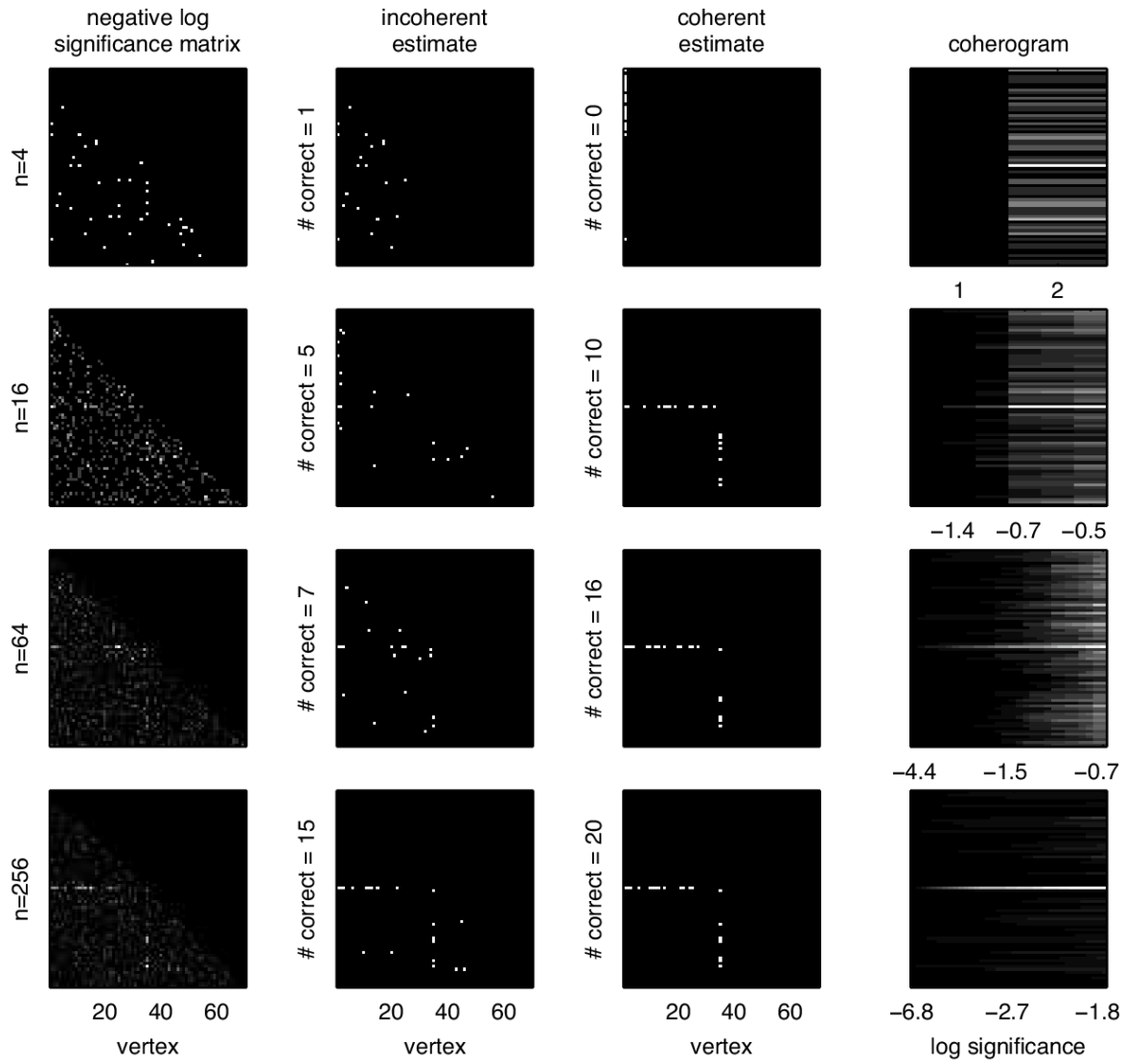


**Figure 2.** An example of the coherent signal-subgraph estimate's improved accuracy over the incoherent signal-subgraph estimate, for a particular homogeneous two-class model specified by: $\mathscr{M}_{70}(1, 20; 0.5, 0.1, 0.3)$. Each row shows the same columns but for increasing the number of graph/class samples. The columns show the negative log-significant matrix (far left), computed using Fisher's exact test (lighter, more significant; each panel is scaled independent of the others because only relative significance matters here); incoherent estimate of the signal-subgraph (column 2); coherent estimate of the signal-subgraph (column 3); and coherogram (far right). As the number of training samples increases (lower rows), both the incoherent and coherent estimates converge toward the truth (the ordinate labels of the middle panels indicate the number of edges correctly identified). For these examples, the coherent estimator tends to find more true edges. The coherogram visually depicts the coherency of the signal; it is also converging toward the truth: The signal-subgraph here contains a single signal-vertex. Reprinted with permission from Vogelstein JT et al. (2013), Figure 2. Copyright 2013, Institute of Electrical and Electronics Engineers.

NOTES

**Algorithm 3.** Pseudocode for training signal-subgraph classifiers.

**Input:** $\mathcal{T}_n$ and a set of constraints $(\vec{s}, \vec{m})$

**Output:** $\hat{\mathcal{S}}_n$, $\{\hat{p}_{uv|y}\}_{(u,v)\in\hat{\mathcal{S}}_n}$, $\{\hat{\pi}_y\}_{y\in\{0,1\}}$

1. Partition the data for the appropriate cross-validation procedure
2. Estimate $p_{uv|y}$ for all $(u, v)$ using Equation 11
3. Estimate $\pi_y$ for all $y$ using Equation 12
4. **for all** $(s, m) \in (\vec{s}, \vec{m})$ **do**
5. Compute $\hat{\mathcal{S}}_n(s, m)$ using Algorithm 1 or 2, as appropriate
6. Compute cross-validated error $\hat{L}_{s,m}$ using Equation 13
7. **end for**
8. Let $\hat{\mathcal{S}}_n = \text{argmin}_{(s,m)} \hat{L}_{s,m}$

## Finite sample evaluation criteria

***Likelihoods and priors***

The likelihood and prior estimators will be evaluated with respect to robustness to model misspecifications, finite samples, efficiency, and complexity.

***Classifier***

We evaluate the classifier's finite sample properties using either held-out or leave-one-out misclassification performance, depending on whether the data are simulated or experimental, respectively. Formally, given C equally sized subsets of the data, $\{\mathcal{T}_1, \ldots, \mathcal{T}_C\}$, the *cross-validated error* is given by

$$\hat{L}_{\hat{h}(\cdot;T_n)} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{|T_n \backslash T_c|}\sum_{G\notin T_c}\mathbb{I}\{\hat{h}(G;T_c) \neq y \quad (13)$$

Given this definition, let $L_{\hat{\pi}}$ be the error of the classifier using only the prior estimates, and let $L_*$ be the error for the Bayes optimal classifier.

To determine whether a classifier is significantly better than "chance" (defined as the performance by a naive classifier), we randomly permute the classes of each graph $\boldsymbol{n}_{\text{MC}}$ times, and then estimate a naive Bayes classifier using the permuted data, yielding an empirical distribution of chance misclassification performance. The $p$-value of a permutation test is the minimum fraction of Monte Carlo permutations that did better than the classifier of interest (Good, 2010).

To determine whether a pair of classifiers are significantly different, we compare the leave-one-out classification results using McNemar's test (McNemar, 1947).

***Signal-subgraph estimators***

To evaluate absolute performance of the signal-subgraph estimators, we define "miss-edge rate" as the fraction of true edges missed by the signal-subgraph estimator:

$$R_n^x = \frac{1}{|\mathcal{S}|}\sum_{(u,v)\in\mathcal{S}}\mathbb{I}\{(u, v)\notin \hat{\mathcal{S}}_n\}. \quad (14)$$

Note that when $|\boldsymbol{\mathcal{S}}|$ is fixed, miss-edge rate is a sufficient statistic for all combinations of false-negative/positive-negative results. Further, we estimate the *relative rate* and *relative efficiency* to evaluate the relative finite sample properties of a pair of consistent estimators. The relative rate is simply $(1 - R_n^{inc})/(1 - R_n^{coh})$. Relative efficiency is the number of samples required for the coherent estimator to obtain the same rate as the incoherent estimator.

## Estimator Properties

***Likelihood and prior estimators***

***Lemma 1.*** $\hat{p}_{uv|y}$ as defined in Equation 11 is an L-estimator.

***Proof.*** Huber defines an L-estimator as an estimator that is a linear combination of (possibly nonlinear functions of) the order statistics of the measurements (Huber, 1981). Indeed, $\hat{p}_{uv|y}$ is a thresholded function of the minimum, maximum, and mean. □

Because L-estimators converge to the MLE, our estimators share all the nice asymptotic properties of the MLE. Moreover, L-estimators are known to be robust to certain model misspecifications (Huber, 1981). The prior estimators are MLEs and therefore also consistent and efficient. Both prior and likelihood estimates are trivial to compute, as closed-form analytic solutions are available for both.

## Signal-subgraph estimators

A variety of test statistics are available for computing the edge-specific class-conditional signal, $T_{uv}^{(n)}$. Fisher's exact test computes the probability of obtaining a contingency table equal to, or more extreme than, the table resulting from the null hypothesis: that the two classes have the same probability of sampling an edge. In other words, Fisher's exact test is the most powerful statistical test

assuming independent edges (Rice, 2001). This leads to the following lemma:

***Lemma 2.*** $\hat{\mathcal{S}}_n(s', m') \to \mathcal{S}$ as $n \to \infty$ when computing $T_{uv}^{(n)}$ via Fisher's exact test, even when $s$ and $m$ are unknown, as long $s' \geq s$ and $m' \geq m$.

***Proof.*** Whenever $p_{uv|0} \neq p_{uv|1}$, the $p$-value of Fisher's exact test converges to zero, whereas whenever $p_{uv|0} = p_{uv|1}$, the distribution of $p$-values converges to the uniform distribution on [0, 1]. Therefore, Fisher's exact test induces a consistent estimator of the signal-subgraph as $n \to \infty$, assuming a fixed and finite $V$. Moreover, as $V \to \infty$, as long as $V/n \to 0$, Fisher's exact test remains consistent (Rice, 2001). □

While most powerful, computing Fisher's exactly is computationally taxing. Fortunately, the chi-squared test is asymptotically equivalent to Fisher's test, and therefore shares those convergence properties (Rice, 2001). Even the absolute difference of MLEs, $|\hat{p}_{uv|1}^{MLE} - \hat{p}_{uv|0}^{MLE}|$, which is trivially easy to compute, is asymptotically equivalent to Fisher's (Rice, 2001) and therefore consistent. Moreover, the signal-subgraph estimators are robust to a variety of model misspecifications. Specifically, as long as all the marginal probability of all the edges in the signal-subgraph are different between the two classes, $p_{uv|1} \neq p_{uv|0}$, and the constraints are upper bounds on the true values, $s' \geq s$ and $m' \geq m$, then any consistent test statistic will yield a consistent signal-subgraph estimator. Estimating the coherent signal-subgraph is more computationally time-consuming than estimating the incoherent signal-subgraph. What is lost by computational time, however, is typically gained by finite sample efficiency whenever the model does not induce too much bias, as will be shown below.

### Bayes plug-in classifier

***Lemma 3.*** The Bayes plug-in classifier, using the signal-subgraph, likelihood, and prior estimators described above, is consistent under the model defined by Equation 2.

***Proof.*** A Bayes plug-in classifier is a consistent classifier whenever the estimates that are plugged in are consistent (Bickel and Doksum, 2000). Because the likelihood, prior, and signal-subgraph estimates are all consistent, the Bayes plug-in classifier is also consistent. □

Note that naive Bayes classifiers often exhibit impressive finite sample performance owing to their winning the bias–variance trade-off relative to other classifiers (Hand and Yu, 2001). In other words, even when edges are highly dependent, because marginal probability estimates are more efficient than joint probability estimates, an independent edge–based classifier will often outperform a classifier based on dependencies.

## Summary

This work makes the following contributions. First, it introduces a novel graph/class model that admits rigorous statistical investigation. Second, it presents two approaches for estimating the signal-subgraph: the first using only vertex label information and the second also utilizing graph structure. The resulting estimators satisfy the five aforementioned desiderata: (model) consistency, robustness to model misspecifications, quadratic complexity, interpretability in terms of the vertices and edges, and state-of-the-art finite sample/empirical performance. Third, simulated data analysis indicates that neither approach dominates the other; rather, the best approach is a function of both the model and the amount of training data. And while the lasso classifier has error properties similar to our incoherent classifier, lasso's computational time is about an order of magnitude longer.

Fourth, these classifiers are applied to an MR connectome sex classification dataset; the coherent classifier performs significantly better than a variety of benchmark classifiers. More specifically, the coherent classifier outperformed a pair of classifiers that use only vertex labels (the naive Bayes and lasso classifiers) as well as a classifier that uses only structural information (the invariant-$k$ nearest-neighbor [$k$NN] classifier). Only the signal subgraph classifier and graph-$k$NN classifier use both vertex labels and graph structure. However, because the graph-$k$NN classifier is universally consistent, it has high variance and therefore takes much longer than the coherent classifier to converge to a good estimate.

Fifth, synthetic data analysis suggests that while we can use the signal-subgraph estimators to improve classification performance, we should not expect that all the edges in the estimated signal-subgraph will be the true signal-edges, even when the model is correct. Moreover, we might expect a drastic improvement in classification performance with only a few additional data samples. Finally, model checking suggests that the independent-edge assumption does not fit the data well.

NOTES

## Acknowledgments

This work was partially supported by the Research Program in Applied Neuroscience. The authors would like to thank Michael Trosset for a helpful suggestion. This chapter was modified from a previously published article of the same title: Vogelstein JT, et al. (2013) IEEE Trans Pattern Anal Mach Intell 35:1539–1551. For a continued explanation of simulations and classifications, please see http://ieeexplore.ieee.org/abstract/document/6341752. Copyright 2013, Institute of Electrical and Electronics Engineers.

## References

Bassett DS, Bullmore ET (2009) Human brain networks in health and disease. Curr Opin Neurol 22:340–347.

Bickel PJ, Doksum KA (2000) Mathematical statistics: basic ideas and selected topics, Vol I, Ed 2, p. 366. Upper Saddle River, NJ: Prentice Hall.

Bunke H, Riesen K (2011) Towards the unification of structural and statistical pattern recognition, Pattern Recognit Lett 33:811–825.

Candès EJ, Wakin M (2008) An introduction to compressive sampling. IEEE Signal Process Mag 25:21–30.

Donoho DLD, Elad M, Temlyakov VNV (2006) Stable recovery of sparse overcomplete representations in the presence of noise. IEEE Trans Inf Theory 52:6–18.

Good PI (2010) Permutation, parametric, and bootstrap tests of hypotheses (Springer series in statistics), Ed 3. New York: Springer.

Hagmann P, Cammoun L, Gigandet X, Gerhard S, Ellen Grant P, Wedeen V, Meuli R, Thiran JP, Honey CJ, Sporns O (2010) MR connectomics: principles and challenges. J Neurosci Methods 194:34–45.

Hand DJ, Yu K (2001) Idiot's Bayes—not so stupid after all? Int Stat Rev 69:385–398.

Huber PJ (1981) Robust statistics (Wiley Series in Probability and Statistics). New York: Wiley.

Ketkar NS, Holder LB, Cook DJ (2009) Empirical comparison of graph classification algorithms. In: IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09), Nashville, TN, March 30–April 2, pp. 259–266.

Kudo T, Maeda E, Matsumoto Y (2005) An application of boosting to graph classification. In: Advances in neural information processing systems 17 (Saul LK, Weiss Y, Bottou L, eds), pp 729–736. Cambridge, MA: MIT Press.

Lasserre JA, Bishop CM, Minka TP (2006) Principled hybrids of generative and discriminative models. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06), New York, NY, June 17–22, Vol 1, pp 87–94.

Lichtman JW, Livet J, Sanes JR (2008) A technicolour approach to the connectome. Nat Rev Neurosci 9:417–422.

McNemar Q (1947) Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika 12:153–157.

Nolte J (2002) The human brain: an introduction to its functional anatomy, Ed 5. Maryland Heights, MO: Mosby.

North G, Greenspan RG (eds) (2007) Invertebrate neurobiology (Cold Spring Harbor Monograph Series). Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press.

Rice JA (2001) Mathematical statistics and data analysis. Belmont, CA: Duxbury Press.

Shepherd JD, Huganir RL (2007) The cell biology of synaptic plasticity: AMPA receptor trafficking. Annu Rev Cell Dev Biol 23:613–643.

Stein CM (1956) Inadmissibility of the usual estimator of the mean of a multivariate normal distribution. In: Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Vol 1. Berkeley, California: University of California Press, 197–206.

Vogelstein JT, Gray Roncal W, Vogelstein RJ, Priebe CE (2013) Graph classification using signal-subgraphs: applications in statistical connectomics. IEEE Trans Pattern Anal Mach Intell 35:1539–1551.

# Data-Intensive Neuroscience: Discovering, Organizing, and Integrating Data for Open, Reproducible Analysis and Modeling

Sean Hill, PhD

Blue Brain Project
École Polytechnique Fédérale de Lausanne, Campus Biotech
Geneva, Switzerland

NOTES

## Introduction

In neuroscience, a major challenge in data-driven analysis and modeling is the heterogeneity of the data in terms of the experimental designs, measurement modalities, and data formats (Akil et al., 2011). This heterogeneity makes it very challenging to identify which data are similar, comparable, and able to be combined or integrated for large-scale analysis or used to constrain a computational model.

Data-driven neuroscience analysis and modeling require organizing, accessing, and integrating highly heterogeneous multimodal and multiscale data and knowledge. The scientific meaning, value, and quality of data are critically dependent not only on the data's semantic identity (what they are) but also on their provenance—where they originated, how they were produced, and who produced them. The semantic identity reflects important information about the type of data, their properties and format, and their relationship to other datasets from the brain. Effective open science approaches also require provenance tracking to support reproducibility, accountability, and attribution for the data, algorithms, and scientists involved. Here we describe the challenge of organizing data and knowledge for analysis and approaches to support data-driven neuroscience analysis and modeling.

## Discovering Data with the Minimal Information for Neuroscience Datasets (MINDS)

In order to discover data, they must be labeled using standardized metadata indexed for searching. A common approach to ensuring data discoverability is to define a minimal set of metadata that is easy to provide accurately and should accompany the publication of every dataset. In neuroscience, although an agreed upon standard has not yet been widely adopted, a proposed standard has been created in "Minimal Information for Neuroscience Datasets," or MINDS (Hill, 2016). MINDS includes metadata describing a variety of attributes: specimen or subject details (e.g., species, age, and strain), contributors; brain location (using a standard atlas or brain parcellation ontology), methods (e.g., method types, analysis methods, equipment, parameters, and specific protocols), data category (e.g., EEG, intracellular recordings, magnetic resonance imaging [MRI], or functional MRI), data-file format, and persistent identifiers showing where the data are stored. However, to ensure consistent discoverability, the values of such a minimal metadata set should adhere to either a controlled vocabulary with clearly defined terms or an ontology, discussed next.

## Organizing Data with Ontologies

Ontologies provide a way of formally representing names, properties, and relationships of a set of entities or concepts within a particular domain. Ontology actually comes from two Greek words: *onto* meaning existence or being real, and *logia* meaning science or study. So ontology is actually the science of what exists or is real. In computer science, ontologies are often used to provide data models and controlled vocabularies for data. Linguistic ontologies may be glossaries, dictionaries, controlled vocabularies, or taxonomies.

Ontologies can serve as useful tools for organizing data and expressing knowledge. For one, they are providing a controlled vocabulary for referring to a set of specific entities or concepts. In addition, they can contain the relationships between these entities and define specific properties of these entities. Using ontologies enables a machine-readable representation of a set of concepts, their properties, and their relationships. For example, defining a cell type first requires formalizing a list of essential properties for describing it (e.g. cell morphology, electrophysiological properties, and synaptic connectivity).

## Ontologies for Neocortical Microcircuitry

For the dataset used in the Blue Brain project for the initial Somatosensory cortex model, a large variety of data covers many different aspects. These include electrophysiological data, per-synaptic responses, and gene expression morphological data. These data must be organized by annotating their structure using structured ontologies in order to understand how to build a model from them, what their key entities are, what their relationships are, and which of their properties you are modeling.

## Online Neuroscience Ontology Resources

It is inadvisable to start from scratch when building a terminology or ontology. Fortunately, many groups have established useful starting points for any neuroscience need.

### National Center for Biomedical Ontology

The National Center for Biomedical Ontology (NCBO) is based at Stanford University but includes collaborators from around the world. Its mission is to develop the software and services to apply ontologies to the biological, medical, and clinical sciences. The organization aims to formalize all

NOTES

knowledge and data that are relevant to improving our understanding of human biology and health. The NCBO emphasizes establishing semantic interoperability in order to enable reproducible and valuable queries and inferences across independently developed knowledge resources. It also recommends standard formats and methodologies for ontology development, maintenance, and usage (https://www.bioontology.org). The primary resource NCBO maintains is the BioPortal, which integrates biomedical ontologies from many sources (https://bioportal.bioontology.org).

## Neuroscience Information Framework

The Neuroscience Information Framework (NIF) (https://neuinfo.org) has established an extensive catalog of neuroscience data resources from around the world (Gardner et al., 2008). It has established an extensive infrastructure for data registration, search, and sharing. Using this infrastructure, the organization provides the NIF Discovery Portal—a semantic search engine that uses standard NIF terminologies to enable users to refine their searches. The NIF Registry is a community resource catalog of curated digital resources of value to researchers and students (https://neuinfo.org/Resources). The NIF Data Sharing service is the largest collection of neuroscience data, biomedical resources, and neuroscience ontology on the web (https://neuinfo.org/data/search?q=*&l=#all). The NIF LinkOut Broker provides links between data published with PubMed IDs and the published articles (https://neuinfo.org/about/linkoutbroker). Finally, NIF has far-reaching expertise and experience in building, enhancing, and maintaining ontologies and vocabularies for neuroscientists (https://neuinfo.org/about/nifvocabularies).

## International Neuroinformatics Coordinating Facility

The International Neuroinformatics Coordinating Facility (INCF) is an international nonprofit organization initiated by the OECD Global Science Forum (Bjaalie and Grillner, 2007). Membership is

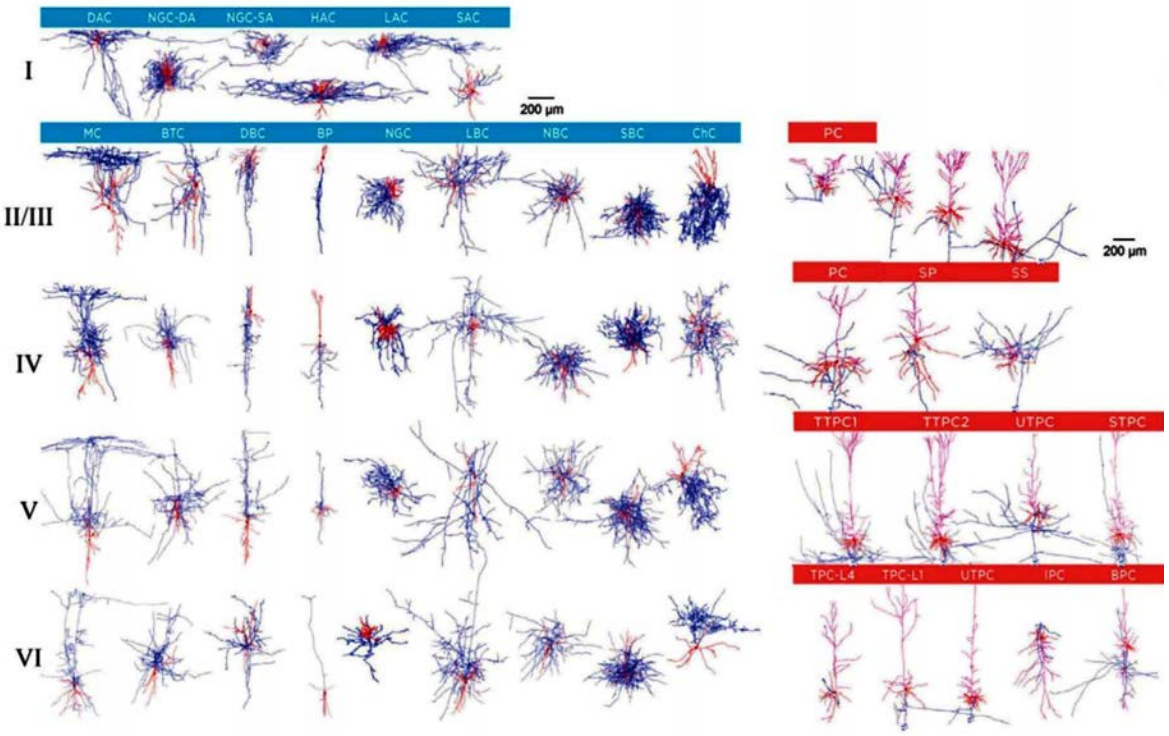


**Figure 1.** 3D reconstruction of 55 m-types of neurons identified from studies of the hindlimb portion of rat primary somatosensory cortex. I, II/III, IV, V, and VI refer to different cortical layers. Axons in blue, dendrites in red. BP, bipolar cell; BPC, pyramidal cell with bipolar apical-like dendrites; BTC, bitufted cell; ChC, chandelier cell; DAC, descending axon cell; DBC, double bouquet cell; HAC, horizontal axon cell; IPC, intermediate neural progenitor cell; LAC, large axon cell; LBC, large basket cell; MC, Martinotti cell; NBC, nest basket cell; NGC-DA, neurogliaform cell–dense axons; NGC-SA, neurogliaform cell–slender axons; PC, pyramidal cell; SAC, small axon cell; SBC, small basket cell; SP, star pyramidal cell; SS, spiny stellate cell; STPC, untufted pyramidal cell; TPC-L1, tufted pyramidal cell, dendrites terminating in L1; TPC-L4, tufted pyramidal cell, dendrites terminating in L4; TTPC1, thick tufted pyramidal cell with a late bifurcating apical tuft; TTPC2, thick tufted pyramidal cell with an early bifurcating apical tuft; UTPC, untufted pyramidal cell. Reprinted with permission from Markram et al. (2015), Fig. 2. Copyright 2015, Elsevier.

NOTES

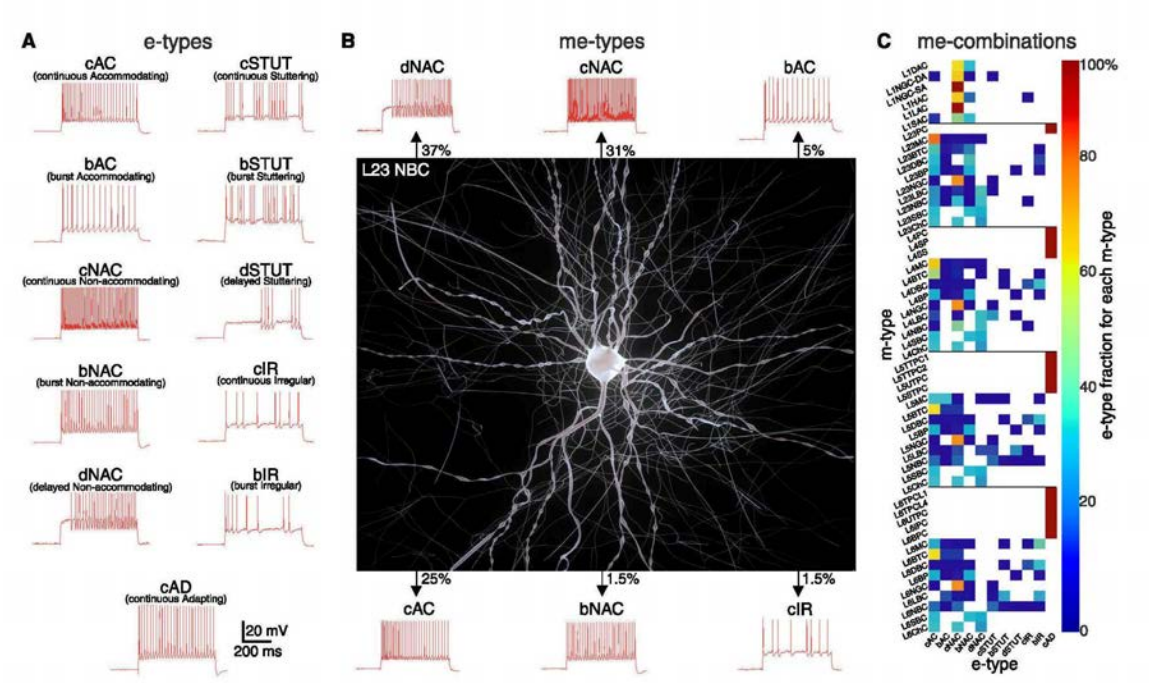


**Figure 2.** *A*, 11 e-types show diverse firing patterns in response to depolarizing step current injections into neocortical neurons. *B,* An exemplar neuron (L23 NBC) with a diversity of 6 e-types. Percentages indicate the relative frequency of e-type occurrence. *C*, Fractions of e-types recorded experimentally in each of the 55 m-types, making up 207 me-combinations. Solid lines, layer boundaries. AC, accommodating; IR, irregular; NAC, non-accommodating; NBC, nest basket cell; STUT, stuttering. *A*, Calibration: vertical, 20 mV; horizontal, 200 ms. Reprinted with permission from Markram et al. (2015), Fig. 4. Copyright 2015, Elsevier.

typically at the national level, with country nodes organizing national activities and international activities coordinated by the INCF secretariat in Stockholm, Sweden. The organization has community-driven interest groups and activities pertaining to data standards, data sharing infrastructure, ontologies, modeling, and brain atlasing (https://www.incf.org).

### KnowledgeSpace

The KnowledgeSpace (https://www.knowledgespace.org) is an open community encyclopedia that links neuroscience concepts to data, models, and literature from around the world. It is the result of a partnership among NIF, INCF, the EU Human Brain Project, and the Blue Brain Project. It provides a search interface to find neuroscience terms (e.g., specific cell types, brain regions, ion channels, and diseases) and uses the NIF infrastructure to search for related data, models, and literature from relevant databases and electronic resources worldwide.

## Neuron Morphology Types (M-Types)

In the Blue Brain Project, we have identified 55 morphological types (m-types) of neocortical neuron (Markram et al., 2015). Figure 1 shows the different types of neurons (for cortical layers I–VI). On the left, you see the inhibitory interneurons, and on the right are the excitatory parameter cells and stellate cells, each with a unique name. At the top of the column are listed the cell-type abbreviations (e.g., in layer I are found neurogliaform cells [NGCs], and in layer V are found Martinotti cells [MCs] and thick tuft pyramidal cells [TTPCs]). These names are used to characterize the properties of neuron morphologies and are useful for organizing and being consistent about how you refer to different types of neurons within a particular brain region.

## Electrical Firing Types (E-Types)

The Blue Brain ontology for electrical firing type (e-type) neurons is rooted in the convention adopted for the Petilla classification for firing properties of interneurons (Petilla Interneuron Nomenclature Group et al., 2008). This classification is based on features of the firing response of a neuron in response to a whole-cell patch-clamp current injection into the soma. This classification can be performed based on the neuron firing response only in an *in vitro* condition; in *in vivo* conditions, the neuron may exhibit significantly different firing behavior.

NOTES

The e-type classification comprises the initial firing properties followed by the sustained firing properties. The initial firing type is therefore either burst (b), continuous (c), or delay (d), which describes the initial response of the cell to a depolarizing current. In some traces, one can see where there may be a brief burst at the beginning, an immediate firing response, or perhaps a brief delay in which no spikes occur. The initial firing type captures these properties.

Similarly, for the sustained firing type, some cells adapt when their interspike interval decreases, whereas others are nonadapting when their interspike interval remains constant. Still others are considered fast-spiking, exhibiting deeper hyperpolarizations after a spike and firing at higher frequencies. Yet other cells show irregular spiking in which their interspike interval may be quite variable.

These properties have been identified as definitive for interneuron firing characteristics and form the basis of the e-type classification. Table 1 relates these interneuron classes to other cell properties and types described in the literature. For example, there are many ways to talk about a cell type and many dimensions along which one can measure a neuron to characterize it. These may include, for example, genes that are expressed or proteins that are present in the cell as molecular markers. Electrical and morphological properties, as discussed, form additional measurement dimensions. Notably, synaptic projection targets and synaptic types may also help to define cell type. Here we focus on characterizing cell types based on their morphological and electrophysiological (morpho-electric) properties.

## Morpho-electric Combinations (Me-Combinations)

The morpho-electric cell-type (me-type) ontology is actually based on the combinations of e-types that occur with m-types. Thus, these "me-combinations" are made up of the two ontologies we talked about earlier. Not all combinations of morphologies and electrical types occur; rather, only a subset of these yields 207 me-combinations. We have compiled data on how many of these occur and with what probability (Fig. 2*c*).

## Synapse Types

The connections among cell types are defined by different categories synapses and the specific properties that govern them. There are excitatory and inhibitory synapses, with each type of synapse mediated by multiple neurotransmitters (e.g.,

**Table 1. Relation of interneuron classes to classification schemes found in the literature.**

| Morphological Type | Neurogliaform Cell (NGC) | Small Basket Cell (SBC) | Double Bouquet Cell (DBC) | Bipolar Cell (BP) | Martinotti Cell (MC) | Bitufted Cell (BTC) | Large Basket Cell (LBC) | Nest Basket Cell (NBC) | Chandelier Cell (ChC) |
|---|---|---|---|---|---|---|---|---|---|
| Other morphological classifications | Dwarf cell, button-type cell | Clutch cell | Bitufted cell/ interneuron, horse-tail cell | Bitufted cell/ interneuron | Bitufted cell/ interneuron | Bitufted interneuron | Common basket cell, typical basket cell | Willow cell, arcade cell, shaft-biased cell, atypical basket cell | Axo-axonic cell |
| Predominantly expressed $Ca^{2+}$-binding proteins and peptides | CB (–), NPY (+), PV (–), VIP (–), CR (–) SOM (–) | CB (++), NPY (+), PV (–), VIP (+++), CR (–) SOM (++) | CB (+), NPY (–), PV (–), VIP (+++), CR (+) SOM (++) | CB (–), NPY (–), PV (–), VIP (+++), CR (++) SOM (++) | CB (++), NPY (++), PV (–), VIP (–), CR (–) SOM (+++) | CB (++), NPY (+), PV (–), VIP (+), CR (++) SOM (++) | CB (++), NPY (+), PV (+++), VIP (+), CR (+) SOM (–) | CB (++), NPY (+), PV (+++), VIP (+), CR (++) SOM (–) | CB (+), NPY (–), PV (–), VIP (–), CR (–) SOM (–) |
| Electrical types | bNAC (7%), cNAC (79%), cSTUT (7%), cAC (7%) | bNAC (36%), cAC (36%), dNAC (29%) | bAC (9%), bIR (37%), bNAC (9%), bSTUT (9%), cAC (9%), cIR (18%), cNAC (9%) | bAC (7%), bIR (14%), bNAC (29%), cAC (29%), cNAC (14%), dSTUT (7%) | bAC (37%), bIR (11%), bSTUT (4%), cAC (37%), cNAC (3%), cSTUT (3%), dNAC (3%) | bAC (17%), cAC (67%), cNAC (17%) | bAC (6%), cAC (12%), cIR (6%), cNAC (17%), dNAC (17%), dSTUT (24%) | bAC (6%), bIR (6%), bSTUT (13%), cAC (20%), cIR (7%), cNAC (20%), cSTUT (20%), dSTUT (7%) | cAC (38%), cNAC (38%), dNAC (25%) |
| Other electrical classifications | Non-fast spiking, late spiking | Fast spiking, non-accommodating, non-adapting | Irregular spiking, regular spiking non-pyramidal, adapting | Late spiking, regular spiking non-pyramidal, adapting | Regular spiking non-pyramidal, burst spiking non-pyramidal, low threshold spiking | Regular spiking non-pyramidal, adapting, burst spiking non-pyramidal | Fast spiking, non-accommodating, non-adapting | Fast spiking, non-accommodating, non-adapting | Fast spiking, late spiking, non-adapting |

Interneurons can be categorized according to which primary marker they express (calcium-binding proteins: parvalbumin [PV], calbindin [CB], and calretinin [CR]; neuropeptides: somatostatin [SOM], vasoactive intestinal polypeptide [VIP], neuropeptide Y [NPY], and cholecystokinin [CCK]). We assign several possible e-types to each m-type based on the Petilla convention (Petilla Interneuron Nomenclature Group et al., 2008). AC, accommodating; IR, irregular; NAC, non-accommodating; STUT, stuttering. Reprinted with permission from Markram et al. (2015), Table 1. Copyright 2015, Elsevier.

NOTES

AMPA, NMDA, $GABA_A$, and $GABA_B$). An important defining characteristic of these synapses is the dynamics of their short-term plasticity in relation to depression and facilitation: the way in which synapses become "stronger" or "weaker" with increased presynaptic firing. As a typical example, depressing synapses produce less quantal release (packets of neurotransmitters) with each subsequent presynaptic release as the available pool of readily releasable vesicles decreases. This characteristic, combined with postsynaptic receptor saturation, results in different short-term plasticity profiles for synapses (Fig. 3). Therefore, synapse types are named after two key properties: the sign of the synapse (excitatory or inhibitory) and the profile of the dynamics (facilitating, depressing, or pseudo-linear). Which synapse type occurs between different me-combinations has been mapped and depends on the presynaptic and postsynaptic me-combinations (Markram et al., 2015).

## Microcircuitry

To define the properties of a microcircuit, we need to combine many of the elements we have described thus far. We start by defining the volumetric boundaries of the microcircuit in the atlas space. The cell densities are described as the number of cells of a given cell type (me-combination) per unit volume. We used this approach to describe a somatosensory cortex microcircuit and observed that, in general, these same properties can be used to describe microcircuits throughout the brain. The minimal boundary sufficient to create a complete microcircuit was defined according to the volume boundary that would enclose a sufficient number of cells (at the correct cell density) to saturate the dendritic density (Markram et al., 2015). The composition recipe gives the relative density of all m-types, e-types, and me-combinations. In addition, the total number of cells within the volume provides the upper constraint on cell density. The map of synaptic types, described earlier, provides further key properties of the microcircuit as it dictates the synapse types that mediate interactions between each me-combination. Finally, a type-specific average bouton density is critical to making a functional conversion to the actual number of synapses (Reimann et al., 2015).

### Semantic data integration

Semantic metadata enable the integration of data in a reproducible and reusable way once the ontology has encoded the relationships and similarity of concepts. For example, similar or related m-types or e-types are categorized in common branches of the ontology. As our knowledge advances and the ontologies are revised, the new version of the ontology may result in different, similar, or related concepts. This new version can change the result of the integration to reflect current knowledge. However, because ontologies are versioned, one can revert to any version as needed to reproduce any analyses.

### Organizing data with brain atlases

A brain region ontology may require a clear spatial boundary, naming, and other properties that capture the relationships between nearby or related brain regions. The Allen Mouse Brain Atlas (a project of the Allen Brain Atlas, http://www.brain-map.org) was built using the Swanson taxonomy to capture the structures within the cortex, the substructures, and the relationship of, for example, the cortex to the diencephalon. In addition, each brain region is linked to three-dimensional (3D) volumetric boundaries in the Allen Mouse Common Coordinate Framework (Oh et al., 2014). Thus, a brain atlas ontology can both use data structures for describing the 3D boundary within the standard space as well as linking these regions to a standard name and abbreviation.

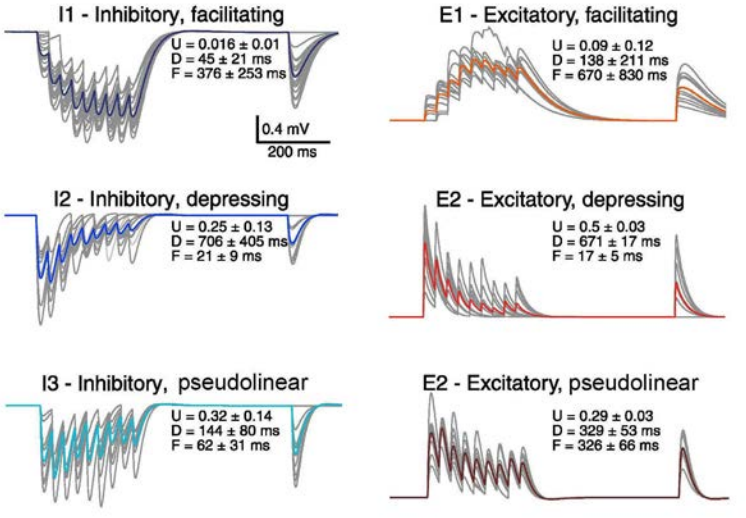


**Figure 3.** The different types of short-term plasticity profiles for cortical synapses. Calibration: vertical, 0.4 mV; horizontal, 200 ms. Reprinted with permission from Markram et al. (2015), Fig. 9*B*. Copyright 2015, Elsevier.

NOTES

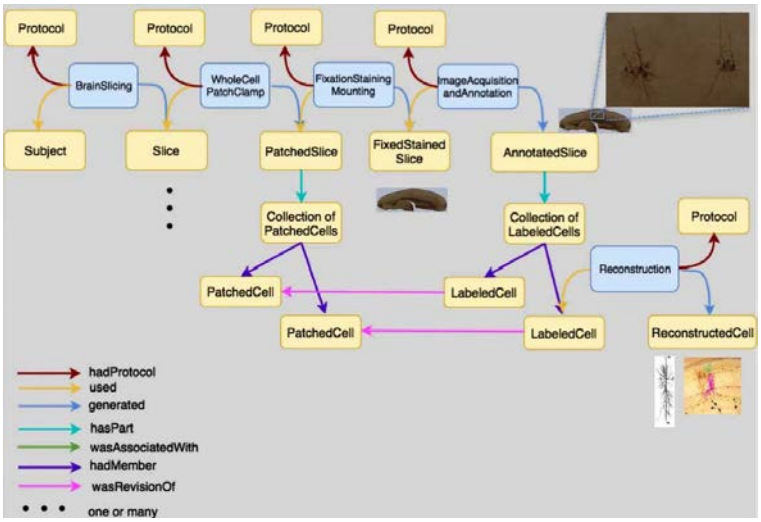


**Figure 4.** The provenance data model for *in vitro*–labeled and reconstructed neuron morphologies. This provenance graph includes entities that represent all the protocols, artifacts, and people involved in generating a reconstructed neuron morphology. These metadata enable the accurate scientific reuse of the reconstructed cell.

## Describing data with provenance

The World Wide Web Consortium (W3C) has developed a standard for describing the provenance of any type of data with an extensible data model and ontology called PROV (https://www.w3.org/TR/prov-overview). An example of a neuroscience-relevant data model developed using PROV is the Neuroimaging Data Model, which supports tracking the provenance of neuroimaging data and generating reproducible analysis results (Maumet et al., 2016). Rich descriptions of the provenance of datasets and models are essential for discoverability, reproducibility, reuse, quality assessment, and attribution. Figure 4 shows an example of another domain-specific provenance data model, developed to describe neuron morphologies.

Other data models are in development to support additional neuroscience datasets, including electrophysiology, brain atlases, and computational models. They reuse or extend community-defined schemas (e.g., Schema.org or Bioschemas.org) and ontologies (e.g., brain parcellation schemes, cell types, taxonomies), providing the basis for validating all submissions according to these schema and their related ontologies. This system enables neuroscientists and modelers to discover data, models, and literature according to metadata properties such as specimen, protocol, brain region, data type, and cell type and makes it possible to discover datasets via semantic relationships (data produced using similar protocols, common brain regions, similar types of data or models, and related cell types). Datasets can also be discovered via provenance relationships (i.e., which datasets originated from the same animal, from the same cell, or from the same laboratory, or which models were built from specific data). The provenance metadata support reproducibility by capturing aspects of the experimental design as well as the protocols, solutions, software, and other tools that were used to generate the dataset or model. Important in the context of team science, such provenance information enables researchers to rapidly summarize the full attribution of datasets (including institution, principal investigator, postdocs, technicians, students, analysts, and curators).

## Conclusion

Organizing data to support data-driven modeling and analysis requires systematic naming and labeling of the data. Standardized names and ontologies that capture the key relationships among classes of entities are essential for reusable querying and computation. Numerous online resources and organizations provide useful starting points and support for developing new vocabularies and ontologies. These ontologies, along with standardized brain atlases, provide useful mechanisms for reproducible data integration of diverse neuroscience data. Provenance-based data models are essential for enabling the description of the experimental context that generated experimental data artifacts as well as the protocols, methods, software, and scientists or engineers that produced them. All these elements are key to discovering, organizing, and integrating heterogeneous data for reproducible analysis and modeling in neuroscience.

## Acknowledgments

Figures and tables for this chapter were excerpted with permission from Markram, Henry, et al. (2015) Reconstruction and simulation of neocortical microcircuitry. Cell 163.2: 456-492. Copyright 2015, Elsevier.

## References


Akil H, Martone ME, Van Essen DC (2011) Challenges and opportunities in mining neuroscience data. Science 331:708–712.

Bjaalie JG, Grillner S (2007) Global neuroinformatics: the International Neuroinformatics Coordinating Facility. J Neurosci 27:3613–3615.

Gardner D, Akil H, Ascoli GA, Bowden DM, Bug W, Donohue DE, Goldberg DH, Grafstein B, Grethe JS, Gupta A, Halavi M, Kennedy DN, Marenco L, Martone ME, Miller PL, Müller HM, Robert A, Shepherd GM, Sternberg PW, Van Essen DC, et al. (2008) The neuroscience information framework: a data and knowledge environment for neuroscience. Neuroinformatics 6:149–160.

Hill SL (2016) How do we know what we know? Discovering neuroscience data sets through minimal metadata. Nat Rev Neurosci 17:735–736.

Maumet C, Auer T, Bowring A, Chen G, Das S, Flandin G, Ghosh S, Glatard T, Gorgolewski KJ, Helmer KG, Jenkinson M, Keator DB, Nichols BN, Poline JB, Reynolds R, Sochat V, Turner J, Nichols TE (2016) Sharing brain mapping statistical results with the neuroimaging data model. Sci Data 3:160102.

Markram H, Muller E, Ramaswamy S, Reimann MW, Abdellah M, Sanchez CA, Ailamaki A, Alonso-Nanclares L, Antille N, Arsever S, Kahou GA, Berger TK, Bilgili A, Buncic N, Chalimourda A, Chindemi G, Courcol JD, Delalondre F, Delattre V, Druckmann S, et al. (2015) Reconstruction and simulation of neocortical microcircuitry. Cell 163:456–492.

Oh SW, Harris JA, Ng L, Winslow B, Cain N, Mihalas S, Wang Q, Lau C, Kuan L, Henry AM, Mortrud MT, Ouellette B, Nguyen TN, Sorensen SA, Slaughterbeck CR, Wakeman W, Li Y, Feng D, Ho A, Nicholas E, et al. (2014) A mesoscale connectome of the mouse brain. Nature 508:207–214.

Petilla Interneuron Nomenclature Group, Ascoli GA, Alonso-Nanclares L, Anderson SA, Barrionuevo G, Benavides-Piccione R, Burkhalter A, Buzsáki G, Cauli B, Defelipe J, Fairén A, Feldmeyer D, Fishell G, Fregnac Y, Freund TF, Gardner D, Gardner EP, Goldberg JH, Helmstaedter M, Hestrin S, et al. (2008) Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex. Nat Rev Neurosci 9:557–568.

Reimann MW, King JG, Muller EB, Ramaswamy S, Markram H (2015) An algorithm to predict the connectome of neural microcircuits. Front Comput Neurosci 9:120.


NOTES

# Fast and Accurate Spike Sorting of High-Channel Count Probes with KiloSort

Marius Pachitariu, PhD, Nicholas Steinmetz, PhD, Shabnam Kadir, PhD, Matteo Carandini, PhD, and Kenneth D. Harris, PhD

Institute of Neurology
University College London
London, United Kingdom

NOTES

## Introduction

New silicon technology is enabling large-scale electrophysiological recordings *in vivo* from hundreds to thousands of channels. Interpreting these recordings requires scalable and accurate automated methods for spike sorting, which should minimize the time required for manual curation of the results. Here we introduce KiloSort, a new integrated spike sorting framework that uses template matching both during spike detection and during spike clustering. KiloSort models the electrical voltage as a sum of template waveforms triggered on the spike times, which allows overlapping spikes to be identified and resolved. Unlike previous algorithms that compress the data using principal component analysis (PCA), KiloSort operates on the raw data, allowing it to construct a more accurate model of the waveforms. Processing times are faster than in previous algorithms thanks to batch-based optimization on graphics processing units (GPUs). We compare KiloSort to an established algorithm and show favorable performance, at much reduced processing times. A novel post-clustering merging step based on the continuity of the templates further substantially reduced the number of manual operations required on these data, for the neurons with near-zero error rates, paving the way for fully automated spike sorting of multichannel electrode recordings.

The oldest and most reliable method for recording neural activity involves lowering an electrode into the brain and recording the local electrical activity around the electrode tip. Action potentials of single neurons can then be observed as a stereotypical temporal deflection of the voltage, called a spike waveform. When multiple neurons close to the electrode fire action potentials, their spikes must be identified and assigned to the correct cell based on the features of the recorded waveforms, a process known as spike sorting (Harris et al., 2000; Hill et al., 2011; Einevoll et al., 2012; Quiroga, 2012; Pillow et al., 2013; Ekanadham et al., 2014; Franke et al., 2015). Spike sorting is substantially helped by the ability to simultaneously measure the voltage at multiple closely spaced sites in the extracellular medium. In this case, the recorded waveforms can be seen to have characteristic spatial shapes determined by each cell's location and physiological characteristics. Together, the spatial and temporal shape of the waveform provides all the information that can be used to assign a given spike to a cell.

New high-density electrodes, currently being tested, can record from several hundred closely spaced recording sites. Fast algorithms are necessary to quickly and accurately spike sort tens of millions of spikes coming from 100 to 1000 cells from recordings performed with such next-generation electrodes in awake, behaving animals. Here we present a new algorithm that provides accurate spike sorting results with run times that scale near-linearly with the number of recording channels. The algorithm takes advantage of the computing capabilities of low-cost, commercially available GPUs to enable approximately real-time spike sorting from 384-channel probes.

### High-density electrophysiology and structured sources of noise

Next-generation high-density neural probes allow the spikes of most neurons to be recorded on 5 to 50 channels simultaneously (Fig. 1*b*). This provides a substantial amount of information per spike, but because other neurons also fire on the same channels, a clustering algorithm is still required to demix the signals and assign spikes to the correct cluster. Although the dense spacing of channels provides a large amount of information for each spike, structured sources of noise can still negatively impact the spike sorting problem. For example, the superimposed waveforms of neurons distant from the electrode (nonsortable units) add up and constitute a continuous random background (Fig. 1*a*) against which the features of sortable spikes (Fig. 1*b*) must be distinguished. In behaving animals, another major confound is given by the movement of the electrode relative to the tissue, which creates an apparent inverse movement of the waveform along the channels of the probe (Fig. 1c).

### Previous work

A traditional approach to spike sorting divides the problem into several stages. In the first stage, spikes are detected that have maximum amplitudes above a predefined threshold, and these spikes are projected into a common low-dimensional space, typically obtained by PCA. In the second stage, the spikes are clustered in this low-dimensional space using a variety of approaches, such as mixtures of Gaussians (Rossant et al., 2016) or peak-density-based approaches (Rodriguez and Laio, 2014). Some newer algorithms also include a third stage of template matching in which overlapping spikes are found in the raw data that may have been missed in the first detection phase. Finally, a manual stage in a GUI is required for awake recordings, to manually perform merge and split operations on the imperfect automated results.

NOTES

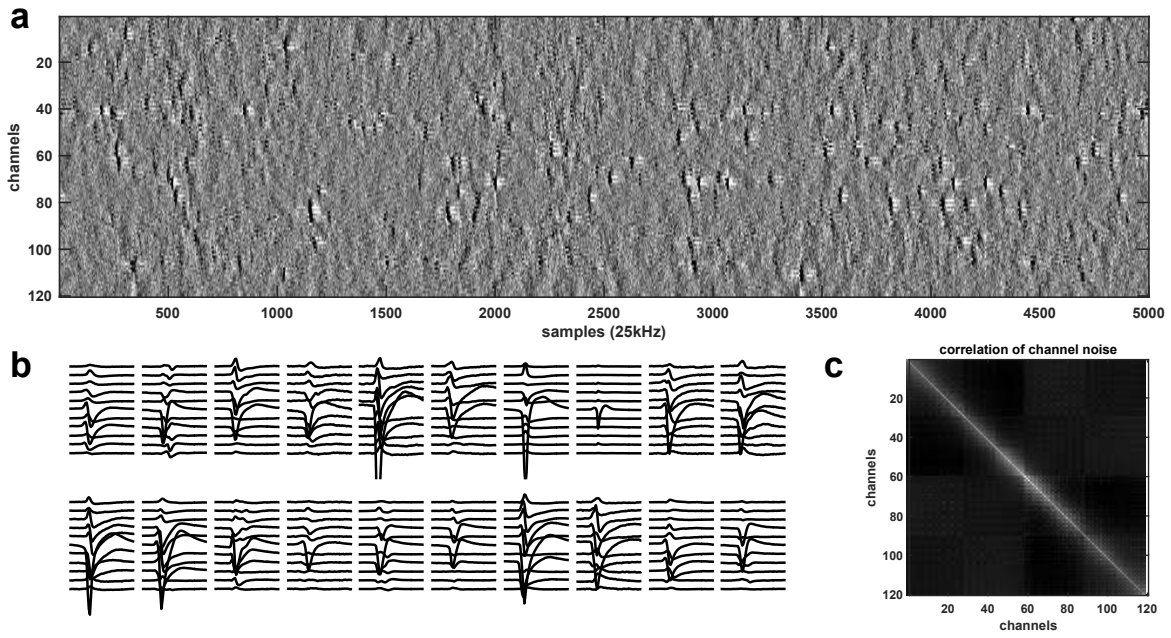


**Figure 1.** Data from high-channel count recordings. ***a***, High-pass filtered and channel-whitened data. Negative peaks are action potentials. ***b***, Example mean waveforms, centered on their peaks. ***c***, Example cross-correlation matrix across channels (before whitening).

Here, we instead combine these steps into a single model with a cost function based on the error of reconstructing the entire raw voltage dataset with the templates of a set of candidate neurons. We derive approximate inference and learning algorithms that can be successfully applied to very large channel count data. This approach is related to a previous study (Ekanadham et al., 2014), but whereas the previous work scale was impractically slow for recordings with large numbers of channels, our further modeling and algorithmic innovations have enabled the approach to be used quickly and accurately on real datasets. We improved on the generative model of Ekanadham and colleagues: from a spiking process with continuous L1-penalized traces, to a model of spikes as discrete temporal events. The approach of Ekanadham et al. (2014) does not scale well to high-channel count probes because it requires the solution of a generic convex optimization problem in many dimensions.

## Model Formulation

We start with a generative model of the raw electrical voltage. Unlike previous approaches, we do not precommit to the times of the spikes, nor do we project the waveforms of the spikes to a lower-dimensional PCA space. Both these steps discard potentially useful information, as we show below.

### Preprocessing: common average referencing, temporal filtering, and spatial whitening

To remove low-frequency fluctuations, such as the local field potential (LFP), we high-pass filter each channel of the raw data at 300 Hz. To diminish the effect of artifacts shared across all channels, we subtract at each timepoint the median of the signal across all recording sites, an operation known as "common average referencing." This step is best performed after high-pass filtering because the LFP magnitude is variable across channels but can be comparable in size to the artifacts.

Finally, we whiten the data in space to remove noise that is correlated across channels (Fig. 1*c*). The correlated noise is caused mostly by far neurons with small spikes (Neto et al., 2016), which have a large spatial spread over the surface of the probe. Because very many such neurons are found at all recording sites, their noise averages out to have normal statistics with a stereotypical cross-correlation pattern across channels (Fig. 1*c*). We distinguish the noise covariance from that of the large, sortable spikes by removing the times of putative spikes (detected with a threshold criterion) from the calculation of the covariance matrix. We use a symmetrical whitening matrix that maintains the spatial structure of the data,

known as ZCA, defined as $W_{ZCA} = \Sigma^{-½} = ED^{-½} E^T$, where $E$, $D$ are the singular vectors and singular values of the estimated covariance matrix $\Sigma$. To regularize $D$, we add a small value to its diagonal. For very large channel counts, estimation of the full covariance matrix $\Sigma$ is noisy, and we therefore compute the columns of the whitening matrix $W_{ZCA}$ independently for each channel, based on its nearest 32 channels.

## Modeling mean spike waveforms with singular value decomposition

When single spike waveforms are recorded across a large number of channels, most channels will have no signal and only noise. To prevent these channels from biasing the spike sorting problem, previous approaches estimated a mask over those channels with sufficient signal-to-noise ratio to be included in a given spike. To further reduce noise and lower the dimensionality of the data for computational reasons, the spikes are usually projected into a small number of temporal principal components (PCs) per channel (typically three). Here we suggest a different method for simultaneous spatial denoising/masking and for lowering the dimensionality of spikes. This method is based on the observation that mean spike waveforms are very well explained by a singular value decomposition (SVD) of their spatiotemporal waveform, with as few as three components (Figs. 2*a*,*b*). However the spatial and temporal components of the SVD vary substantially from neuron to neuron; hence, the same set of temporal basis functions per channel cannot be used to model all neurons (Figs. 2*a*,*b*), as typically done in standard approaches. We analyzed the ability of the classical and proposed methods for dimensionality reduction and found that the proposed decomposition can reconstruct waveforms with approximately five times less residual variance than the classical approach. This allows the decomposition to capture small but highly distinguishable features of the spikes, which ultimately can help distinguish among neurons with very similar waveforms.

## Integrated template matching framework

To define a generative model of the electrical recorded voltage, we take advantage of the approximately linear additivity of electrical potentials from different sources in the extracellular medium. We combine the spike times of all neurons into an $N_{spikes}$-dimensional vector $\mathbf{s}$, such that the waveforms start at time samples $\mathbf{s} + 1$. We define the cluster identity of spike $k$ as $\sigma(k)$, taking values into the set {1, 2, 3, ..., $N$}, where $N$ is the total number of neurons. We define the unit-norm waveform of neuron $n$ as the matrix $K_n = U_n W_n$, of size number of channels by number of sample timepoints $t_s$ (typically 61). The matrix $K_n$ is defined by its low-dimensional deconstruction into three pairs of spatial and temporal basis functions, $U_n$ and $W_n$, such that the norm of $U_n W_n$ is 1. The value of the electrical voltage at time $t$ on channel $i$ is defined by

$$\begin{aligned} V(i,t) &= V_0(i,t) + N(0, \varepsilon) \\ V_0(i,t) &= \sum_{k, s(k) < t}^{s(k) \geq t - t_s} x_k K_{\sigma(k)}(i, t - s(k)) \\ x_k &\sim N\left(\mu_{\sigma(k)}, \lambda \mu^2_{\sigma(k)}\right), \end{aligned} \tag{1}$$

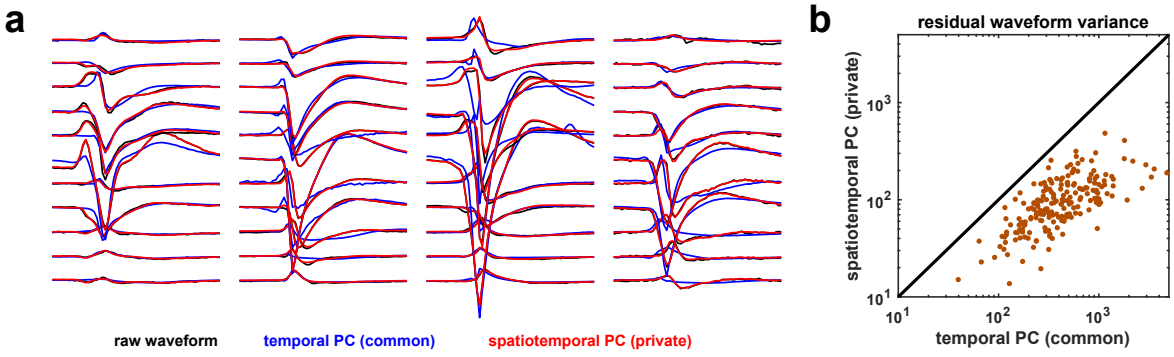


**Figure 2.** Spike reconstruction from three private PCs. ***a***, Four example average waveforms (black) with their respective reconstruction with three common temporal PCs/channel (blue) and with reconstruction based on three spatiotemporal PCs (red), private to each spike. The red traces mostly overlap the black traces. ***b***, Summary of residual waveform variance for all neurons in one dataset.

where $xk > 0$ is the amplitude of spike $k$. Spike amplitudes in the data can vary significantly even for spikes from the same neuron, due to factors like burst adaptation and drift. We modeled the mean and variance of the amplitude variability, with the variance of the distribution scaling with the square of the mean. $\lambda$ and $E$ are hyperparameters that control the relative scaling with respect to each other of the reconstruction error and the prior on the amplitude. In practice, we set these constant for all recordings.

This model formulation leads to the following cost function, which we minimize with respect to the spike times, cluster assignments, amplitudes and templates

$$\mathscr{L}(\mathrm{s},\mathrm{x},K,\sigma) = \left\| V - V_0 \right\|^2 + \frac{\varepsilon}{\lambda} \sum_k \left( \frac{x_k}{\mu_{\sigma k}} - 1 \right)^2 \quad (2)$$

## Learning and Inference in the Model

To optimize the cost function, we alternate between finding the best spike times **s**, cluster assignments $\sigma$, and amplitudes **x** (template matching) and optimizing the template $K$ parametrization with respect to **s**, $\sigma$, **x** (template optimization). We initialize the templates using a simple scaled K-means clustering model, which we in turn initialize with prototypical spikes determined from the data. After the final spike times and amplitudes have been extracted, we run a final post-optimization merging algorithm that finds pairs of clusters whose spikes form a single continuous density. These steps are separately described in detail below.

### Stacked initializations with scaled K-means and prototypical spikes

The density of spikes can vary substantially across the probe, depending on the location of each recording site in the brain. Thus, initializing the optimization in a density-dependent way can assign more clusters to regions that require more, relieving the main optimization from the local minima–prone problem of moving templates from one part of the probe to another. For the initialization, we therefore start by detecting spikes using a threshold rule, and as we load more of the recording, we keep a running subset of prototypical spikes that are sufficiently different from each other by an L2 norm criterion. We avoid overlapping spikes to be counted as prototypical spikes by enforcing a minimum spatiotemporal peak isolation criterion on the detected spikes. Of the prototypical spikes thus detected, we consider a fixed number $N$ that had the most matches to other spikes in the recording.

We then use this initial set of spikes to initialize a scaled K-means algorithm. This algorithm uses the same cost function described in Equation 2, with spike times s fixed to those found by a threshold criterion. Unlike standard K-means, each spike is allowed to have variable amplitude (Coates et al., 2011).

### Learning the templates via stochastic batch optimization

The main optimization reestimates the spike times **s** at each iteration. The "online" nature of the optimization helps to accelerate the algorithm and to avoid local minima. For template optimization, we use a simple running average update rule

$$A_n^{\mathrm{new}}(i, t_0) \leftarrow (1-p)^{j_n} A_n^{\mathrm{old}}(i, t_0) + (1-(1-p)^{j_n}) \sum_{k \in \mathrm{batch}}^{\sigma(k)=n} V(i, s(k)+t_0), \quad (3)$$

where $A_n$ is the running average waveform for cluster $n$, $j_n$ represents the number of spikes from cluster $n$ identified in the current batch, and the running average weighs past samples exponentially with a forgetting constant $p$. Thus, $A_n$ approximately represents the average of the past $p$ samples assigned to cluster $n$. Note that different clusters will therefore update their mean waveforms at different rates, depending on their number of spikes per batch. Since firing rates vary over two orders of magnitude in typical recordings (from < 0.5 to 50 spikes/s), the adaptive running average procedure allows clusters with rare spikes to nonetheless average enough of their spikes to generate a smooth average template.

Like most clustering algorithms, the model we developed here is prone to nonoptimal local minima. We use several techniques to ameliorate this problem. First, we anneal several parameters during learning to encourage exploration of the parameter space, which stems from the randomness induced by the stochastic batches. We anneal the forgetting constant $p$ from a small value (typically 20) at the beginning of the optimization to a large value at the end (typically several hundred). We also anneal from small to large the ratio $\varepsilon/\lambda$, which controls the relative impact of the reconstruction term and amplitude bias term in Equation 2. Therefore, at the beginning of the optimization, spikes assigned to the same cluster are allowed to have more variable amplitudes. Finally, we anneal the threshold for spike detection (see below) to allow a greater mismatch between spikes and the available templates at the beginning of the optimization. As optimization progresses, the templates become more precise, and spikes increase

NOTES

their projections onto their preferred template, thereby allowing higher thresholds to separate them from the noise.

## Inferring spike times and amplitudes via template matching

The inference step of the proposed model attempts to find the best spike times, cluster assignments, and amplitudes, given a set of templates $\{K_n\}_n$ with low-rank decompositions $K_n = U_n W_n$ and mean amplitudes $\mu n$. The templates are obtained from the running average waveform $A_n$, after an SVD decomposition to give $A_n \sim \mu_n K_n = \mu_n U_n W_n$, with $\|U_n W_n\| = 1$, with $U_n$ orthonormal and $W_n$ orthogonal. The primary roles of the low-rank representation are to guarantee fast inferences and to regularize the waveform model.

We adopt a parallelized matching pursuit algorithm to iteratively estimate the best fitting templates and subtract them off from the raw data. In standard matching pursuit, the best fitting template is identified over the entire batch, its best reconstruction is subtracted from the raw data, and then the next best fitting template is identified iteratively until the amount of explained variance falls below a threshold, which constitutes the stopping criterion. To find the best fitting template, we estimate for each time $t$ and each template $n$ the decrease in the cost function obtained by introducing template $n$ at location $t$, with the best fitting amplitude $x$. This is equivalent to minimizing a standard quadratic function of the form $ax^2 - 2bx + c$ over the scalar variable $x$, with $a$, $-2b$, and $c$ derived as the coefficients of $x^2$, $x$, and 1 from Equation 2

$$a = 1 + \frac{\varepsilon}{\lambda \mu_n^2}; \; b = (K_n \star V)(t) + \frac{\varepsilon}{\lambda \mu_n}; \; c = \lambda \mu_n^2 \qquad (4)$$

where $\star$ represents the operation of temporal filtering (convolution with the time-reversed filter). Here the filtering is understood as channel-wise filtering followed by a summation of all filtered traces, which computes the dot product between the template and the voltage snippet starting at each timepoint $t$. The decrease in cost $dC(n, t)$ that would occur if a spike of neuron $n$ were added at time $t$, and the best $x$ are given by

$$x_{\text{best}} = \frac{b}{a}$$
$$dC(n,t) = \frac{b^2}{a} - c \qquad (5)$$

Computing $b$ requires filtering the data $V$ with all the templates $K_n$, which amounts to a very large number of operations, particularly when the data have many channels. However, our low-rank decomposition allows us to reduce the number of operations by a factor of $N_{\text{chan}}/N_{\text{rank}}$, where $N_{\text{chan}}$ is the number of channels (typically >100) and $N_{\text{rank}}$ is the rank of the decomposed template (typically 3). This follows from the observation that

$$V \star K_n = V \star (U_n W_n) = \sum_j (U_n(:, j)^T \cdot V) \star W_n(j, :), \qquad (6)$$

where $U_n(:, j)$ is understood as the $j$-th column of matrix $U_n$, and similarly $W_n(j, :)$ is the $j$-th row of $W_n$. We have thus replaced the matrix convolution $V \star K_n$ with a matrix product $U_n^T V$ and $N_{\text{rank}}$ one-dimensional convolutions. We implemented the matrix products and filtering operations efficiently using consumer GPU hardware. Iterative updates of $dC$ after template subtraction can be obtained quickly using precomputed cross-template products, as typically done in matching pursuit. The iterative optimization stops when a predefined threshold criterion on $dC$ is larger than all elements of $dC$.

Owing to its greedy nature, matching pursuit can perform badly at reducing the cost function in certain problems. It is, however, appropriate to our problem because spikes are very rare events, and overlaps are typically small—particularly in high dimensions over the entire probe. Further, typical datasets contain millions of spikes, and only the simple form of matching pursuit can be efficiently employed. We implemented the simple matching pursuit formulation efficiently on consumer GPU hardware. Consider the cost improvement matrix $dC(n, t)$. When the largest element of this matrix is found and the template subtracted, no values of $dC$ need to change except those very close in time to the fitted template ($t_s$ samples away). Thus, instead of finding the global maximum of $dC$, we can find local maxima above the threshold criterion and impose a minimal distance ($t_s$) between such local maxima. The identified spikes can then be processed in parallel without affecting each other's representations.

We found it unnecessary to iterate the (relatively expensive) parallel matching pursuit algorithm during the optimization of the templates. We obtained similar templates when we aborted the parallel matching pursuit after the first parallel detection step, without detecting any further overlapping spikes. To improve the efficiency of the optimization, we therefore applied the full parallel template matching algorithm only on the final pass, thus obtaining the overlapping spikes.

# Benchmarks

First, we timed the algorithm on several large-scale datasets. The average run times for 32-, 128-, and

384-channel recordings were 10, 29, and 140 min, respectively, on a single GPU-equipped workstation. These were significant improvements over an established framework called KlustaKwik (Rossant et al., 2016), which needed approximately 480 channel recordings and 10,000–20,000 min when run on 32- and 128-channel datasets on a standard CPU cluster (we did not attempt to run KlustaKwik on 384-channel recordings).

The significant improvements in speed could have come at the expense of accuracy losses. We compared KiloSort and KlustaKwik on 32- and 128-channel recordings using a technique known as "hybrid ground truth" (Rossant et al., 2016). To create these data, we first selected all the clusters from a recording that had been previously analyzed with KlustaKwik and curated by a human expert. For each cluster, we extracted its raw waveform and denoised it with an SVD decomposition (keeping the top seven dimensions of variability). We then added the denoised waveforms at a different but nearby spatial location on the probe with a constant channel shift, randomly chosen for each neuron. To avoid increasing the spike density at any location on the probe, we also subtracted from the denoised waveform from its original location.

Finally, we ran both KiloSort and KlustaKwik on 16 instantiations of the hybrid ground truth. We matched ground truth cells with clusters identified by the algorithms to find the maximizer of the score = 1 − false-positive rate − miss rate, where the false-positive rate was normalized by the number of spikes in the test cluster, and the miss rate was normalized by the number of spikes in the ground truth cluster. Values close to 1 indicate well-sorted units. Both KiloSort and KlustaKwik performed well, though KiloSort produced significantly more cells with well-isolated clusters (53% vs 35% units with scores > 0.9).

We also estimated the best achievable score following manual sorting of the automated results. To minimize human operator work, algorithms are typically biased toward producing more clusters than can be expected in the recording because manually merging an oversplit cluster is easier, less time-consuming, and less error-prone than splitting an overmerged cluster (the latter requires choosing a carefully defined separation surface). Both KiloSort and KlustaKwik had such a bias, producing between two and four times more clusters than the expected number of neurons.

To estimate the best achievable score after operator merges, we took advantage of the ground truth data and automatically merged candidate clusters (Figs. 3*a–c*) so as to greedily maximize their score. Final best results as well as the required number of matches are shown in Figures 3*d–g* (KiloSort vs KlustaKwik, 69% vs 60% units with scores > 0.9). The relative

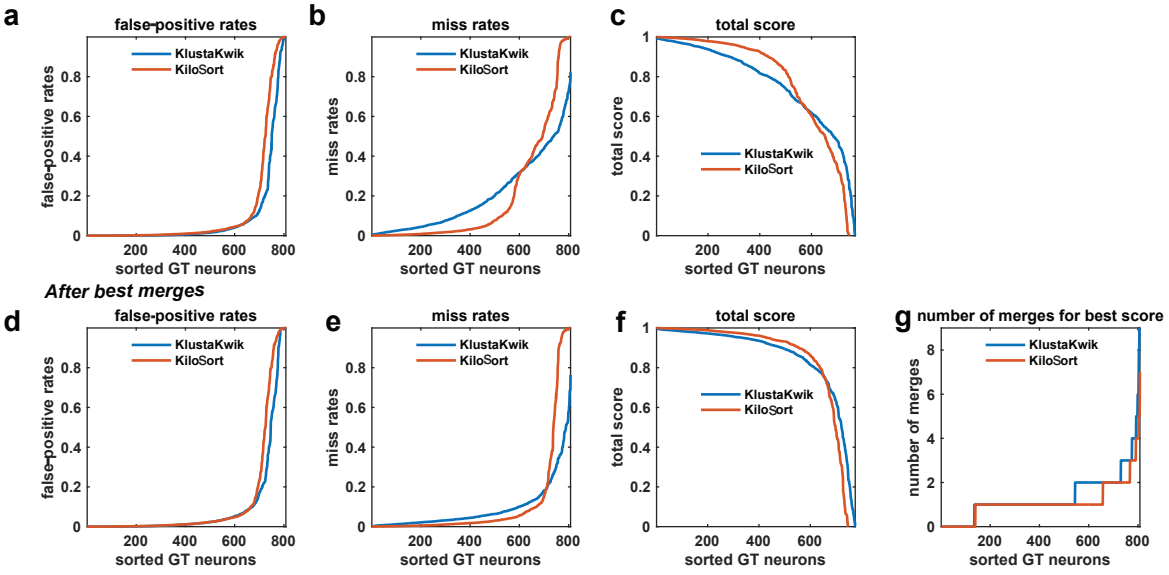


**Figure 3.** Hybrid ground truth performance of proposed (KiloSort) versus established (KlustaKwik) spike sorting algorithm. ***a***, Distribution of false-positive rates. ***b***, Distribution of misses. ***c***, Total score. ***d***, ***e***, ***f***, Same as ***a***, ***b***, and ***c*** after greedy best possible merges. ***g***, Number of merges required to reach best score. GT, ground truth.

performance improvement of KiloSort is clearly driven by fewer misses (Fig. 3*e*), which are likely the result of its ability to detect overlapping spikes.

## Extension: Post Hoc Template Merging

We found that we can further reduce human operator work by performing most of the merges in an automated way. The most common oversplit clusters show remarkable continuity of their spike densities (Fig. 4). In other words, no discrimination boundary can be identified orthogonal to which the oversplit cluster appears bimodal. Instead, these clusters arise as a consequence of the algorithm partitioning clusters with large variance into multiple templates, so as to better explain their total variance. In KiloSort, we can exploit the fact that the decision boundaries between any two clusters are in fact planes (which we show below). If two clusters belong to the same neuron, their one-dimensional projections in the space orthogonal to the decision boundary will show a continuous distribution (Figs. 4*c*,*d*,*g*,*h*), and the clusters can be merged. We use this idea to sequentially merge any two clusters with continuous distributions in their two-dimensional feature spaces. Note that the best PCs for each cluster's main channel are much less indicative of a potential merge (Figs. 4*b*,*f*).

To see why the decision boundaries in KiloSort are linear, consider two templates *Ki* and *Kj*, and consider that we have arrived at the instance of template matching in which a spike $k$ needs to be assigned to one of these two templates. Their respective cost function improvements are $dC(i, t) = a_i^2/b_i$ and $dC(j, t) = a_j^2/b_j$, using the convention from Equation 4. The decision of assigning spike $k$ to one or the other of these templates is then equivalent to determining the sign of $dC(i, t) - dC(j, t)$, which is a linear discriminant of the feature projections

$$\text{sign}(dC(i, t) - dC(j, t)) = \text{sign}(a_i/b_i^{1/2} - a_j/b_j^{1/2}), \quad (7)$$

where $b_i$ and $b_j$ do not depend on the data, and $a_{i,j}$ are linear functions of the raw voltage; hence, the decision boundary between any two templates is linear (Fig. 4).

## Discussion

We have demonstrated here a new framework for spike sorting of high-channel count electrophysiology data, which offers substantial accuracy and speed improvements over previous frameworks while reducing the amount of manual work required to isolate single units. KiloSort is currently enabling spike sorting of ≤1000 neurons recorded simultaneously in awake behaving animals and will help to enable the next generation of large-scale neuroscience. The code is available online at https://github.com/cortex-lab/KiloSort.

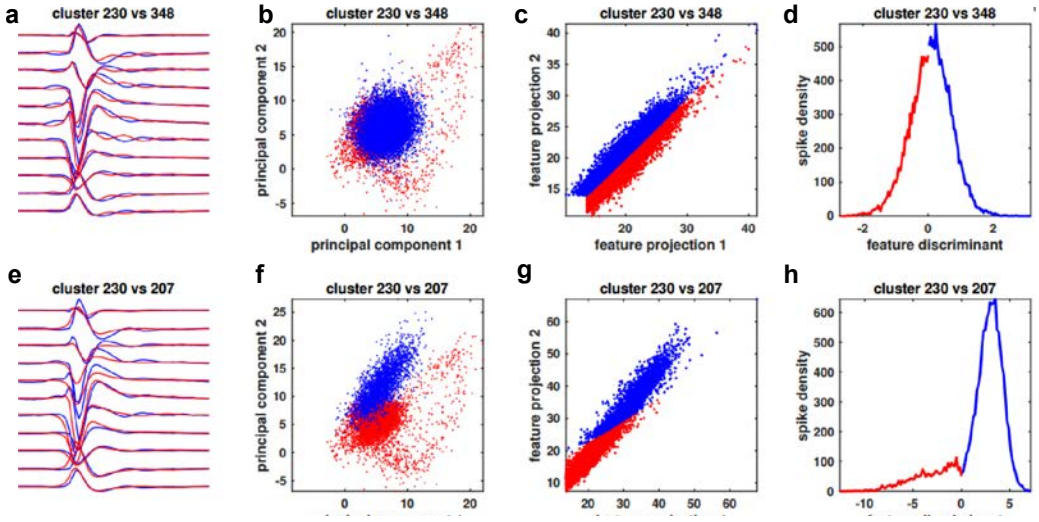


**Figure 4.** PC and feature-space projections of two pairs of clusters that should be merged. ***a***, ***e***, Mean waveforms of merge candidates. ***b***, ***f***, Spike projections into the top PCs of each candidate cluster. ***c***, ***g***, Template feature projections for the templates corresponding to the candidate clusters. ***d***, ***h***, Discriminant of the feature projections from ***c*** and ***g*** (see Eq. 7).

## Acknowledgments

This chapter was first presented at the 30th Conference on Neural Information Processing Systems (NIPS 2016), December 5–10, Barcelona, Spain.

## References

Coates A, Ng AY, Lee H (2011) An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR 15:215–223.

Einevoll GT, Franke F, Hagen E, Pouzat C, Harris KD (2012) Towards reliable spike-train recordings from thousands of neurons with multielectrodes. Curr Opin Neurobiol 22:11–17.

Ekanadham C, Tranchina D, Simoncelli EP (2014) A unified framework and method for automatic neural spike identification. J Neurosci Methods 222:47–55.

Franke F, Pröpper R, Alle H, Meier P, Geiger JR, Obermayer K, Munk MH (2015) Spike sorting of synchronous spikes from local neuron ensembles. J Neurophysiol 114:2535–2549.

Harris KD, Henze DA, Csicsvari J, Hirase H, Buzsáki G (2000) Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. J Neurophysiol 84:401–414.

Hill DN, Mehta SB, Kleinfeld D (2011) Quality metrics to accompany spike sorting of extracellular signals. J Neurosci 31:8699–8705.

Neto JP, Lopes G, Frazão J, Nogueira J, Lacerda P, Baião P, Aarts A, Andrei A, Musa S, Fortunato E, Barquinha P, Kampff AR (2016) Validating silicon polytrodes with paired juxtacellular recordings: method and dataset. J Neurophysiol 116:892–903.

Pillow JW, Shlens J, Chichilnisky EJ, Simoncelli EP (2013) A model-based spike sorting algorithm for removing correlation artifacts in multi-neuron recordings. PLoS One 8:e62123.

Quiroga RQ (2012) Spike sorting. Curr Biol 22:R45–R46.

Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Science 344:1492–1496.

Rossant C, Kadir SN, Goodman DFM, Schulman J, Hunter MLD, Saleem AB, Grosmark A, Belluscio M, Denfield GH, Ecker AS, Tolias AS, Solomon S, Buzsáki G, Carandini M, Harris KD (2016) Spike sorting for large, dense electrode arrays. Nat Neurosci 19:634–641.

NOTES

# The Montreal Neurological Institute Ecosystem: Enabling Reproducible Neuroscience from Collection to Analysis in the Web

Gregory Kiar, MS,[1,3] Carolina Makowski, BS,[1,4] Jean-Baptiste Poline, PhD,[1,2,5] Samir Das, BSc,[1] and Alan C. Evans, PhD[1,3,4,5]

[1]Montreal Neurological Institute and Hospital
McGill University
Montreal, Canada

[2]University of California, Berkeley
Berkeley, California

[3]Department of Biomedical Engineering
McGill University
Montreal, Canada

[4]Integrated Program in Neuroscience
McGill University
Montreal, Canada

[5]Department of Neurology and Neurosurgery
McGill University
Montreal, Canada

NOTES

## Introduction

It is well recognized by the scientific community that neuroscience has gained a great deal of momentum in recent years, with new funding agencies and opportunities following suit (Society for Neuroscience, n.d.). As a result, data collection is occurring at unprecedented rates across a wide array of populations and scales: from healthy human populations (Mennes et al., 2013; Van Essen et al., 2013; Sudlow et al., 2015; National Institute of Mental Health, n.d.) to those with neurological and psychiatric disorders (Jack et al., 2008; Di Martino et al., 2014; Wang et al., 2016) to small mammalian species (Kreshuk et al., 2014; Kasthuri et al., 2015). These vast supplies of data naturally require similar advances in technology for organization, processing, and visualization and must be multifaceted in order to intuitively allow users to harness the full potential of their datasets.

An exemplar of such tools is BrainBrowser (Sherif et al., 2014), which allows for interactive visualization, enabling familiar and comprehensive interaction with images. Once a user is well versed in the quality and nature of their data, the next step entails choosing a tool for analysis to "decode" the raw data to try answering specific, and often challenging, scientific questions. It has historically been common practice for a single research group to pose and attempt to answer such questions using their own datasets; however, this approach is increasingly being questioned as issues with reproducibility and statistical power ensue. As a result, data sharing (or data publishing) has emerged as an increasingly popular practice in neuroscience, allowing researchers to pool data and build sufficiently powered datasets to develop more impactful and generalizable findings.

Taking this in stride, the Montreal Neurological Institute and Hospital recently announced its initiative to become the first fully open institute (Montreal Neurological Institute and Hospital, n.d.) in the hopes that enabling open access to various biospecimen, neuroimaging, behavioral, and clinical data will propel scientific discovery. This sentiment is already widely embraced by the data-sharing neuroscientific community, but several challenges have become apparent in the pursuit of this approach. Among these challenges is developing computational infrastructure as a critical backbone for such initiatives. Several efforts have been launched globally to lower the barrier to data sharing and deploying software pipelines for neuroimaging data (Rex et al., 2003; Marcus et al., 2006; Scott et al., 2011; Das et al., 2011, 2016; Sherif et al., 2014). The BigBrain project (Amunts et al., 2013) is a key example that demonstrates the full life cycle of the open science process: from data collection and tool development to the creation of an open access dataset for the community to explore and exploit.

This chapter explores the tools and platforms developed at the Montreal Neurological Institute (MNI) to enhance the goals of reproducibility and the proliferation of a wide array of shared datasets. It is this infrastructure that aided in the success of the BigBrain project, including several other big data projects, and will enable similar initiatives to be launched in future. The MNI ecosystem is an amalgamation of several complementary platforms: the Longitudinal Online Research and Imaging System (LORIS), a neuroimaging visualization tool (BrainBrowser), a descriptive command-line framework for increasing pipeline portability (Boutiques), and a centralized resource for housing and analyzing neuroimaging data (CBRAIN).

## Data Sharing and Reproducibility

In order to adopt best practices in data sharing, as well as the interoperability of platforms leveraged to do so, the development of standards helps researchers meet minimum requirements and greatly improves efficiency. One could consider as an example how inefficient the web would be if the community did not agree on the HTML standard (cf. the recommendations of the World Wide Web Consortium). Organizations such as the International Conference on Cognitive Neurodynamics (Bjaalie, 2008) focus resources and hold workshops to help develop and improve such standards, and such collaborative efforts have continued to gain traction and influence in the neuroimaging community. A tangible product of this movement can be seen in the recent COBIDAS manuscript (published by the Organization for Human Brain Mapping's Committee on Best Practice in Data Analysis and Sharing) (Nichols et al., 2017), which suggests best practices for data sharing and analysis. With this quest for interoperability under way, a data organization standard, the Brain Imaging Data Structures (BIDS) (Gorgolewski et al., 2016), has emerged that enables datasets and repositories to be easily interchanged for analysis with an increasing array of compatible tools (Gorgolewski et al., 2017).

As these standards and their ecosystem spread, publishing datasets and acquiring DOIs for other research products (e.g., code) accelerates the potential for feedback, contribution, adoption, and citation of one's work within the community, leading to increased efficiency. Similarly, because quantifying the reproducibility of analyses has

NOTES

Table 1. Data sharing tools that contribute to reproducible neuroscience.

| Tool | Brief Description | URLs |
|---|---|---|
| LORIS | Platform for data collection, management, quality control, and sharing | http://loris.ca/<br>https://github.com/aces/Loris/ |
| BrainBrowser | Web-based viewer for volumetric and surface-based neuroimaging data | https://brainbrowser.cbrain.mcgill.ca<br>https://github.com/aces/brainbrowser |
| CBRAIN | Platform for tool and data management for web-accessible deployment in HPC environments | http://mcin-cnim.ca/technology/cbrain/<br>https://github.com/aces/cbrain |
| Boutiques | Descriptive command-line framework for repeatedly deploying pipelines | http://boutiques.github.io/<br>https://github.com/boutiques/boutiques |
| BigBrain | Human brain scan collected at 20 μm isotropic resolution | http://bigbrainviewer.acelab.ca/<br>https://mcin-cnim.ca/research/bigbrain/ |

become increasingly desirable, platforms such as CBRAIN (Sherif et al., 2014) and LORIS (Das et al., 2011), which keep strict provenance records of data, tools, and execution instructions, allow tools and datasets to be evaluated robustly. Thus, these tools encourage a level of transparency and provenance tracking in all steps of data management and analysis and contribute to the execution of reproducible neuroscience.

Developing and adopting standards, as well as openly sharing or publishing data and code and other research products, all propagate the idea of open science. In turn, this process lowers the financial and technical barrier for entry to performing high-quality, big-data neuroscience research.

In the following sections, we will explore several data sharing tools (Table 1) that represent various pieces of this puzzle. By integrating these tools (as well as ensuring interoperability with other similar emerging efforts), the requirements for performing computational neuroscience are quickly being reduced to simply possessing a web browser.

## LORIS: data collection and management

The platform used to host the BigBrain is LORIS, an online databasing resource created and maintained by our group at the McGill Centre for Integrative Neuroscience. In addition to brain specimen data such as BigBrain, LORIS has the functionality to store genetic, behavioral, and clinical data and uses the aforementioned BrainBrowser tool for visualizing neuroimaging data.

The LORIS system (Das et al., 2011, 2016) was designed specifically for heterogeneous data acquisition, curation, and dissemination and serves as the backbone for the Tannenbaum Open Science Initiative at MNI. It is a web-based data management system, freely available on GitHub as open-source software. LORIS has a modular, extensible architecture that can support multiple data modalities, including demographic metadata and behavioral/clinical, neuroimaging, and genomic data, and provides a flexible and robust platform for many types of multisite studies and projects. LORIS also provides a built-in multimodal querying web browser for elaborate population subsampling.

## BrainBrowser: visualization

BrainBrowser is an HTML5 visualization tool that leverages the capabilities of WebGL and provides a web-based exploration of volumetric and surface-based datasets (Sherif et al., 2014). Applications of BrainBrowser include quality control and annotation of neuroimaging datasets, and it has been included in both the LORIS and CBRAIN ecosystems. Several features of BrainBrowser are being developed, such as tagging, intensity thresholding, and enhanced three-dimensional (3D) overlays. In addition, new functionalities are being built to enhance the user experience and enable the visualization and streaming of larger datasets without significant performance issues.

## CBRAIN and Boutiques: pipeline standardization, deployment, and high-performance computing management

Given high-performance computing (HPC) infrastructure, it is still often difficult for researchers to effectively deploy software tools in a manner that scales for large datasets, as doing so requires knowledge about cluster/supercomputer configuration and credentials. To meet researchers' needs in this area, the CBRAIN platform provides an interface to both tools and HPC and processing capabilities, and it

NOTES

abstracts the complexities of implementing and configuring software. By leveraging the Boutiques descriptive framework (Glatard et al., 2015), which enables tool users or developers to specify instructions for how command-line arguments can be formulated, we are able to programmatically create views, validate inputs, and launch tasks on HPC resources.

Container environments such as Docker and Singularity enable these descriptors to be packaged and shipped to resources without manually configuring tools. In this way, they can increase the portability and scalability of analysis as well as ensure that the exact same environment has been reused (albeit to the limit of the underlying platform's lowest layer). The CBRAIN platform (Sherif et al., 2014) enables users to upload and manage their data from a web browser and submit tasks to HPC nodes. Although CBRAIN is an open-source project that can be launched by any institute, the main Canadian installation operating on Compute Canada and Calcul Québec resources has more than 500 users from more than 145 sites across 22 countries, has registered over 1 million user files, and run jobs totaling over 24 million CPU core hours to date. Popular workflows and tools such as CIVET, FreeSurfer Software Suite, FMRIB Software Library (FSL), NITRC's Neuroimaging Analysis Kit (NIAK), NeuroData's MR Graphs package (ndmg), and others have been integrated with CBRAIN, enabling users to analyze their data using a variety of tools with ease.

### BigBrain: brain imaging at an unprecedented scale

Enabling an unprecedented look at the human brain, BigBrain (Amunts et al., 2013) spans microscopic and macroscopic scales. Whereas previously available reference brains were restricted to a single scale (e.g., whole-brain magnetic resonance imaging in humans or electron microscopy of small sections from mice), BigBrain is an ultra-high-resolution 3D model of a full human brain at 20 μm resolution, coming closer to visualizing both spaces than any previous dataset.

BigBrain is free, publicly available, and provides the opportunity for considerable neuroanatomical insights because it allows features to be extracted at high resolution for modeling and simulation. Supporting tools such as Atelier3D and BigBrain Viewer enable users to explore the BigBrain as well as overlay their annotations or segmentations for evaluation, refinement, and sharing. Ongoing development continues to make these data and interactions more accessible for users who wish to query, download, or process BigBrain with limited computational expertise or resources. These features make the BigBrain a unique resource for the benefit of the entire neuroscience community.

## Workshop Demonstrations

Participants of this workshop will have the opportunity to cycle among three demonstrations: (1) LORIS and BrainBrowser, (2) CBRAIN and Boutiques, and (3) BigBrain. Table 2 summarizes suggested preparations to be made by attendees in order to get the most out of the workshop. Together we will:

1. Demonstrate LORIS and BrainBrowser as an efficient and featureful solution for storing, manipulating, managing, and sharing neuroimaging data;

2. Demonstrate CBRAIN and Boutiques as a powerful coupling that enables rapidly going from tool development or execution on a small scale to HPC environments; and

3. Explore the unique BigBrain dataset through an interactive and responsive viewer.

Table 2. List of workshop demonstrations and suggested preparations for each.

| Demo | Suggested Preparation |
|---|---|
| LORIS + BrainBrowser | Small (~2–3 subjects/session) DICOM dataset for ingesting, preprocessing, and visualization |
| CBRAIN + Boutiques | An example command line (e.g., bet -m /my/image.nii.gz) for a tool you regularly use or wish to use |
| BigBrain | Curiosity about a brain region you wish to explore at 20 μm resolution |

## Acknowledgments

We would like to thank all members of our team, our collaborators, and our mentors. We would also like to graciously thank our funders and sponsors, a list of which can be found on our website at the McGill Centre for Integrative Neuroscience: http://mcin-cnim.ca/.

## References

Amunts K, Lepage C, Borgeat L, Mohlberg H, Dickscheid T, Rousseau MÉ, Bludau S, Bazin PL, Lewis LB, Oros-Peusquens AM, Shah NJ, Lippert T, Zilles K, Evans AC (2013) BigBrain: an ultrahigh-resolution 3D human brain model. Science 340:1472–1475.

NOTES

Bjaalie JG. (2008) The Neuroinformatics Portal of the International Neuroinformatics Coordination Facility. In: Advances in cognitive neurodynamics ICCN 2007: Proceedings of the International Conference on Cognitive Neurodynamics. (Wang R, Shen E, Gu F, eds), pp 667–672. Dordrecht: Springer.

Das S, Glatard T, MacIntyre LC, Madjar C, Rogers C, Rousseau ME, Rioux P, MacFarlane D, Mohades Z, Gnanasekaran R, Makowski C, Kostopoulos P, Adalat R, Khalili-Mahani N, Niso G, Moreau JT, Evans AC (2016) The MNI data-sharing and processing ecosystem. Neuroimage 124(Pt B): 1188–1195.

Das S, Zijdenbos AP, Harlap J, Vins D, Evans AC (2011) LORIS: a web-based data management system for multi-center studies. Front Neuroinform 5:37.

Di Martino A, Yan CG, Li Q, Denio E, Castellanos FX, Alaerts K, Anderson JS, Assaf M, Bookheimer SY, Dapretto M, Deen B, Delmonte S, Dinstein I, Ertl-Wagner B, Fair DA, Gallagher L, Kennedy DP, Keown CL, Keysers C, Lainhart JE, et al. (2014) The Autism Brain Imaging Data Exchange: towards large-scale evaluation of the intrinsic brain architecture in autism. Mol Psychiatry 19:659–667.

Glatard T, Da Silva RF, Boujelben N, Adalat R, Beck N, Rioux P, Rousseau M, Deelman E, Evans AC (2015) Boutiques: an application-sharing system based on Linux containers. Poster presented at Neuroinformatics 2015, Cairns, Australia, August 20–22.

Gorgolewski KJ, Alfaro-Almagro F, Auer T, Bellec P, Capotă M, Chakravarty MM, Churchill NW, Cohen AL, Craddock RC, Devenyi GA, Eklund A, Esteban O, Flandin G, Ghosh SS, Guntupalli JS, Jenkinson M, Keshavan A, Kiar G, Liem F, Raamana PR, et al. (2017) BIDS apps: improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods. PLoS Comput Biol 13:e1005209.

Gorgolewski KJ, Auer T, Calhoun VD, Craddock RC, Das S, Duff EP, Flandin G, Ghosh SS, Glatard T, Halchenko YO, Handwerker DA, Hanke M, Keator D, Li X, Michael Z, Maumet C, Nichols BN, Nichols TE, Pellman J, Poline JB, et al. (2016) The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. Sci Data 3:160044.

Jack CR, Bernstein MA, Fox NC, Thompson P, Alexander G, Harvey D, Borowski B, Britson PJ, L Whitwell J, Ward C, Dale AM, Felmlee JP, Gunter JL, Hill DL, Killiany R, Schuff N, Fox-Bosetti S, Lin C, Studholme C, DeCarli CS, et al. (2008) The Alzheimer's Disease Neuroimaging Initiative (ADNI): MRI methods. J Magn Reson Imaging 27:685–691.

Kasthuri N, Hayworth KJ, Berger DR, Schalek RL, Conchello JA, Knowles-Barley S, Lee D, Vázquez-Reina A, Kaynig V, Jones TR, Roberts M, Morgan JL, Tapia JC, Seung HS, Roncal WG, Vogelstein JT, Burns R, Sussman DL, Priebe CE, Pfister H, et al. (2015) Saturated reconstruction of a volume of neocortex. Cell 162:648–661.

Kreshuk A, Koethe U, Pax E, Bock DD, Hamprecht FA (2014) Automated detection of synapses in serial section transmission electron microscopy image stacks. PloS One 9:e87351.

Marcus DS, et al. (2006) XNAT: a software framework for managing neuroimaging laboratory data. Proceedings of the 12th Annual Meeting of the Organization for Human Brain Mapping, Florence, Italy.

Mennes M, Biswal BB, Castellanos FX, Milham MP (2013) Making data sharing work: the FCP/INDI experience. Neuroimage 82:683–691.

Montreal Neurological Institute and Hospital (n.d.) Open Science data sharing initiative. Available at https://www.mcgill.ca/neuro/open-science-0.

National Institute of Mental Health (NIMH) (n.d.) Adolescent Brain Cognitive Development (ABCD) data repository. Available at https://data-archive.nimh.nih.gov/abcd.

Nichols TE, Das S, Eickhoff SB, Evans AC, Glatard T, Hanke M, Kriegeskorte N, Milham MP, Poldrack RA, Poline JB, Proal E, Thirion B, Van Essen DC, White T, Yeo BT (2017) Best practices in data analysis and sharing in neuroimaging using MRI. Nat Neurosci 20:299–303.

Poldrack RA, Barch DM, Mitchell JP, Wager TD, Wagner AD, Devlin JT, Cumba C, Koyejo O, Milham MP (2013) Toward open sharing of task-based fMRI data: the OpenfMRI project. Front Neuroinform 7:12.

Rex DE, Ma JQ, Toga AW (2003) The LONI pipeline processing environment. Neuroimage 19:1033–1048.

NOTES

Scott A, Courtney W, Wood D, de la Garza R, Lane S, King M, Wang R, Roberts J, Turner JA, Calhoun VD (2011) COINS: an innovative informatics and neuroimaging tool suite built for large heterogeneous datasets. Front Neuroinform 5:33.

Sherif T, Kassis N, Rousseau MÉ, Adalat R, Evans AC (2014) BrainBrowser: distributed, web-based neurological data visualization. Front Neuroinform 8:89.

Sherif T, Rioux P, Rousseau ME, Kassis N, Beck N, Adalat R, Das S, Glatard T, Evans AC (2014) CBRAIN: a web-based, distributed computing platform for collaborative neuroimaging research. Front Neuroinform 8:54.

Society for Neuroscience (n.d.) Global funding sources. Available at https://www.sfn.org/awards-and-funding/global-funding-sources.

Sudlow C, Gallacher J, Allen N, Beral V, Burton P, Danesh J, Downey P, Elliott P, Green J, Landray M, Liu B, Matthews P, Ong G, Pell J, Silman A, Young A, Sprosen T, Peakman T, Collins R (2015) UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS Med 12:e1001779.

Van Essen DC, Smith SM, Barch DM, Behrens TE, Yacoub E, Ugurbil K; WU-Minn HCP Consortium (2013) The WU-Minn human connectome project: an overview. Neuroimage 80:62–79.

Wang L, Alpert KI, Calhoun VD, Cobia DJ, Keator DB, King MD, Kogan A, Landis D, Tallis M, Turner MD, Potkin SG, Turner JA, Ambite JL (2016) SchizConnect: mediating neuroimaging databases on schizophrenia and related disorders for large-scale integration. Neuroimage 124(Pt B):1155–1167.

# BigBrain: An Ultra-High-Resolution 3D Human Brain Model

Katrin Amunts, MD, PhD,[1–4] Claude Lepage, PhD,[5] Louis Borgeat, PhD,[6] Hartmut Mohlberg, PhD,[1,2] Timo Dickscheid, PhD,[1,2] Marc-Étienne Rousseau, PhD,[5] Sebastian Bludau, PhD,[1,2] Pierre-Louis Bazin, PhD,[7] Lindsay B. Lewis, PhD,[5] Ana-Maria Oros-Peusquens, PhD,[1,2] Nadim J. Shah, PhD,[1,2] Thomas Lippert, PhD,[8] Karl Zilles, MD, PhD,[1–4] and Alan C. Evans, PhD[51]

[1]Institute of Neuroscience and Medicine (INM-1, INM-4)
Research Centre Jülich
Jülich, Germany

[2]Jülich Aachen Research Alliance (JARA)
JARA-BRAIN
Jülich, Germany

[3]Section Structural–Functional Brain Mapping
Department of Psychiatry, Psychotherapy and Psychosomatics
School of Medicine, RWTH Aachen University
Aachen, Germany

[4]Cécile and Oskar Vogt Institute for Brain Research
Heinrich Heine University Düsseldorf
University Hospital Düsseldorf
Düsseldorf, Germany

[5]Montreal Neurological Institute and Hospital
McGill University
Montreal, Canada

[6]National Research Council of Canada
Ottawa, Canada

[7]Max Planck Institute for Human Cognitive and Brain Sciences
Leipzig, Germany

[8]Jülich Supercomputing Centre
Research Centre Jülich
Jülich, Germany

NOTES

## Introduction

Reference brains are indispensable tools in human brain mapping, enabling integration of multimodal data into an anatomically realistic standard space. Available reference brains, however, are restricted to the macroscopic scale and do not provide information on the functionally important microscopic dimension. We created an ultra-high-resolution three-dimensional (3D) model of a human brain at a nearly cellular resolution of 20 μm, based on the reconstruction of 7404 histological sections. "BigBrain" is a free, publicly available tool that provides considerable neuroanatomical insight into the human brain, thereby allowing the extraction of microscopic data for modeling and simulation (http://bigbrain.cbrain.mcgill.ca). BigBrain enables the testing of hypotheses on optimal path lengths between interconnected cortical regions or on spatial organization of genetic patterning, thereby redefining traditional neuroanatomy maps such as those of Brodmann and von Economo (Brodmann, 1909; von Economo and Kosinkas, 1925).

Brain organization on multiple scales and regional segregation are key elements for the development of a realistic model of the human brain. Multiscale organization requires the integration of both multilevel and multimodal data, from the level of cells with their specific connectivity to the level of cognitive systems and the whole brain. Magnetic resonance imaging (MRI) enables the study of the structure and function of the living human brain, with a spatial resolution in the range of 1 mm for structural imaging and somewhat larger for functional MRI (fMRI) (Roland and Zilles, 1994; Toga et al., 2006). This resolution is well above the cellular scale but has been sufficient for establishing human brain atlases to capture information at the level of brain areas, subcortical nuclei, gyri, and sulci (Talairach and Tournoux, 1988; Roland et al., 1994; Toga et al., 2006; Evans et al., 2012).

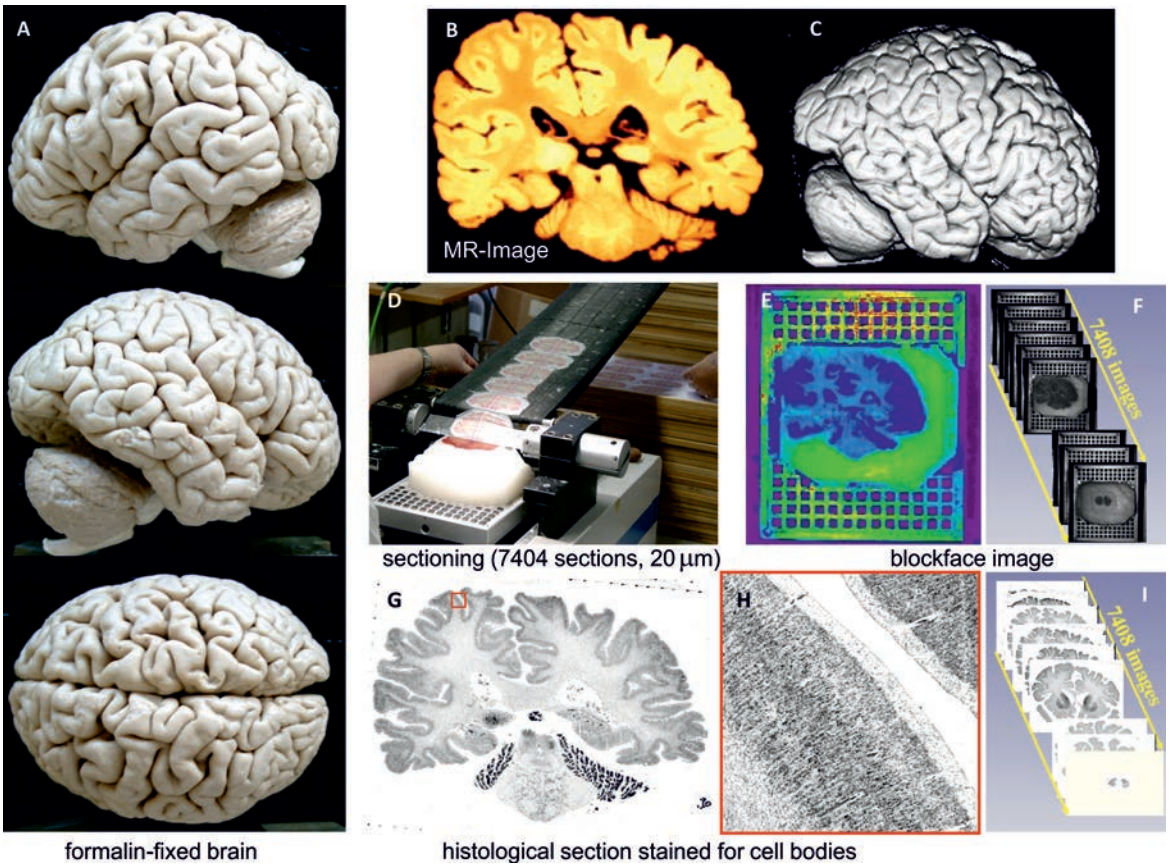


**Figure 1.** Illustration of tissue and image processing. ***A***, Photographs of the fixed brain: lateral left (top), lateral right (middle), and dorsal (bottom) views. ***B***, MRI (coronal view) and ***C***, 3D-reconstructed MRI volume of the fixed brain. ***D***, Histological sectioning. ***E***, Blockface image of a section (pseudocolored) resting on the mounting grid that served to align the blockface images. ***F***, Series of blockface images. ***G***, Cell-body-stained histological sections with the region of interest denoted (red box). This area is shown with higher magnification in ***H***. ***I***, Series of histological images that were 3D reconstructed using the blockface images ***F*** and the MRI ***C***. Reprinted with permission from Amunts K et al. (2013), Figure 1. Copyright 2013, American Academy for the Advancement of Sciences.

NOTES

Cytoarchitectonic probabilistic maps enable the identification of microstructural correlates involved in a specific brain function, as determined by fMRI, e.g., during a cognitive task (Eickhoff et al., 2006; Zilles and Amunts, 2010). This approach is supported by combined physiological and imaging studies showing that the response properties of neurons change at the border of two areas (Luppino et al., 1991; Nelissen et al., 2005). Existing human brain atlases do not allow for the integration of information at the level of cortical layers, columns, microcircuits, or cells (Fig. S1, available at http://www.sciencemag.org/cgi/content/full/340/6139/1472/DC1), as has been shown recently for mouse or invertebrate brains (Li et al., 2010; Peng et al., 2010). Still, fine-grained anatomical resolution is a necessary prerequisite to fully understand the neurobiological basis of cognition, language, emotions, and other processes as well as to bridge the gap between large-scale neural networks and local circuitry within the cerebral cortex and subcortical nuclei.

## Creation of a Human Brain Model

We sought to create a human brain model at nearly cellular resolution by going considerably beyond the 1 mm resolution of presently available atlases, taking advantage of recent progress in computing capacities and image analysis, and relying on our experience in processing histological sections of the complete brain. Major challenges include, but are not limited to, the highly folded cerebral cortex, the large number of areas, considerable variability among brains, and the sheer size of the brain, with its nearly 86 billion neurons and the same number of glial cells (Hilgetag and Barbas, 2009; Herculano-Houzel, 2012). Compared with rodent or invertebrate brains, the human brain is extremely complex: For example, the volume of a human cerebral cortex is ~7500× larger than a mouse cortex, and the amount of white matter is 53,000× larger in humans than in mice. The recently published dataset of the digitized mouse brain with 1 µm resolution has a total uncompressed volume data of 8 TB (Li et al., 2010). The creation of a volume with similar spatial resolution for the human brain would result in ~21,000 TB. The interactive exploration (as opposed to simple storage) of such a dataset is beyond the capacities of current computing. Thus, among other methodological problems, data processing becomes a major challenge for any project aiming at the reconstruction of a human brain at cellular resolution.

To create the brain model, we used a large-scale microtome to cut a complete paraffin-embedded, 65-year-old brain (male) coronally (Fig. 1). We then acquired 7400 sections at 20 µm thickness and stained them for cell bodies (Merker, 1983). Histological sections were digitized, resulting in images of maximally 13,000 × 11,000 pixels (10 × 10 µm pixel size). The total volume of this dataset was 1 TByte. The uninterrupted data acquisition time was ~1000 h. To generate a dataset with isotropic resolution, we downscaled all images to 20 × 20 µm to match the section thickness of 20 µm.

Histological processing inevitably introduces artifacts, which pose problems at all stages of the 3D reconstruction process. Defects include rips, tears, folds, missing and displaced pieces, distortion (shear), stain inhomogeneity, and crystallization. We performed both manual and automatic repairs to restore the integrity of all sections before the 3D reconstruction of the whole brain as a contiguous volume (Figs. S2–S4, available at http://www.sciencemag.org/cgi/content/full/340/6139/1472/DC1). The repaired sections were registered to the MRI, which served as an undistorted frame of reference, and further aligned section-to-section with the use of nonlinear registration. All calculations were performed on high-performance computing (HPC) facilities within the Compute Canada network and were run on Jülich Research on Petaflop Architecture (JuRoPA) at the Jülich Supercomputing Centre (supplementary material, available at http://www.sciencemag.org/cgi/content/full/340/6139/1472/DC1).

## 3D Analysis

Figure 2 shows three sample regions from primary sensory and motor cortices in the original coronal plane and the reconstructed sagittal and horizontal planes. Note the smooth contours in the virtual sections, confirming the high quality of the 3D reconstruction. The images in all three planes at 20 µm reveal differences in the laminar pattern among brain areas and enable an observer-independent definition of borders between them (Schleicher et al., 2009). To prove the feasibility of our mapping approach in higher associative cortices with more subtle architectonic differences in between, we defined a border between Brodmann area (BA) 10 of the frontal pole and BA32 (dorsal anterior cingulate area 32) (Fig. 3). Although some artifacts caused by residual mismatches between aligned sections still exist, the border of interest has been detected in the original and the horizontal virtual plane at the identical location. Thus, the present "BigBrain" model allows for the recognition of the borders not only between primary cortical areas (feasible, at least to some extent with advanced MRI technology)

NOTES

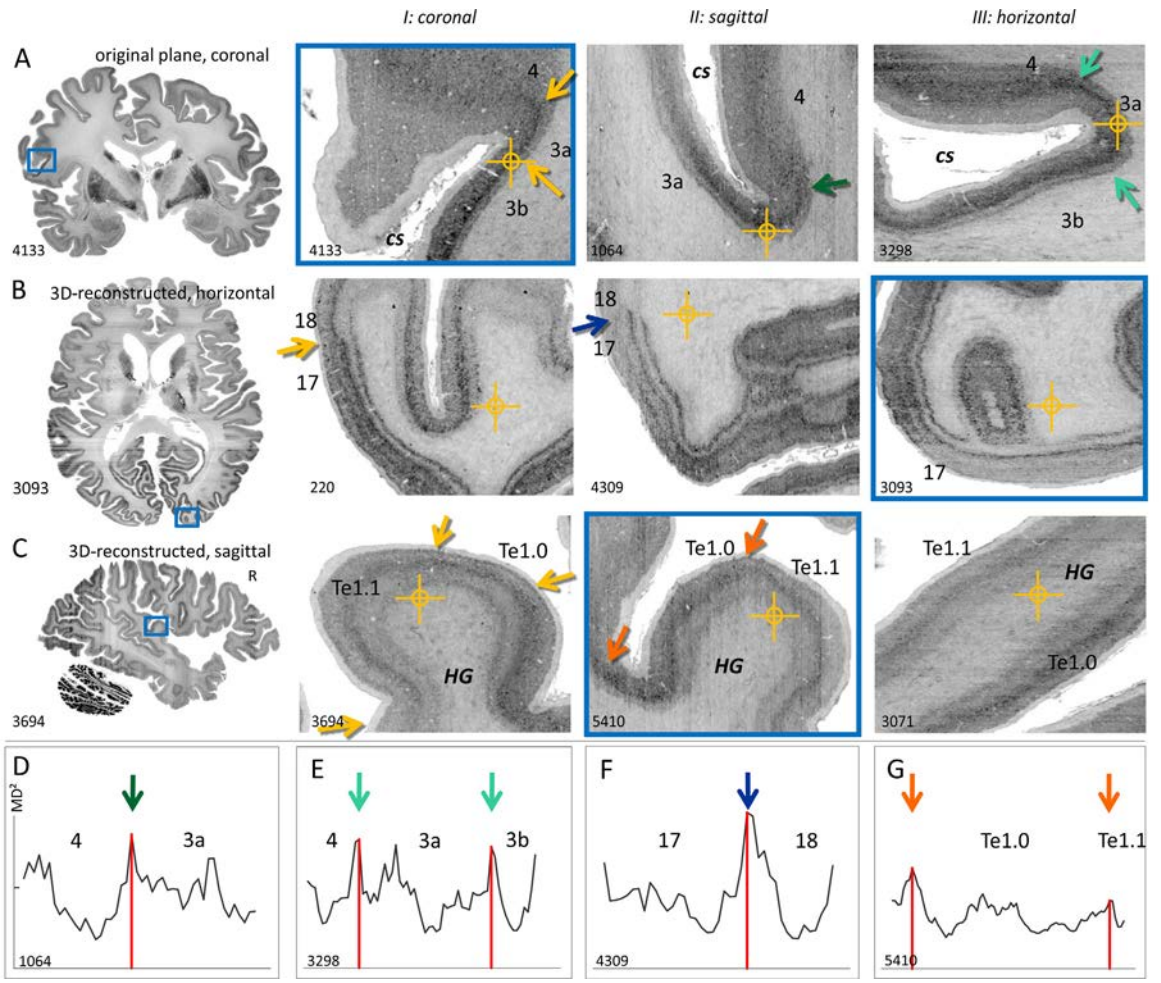


**Figure 2.** Primary cortical regions in the three planes of section. ***A***, Sensorimotor (BA4, 3a and 3b); ***B***, visual (BA17 [V1] and BA18 [V2]); and ***C***, auditory cortex (areas Te1.0, Te1.1 as part of BA41) (left column). Overviews of the whole-brain sections in the original plane (***A***) and the 3D-reconstructed horizontal (***B***) and sagittal (***C***) planes. Crosshairs denote identical positions within a row. Columns I–III show coronal, sagittal, and horizontal planes, respectively. Section numbers are shown in the lower-left corner of each panel. ***D–G***, Definition of borders for regions of interest from ***A*** to ***C***, based on the Mahalanobis distance (Schleicher et al., 2009). Corresponding borders are labeled by identically colored arrows (see also supplementary materials and Fig. 3). Reprinted with permission from Amunts K et al. (2013), Figure 2. Copyright 2013, American Academy for the Advancement of Sciences.

(Fatterpekar et al., 2002; Walters et al., 2007; Glasser and van Essen 2011; Sánchez-Panschuelo et al., 2012) but also between higher associative areas. Until now, the recognition of the latter borders, based on their laminar pattern, was accessible in two-dimensional (2D) histological sections and light-microscope images, but only at those locations where the cortex was cut orthogonal to the pial surface. The latter condition is often not fulfilled (Fig. 2A, coronal), thus making border definition based on quantitative criteria throughout the whole cortical ribbon in 2D sections impossible.

The 3D analysis indicates that the relationship among cortical folds and borders of cytoarchitectonic areas is heterogeneous. Whereas this relationship is considerably close for some areas, it seems to be less well defined for others. For example, the border between the primary motor and somatosensory cortex is localized in the fundus of the central sulcus (cs), independently from the orientation of the cutting plane (Fig. 2A). This is not the case for the primary auditory area Te1 (BA41), which is more or less restricted by Heschl's gyrus (HG) in two planes (Fig. 2C) but has no sulcal landmark in the third plane (Morosan et al., 2001). Whereas the sulcal pattern is associated with areal borders in other, nonprimary areas (e.g., BA35) (Augustinack et al., 2013), the border between the primary visual area BA17 (V1) and neighboring BA18 (V2) (Fig. 2B) does not seem to be related to a sulcus. The same is true for the border between BA10 and the neighboring cingulate cortex (Fig. 3). This variable relationship among cytoarchitectonic borders and macroscopic

NOTES

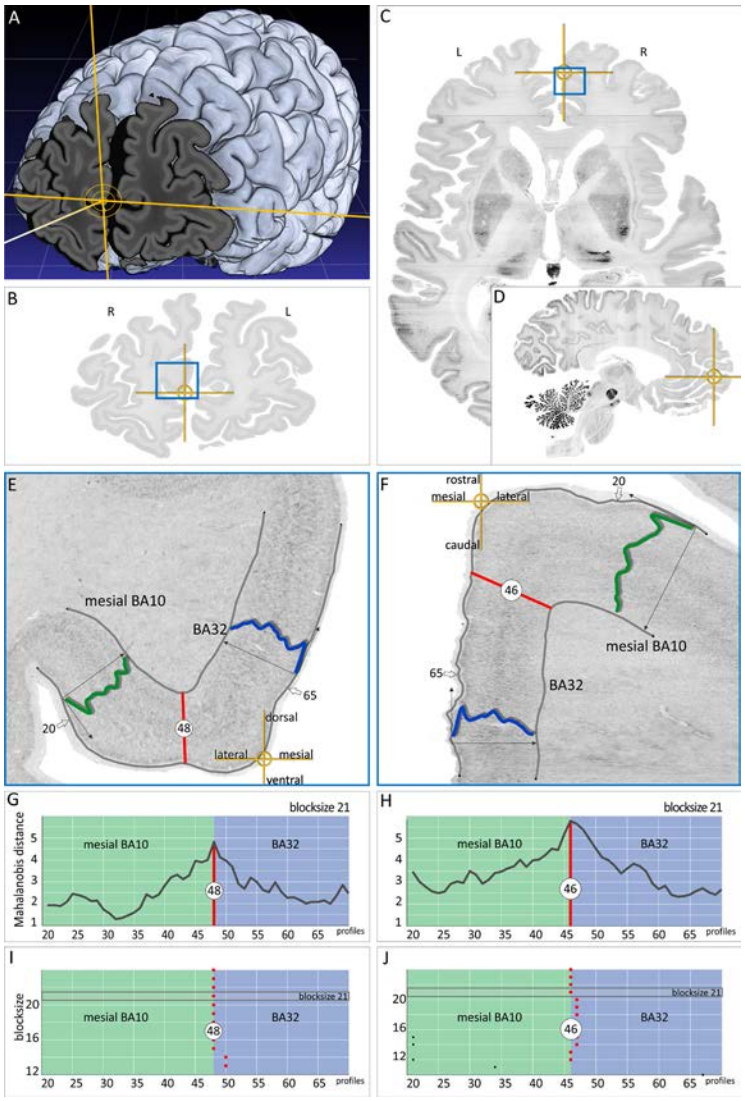


**Figure 3.** Definition of a boundary (Schleicher et al., 2009) in the frontal cortex. ***A***, Surface rendering of the 3D reconstructed brain (rostral view) with the frontal pole removed. ***B***, Coronal section 6704. R, right; L, left. ***C***, 3D reconstructed horizontal (2740) and ***D***, sagittal (3588) sections. Yellow crosshairs are at identical position in ***A–F***. ***E–J,*** Border definition in ***B*** and ***C***. Green, mean profile of mesial BA10 (Bludau et al., 2014); blue, mean profile of BA32 (***E, F***). ***G, H***, Mahalanobis distance as a function of position along the cortical ribbon. ***I, J,*** Localization of significant peaks ($p < 0.01$) in the Mahalanobis distance (see also supplementary materials). Reprinted with permission from Amunts K et al. (2013), Figure 3. Copyright 2013, American Academy for the Advancement of Sciences.

landmarks has been analyzed in the past (Amunts et al., 2004; Fischl et al., 2008), but not in 3D space.

The spatial dimension, however, is relevant, because the directionality of hemispheric growth during embryonic and fetal development and the coupling of cortical areas via fiber tracts define the spatial organization of cortical areas and their connections, as well as sulci and gyri in the adult brain. The effect of early cortical regionalization on folding has been modeled by introducing geometric, mechanical, and growth asymmetries in the model (Toro et al., 2008). Another model considered variability between brains during ontogeny (Lefèvre and Mangin, 2010). A recent study has emphasized the strong geometric structure of fibers and pathways as a result of early development (Wedeen, 2012). A subsequent study reported that fiber connection patterns closely follow gyral folding patterns in the direction tangential to the cortical sphere (Chen et al., 2013). The concept of the tension-based morphogenesis effect provides a theory of folding processes caused by the tension of fiber tracts connecting brain regions (van Essen, 1997; Kriegsteine et al., 2006), whereas other theories identify differences in the relationship between supragranular and infragranular layers (i.e., cytoarchitectonic differences) as factors shaping cortical folding (Amstrong et al., 1991). The validation of all these concepts requires high-resolution spatial models of the human brain for testing the underlying hypotheses.

## Implications of Findings and the BigBrain Dataset

The present findings and data on the localization of cortical areas with respect to gyri and sulci support the notion that their topographical relationship is not merely a pure geometric phenomenon but rather the result of an interference of developmental processes and the internal structure of areas, including their connectivity (Zilles and Amunts, 2012). A systematic analysis of cortical borders across the whole cortical ribbon is urgently needed. The variability in this relationship across individuals requires the generation of additional BigBrain datasets in the future: labor-intensive work that is currently under way.

To consider intersubject variability in the present dataset, vector fields have been calculated based on

NOTES

a 400 μm isotropic downsampled volume, to define a homeomorphic transformation between the BigBrain and the Montreal Neurological Institute and Hospital (MNI) space, which embeds information about intersubject variability (supplementary material, available at http://www.sciencemag.org/cgi/content/full/340/6139/1472/DC1). Thus, cytoarchitectonic or functional probability maps in MNI space can be mapped to the BigBrain dataset. We plan to establish links to other reference systems so as to combine high-resolution cytoarchitectonic data with, for example, gene expression maps (Jones et al., 2009), neural projections (Kasthuri and Lichtman, 2007), or future brain-activity maps (Alivisatos et al., 2013).

The BigBrain dataset will be made publicly available to promote the development of new tools for defining 3D cytoarchitectonic borders. BigBrain allows the extraction of parameters of cortical organization by enabling measurements parallel to cell columns (e.g., cortical thickness, densities of cell bodies per column, surface measures). In this way, it provides a "gold standard" for calibrating *in vivo* measurements of cortical thickness and other measures.

The BigBrain dataset represents a new reference brain, moving from a macroanatomical perspective to microstructural resolution. This model provides a basis for addressing stereotaxic and topological positions in the brain at the micrometer range (e.g., with respect to cortical layers and sublayers). BigBrain will make it possible to localize findings obtained in cellular neuroscience and mapping studies targeting transmitter receptor distributions (Zilles and Amunts, 2009), fiber bundles (Axer et al., 2011), and genetic data (Jones at al., 2009; Shen et al., 2012). The BigBrain model can also be exploited as a source for generating realistic input parameters for modeling and simulation. It thus represents a reference frame with nearly cellular resolution—a capability that has not been previously available for the human brain—while considering the regional heterogeneity of human brain organization.

## Acknowledgments

We acknowledge funding support from Canada's Advanced Research and Innovation Network (www.canarie.ca) for the software development of the CBRAIN portal. We also thank Compute Canada (www.computecanada.ca) for continued support through extensive access to the Compute Canada HPC grid. We thank P. Morosan and U. Pietrzyk for helpful discussion, as well as F. Kocaer and U. Blohm for technical assistance. We are particularly grateful to R. Adalat for his efforts in managing and coordinating this collaboration. This work was supported by the Portfolio project "Supercomputing and Modeling for the Human Brain," funded by the Helmholtz Association of German Research Centres, and the Human Brain Project, a European Union Future and Emerging Technologies Flagship project. The normalized BigBrain data are available at http://bigbrain.cbrain.mcgill.ca (upon free subscription).

This chapter was modified from a previously published article of the same title: Amunts K, et al. (2013) Science 340:1472–1475. Supplementary materials can be accessed at http://www.sciencemag.org/cgi/content/full/340/6139/1472/DC1, including Materials and Methods, Figures S1–S6, References, and Movie S1 (visualization of the 3D-reconstructed dataset of the BigBrain from Atelier3D). 

## References

Alivisatos AP, Chun M, Church GM, Deisseroth K, Donoghue JP, Greenspan RJ, McEuen PL, Roukes ML, Sejnowski TJ, Weiss PS, Yuste R (2013) Neuroscience. The brain activity map. Science 339:1284–1285.

Amunts K, Lepage C, Borgeat L, Mohlberg H, Dickscheid T, Rousseau MÉ, Bludau S, Bazin PL, Lewis LB, Oros-Peusquens AM, Shah NJ, Lippert T, Zilles K, Evans AC (2013) BigBrain: an ultrahigh-resolution 3D human brain model. Science 340:1472–1475.

Amunts K, Weiss PH, Mohlberg H, Pieperhoff P, Eickhoff B, Gurd JM, Marshall JC, Shah NJ, Fink GR, Zilles K (2004) Analysis of neural mechanisms underlying verbal fluency in cytoarchitectonically defined stereotaxic space—the roles of Brodmann areas 44 and 45. Neuroimage 22:42–56.

Armstrong E, Curtis M, Buxhoevedn DP, Fregoe C, Zilles K, Casanova MF, McCarthy WF (1991) Cortical gyrification in the rhesus monkey: a test of the mechanical folding hypothesis. Cereb Cortex 1:426–432.

Augustinack KC, Huber KE, Stevens AA, Roy M, Frosch MP, van der Kouwe AJ, Wald LL, Van Leemput K, McKee AC, Fischl B, Alzheimer's Disease Neuroimage Initiative (2013) Predicting the location of human perirhinal cortex, Brodmann's area 35, from MRI. Neuroimage 64:32–42.

NOTES

Axer M, Grässel D, Kleiner M, Dammers J, Dickscheid T, Reckfort J, Hütz T, Eiben B, Pietrzyk U, Zilles K, Amunts K (2011) High-resolution fiber tract reconstruction in the human brain by means of three-dimensional polarized light imaging (3D-PLI). Front Neuroinform 5:34.

Bludau S, Eickhoff SB, Mohlberg H, Caspers S, Laird AR, Fox PT, Schleicher A, Zilles K, Amunts K (2014) Cytoarchitecture, probability maps and functions of the human frontal pole. Neuroimage 93(Pt 2):260–275.

Brodmann K (1909) Vergleichende Lokalisationslehre der Großhirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues. Leipzig: Barth. In: Brodmann's localisation in the cerebral cortex (Garey LJ, ed, 1994). London: Smith-Gordon.

Chen H, Zhang T, Guo L, Li K, Yu X, Li L, Hu X, Han J, Hu X, Liu T (2013) Coevolution of gyral folding and structural connection patterns in primate brains. Cereb Cortex 23:1208–1217.

Eickhoff SB, Heim S, Zilles K, Amunts K (2006) Testing anatomically specified hypotheses in functional imaging using cytoarchitectonic maps. Neuroimage 32:570–582.

Evans AC, Janke AL, Collins DL, Baillet S (2012) Brain templates and atlases. Neuroimage 62:911–922.

Fatterpekar GM, Naidich TP, Delman BN, Aguinaldo JG, Gultekin SH, Sherwood CC, Hof PR, Drayer BP, Fayad ZA (2002) Cytoarchitecture of the human cerebral cortex: MR microscopy of excised specimens at 9.4 Tesla. AJNR Am J Neurorad 23:1313–1321.

Fischl B, Rajendran N, Busa E, Augustinack J, Hinds O, Yeo BT, Mohlberg H, Amunts K, Zilles K (2008) Cortical folding patterns and predicting cytoarchitecture. Cereb Cortex 18:1973–1980.

Glasser MF, Van Essen DC (2011) Mapping human cortical areas *in vivo* based on myelin content as revealed by T1- and T2-weighted MRI. J Neurosci 31:11597–11616.

Herculano-Houzel S (2012) The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost. Proc Natl Acad Sci USA 109(Suppl 1):10661–10668.

Hilgetag CC, Barbas H (2009) Are there ten times more glia than neurons in the brain? Brain Struct Funct 213:365–366.

Jones AR, Overly CC, Sunkin SM (2009) The Allen Brain Atlas: 5 years and beyond. Nat Rev Neurosci 10:821–828.

Kasthuri N, Lichtman JW (2007) The rise of the 'projectome'. Nat Methods 4:307–308.

Kriegstein A, Noctor S, Martínez-Cerdeño V (2006) Patterns of neural stem and progenitor cell division may underlie evolutionary cortical expansion. Nat Rev Neurosci 7:883–890.

Lefèvre J, Mangin JF (2010) A reaction-diffusion model of human brain development. PLoS Comput Biol 6:e1000749.

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

Luppino G, Matelli M, Camarda RM, Gallese V, Rizzolatti G (1991) Multiple representations of body movements in mesial area 6 and the adjacent cingulate cortex: an intracortical microstimulation study in the macaque monkey. J Comp Neurol 311:463–482.

Merker B (1983) Silver staining of cell bodies by means of physical development. J Neurosci Methods 9:235–241.

Morosan P, Rademacher J, Schleicher A, Amunts K, Schormann T, Zilles K (2001) Human primary auditory cortex: cytoarchitectonic subdivisions and mapping into a spatial reference system. Neuroimage 13:684–701.

Nelissen K, Luppino G, Vanduffel W, Rizzolatti G, Orban GA (2005) Observing others: multiple action representation in the frontal lobe. Science 310:332–336.

Peng H, Ruan Z, Long F, Simpson JH, Myers EW (2010) V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. Nat Biotechnol 28:348–353.

Roland PE, Zilles K (1994) Brain atlases—a new research tool. Trends Neurosci 17:458–467.

Roland PE, Graufelds CJ, Wahlin J, Ingelman L, Andersson M, Ledberg A, Pedersen J, Akerman S, Dabringhaus A, Zilles K (1994) Human brain atlas: for high-resolution functional and anatomical mapping. Hum Brain Mapp 1:173–184.

Sánchez-Panchuelo RM, Francis ST, Schluppeck D, Bowtell RW (2012) Correspondence of human visual areas identified using functional and anatomical MRI *in vivo* at 7 T. J Magn Reson Imaging 35:287–299.

NOTES

Schleicher A, Morosan P, Amunts K, Zilles K (2009) Quantitative architectural analysis: a new approach to cortical mapping. J Autism Dev Disord 39:1568–1581.

Shen EH, Overly CC, Jones AR (2012) The Allen Human Brain Atlas: comprehensive gene expression mapping of the human brain. Trends Neurosci 35:711–714.

Talairach J, Tournoux P (1988) Co-planar stereotaxic atlas of the human brain: 3-D proportional system: an approach to cerebral imaging. Stuttgart, Germany: Thieme.

Toga AW, Thompson PM, Mori S, Amunts K, Zilles K (2006) Towards multimodal atlases of the human brain. Nat Rev Neurosci 7:952–966.

Toro R, Perron M, Pike B, Richer L, Veillette S, Pausova Z, Paus T (2008) Brain size and folding of the human cerebral cortex. Cereb Cortex 18:2352–2357.

Van Essen DC (1997) A tension-based theory of morphogenesis and compact wiring in the central nervous system. Nature 385:313–318.

von Economo C, Koskinas GN (1925) Die Cytoarchitektonik der Hirnrinde des erwachsenen Menschen. Berlin: Springer.

Walters NB, Eickhoff SB, Schleicher A, Zilles K, Amunts K, Egan GF, Watson JD (2007) Observer-independent analysis of high-resolution MR images of the human cerebral cortex: *in vivo* delineation of cortical areas. Hum Brain Mapp 28:1–8.

Wedeen VJ, Rosene DL, Wang R, Dai G, Mortazavi F, Hagmann P, Kaas JH, Tseng WY (2012) The geometric structure of the brain fiber pathways. Science 335:1628–1634.

Zilles K, Amunts K (2009) Receptor mapping: architecture of the human cerebral cortex. Curr Opin Neurol 22:331–339.

Zilles K, Amunts K (2010) Centenary of Brodmann's map—conception and fate. Nat Rev Neurosci 11:139–145.

Zilles K, Amunts K (2012) Neuroscience. Segregation and wiring in the brain. Science 335:1582–1584.