

# Tropism Analysis Package: Interactive Machine Learning Software to Identify Viral Host Factors Through Single-Cell Host-Virus mRNA Profiling

## Authors

Kenny Pavan<sup>1,2</sup>, Lamy Ben Ameer<sup>1</sup>, Zach Goode<sup>1</sup>, Emily Tiedemann<sup>1</sup>, Elizabeth Kolb<sup>1</sup>, Arpiar Saunders<sup>1#</sup>

## Affiliations

1. Vollum Institute, Oregon Health & Science University, Portland, OR, 97239, USA

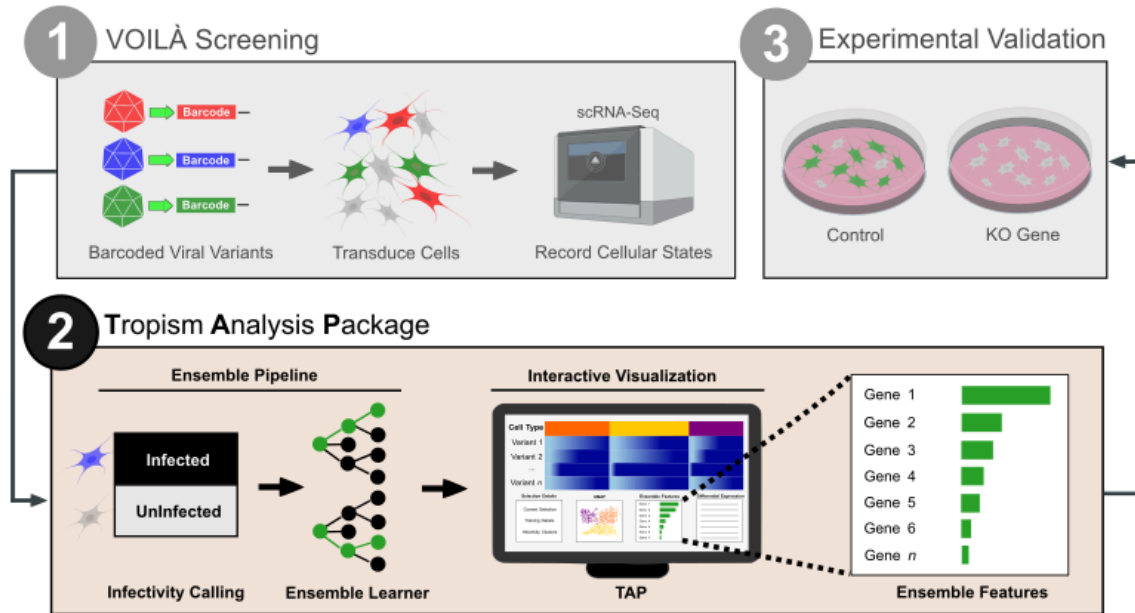
2. Biomedical Engineering Graduate Program, Oregon Health and Science University, Portland, OR, 97239, USA

# Correspondence: saundear@ohsu.edu

## Abstract

Understanding the cellular host factors that control viral tropism and infection outcomes remain a central challenge in both virology and viral vector-based gene delivery. Although genetic screens and biochemical analysis in immortalized cell lines have proven successful at identifying pro-viral entry receptors, transcriptional/replicational regulators or anti-viral restriction factors, such approaches are a challenge to apply in diverse primary cells that constitute complex tissue. To address this gap, we recently developed VOILÀ (Virus-Host Factor Interaction Learning Analysis), a framework that uses single-cell RNA-seq to nominate host factor proteins by analyzing cell-to-cell variability in host and virus gene expression. Here, we describe the Tropism Analysis Package (TAP), a flexible, interactive ensemble machine learning package for analysis and visualization of host-virus scRNA-seq data, including VOILÀ datasets. TAP integrates seamlessly with standard Scanpy AnnData objects, supports high-throughput screening implemented through RNA barcoding, and enables dynamic visualization, multi-level filtering, and interpretable ensemble learning to identify genes predictive of viral entry and replication. Using scRNA-seq data from mouse primary brain cells *in vitro* and *in vivo*, we showcase TAP by nominating host factors across structurally diverse capsids of non-replicating adeno-associated virus (AAV) and for replicating rabies virus. TAP provides a scalable, open-source framework to accelerate systematic discovery of viral host factors from single-cell RNA data.

## Graphical Abstract



## Availability and Implementation

TAP is a pip installable package that can be found at <https://github.com/ArpiarSaundersLab/TAP> along with documentation and usage tutorials. Our results can be interactively explored at: <https://www.aavdb.com>.

## 1. Introduction

A central challenge in virology and viral vector-based gene delivery is to understand the molecular mechanisms that dictate tropism and control infection outcomes in complex tissue. Tropism refers to the capacity of a virus to selectively infect particular cell types, reflecting the cumulative influence of host-encoded molecules that enable or restrict viral attachment, entry, replication, and spread. Much of what is known about these molecular determinants stems from genetic screens and biochemical interaction studies using immortalized cell lines, which have identified core receptors, co-receptors, and restriction factors for a limited number of viruses, including adeno-associated virus (AAV) and rabies viruses.<sup>1–8</sup> While these approaches have been instrumental in revealing proteins responsible for highly conserved host–virus interactions, such simplified cell systems are poorly suited to discover host factors across molecularly diverse primary cell types in a more physiologically relevant tissue context. Consequently, mechanisms inferred from cell lines may not fully explain the cell-type-specific patterns of viral transduction observed in experimental animals or human patients, which are of great interest to researchers and clinicians.

Improvements in the sensitivity and throughput of single-cell RNA sequencing have opened new avenues to characterize host-virus transcriptional interactions in primary cells, including the discovery of viral host factors. To date, such datasets have largely been used to describe tropism at the level of cell populations. However, quantitative, genome-wide host-virus co-expression data across thousands of individual cells offers new opportunities to gain insight into molecular mechanisms that dictate viral infection properties.

For example, we recently developed VOILÀ (Virus–Host Factor Interaction Learning Analysis), a single-cell RNA-seq framework that systematically links cell-to-cell variability in capsid-specific AAV and host gene expression to nominate candidate genes whose protein products determine viral tropism in primary cells [Ben Ameer et al] (**Figure 1A, S1A**). In this study, we used VOILÀ to screen for host factors for eleven structurally distinct AAV capsids in mouse primary brain cell cultures, leveraging capsid-specific RNA barcodes for multiplexed analysis. VOILÀ nominated specific members of the Complement 1q-like (C1QL) protein family as selective host factors for the AAV.Retro capsid, which we subsequently validated as the key receptors responsible for AAV.Retro entry into neuronal axons through a novel pathway [Ben Ameer et al]. VOILÀ data demonstrate that viral–host gene expression signatures across thousands of individual cells constitute a high-information-content data type from which machine learning can nominate host factors for focused experimental testing. However, computational tools that facilitate the analysis, exploration and visualization of host-virus scRNA-seq data are lacking.

To help fill this need, we developed the Tropism Analysis Package (TAP), which provides a unified computational environment for both analysis and interactive exploration of host-virus scRNA-seq data, such as VOILÀ datasets. While gene-module methods and advanced feature-selection frameworks can capture high-dimensional co-expression structure, many current implementations depend on deep architectures, or graph neural formulations that complicate biological interpretation and hinder straightforward validation of single host factors nomination in the context of a classification problem<sup>9,10</sup>. On the back end, TAP implements an interpretable ensemble machine learning pipeline that systematically evaluates host gene expression features across categorical subsets - such as viral strain, capsid type, or transduced cell type - to identify genes predictive of viral mRNA expression, such as entry receptors. The platform integrates seamlessly with Scanpy AnnData objects<sup>11,12</sup>, enabling use with standard single-cell RNA-seq workflows. Additionally, TAP supports multiplexed viral variant analysis by tracking variant-specific RNA barcodes. On the front end, TAP supports host factor discovery through an interface for interactive visualization that dynamically summarizes viral variant infectivity results across cell populations. This is accomplished using dynamic multi-level filtering to control variant expression plots, host gene feature ranking and model summaries flexibly across different types of experimental replicate structures.

Here we describe the TAP computational framework and showcase TAP functionality through three example datasets, focusing on structurally-diverse AAV capsid or rabies virus transduction of primary mouse brain cells. In the first example, we illustrate how TAP analysis of the initial VOILÀ screen (transduction of cultured mouse cortical cells by 11 structurally-distinct AAV capsids) nominated *C1ql3* as a host factor for the AAV2.Retro capsid. In the second example, we use TAP to re-analyze a scRNA-seq derived from embryonic mouse brain after *in vivo* transduction of 14 structurally-distinct AAV capsids. TAP analysis nominated LRFN5<sup>13</sup> and HS3ST4<sup>14</sup> as candidate host factors for AAV.SCH9, an AAV capsid distinguished by fast-onset expression and neural stem cell tropism *in vivo*<sup>15</sup>. In the third example, we highlight TAP's applicability to replicating viruses, by analysing rabies virus transduction of cultured mouse cortical cells. TAP results corroborate that DYNLL1/2 are positive regulators of rabies virus expression<sup>16–19</sup>. Together, these results highlight TAP's flexibility and functionality for analyzing host-virus interactions through mRNA co-expression signatures in primary cells, helping advance single-cell RNA-seq as a broadly applicable dataset for viral host factors discovery to complement more traditional genetic screens in cell lines.

## 2. Results

### 2.1a Package Capabilities and Data Exploration Overview

The Tropism Analysis Package (TAP) is a pip-installable, object-oriented toolkit that integrates an ensemble machine learning and feature-selection pipeline with interactive visualization of single-cell host–virus co-expression datasets. The central intuition behind TAP is that genes whose expression best discriminates infected from uninfected cells of the same type, as learned by the classifier, correspond to candidate host factors whose protein products modulate viral mRNA abundance in important ways, such as regulating viral entry, transcription or replication. TAP accepts AnnData<sup>12</sup> or AnnSQL<sup>20</sup> objects with a two-dimensional embedding (such as UMAP or tSNE) as input, which are validated and lightly preprocessed according to user-defined parameters. An overview of the TAP computational workflow is shown in **Figure 1A**.

Within TAP, each combination of cell type and viral variant, referred to as a “categorical subset”, is formulated as an independent classification problem. Users may provide prior cell type annotations or perform automated classification using the integrated CellTypist module<sup>21,22</sup>. TAP includes all available CellTypist tissue models as well as a custom-trained model of the adult mouse brain based on single-nuclei RNA-seq analysis<sup>23</sup>. Each categorical subset is iteratively processed through an ensemble pipeline that includes: infected versus uninfected label determination, host gene Feature Importance score assignment using random forest classification based on infection labels, and differential expression testing across infection label groups. To match different types of viruses and experimental design, each step in this process is highly configurable. For example, cluster “marker” genes (sources of false positive associations) can be optionally flagged for removal and imbalanced representation of single mRNA profiles across infected and uninfected classes can be flexibly balanced (**Figure 1B**). All results are stored in JSON format within an HTML output file which are then used to compose the interactive user interface (UI) via custom javascript, html, and css code.

### 2.1b Ensemble Pipeline

The ensemble pipeline is composed of three major components that are applied to each cell type/viral variant categorical subset. The first component classifies each scRNA-seq profile into infected and uninfected groups. Infection labels are determined by considering mRNA counts associated with each viral variant versus the total mRNA counts for each cellular RNA profile. Three options are available as a parameter choice: a single threshold (i.e  $> x$  infected), multiple thresholds (i.e serotype counts  $> x =$  infected,  $< x =$  uninfected), or through unsupervised clustering using a Gaussian Mixture Model as implemented by SciKit Learn.<sup>24,25</sup> Once infectivity labels are assigned, class imbalance is assessed and optionally adjusted. By default, imbalance correction is implemented using SMOTE, however, additional options implemented in Imbalance Learn are included as TAP parameters.<sup>26,27</sup> Next, scRNA-seq data are split into 70% training and 30% testing data to train a random forest classifier to calculate Feature Importance scores for each host gene. Feature Importance scores capture how gene-specific mRNA levels contribute to infection label classification and are calculated using the impurity-based method included with the SciKit Learn. The performance of each categorical subset infection classification model is quantified using area-under-the-curve (AUC) values and accuracy scores, with higher AUC values and accuracy scores indicating a stronger predictive accuracy and separation of infection states. Both metrics

are included for each categorical subset and contained in the UI. Parameters for permutation testing and random forest hyperparameter tuning are included in TAP. Lastly, differential expression testing is performed for each categorical subset by comparing scRNA profiles grouped by infectivity label using the Scanpy *rank\_gene\_groups* method. All results are stored in JSON format and structured with categorical subset combination strings as keys so that they can be accessed  $O(1)$  in the UI.

### 2.1c Interactive User Interface (UI)

The interactive UI, primarily composed of HTML and CSS code, controls custom Javascript using the Bootstrap CSS framework. The UI contains five panels. The first panel, which displays the “Expression Heatmap”, shows categorical subset observations as colored blocks across the top (cell types) and viral variant mRNA expression levels in each row. User selection of each categorical subset filters the selection of cells and updates the other four result summary panels accordingly (**Methods**). Notably, in the Expression Heatmap, viral variant mRNA expression values scale according to screen viewport through aggregation of cells into available pixels ( $cells\ per\ pixel = total\ cells / total\ pixels$ ), providing a visual clarity not achievable with static heatmap or dotplot representations. The “Selection Details” panel contains information related to the selected categorical subset. These data include the total cells selected, cells per pixel, and cells with expression greater than 0; an infection label summary plot based on variant mRNA versus total mRNAs for each cell; and hyperparameters of the random forest model. The “UMAP” panel displays a responsive UMAP plot of viral variant mRNA levels. The “Ensemble Features” panel lists gene features sorted descending Feature Importance score (**Methods**). To facilitate the identification of genes with robust viral variant associations, each gene with non-zero Feature Importance scores may also have up to three colored pepper icons. Pepper color indicates if the gene is (1) also observed as differentially expressed across infected versus uninfected cell groups (yellow); (2) has non-zero Feature Importance scores and is differentially expressed across all experimental replicates (orange); and (3) encodes a protein that matches keywords related to “virus” or “transmembrane” proteins. Lastly, differential gene expression results are displayed in a table format (cluster label, log fold-change (logfc), and adjusted p-value). Combined, this five panel UI allows users to explore candidate viral variant-specific host factor results in a fast, dynamic and intuitive way that is also easily shareable for online analysis.

## 2.2 Utility of TAP for Host Factor Screening

To showcase TAP’s flexibility and performance for host factor screening, we applied TAP to three distinct scRNA-seq datasets: non-replicating AAV of diverse capsid structures, applied either to mouse brain cells *in vitro* [Ben Ameuer et al] or *in vivo*<sup>28</sup>, as well as replicating rabies virus (RABV) applied to mouse brain cells *in vitro*<sup>29</sup>. Each dataset was preprocessed using a Scanpy pipeline (**Methods**) and executed through the TAP package, resulting in a single compressed HTML output file that preserves all analysis metadata, including UMAP embeddings, infectivity labels, ranked gene Feature Importance score lists across categorical subsets. A web browser is used to filter and explore a cache of results in real time (**Figure 1D**). Select results from each analysis are reported below. Full results from these three datasets can be explored at <https://www.aavdb.com/>.

### 2.2a Analysis of 11 AAV Capsid Serotypes Using Mouse Brain Cells *In Vitro*

To introduce TAP’s utility for AAV capsid-specific host factor screening, we first focus on a VOILÀ dataset in which 11 distinct AAV capsids were used to transduce mouse cortical cell cultures *in vitro*

(**Figure 2A** and **Methods**). Capsid-specific expression is tracked through an RNA barcode incorporated into the 3' region of an mRNA transcript encoding nuclear-localized GFP driven by the CAG promoter. This capsid collection consisted of 8 naturally-occurring variants - for which conserved primary attachment and protein entry receptors are described<sup>30,31</sup> - as well as three structurally-engineered capsids (AAV2.Retro<sup>32</sup>, AAV.Shh10<sup>33</sup> and AAV.Anc80<sup>34</sup>) that exhibit unique qualities of brain cell tropism through unknown mechanisms. Capsids were bath-applied at equal titer at seven days *in vitro* (DIV) and scRNA-seq libraries were generated seven days later (**Methods**). Three biological replicates totalled 12,071 scRNA-seq profiles after preprocessing, quality control and cell calling were collected and analysed with TAP (Replicate 1, n= 12,071; Replicate 2, n = 9,643 and Replicate 3, n=7,789 ). The presence of transcriptionally-diverse excitatory (glutamatergic) and inhibitory (GABAergic) neurons, glia (astrocytes, microglia, and oligodendrocyte precursor cells) and neural precursor cells, were confirmed from *de novo* clustering and cross-cluster analysis of known cell type “marker” genes, as well as CellTypist assignment within TAP (**Figure 2B,C**). Each of the eleven capsids showed quantitative differences in RNA barcode expression across brain cell type populations (**Figure 2D**).

The data were then processed through the TAP ensemble pipeline. Default parameters were used, save for infectivity calling, in which infectivity labels were assigned for each replicate using the buffered threshold approach ( >5 UMIs classified an RNA profile as infected and 0 UMIs as uninfected; **Figure 2F** and **Figure S2A**). To illustrate TAP functionality and results summaries, we focus on excitatory (glutamatergic) and inhibitory (GABAergic). Across 66 total categorical subsets (11 capsids x 2 cell populations x 3 replicates), 30 (10 categories in each replicate) had sufficient mRNA profiles to define infected and uninfected group labels for host factor analysis via ensemble feature selection (**Figure S2B**). Categorical subsets simultaneously containing non-zero Feature Importance scores while being also differentially expressed totaled 196 genes across all replicates (adjusted p value < 0.05; yellow pepper criterion)(**Methods**). To better exclude false positive associations, we filtered this gene set to include capsid/cell population associations across all replicates (orange pepper criterion) and for genes that encoded transmembrane receptors (red pepper criterion; **Methods**). We identified a single hit, *Clql3* in excitatory neurons transduced by AAV2.Retro (yellow, orange, red pepper criterion)(**Figure S2C**). As expected, levels of *Clql3* mRNA and the AAV2.Retro RNA barcode co-varied similarly in a UMAP representation of the experiment (**Figure 2G**).

We use the *Clql3*/AAV2.Retro association in excitatory neurons to highlight how TAP categorical subset results are explored and critically evaluated using the TAP UI (**Figure 2E-G**). Within the “Expression Heatmap” panel, selection of the AAV2.Retro (row) and excitatory neuron population (column) dynamically updates the “Infectivity Calling” and “Ensemble Features” panels (**Figure 2F,G**). The “Selection Details” panel is also updated, providing critical details of the analysis, including model accuracy (**Figure S2A**). *Clql3* showed ensemble Feature Importance scores of 0.71, 0.43, and 0.15 across replicates, with corresponding AUC scores of 0.79, 0.82, and 0.80 (**Figure S2A**). Collectively, this analysis underscores that *Clql3* expression is a strong and reproducible predictor of AAV2.Retro RNA barcode expression in excitatory neurons, exemplifying how TAP ensemble pipeline and interactive UI enables quantitative, model-informed nomination of host factor–capsid relationships.

## 2.2b Analysis of 14 AAV Capsid Serotypes Using Mouse Brain Cells *In Vivo*

To test TAP's robustness and flexibility to nominate AAV capsid-specific host factors from an *in vivo* experimental context, we re-analyzed a published dataset in which 14 unique AAV capsids were barcoded and transduced into the embryonic mouse brain<sup>28</sup> (**Figure 3A**). Capsid-specific expression was tracked through one of three unique RNA barcodes incorporated into the 3' region of an mRNA transcript encoding GFP and driven by the CAG promoter. AAV capsids were mixed at equal titer, injected into the lateral ventricle at embryonic day (E)13.5 and cells were sampled for scRNA-seq analysis 48 hours later E15.5 (for detailed Methods, see Zheng et al <sup>28</sup>). These scRNA-seq data include a diverse set of developing neuron and non-neuronal cell populations and showed quantitative differences in capsid-specific RNA barcode, as illustrated by AAV.SCH9, AAV2.NN and SCH9.G160D examples (**Figure 3B,C**). This dataset is noteworthy for TAP-based host factor analysis in four ways. First, AAVs were transduced systemically *in vivo* as compared to bath-applied *in vitro* (**Figure 2**). Second, the 14 capsids were pre-selected for fast-onset transduction and consisted of 12 engineered and two naturally-occurring capsid variants; engineered capsids with modified tropism suggest the potential for novel host factor associations. Third, each capsid was associated with three distinct barcodes (denoted "BCNa", "BCMb", and "BCNc"), a distinct experimental replicate structure from the previous *in vitro* dataset which is important for TAP to track. Fourth, GFP+ cells were enriched via fluorescent sorting before library preparation and include a mix of capsid-specific transduced and non-transduced populations (n=7,205 AAV-transduced scRNA-profiles; n=7,630 untransduced scRNA-profiles).

The data were then processed through the TAP ensemble pipeline. Default settings were used for the ensemble pipeline except for two parameters. First, infected and uninfected groups were defined using custom infectivity thresholding (>2 UMIs infected, 0=uninfected). Second, replicate mode was configured to track the three RNA barcodes associated with the same capsid serotype. Across 54 total categorical subsets (3 capsids x 6 cell populations x 3 replicates), 42 (16, 14, and 12 in replicates 1, 2 and 3, respectively) had sufficient mRNA profiles to define infected and uninfected group labels for host factor analysis via ensemble feature selection (**Figure S3C**). Categorical subsets simultaneously containing non-zero Feature Importance scores while being also differentially expressed totaled 305 genes across all replicates (adjusted p value < 0.05; yellow pepper criterion)(**Methods**). To better exclude false positive associations, we filtered this gene set to include capsid/cell population associations across all replicates (orange pepper criterion) and for genes that encoded transmembrane receptors (red pepper criterion; **Methods**). In ULPNs, we identified one hit (yellow, orange, red pepper criterion) at the intersection of all replicates, *Lrfr5* transduced by AAV-SCH9 (**Figure S3D**). As expected, levels of *Lrfr5* mRNA and the AAV-SCH9 RNA barcode co-varied similarly in a UMAP representation of the experiment (**Figure 3F**). Additionally, we observed *Hs3st4* in Migrating Neurons as a candidate feature (**Figure 3E, S3E**).

We used the TAP UI interface to explore candidate host factors for AAV-SCH9 (BC1a, BC1b, and BC1c barcodes) across two different categorical subsets (**Figure 3C-E**). By selecting the "Upper Layer Projection Neuron" (ULPN) population, TAP revealed that the Leucine Rich Repeat And Fibronectin Type III Domain Containing (*Lrfr5*) gene had Feature Importance scores of 0.45, 0.29, 0.56 and AUC scores of 0.77, 0.79, and 0.80 across barcode replicates, respectively (**Figure 3D** and **Figure S3A**). By selecting the "Migrating Neuron" population, TAP revealed the Heparan Sulfate-Glucosamine 3-Sulfotransferase 4 (*Hs3st4*) gene as the top scoring gene feature across all barcode replicates, with Importance Scores of 0.29, 0.05, and 0.01 and AUC scores of 0.69, 0.75, and 0.80, respectively (**Figure 3E** and **Figure S3B**). As expected, summed AAV-SCH9 barcode levels co-varied with *Hs3st4* and *Lrfr5*

mRNAs on a UMAP plot of the experiment (**Figure 3F**) and AAV-SCH9 RNAs highly overlapped with cells expressing *Hs3st4* in the Migrating Neuron population and with *Lrfr5* in ULPN population (**Figure S3F-G**). As we detail in the Discussion, these results highlight TAP's capacity to nominate promising context-dependent AAV capsid host factors from *in vivo* experiments and offer candidate proteins that explain the enhanced tropism of AAV-SCH9 for neural stem cells.

### 2.2c Analysis of Rabies Virus SAD-B19ΔG Using Mouse Brain Cells *In Vitro*

To test TAP's nomination of host factors outside of the non-replicating AAV *Dependoparvovirus* genus, we next focused on rabies virus (RABV). The *Lyssavirus* RABV has a negative stranded RNA genome with five genes: *N*, *P*, *M*, *G* and *L* and actively replicates in infected cells. Our experiments took advantage of a mutant RABV in which the glycoprotein gene (*G*) was replaced with GFP and pseudotyped with the EnvA coat protein (SAD-B19ΔG-GFP(EnvA))(**Figure 4**). This biosafe RABV system allows replication-competent RABV to target TVA receptor expression cells without secondary infections and conveniently allows viral gene expression to be monitored by GFP fluorescence<sup>35</sup>. While important host factors that promote or restrict RABV replication or transcription have been described in immortalized cell lines, what host factors are critical in primary brain cells are less well understood<sup>17</sup>. We used TAP re-analyze three scRNA-seq replicates totalling 12,680 mRNA profiles derived from cultured embryonic mouse cortex 72 hours after SAD-B19ΔG-GFP(EnvA) transduction (Replicate 1, n= 4,262; Replicate 2, n = 3,612 and Replicate 3, n=4,806 )<sup>29</sup>. Marker gene and CellTypist analysis confirmed the presence of major neuronal excitatory (glutamatergic), inhibitory (GABAergic) and astrocyte populations (**Figure 4B** and **Methods**). Analysis of individual and aggregated RABV gene expression levels confirmed broad expression across cell populations with mild quantitative differences (**Figure 4C**).

The data were then processed through the TAP ensemble pipeline. Random Forest regression was applied to assign gene Feature Importance scores using continuous variation in RABV RNAs as independent variables. Across 54 total categorical subsets (6 gene categories x 3 cell populations x 3 replicates), 20 (7, 6, and 7 in replicates 1, 2 and 3, respectively) had sufficient mRNA profiles to define via Gaussian Mixture Model, infected and uninfected group labels for host factor analysis via ensemble feature selection (**Figure S4B**). Categorical subsets simultaneously containing non-zero Feature Importance scores while being also differentially expressed totaled 158 genes across all replicates (adjusted p value < 0.05; yellow pepper criterion)(**Methods**). To better exclude false positive associations, we filtered this gene set to include capsid/cell population associations across all replicates (orange pepper criterion) and for genes that encoded transmembrane receptors (red pepper criterion; **Methods**). In the *total\_counts* row (simulated gene of total gene counts), we identified one hit (yellow, orange, red pepper criterion) at the intersection of all replicates in Glutamatergic and Interneurons, DYNLL1. (**Figure S4E**).

The corresponding TAP interface export displayed interactive selections of "Glutamatergic + Total Viral UMIs" and "Interneurons + Total Viral UMIs," each linked to their infectivity classifications and ensemble feature outputs (**Figure 4C**). Using TAP's ensemble feature pipeline, we identified known rabies virus host factors such as DYNLL1 and DYNLL2 within infected neuronal subsets, recapitulating prior findings that dynein light-chain proteins facilitate rabies polymerase transport and transcription. In glutamatergic neurons, TAP achieved a regression R<sup>2</sup> score of 0.77, 0.5, and 0.53 for replicates 1-3, respectively and assigned DYNLL1 importance scores of 0.83, 0.73, and 0.80. In interneurons, TAP

achieved a  $R^2$  score of 0.48, 0.46, and 0.32 for each replicate and assigned DYNLL1 importance scores of 0.29, 0.31, and 0.60 (**Figure 4C** Bottom panels).

Together with the AAV dataset, this analysis illustrates TAP's versatility in handling both non-replicating vector systems and replicating neurotropic viruses. By integrating automated infectivity calling, ensemble feature discovery, and cell type-specific visualization, TAP enables rapid and interpretable identification of genes that shape viral tropism and host-pathogen interactions at single-cell resolution.

## 4. Discussion

Systematic identification of specific proteins that influence viral tropism in primary cells remains a major unmet need in both virology and viral vector research. Advances in the sensitivity and throughput of single-cell RNA sequencing support high information content descriptions of host-virus co-expression relationships for many virus types and cell populations from diverse tissues. In principle, such cell-to-cell molecular variability could nominate viral host factors by identifying host genes whose quantitative differences in RNA level lead to protein level changes that play a privileged role in shaping viral RNA expression. Consistent with this idea, VOILA scRNA-seq analysis of barcoded AAV capsids nominated *C1ql1/3* genes as AAV2.Retro host factors. We subsequently validated that C1QL1/3 proteins are receptors for AAV2.Retro that are necessary for axonal uptake [Ben Ameur et al]. Despite the promise of RNA co-expression as means to viral host factors, computational tools specifically designed to prioritize gene-level hypotheses for experimental testing are needed. We developed the Tropism Analysis Package (TAP) to fill this gap. To highlight TAP's mechanics, we walk through the VOILA dataset including the AAV2.Retro/*C1ql3* association. To highlight TAP's broad applicability, we identify either high-quality candidate or validated viral host factors from two additional scRNA-seq datasets, spanning viral types and experimental designs.

TAP is a pip installable Python package that inputs standard AnnData or AnnSQL data into an ensemble pipeline that treats each cell type-virus as a categorical subset for supervised classification. Specifically, host gene mRNA levels are used to predict infectivity status across individual cells. TAP's pipeline parameters are tunable to accommodate different virus types and experimental designs, including viral variant multiplexing through variant-specific RNA barcoding. For example, infectivity labels can be assigned by thresholding viral mRNA counts or by gaussian mixture models that account for differences in the size of each cell's RNA library. The resulting infectivity labels can be optionally balanced prior to differential expression and Random Forest classifier analysis. We chose Random Forest classification because it is a standard, interpretable machine learning method to assign Feature Importance scores for each gene in the genome. Additionally, TAP supports Random Forest regression. Random Forest regression assigns genes Feature Importance scores as a function of continuous viral expression data rather than categorical labels, and thus is amenable for replicating viruses or those with otherwise high RNA expression. To help highlight robust host factor candidates, TAP implements experiment-specific metrics, visualized with convenient colored pepper icons; for example, to flag genes that classify infection labels cells across replicates or encode proteins annotated as localized to the extracellular compartment, plasma membrane, or known viral interactors. The TAP pipeline can be run locally, on a high-performance computer cluster or a cloud-based server. The latter two options help overcome memory requirements associated with large datasets (~10Gb per 10k cells; **Figure S1C**).

Host factors may influence viral mRNA expression in myriad ways in cells, necessitating analysis tools to explore, scrutinize and share TAP results with ease. We developed an interactive TAP UI that inputs a single sharable HTML file generated by the TAP pipeline. This HTML file is directly shareable and can be run on the web browser of a standard computer. Alternatively, the HTML file can be uploaded to a cloud-based server for access to large datasets. For example, the TAP results reported here are available for exploration at [aavdb.com](http://aavdb.com). Each TAP output file is accompanied by a meta data JSON file that serves as a lookup resource for building searchable databases, agentic artificial intelligence, or for integration with other analysis tools. Using the dynamic UI interface, users filter active TAP categorical subsets through the viral Expression Heatmap panel, which updates results from the selected analysis, including model performance and gene Feature Importance scores. Additional RNA barcode tracking supports replicate identification or high-throughput viral variant screening.

To explore single cell virus-host co-expression data as broadly applicable strategy host factor discovery, we used TAP to analyze two additional experiments: an *in vivo* scRNA-seq analysis of 14 RNA-barcoded AAV capsid serotypes and an *in vitro* analysis of replicating rabies virus (RABV). In the first case, TAP nominated high-quality host factor candidates for capsid-specific AAVs. In the second case, TAP nominated RABV host factors that had been identified by previous functional analysis. These results underscore TAP's utility and application to *in vivo* analysis and analysis related to replicating viruses.

For example, TAP nominated two cell type-specific host factors for AAV.SCH9, an engineered capsid with potent tropism for neural stem cells and other brain cell types through largely unknown molecular mechanisms<sup>15</sup>. In developing upper-layer projection neurons (ULPNs), TAP nominated *Lrfrn5*. LRFN5 is a glycosylated transmembrane protein containing six N-terminal extracellular glycosylation sites<sup>36</sup>, terminal galactose moieties that we speculate could serve as AAV attachment sites. In the postnatal neurons, LRFN5 is a postsynaptic protein involved in synaptogenesis<sup>37</sup>. Embryonic *Lrfrn5* expression begins around E11.5 and peaks between E13.5 and E17.5<sup>13</sup>, coincident with AAV intra-ventricular injection at E13.5<sup>28</sup>. In the Migrating Neuron population, TAP nominated a distinct association with *Hs3st4*. HS3ST4 encodes a 3-O-sulfotransferase, which catalyzes 3-O-sulfated glucosaminyl residues to heparan sulfate proteoglycans (HSPGs)<sup>15,38</sup>. Of note, AAV.SCH9 is a chimeric capsid engineered from AAV2/6/8/9 capsid components and has been shown experimentally to bind both galactose and HSPGs glycans on cellular membranes<sup>15</sup>. Additional experiments are required to determine if LRFN5 and HS3ST4 facilitate cell type-specific attachment and internalization of AAV.SCH9 *in vivo*.

Lastly, we applied TAP to study replicating SAD-B19ΔG RABV in mouse primary cortical cultures. Focusing on total RABV mRNA levels, TAP identified high and robust Feature Importance scores of *Dynll1* in both excitatory and inhibitory neurons, and lower but consistent scores for *Dynll2* in inhibitory neurons. DYNLL1/2 are non-canonical dynein light-chain subunits that bind to RABV P and L proteins and facilitate RABV replication and transcription in cell lines<sup>16,19,39-41</sup>. More recently DYNLL1, was identified as an important pro-viral transcriptional regulator in human neurons<sup>17</sup>. TAP also identified a lesser but consistent association of *Mgst3* in excitatory neurons. MGST3 encodes an enzyme that uses glutathione to neutralize lipid peroxidation products, reducing oxidative injury and maintaining neuronal health. The roles of MGST3 in neuronal response to RABV infection and DYNLL2 in promoting RABV expression in cortical inhibitory neurons merit further investigation.

## Limitations of the study

Like other types of genetic screens, host-virus co-expression data has a number of intrinsic limitations for host factor discovery which informed our development of TAP. First, although host factors are proteins, co-expression analysis quantifies genome-wide mRNA levels. Importantly, while protein translation rates vary considerably across genes<sup>42</sup>, but likely less so for the same gene across cells, cell-to-cell variability in mRNA is likely a reasonable proxy for cell-to-cell variability in protein. Second, cell-to-cell co-expression variability is a product of both intrinsic differences and responses to infection. In the case of non-replicating AAVs, intrinsic differences appear dominant and show substantial hysteresis between the time of transduction and time of sampling (7 and 2 days in the analyzed AAV datasets). TAP's flexible ensemble pipeline and interactive UI is designed to assist the user in scrutinizing the nature of promising host factor candidates. Third, although Random Forest provided an easily implemented and interpretable Feature Importance method, future work might incorporate additional machine learning and statistical frameworks. For instance, Support Vector Machines with linear kernels also provide interpretable feature weights, while logistic regression with regularization methods such as LASSO can perform embedded feature selection. Gradient boosted tree models, such as XGBoost, offer refined feature importance metrics and improved predictive power. Even simpler approaches such as k-nearest neighbors can be adapted with feature weighting schemes to highlight variable contributions. These alternative classifier methods could extend alternative advantages for prioritizing specific host factor genes in a TAP framework.

## Conflict of Interests

Authors declare no conflicts of interest.

## Author Contributions

Conceptualization (KP); Data Curation (KP); Formal Analysis (KP); Funding acquisition (AS); Investigation (KP); Methodology (KP, AS); Project Administration (AS); Resources (KP, AS); Software (KP); Supervision (AS); Validation (KP); Visualization (KP); Writing – original draft (KP); Writing – review & editing (KP, AS).

## Acknowledgements

The authors thank Dr. Michael Mooney, Dr. Andrew Adey, Dr. Guanming Wu, Connor Frankston in addition to the members of the Saunders lab, Andrew Adey lab, and Brian O’Roak lab for helpful advice and/or discussions.

## Funding

This work has been supported by the NIH Brain Initiative (R01 MH130464); Sloan Foundation, and Simons Foundation Autism Research Initiative.

## 3. Methods

### 3.1 Package Software Structure

Developed as an object-oriented Python pip installable package, the TAP class implements a recursive categorical parsing technique to read the standard AnnData *X* layer cell by gene expression along with user-defined *obs* layer categories. The class can be instantiated with a variety of parameters all

documented on the corresponding Github (url) to tune functionality of the html output containing the interactiveUI. The initial default steps include: data validation, conversion to raw expression counts, filtering of cells with low expression, then normalizing and log transforming the dataset, achieved using scanpy preprocessing functionality. Once completed, the dataset can utilize automated cell type calling using the included Cell Typist package or use manual cell type definitions provided in the AnnData object. All default Cell Typist models are included and can be accessed by using the runCell Typist parameter when the TAP class is instantiated.

Once categorical cell identifiers (cell type, replicates, etc...) are defined, all possible subsets of viral expression counts and categorical observations are then processed through the pipeline described in **Figure 1B** and stored for O(1) access by the user interface. TAP considers the single-cell AnnData input as the following:

$$X \in \mathbb{R}^{n \times p}$$

Where  $n$  is the number of cells and  $p$  is the number of genes. It is expected that each cell  $i$  provided as input contains metadata, represented as  $m$  for categorical observations (e.g. cell type, replicates, etc) and  $C$  for containing all possible combinations of categories (Vector, Category, Vector + Category, Vector + Replicate + Category, etc) belonging to  $C^{(i)}$  as defined below.

$$m_i = (m_i^{(1)}, m_i^{(2)}, \dots, m_i^{(k)}), m_i^{(j)} \in C^{(j)}$$

The TAP software then recursively iterates each unique categorical combination and performs threshold filtering to confirm non-empty subsets of data described as:

$$X_c = \{x_i \in X \mid \forall j, m_i^{(j)} = c^{(j)}\} \text{ for each } c \in C$$

Where  $c = (c^{(1)}, \dots, c^{(k)})$  is a unique categorical combination of a the Cartesian product of all selected metadata variables  $C = (c^{(1)} \times \dots \times c^{(k)})$  and  $X_c$  is the subset of cells matching that particular combination. Each  $X_c$  subset is then passed through the highly configurable TAP analysis pipeline (**Figure 1B**) consisting of data anomaly validations, preprocessing, outlier removal, and infectivity clustering leading to important features for each  $X_c$  classification.

### 3.2 Infectivity Calling of Each Categorical Subset

TAP treats each categorical subset ( $X_c$ ) of cells as an individual dataset and classification problem to be processed by its algorithmic ensemble pipeline. The first step is to frame each subset ( $X_c$ ) into a classification problem by clustering infectivity, yielding binary labels: infected vs. uninfected. All infectivity calls consider only viral expression counts and total expression counts. TAP supports three infectivity calling strategies: (1) Simplistic infectivity thresholding accepts a scalar parameter that will denote all cells greater than that value will be considered infected. (2) The buffered infectivity thresholding option accepts two scalar values. A greater than value and a less than value. Depending on the vector and its properties, this option may provide the ability to capture more extreme transcriptional states of the two classes by setting the parameters with a buffered zone between greater than and less than

values. (3) Lastly, the Scikit Learn Gaussian Mixture Model (GMM) implementation is included as a third option for clustering infectivity.<sup>24</sup> This option deploys the `GaussianMixture` class with the default parameters and `n_components = 2` to classify the infected and uninfected labels. Additionally, the TAP implementation of the GMM aims to increase the transcriptional distinctiveness of each label by only including cells with a 95% or greater probability of existing in each class.

### 3.3 Ensemble Implementation and Feature Importance

A Random Forest (RF) classifier implemented by Scikit Learn is used by default and invoked in the `FeatureSelection` class, which is instantiated in the TAP class and trained on each subset of categorical cells ( $X_c$ ) after clustering labels are determined. Each subset ( $X_c$ ) is evaluated for class imbalance. The Imbalanced Learn package is utilized to provide TAP parameters for data balancing.

Once classification labels are assigned, TAP trains a Random Forest (RF) model on each categorical subset ( $X_c$ ). The Random Forest (RF) classifier *RandomForestClassifier*, implemented by Scikit Learn is used by default and invoked in the *FeatureSelection* class, which is instantiated in the TAP class. Optional parameters also exist for feature and gene filtering prior to training, removing specified viral serotypes, marker genes, and other unwanted features. To address class imbalance, TAP supports several resampling techniques via the imbalanced-learn package, including SMOTE, ADASYN, Borderline-SMOTE, SVM-SMOTE, KMeans-SMOTE, and random oversampling.<sup>26,43–46</sup> Subsets with insufficient class diversity (e.g. only one class present) are skipped automatically.

The default parameters for training the *RandomForestClassifier* on each subset ( $X_c$ ) include `n_estimators=20`, `max_depth=2`, and `min_samples_split=2`, though all hyperparameters can be customized by the user. TAP also supports automated hyperparameter tuning via randomized grid search (*RandomizedSearchCV*), exploring combinations of tree depth, number of estimators, maximum features, and minimum samples per split. The *rfHyperParameterTune* parameter may be used to access the *RandomizedSearchCV* functionality, however, this will increase the runtime of the TAP build. Model performance is evaluated and stored using accuracy and AUC. Additionally, TAP can optionally compute permutation-based feature importances using Scikit Learn's `permutation_importance`, repeated two times, user-defined, or can be skipped entirely. This produces a ranked list of genes most predictive of infectivity for each categorical subset ( $X_c$ ). All feature related metrics are stored in JSON format and for downstream composition of the user interface.

## 3.4 Visualization Development

### 3.4.1 Overview of Interface Construction

TAP constructs a user interface in three parts. (1) The `AnnData` object undergoes extensive preprocessing, infectivity clustering, and ensemble feature selection, while caching the data results and base64 encoding supporting images into JSON format (**Methods 3.1-3.3**). (2) The preprocessed cell by gene matrix is converted to a CSV format. (3) Both JSON and CSV results are converted to one line representations, then pushed into a singular HTML output file which serves as a template for the visualization. The TAP visualization interface is built as a fully self-contained HTML output using Bootstrap and custom vanilla JavaScript. It supports interactive exploration of infectivity classifications and gene or virus level features across categorical cell subsets ( $X_c$ ).

### 3.4.2 Interactive User Interface Development

As the main component of the UI, a dynamic and responsive heatmap is generated directly client side by parsing all JSON and CSV components using the D3 javascript library.<sup>47</sup> The main heatmap is crafted by calculating the amount of columns necessary for each cell to fit in the heatmap container, then the amount of table columns and rows necessary to build the heatmap are generated, which is responsive to various browser sizes; however, is not intended to be mobile optimized. Each top category bar (horizontally displayed) and gene/vector (left column, vertically displayed) are assigned unique IDs. Javascript on-click listeners detect user click interactions with those identifiers and facilitate loading selection specific content from the JSON cache in each of the four panels below the main heatmap.

### 3.4.3 Selection Details and Metrics

When responding to an on-click event, the first actionable task is to rebuild the “Selection Details” lower left block of the web application. Javascript will set a selection state which denotes the user interactive choice ( $C^{(i)}$ ), then use it as a cache key to O(1) lookup details of corresponding metadata in the embedded JSON. The JSON provided details are used to update all values in the “Selection Details” UI element. Additionally, an image of the cluster results for the specific cell subset ( $X_c$ ) is shown. All clustering images described in **Methods 3.2** are base64 encoded and embedded into the JSON, so they can be rendered in HTML. This section augments the other UI components by providing contextual validation of each subset ( $X_c$ ) and UI components including: RF model parameters, accuracy, AUC; along with clustering results, and selection details which can be used as a first step toward evaluating results.

### 3.4.4 Ensemble Feature Importance, Differential Expression, and Highly Important Indicators

The two UI components, “Ensemble Features” and “Differential Expression” display gene lists constructed from each subset of cells ( $X_c$ ). Each component uses javascript to draw a HTML table and populate it with results from **Methods 3.1-3.3**, based on the current user selection. All listed gene results are hyperlinked directly to the Gene Cards verbose description.<sup>48</sup> Ensemble feature results are displayed in descending order importance (highest on top) and include a barplot as a visual indicator, along with yellow, orange, and red pepper icons to indicate potential relevance to tropism (**Figure 1D**). These three icons are determined by the following logic: (1) A yellow icon denotes a particular gene that exists in both the Ensemble Features and the Differential Expression lists for the current selection. (2) An orange icon indicates a specific gene is present across all replicates in both lists; and (3) a red icon flags genes whose descriptions are fuzzy matches for keywords relating to receptors, plasma membrane, transmembrane, adhesion, or an extracellular molecule. The keyword matching is accomplished upstream of the UI during **Methods 3.1-3.3** using gene descriptions retrieved from the BioThings API.<sup>49</sup> The approach simply parses the descriptions for the relevant keywords. Each icon level is then indicated in the JSON for a particular gene contained with the subset selection ( $X_c$ ).

### 3.4.5 Configurations and Parameters

Visual configurations options, including heatmap color themes and aesthetic visual UI styling were developed in pure javascript and are accessible via the gear icon in the interface. When selecting an option, Javascript updates html element IDs to apply the selected styles dynamically. The heatmap supports multiple themes that were developed to closely resemble popular Matplotlib color themes: Cividis, Viridis, Plasma, Inferno, Magma and a custom TAP theme that maps values from black to red.<sup>50</sup>

Although these settings exist and are directly accessible in the UI, additional configuration options, including deeper parameter control over aesthetics and functionality and can be accessed in our documentation found at: <https://github.com/ArpiarSaundersLab/tap>

### 3.5 Replicate Mode

TAP includes the boolean parameter “*replicate\_mode*” which can be used to process the dataset that considers replicates as individual entities. Replicate mode starts by examining categorical observations added by the user and searches for text indicating a category is part of a series of values (e.g. *replicate\_1*, *replicate\_2*, ... , *replicate\_n*). Replicate mode also searches genes/vectors for potential replicates in the same manner (e.g. *aav1\_bc1*, *aav1\_bc2*, ... , *aav1\_bcn*). Once each replicate is determined, they are processed in the main pipeline described in **Figure 1** and **Methods 3.1**, however, each subset ( $X_c$ ) now includes an additional term specifically for each replicate. Once important features are determined individually for each replicate, they are then examined for the intersection of features existing in each replicate. Common features are denoted as described in **Methods 3.4.4**. Replicates are then represented in the UI as categories in the top bar (**Figure 1D,2E**) and parsed directly from the embedded JSON object.

### 3.6 Performance and Usability

Building the HTML output of TAP requires the entire pipeline (**Figure 1B**) to be run recursively through all categorical combinatorial subsets of cells. TAP requires ~10Gb of memory for every 10,000 cells in the input dataset. TAP is multi-threaded and will use 2 CPUs by default. This value can be modified by utilizing the “*cpus*” parameter. Once the HTML output is built, the file can be opened in any modern browser. During the building process, all results are cached in the final output which makes the UI experience fluid on most modern computers.

### 3.7 Multiplexed AAV Barcoding, Primary Cortical Cell Culture, and AAV transduction.

AAV vectors were based on a plasmid backbone encoding EGFP fused to a nuclear localization signal (NLS) under the control of the CAG promoter. A unique 300 bp synthetic barcode was inserted into the 3' UTR of each construct to allow transcript-based discrimination of capsid variants. Barcode sequences were designed to minimize secondary structure and avoid homopolymer runs. Each barcoded plasmid was cloned individually and sequence-verified via Sanger or whole-plasmid sequencing. Recombinant AAVs were produced using 11 distinct capsid variants (**Figure 2A, S1A**), each packaging an identical genome differing only in barcode identity. Production and titration were performed by the UNC NeuroTools Viral Vector Core Facility. Viral preparations were pooled in equimolar proportions to a final concentration of  $1 \times 10^{12}$  vg/mL.

Cortical tissue was isolated from E16.5 C57BL/6J mouse embryos and dissociated to obtain a mixed population of brain cell types, including neurons, astrocytes, oligodendrocyte precursor cells (OPCs), and microglia. Cells were cultured in Neurobasal medium supplemented with B27 and maintained at 37°C with 5% CO<sub>2</sub>. At DIV7, cultures were transduced with the pooled AAV cocktail. At DIV14, cells were enzymatically dissociated using papain and filtered to obtain single-cell suspensions. Single-cell RNA-seq libraries were generated using the 10x Genomics Chromium Single-cell 3' v3.1 platform following the manufacturer's protocol.

Single-cell preprocessing was completed using the default Cell Ranger pipeline.<sup>51</sup> Ambient RNA signal and duplicate cell captures were removed using Cell Bender with the default parameters, adjusted only for expected cells of each library.<sup>52</sup> Barcodes were demultiplexed based on their unique 3' UTR barcode to determine which serotype transduced each individual cell. Using a custom script, barcodes were then added to the cell by gene matrix to be accessed downstream alongside the full gene panel. A standard Scanpy preprocessing pipeline was then followed for each dataset, along with PCA, integration using the Harmony<sup>21</sup>, followed by UMAP and manual cell type identification based on the Allen Brain Map Mouse Whole Cortex Markers.

### 3.8 Cell Typist Model Training

We provide automatic cell type prediction included with TAP using the Cell Typist package. Included with TAP are all default models provided by the package maintainers as well as two additional models we provide for convenience. The first model was trained using an adult mouse brain single-cell atlas<sup>23</sup> and the second is from a Rhesus macaque single-cell amygdala dataset<sup>53</sup>. For both models, normalization and scaling were performed as recommended by Cell Typist. A grid parameter search was then deployed to determine optimal model training parameters of the following: *alpha*, *epochs*, *use\_SGD*, *mini\_batch*, *with\_mean*, and *feature\_selection*. Trained models were then assessed using confidence scores provided by Cell Typist. Models with the mean highest confidence scores were selected.

## References

1. See, W.R., Yousefi, M., and Ooi, Y.S. (2024). A review of virus host factor discovery using CRISPR screening. *mBio* 15, e03205-23. <https://doi.org/10.1128/mbio.03205-23>.
2. McDougall, W.M., Perreira, J.M., Reynolds, E.C., and Brass, A.L. (2018). CRISPR genetic screens to discover host–virus interactions. *Curr. Opin. Virol.* 29, 87–100. <https://doi.org/10.1016/j.coviro.2018.03.007>.
3. Pillay, S., Meyer, N.L., Puschnik, A.S., Davulcu, O., Diep, J., Ishikawa, Y., Jae, L.T., Wosen, J.E., Nagamine, C.M., Chapman, M.S., et al. (2016). An essential receptor for adeno-associated virus infection. *Nature* 530, 108–112. <https://doi.org/10.1038/nature16465>.
4. Dudek, A.M., Zabaleta, N., Zinn, E., Pillay, S., Zengel, J., Porter, C., Franceschini, J.S., Estelien, R., Carette, J.E., Zhou, G.L., et al. (2020). GPR108 Is a Highly Conserved AAV Entry Factor. *Mol. Ther.* 28, 367–381. <https://doi.org/10.1016/j.ymthe.2019.11.005>.
5. Dhungel, B.P., Xu, H., Nagarajah, R., Vitale, J., Wong, A.C.H., Gokal, D., Feng, Y., Tabar, M.S., Metierre, C., Parsania, C., et al. (2025). An alternate receptor for adeno-associated viruses. *Cell* 188, 4924–4935.e23. <https://doi.org/10.1016/j.cell.2025.06.026>.
6. Li, B., Clohisey, S.M., Chia, B.S., Wang, B., Cui, A., Eisenhaure, T., Schweitzer, L.D., Hoover, P., Parkinson, N.J., Nachshon, A., et al. (2020). Genome-wide CRISPR screen identifies host dependency factors for influenza A virus infection. *Nat. Commun.* 11, 164. <https://doi.org/10.1038/s41467-019-13965-x>.
7. Biering, S.B., Sarnik, S.A., Wang, E., Zengel, J.R., Leist, S.R., Schäfer, A., Sathyan, V., Hawkins, P., Okuda, K., Tau, C., et al. (2022). Genome-wide bidirectional CRISPR screens identify mucins as host

- factors modulating SARS-CoV-2 infection. *Nat. Genet.* **54**, 1078–1089. <https://doi.org/10.1038/s41588-022-01131-x>.
8. Park, R.J., Wang, T., Koundakjian, D., Hultquist, J.F., Lamothe-Molina, P., Monel, B., Schumann, K., Yu, H., Krupczak, K.M., Garcia-Beltran, W., et al. (2017). A genome-wide CRISPR screen identifies a restricted set of HIV host dependency factors. *Nat. Genet.* **49**, 193–203. <https://doi.org/10.1038/ng.3741>.
  9. Su, C., Xu, Z., Shan, X., Cai, B., Zhao, H., and Zhang, J. (2023). Cell-type-specific co-expression inference from single cell RNA-sequencing data. *Nat. Commun.* **14**, 4846. <https://doi.org/10.1038/s41467-023-40503-7>.
  10. van Dam, S., Vösa, U., van der Graaf, A., Franke, L., and de Magalhães, J.P. (2018). Gene co-expression analysis for functional classification and gene–disease predictions. *Brief. Bioinform.* **19**, 575–592. <https://doi.org/10.1093/bib/bbw139>.
  11. Wolf, F.A., Angerer, P., and Theis, F.J. (2018). SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15. <https://doi.org/10.1186/s13059-017-1382-0>.
  12. Virshup, I., Rybakov, S., Theis, F.J., Angerer, P., and Wolf, F.A. (2021). anndata: Annotated data. Preprint at bioRxiv, <https://doi.org/10.1101/2021.12.16.473007>.
  13. Comparative analysis of structure, expression and PSD95-binding capacity of Lrln, a novel family of neuronal transmembrane proteins (2006). *Gene* **380**, 72–83. <https://doi.org/10.1016/j.gene.2006.05.014>.
  14. Cagno, V., Tseligka, E.D., Jones, S.T., and Tapparel, C. (2019). Heparan Sulfate Proteoglycans and Viral Attachment: True Receptors or Adaptation Bias? *Viruses* **11**, 596. <https://doi.org/10.3390/v11070596>.
  15. Ojala, D.S., Sun, S., Santiago-Ortiz, J.L., Shapiro, M.G., Romero, P.A., and Schaffer, D.V. (2018). In Vivo Selection of a Computationally Designed SCHEMA AAV Library Yields a Novel Variant for Infection of Adult Neural Stem Cells in the SVZ. *Mol. Ther.* **26**, 304–319. <https://doi.org/10.1016/j.ymthe.2017.09.006>.
  16. Bauer, A., Nolden, T., Nemitz, S., Perlson, E., and Finke, S. (2015). A Dynein Light Chain 1 Binding Motif in Rabies Virus Polymerase L Protein Plays a Role in Microtubule Reorganization and Viral Primary Transcription. *J. Virol.* **89**, 9591–9600. <https://doi.org/10.1128/jvi.01298-15>.
  17. Feige, L., Young, K., Cerapio, J.P., Kozaki, T., Kergoat, L., Libri, V., Ginhoux, F., Hasan, M., Ameer, L.B., Chin, G., et al. (2025). Human brain cell types shape host-rabies virus transcriptional interactions revealing a preexisting pro-viral astrocyte subpopulation. *Cell Rep.* **44**. <https://doi.org/10.1016/j.celrep.2025.116159>.
  18. Tan, G.S., Preuss, M.A.R., Williams, J.C., and Schnell, M.J. (2007). The dynein light chain 8 binding motif of rabies virus phosphoprotein promotes efficient viral transcription. *Proc. Natl. Acad. Sci.* **104**, 7229–7234. <https://doi.org/10.1073/pnas.0701397104>.

19. Jespersen, N.E., Leyrat, C., Gérard, F.C., Bourhis, J.-M., Blondel, D., Jamin, M., and Barbar, E. (2019). The LC8-RavP ensemble Structure Evinces A Role for LC8 in Regulating Lyssavirus Polymerase Functionality. *J. Mol. Biol.* *431*, 4959–4977. <https://doi.org/10.1016/j.jmb.2019.10.011>.
20. Pavan, K., and Saunders, A. (2025). AnnSQL: a Python SQL-based package for fast large-scale single-cell genomics analysis using minimal computational resources. *Bioinform. Adv.* *5*, vbaf105. <https://doi.org/10.1093/bioadv/vbaf105>.
21. Xu, C., Prete, M., Webb, S., Jardine, L., Stewart, B.J., Hoo, R., He, P., Meyer, K.B., and Teichmann, S.A. (2023). Automatic cell-type harmonization and integration across Human Cell Atlas datasets. *Cell* *186*, 5876–5891.e20. <https://doi.org/10.1016/j.cell.2023.11.026>.
22. Conde, C.D., Xu, C., Jarvis, L.B., Rainbow, D.B., Wells, S.B., Gomes, T., Howlett, S.K., Suchanek, O., Polanski, K., King, H.W., et al. (2022). Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science*. <https://doi.org/10.1126/science.abl5197>.
23. Langlieb, J., Sachdev, N.S., Balderrama, K.S., Nadaf, N.M., Raj, M., Murray, E., Webber, J.T., Vanderburg, C., Gazestani, V., Tward, D., et al. (2023). The molecular cytoarchitecture of the adult mouse brain. *Nature* *624*, 333–342. <https://doi.org/10.1038/s41586-023-06818-7>.
24. Pedregosa, F., Pedregosa, F., Varoquaux, G., Varoquaux, G., Org, N., Gramfort, A., Gramfort, A., Michel, V., Michel, V., Fr, L., et al. Scikit-learn: Machine Learning in Python. *Mach. Learn. PYTHON*.
25. Maximum Likelihood from Incomplete Data via the EM Algorithm on JSTOR <https://www-jstor-org.liboff.ohsu.edu/stable/2984875?seq=1>.
26. SMOTE: Synthetic Minority Over-sampling Technique | Journal of Artificial Intelligence Research <https://www.jair.org/index.php/jair/article/view/10302>.
27. Lemaître, G., Nogueira, F., and Aridas, C.K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *J. Mach. Learn. Res.* *18*, 1–5.
28. Zheng, X., Wu, B., Liu, Y., Simmons, S.K., Kim, K., Clarke, G.S., Ashiq, A., Park, J., Li, J., Wang, Z., et al. (2024). Massively parallel in vivo Perturb-seq reveals cell-type-specific transcriptional networks in cortical development. *Cell* *187*, 3236–3248.e21. <https://doi.org/10.1016/j.cell.2024.04.050>.
29. Saunders, A., Huang, K.W., Vondrak, C., Hughes, C., Smolyar, K., Sen, H., Philson, A.C., Nemesh, J., Wysoker, A., Kashin, S., et al. (2022). Ascertaining cells' synaptic connections and RNA expression simultaneously with barcoded rabies virus libraries. *Nat. Commun.* *13*, 6993. <https://doi.org/10.1038/s41467-022-34334-1>.
30. Zhang, R., Xu, G., Cao, L., Sun, Z., He, Y., Cui, M., Sun, Y., Li, S., Li, H., Qin, L., et al. (2019). Divergent engagements between adeno-associated viruses with their cellular receptor AAVR. *Nat. Commun.* *10*, 3760. <https://doi.org/10.1038/s41467-019-11668-x>.
31. Walkey, C.J., Snow, K.J., Bulcha, J., Cox, A.R., Martinez, A.E., Ljungberg, M.C., Lanza, D.G., Giorgi, M.D., Chuecos, M.A., Alves-Bezerra, M., et al. (2025). A comprehensive atlas of AAV tropism in the mouse. *Mol. Ther.* *33*, 1282–1299. <https://doi.org/10.1016/j.ymthe.2025.01.041>.
32. Tervo, D.G.R., Hwang, B.-Y., Viswanathan, S., Gaj, T., Lavzin, M., Ritola, K.D., Lindo, S., Michael, S., Kuleshova, E., Ojala, D., et al. (2016). A Designer AAV Variant Permits Efficient

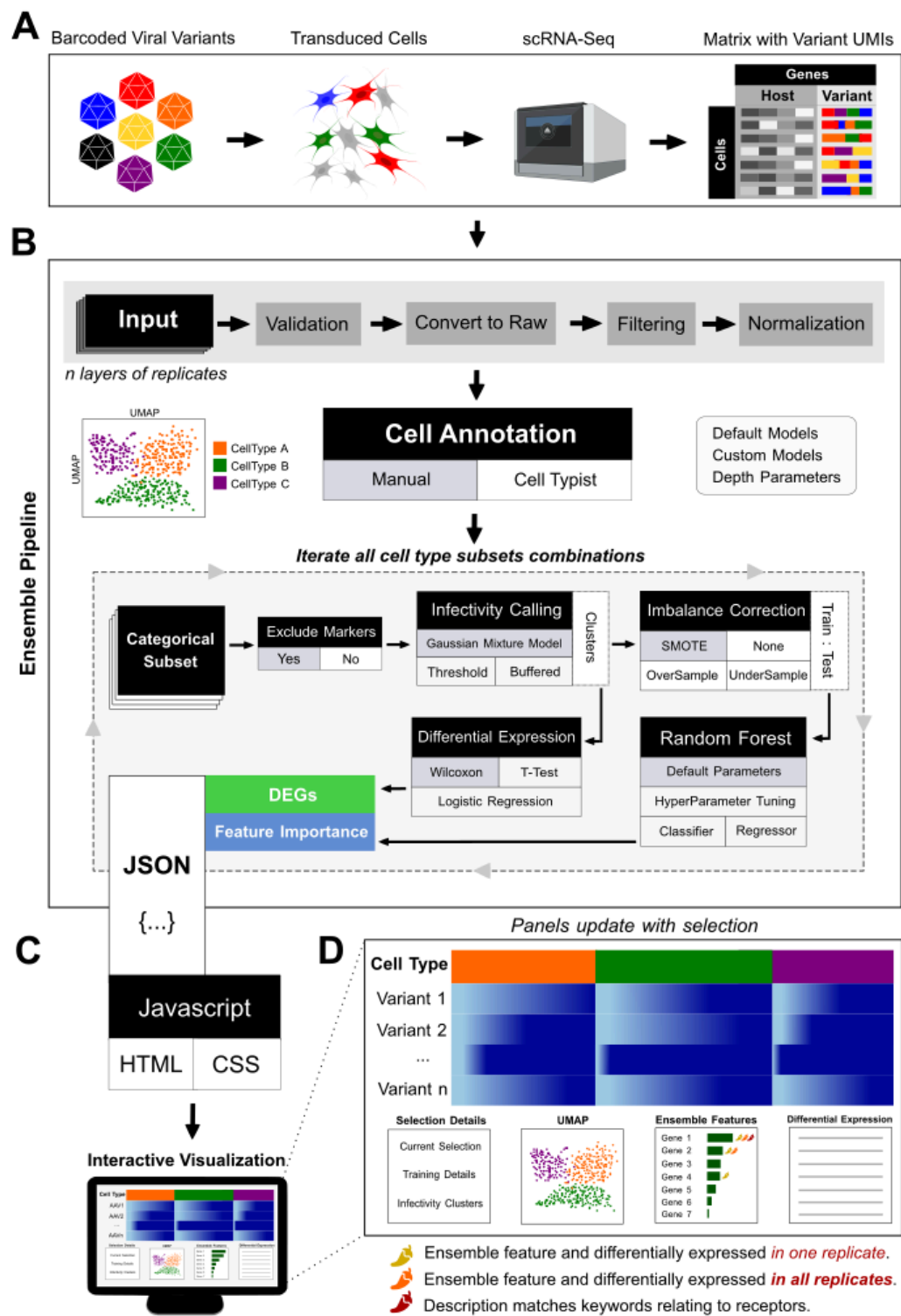
Retrograde Access to Projection Neurons. *Neuron* 92, 372–382. <https://doi.org/10.1016/j.neuron.2016.09.021>.

33. Klimczak, R.R., Koerber, J.T., Dalkara, D., Flannery, J.G., and Schaffer, D.V. (2009). A Novel Adeno-Associated Viral Variant for Efficient and Selective Intravitreal Transduction of Rat Müller Cells. *PLOS ONE* 4, e7467. <https://doi.org/10.1371/journal.pone.0007467>.
34. Zinn, E., Pacouret, S., Khaychuk, V., Turunen, H.T., Carvalho, L.S., Andres-Mateos, E., Shah, S., Shelke, R., Maurer, A.C., Plovie, E., et al. (2015). In Silico Reconstruction of the Viral Evolutionary Lineage Yields a Potent Gene Therapy Vector. *Cell Rep.* 12, 1056–1068. <https://doi.org/10.1016/j.celrep.2015.07.019>.
35. Wickersham, I.R., Finke, S., Conzelmann, K.-K., and Callaway, E.M. (2007). Retrograde neuronal tracing with a deletion-mutant rabies virus. *Nat. Methods* 4, 47–49. <https://doi.org/10.1038/nmeth999>.
36. Nam, J., Mah, W., and Kim, E. (2011). The SALM/Lrfr family of leucine-rich repeat-containing cell adhesion molecules. *Semin. Cell Dev. Biol.* 22, 492–498. <https://doi.org/10.1016/j.semcdb.2011.06.005>.
37. Choi, Y., Nam, J., Whitcomb, D.J., Song, Y.S., Kim, D., Jeon, S., Um, J.W., Lee, S.-G., Woo, J., Kwon, S.-K., et al. (2016). SALM5 trans-synaptically interacts with LAR-RPTPs in a splicing-dependent manner to regulate synapse development. *Sci. Rep.* 6, 26676. <https://doi.org/10.1038/srep26676>.
38. Kremer, L.P.M., Cerrizuela, S., Dehler, S., Stiehl, T., Weinmann, J., Abendroth, H., Kleber, S., Laure, A., Andari, J.E., Anders, S., et al. (2021). High throughput screening of novel AAV capsids identifies variants for transduction of adult NSCs within the subventricular zone. *Mol. Ther. Methods Clin. Dev.* 23, 33–50. <https://doi.org/10.1016/j.omtm.2021.07.001>.
39. Raux, H., Flamand, A., and Blondel, D. (2000). Interaction of the Rabies Virus P Protein with the LC8 Dynein Light Chain. *J. Virol.* 74, 10212–10216. <https://doi.org/10.1128/jvi.74.21.10212-10216.2000>.
40. Tan, G.S., Preuss, M.A.R., Williams, J.C., and Schnell, M.J. (2007). The dynein light chain 8 binding motif of rabies virus phosphoprotein promotes efficient viral transcription. *Proc. Natl. Acad. Sci.* 104, 7229–7234. <https://doi.org/10.1073/pnas.0701397104>.
41. Poisson, N., Real, E., Gaudin, Y., Vaney, M.-C., King, S., Jacob, Y., Tordo, N., and Blondel, D. (2001). Molecular basis for the interaction between rabies virus phosphoprotein P and the dynein light chain LC8: dissociation of dynein-binding properties and transcriptional functionality of P. *J. Gen. Virol.* 82, 2691–2696. <https://doi.org/10.1099/0022-1317-82-11-2691>.
42. Buccitelli, C., and Selbach, M. (2020). mRNAs, proteins and the emerging principles of gene expression control. *Nat. Rev. Genet.* 21, 630–644. <https://doi.org/10.1038/s41576-020-0258-4>.
43. He, H., Bai, Y., Garcia, E.A., and Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328. <https://doi.org/10.1109/IJCNN.2008.4633969>.

44. Han, H., Wang, W.-Y., and Mao, B.-H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In *Advances in Intelligent Computing*, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, eds. (Springer), pp. 878–887. [https://doi.org/10.1007/11538059\\_91](https://doi.org/10.1007/11538059_91).
45. Adaptive SV-Borderline SMOTE-SVM algorithm for imbalanced data classification (2024). *Appl. Soft Comput.* 150, 110986. <https://doi.org/10.1016/j.asoc.2023.110986>.
46. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE (2018). *Inf. Sci.* 465, 1–20. <https://doi.org/10.1016/j.ins.2018.06.056>.
47. Bostock, M., Ogievetsky, V., and Heer, J. (2011). D<sup>3</sup> Data-Driven Documents. *IEEE Trans. Vis. Comput. Graph.* 17, 2301–2309. <https://doi.org/10.1109/TVCG.2011.185>.
48. Stelzer, G., Rosen, N., Plaschkes, I., Zimmerman, S., Twik, M., Fishilevich, S., Stein, T.I., Nudel, R., Lieder, I., Mazor, Y., et al. The GeneCards Suite: From Gene Data Mining to Disease Genome Sequence Analyses. <https://doi.org/10.1002/cpbi.5>.
49. Lelong, S., Zhou, X., Afrasiabi, C., Qian, Z., Cano, M.A., Tsueng, G., Xin, J., Mullen, J., Yao, Y., Avila, R., et al. (2022). BioThings SDK: a toolkit for building high-performance data APIs in biomedical research. *Bioinformatics* 38, 2077–2079. <https://doi.org/10.1093/bioinformatics/btac017>.
50. Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* 9, 90–95. <https://doi.org/10.1109/MCSE.2007.55>.
51. Zheng, G.X.Y., Terry, J.M., Belgrader, P., Ryvkin, P., Bent, Z.W., Wilson, R., Ziraldo, S.B., Wheeler, T.D., McDermott, G.P., Zhu, J., et al. (2017). Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* 8, 14049. <https://doi.org/10.1038/ncomms14049>.
52. Fleming, S.J., Chaffin, M.D., Arduini, A., Akkad, A.-D., Banks, E., Marioni, J.C., Philippakis, A.A., Ellinor, P.T., and Babadi, M. (2023). Unsupervised removal of systematic background noise from droplet-based single-cell experiments using CellBender. *Nat. Methods* 20, 1323–1335. <https://doi.org/10.1038/s41592-023-01943-7>.
53. Totty, M.S., Juanes, R.C., Bach, S.V., Ameur, L.B., Valentine, M.R., Simons, E., Romac, M., Trinh, H., Henderson, K., Rosario, I.D., et al. (2024). Transcriptomic diversity of amygdalar subdivisions across humans and nonhuman primates. Preprint at bioRxiv, <https://doi.org/10.1101/2024.10.18.618721> <https://doi.org/10.1101/2024.10.18.618721>.

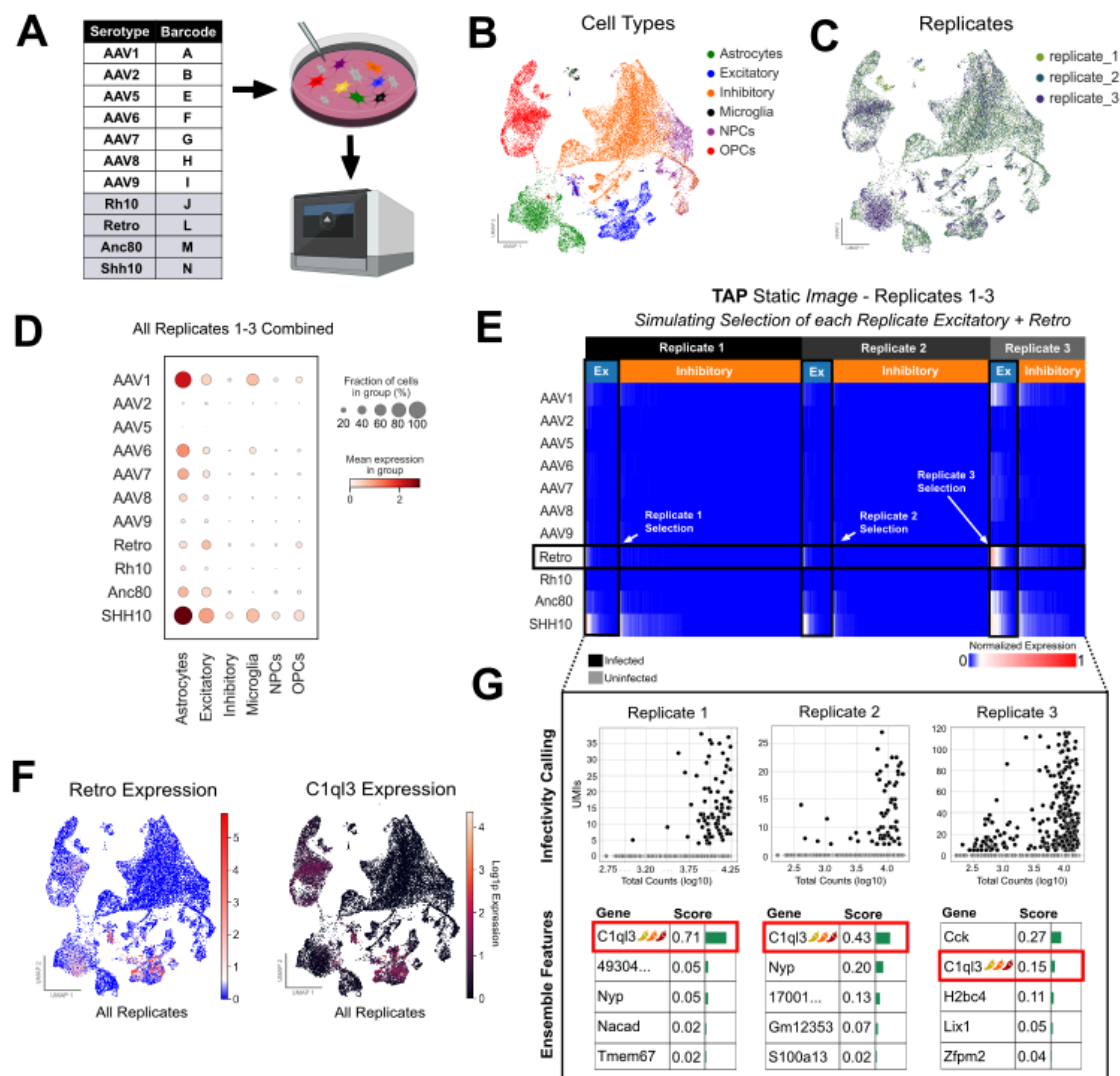


Figure 1



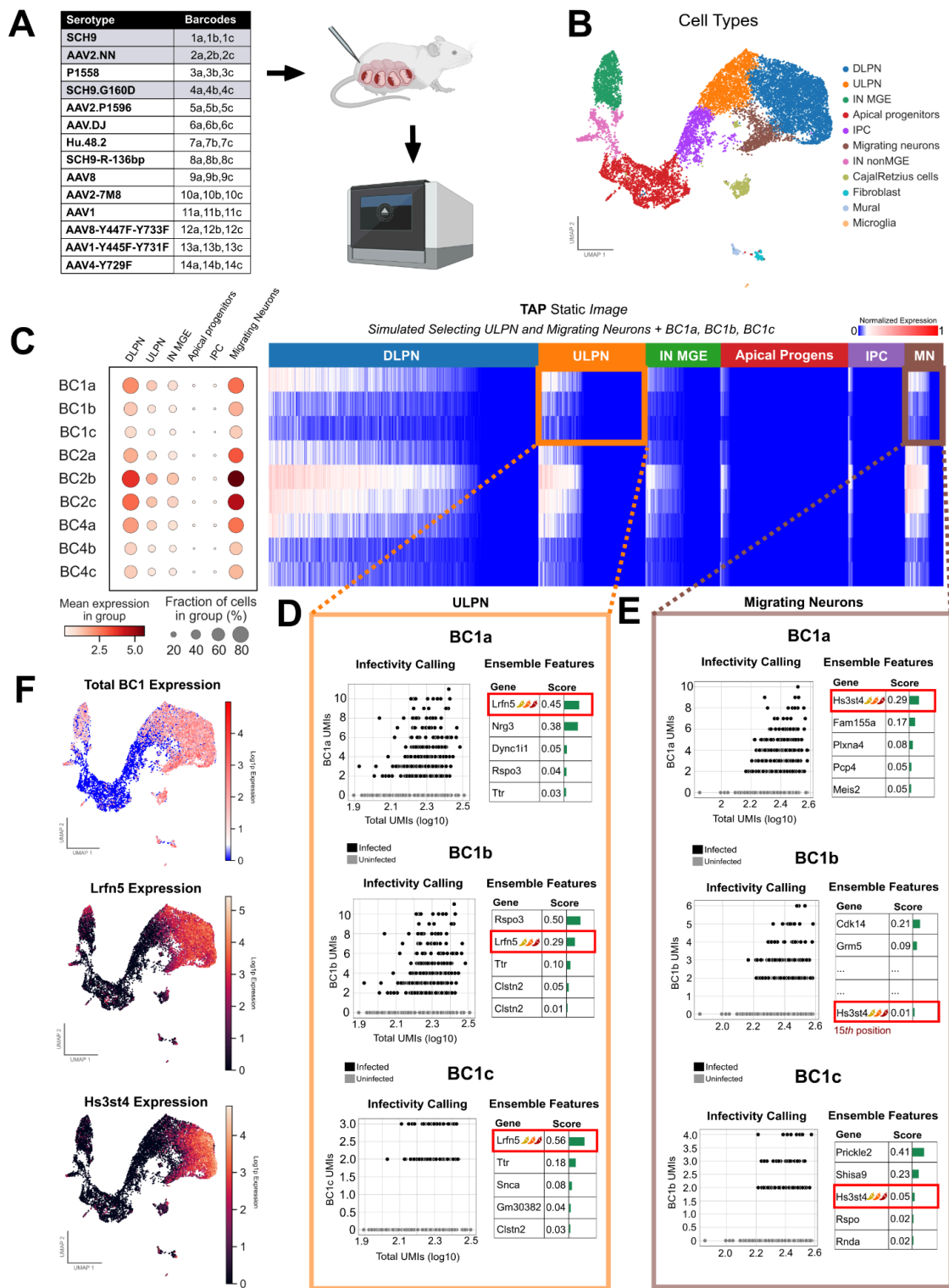
**Figure 1. Schematic of TAP ensemble pipeline and interactive user interface.** (A) Overview of the VOILÀ experimental workflow. Primary cells *in vitro* or *in vivo* are transduced by either a single virus or multiple viral variants harboring variant-specific RNA barcodes (shown). Standard single-cell RNA sequencing is then performed, resulting in a host/virus gene expression matrix in AnnData or AnnSQL format, which is compatible with the TAP build process. (B) Overview of the TAP ensemble machine learning pipeline for host factor discovery. Input data are pre-processed through validation, filtering, and cell library normalization steps. Cell Annotations are included in AnnData by *a priori* analysis or through assignment automatically using CellTypist models. The ensemble pipeline is then iteratively applied to each annotation category. The output of this process 1) Ensemble Feature Importance Scores and Differential Expression results for each expressed gene across infected and uninfected cell populations. Default parameters are highlighted in light gray. Results are stored in a JSON format. (C) Overview of the TAP user interface. The cached JSON results are parsed and composed into a custom user interface using HTML and CSS. (D) Key features of the interactive TAP user interface. Cell types horizontally span the top bar, while viral variants vertically span the left column. User selection of either cell type/viral variant categories dynamically filters the dataset and updates four results panels: (1) Selection details and model diagnostics; 2) UMAP plot of viral variant expression; 3) Ensemble Feature Importance Scores (rank ordered); and 4) Differential Expression test results. A colored pepper icon system is used to highlight genes of interest (yellow = differentially expressed in at least 1 replicate; orange = non-zero importance score and differentially expressed in all experimental replicates; red = potential membrane receptors or known viral function).

Figure 2



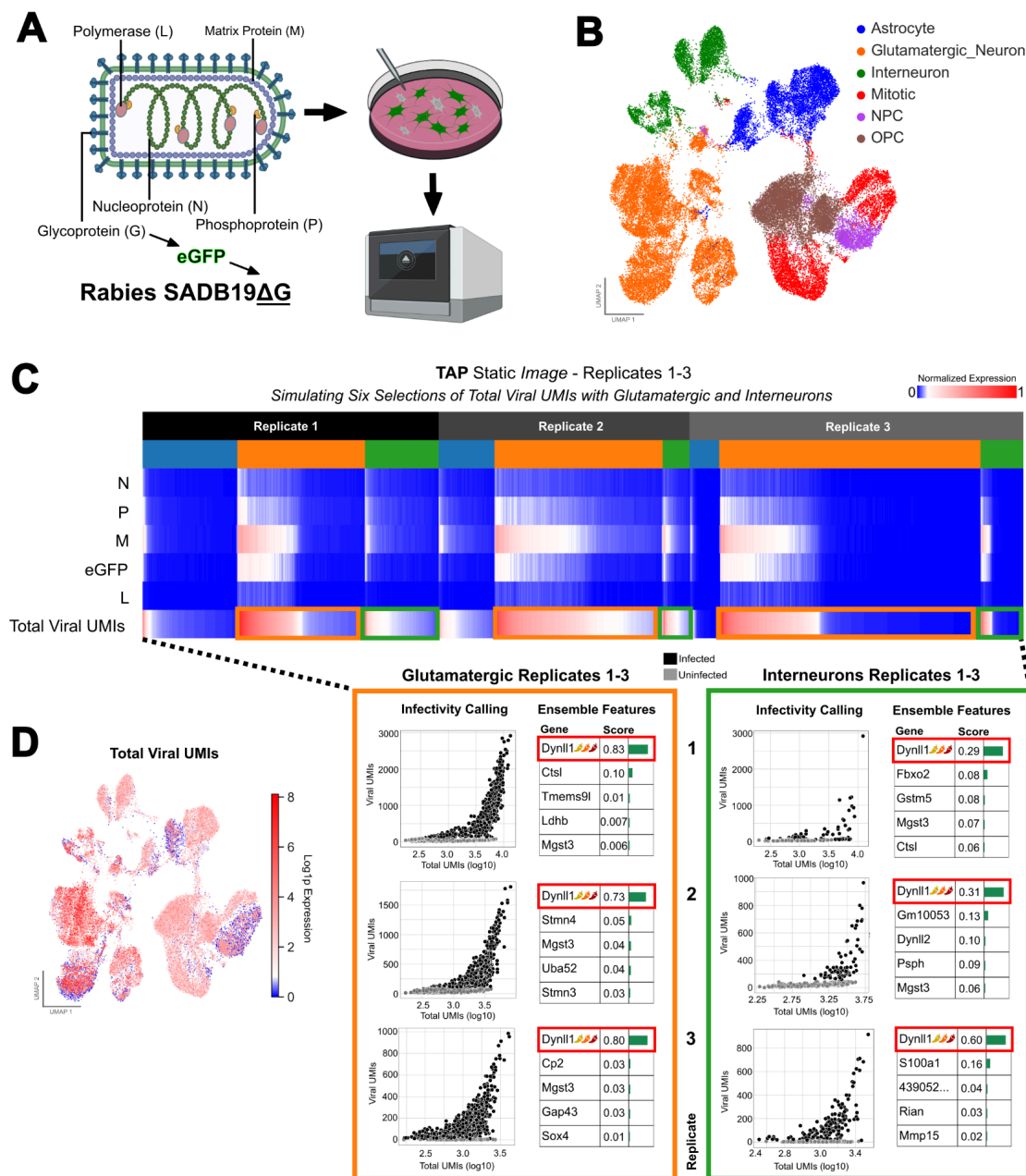
**Figure 2. TAP analysis of 11 AAV capsids using mouse brain cells *in vitro* nominates C1QL3 as a host factor for AAV2.Retro** (A) 11 Individually barcoded serotypes are introduced to primary cell culture, single-cell RNA sequencing is performed, then preprocessed as defined in **Methods**. (B) UMAP of 3 combined replicates displaying cell type annotations (**Methods**). (C) UMAP integration colored by each replicate. (D) Dotplot of all replicates combined showing each variant and cell type annotation. (E) TAP static image showcasing its ability to nest categorical replicate (3) and cell type observations. Each replicate is represented in the horizontal top bar, while cell types are in the second bar. Individual serotypes are stacked vertically on the left of the plot. The static image is simulating the selection of Excitatory Neurons in Replicate 1 and 2. (F) UMAP plots of the serotype AAV2.Retro and its top feature C1ql3 show colocalization in Excitatory Neurons. (G) Top: Infectivity clustering of replicate 1 and 2 selections from (E). Showing infected cells in black and uninfected cells in gray. Bottom: The top five features displayed in the TAP UI for each of the two selections. Notably, *C1ql3* contains three pepper icons which indicates the feature is present in all replicates and matches for keywords related to tropism (**Methods**).

Figure 3



**Figure 3. TAP analysis of 14 AAV capsids in developing mouse brain nominates HS3ST4 and LRFN5 as host factors for AAV.SCH9** (A) Existing dataset where 14 individual serotypes, with 3 barcodes per serotype are introduced to the lateral ventricles of a pregnant adult mouse, then sequenced and processed as defined in Zheng et al 2024. This dataset was then processed using TAP (**Methods**). (B) UMAP projection of 14,835 cells and their corresponding cell type annotations. (C) (Left) Dotplot generated using Scanpy of each of the 3 barcodes for BC1, BC2, and BC4 serotypes. (Right) Static export of TAP dynamic heatmap with manual highlighted selections of BC1a + ULPNs and BC1a + Migrating Neurons. (D) (Top) Infectivity calling of BC1a + ULPN selection along with top ensemble features determined from model training. Lrnf5 is a top candidate in BC1a along with both other barcodes for BC1. (Middle) UMAP projection displaying Lrnf5 expression in all cells. (Bottom) Percentage composition of BC1 and Lrnf5 in ULPNs. (E) (Top) Infectivity calling of BC1a + Migrating Neurons, along with top ensemble features determined from model training. (Middle) UMAP expression of top Hs3st4, the top ensemble candidate feature. (Bottom) Percent composition of BC1 and Hs3st4 in Migrating Neurons. (F) (Top) UMAP projection of the sum of all BC1 barcodes. (Bottom) Violin plots showing individual expression profiles of BC1a, BC1b, and BC1c.

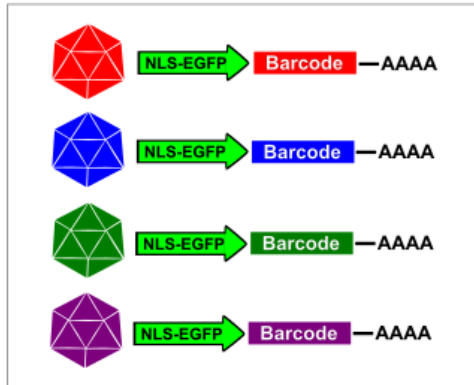
Figure 4



**Figure 4: Rabies Virus Tropism in Primary Cell Culture**  
(A) Rabies SADB19ΔG introduced into primary cells, then cellular states recorded via single-cell sequencing.  
(B) Cell annotations projected into UMAP space. (C) Static export image of TAP illustrating Glutamatergic and Interneuron selections across each replicate, each coupled with “Total Viral UMIs.” Bottom left displaying each Glutamatergic replicate infectivity clustering results, along with top 5 ensemble features of each replicate. Bottom right showing each replicate of Interneurons, along with top 5 ensemble features. (D) UMAP projection showing Total Viral UMI expression (ln). (D) UMAP showing expression of total viral UMI counts.

## Supplemental 1

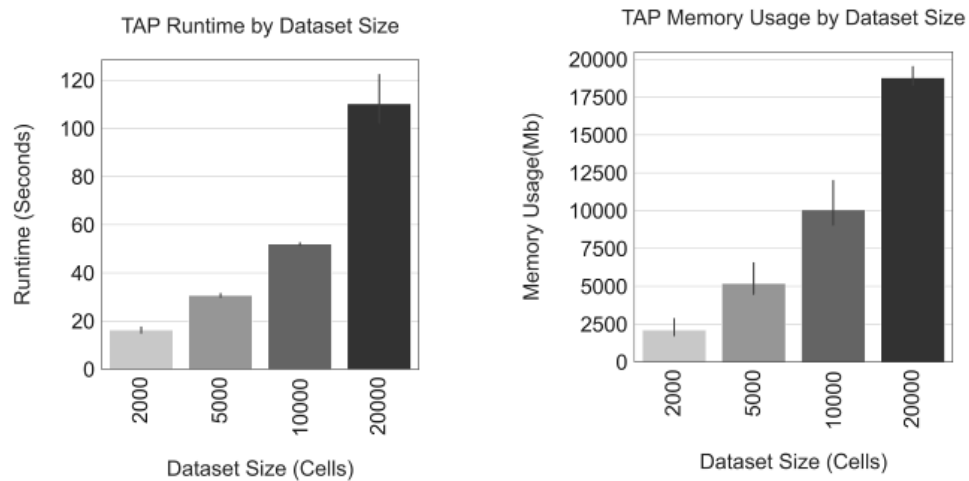
**A**



**B**

```
1 import tap as t
2
3 parameters = {
4     "filename": "/path/to/input/file/filename.h5ad",
5     "name": "AAV Tropism Interactive UI",
6     "genes": ["AAV2", "AAV5", "AAV6", "AAV7", "AAV8", "AAV9", ...],
7     "categories": ["replicates", "cell_type"],
8     "exclude": ["Microglia", "NPCs", "OPCs"],
9     "outputPath": "/path/to/output/",
10    "outputName": "filename.html",
11    "excludeMarkers": True,
12    "clusterMethod": "threshold",
13    "clusterThreshold": 1,
14    "balance": "smote",
15    "rfPermuteFeatureImportance": True,
16 }
17
18 results = t.TAP(**parameters)
```

**C**



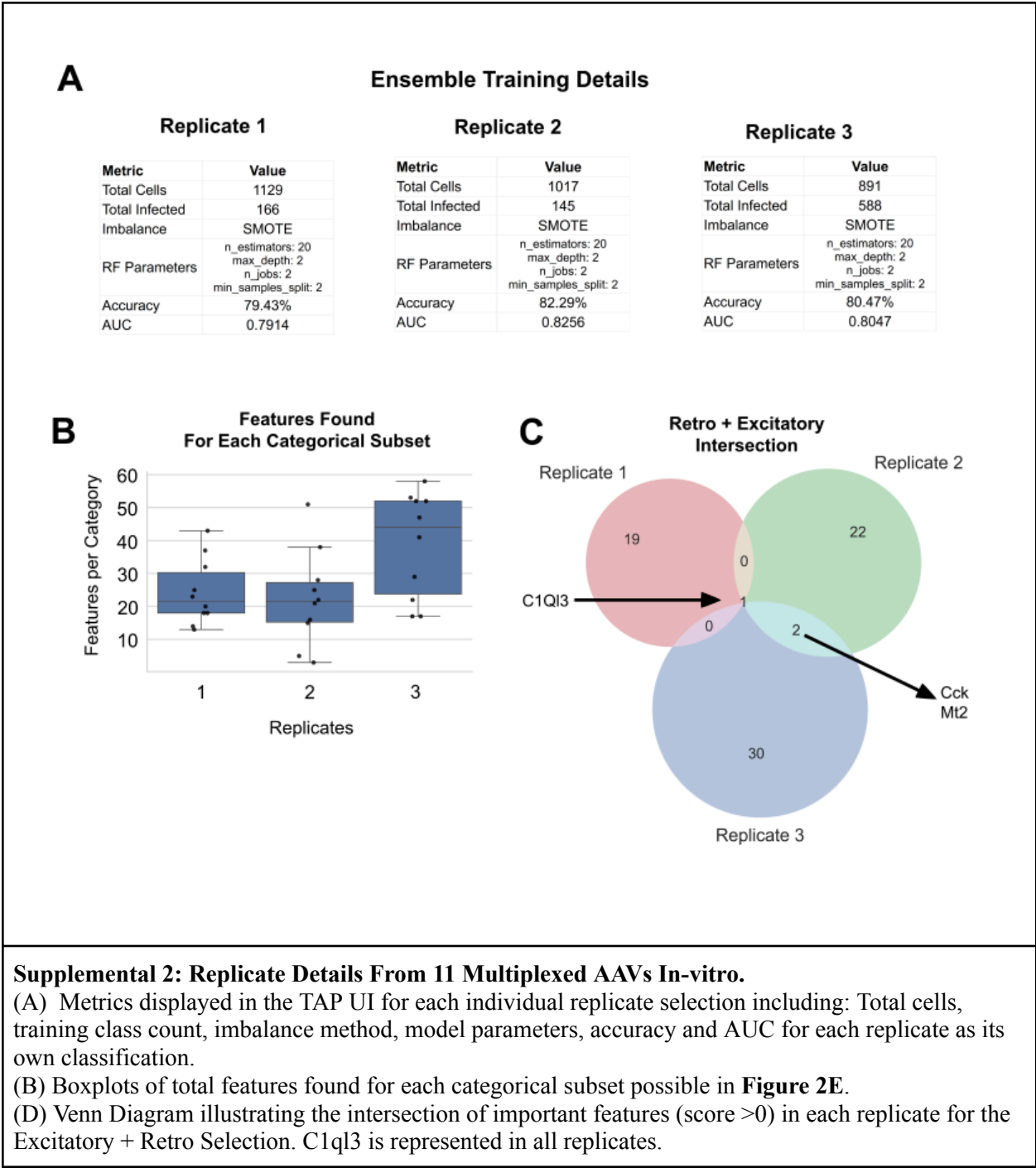
**Supplemental 1: TAP multiplexed configuration and Build Example.**

(A) Illustration of multiplexed AAV configuration showing each serotype with a unique orthogonal barcode following the eGFP reporter.

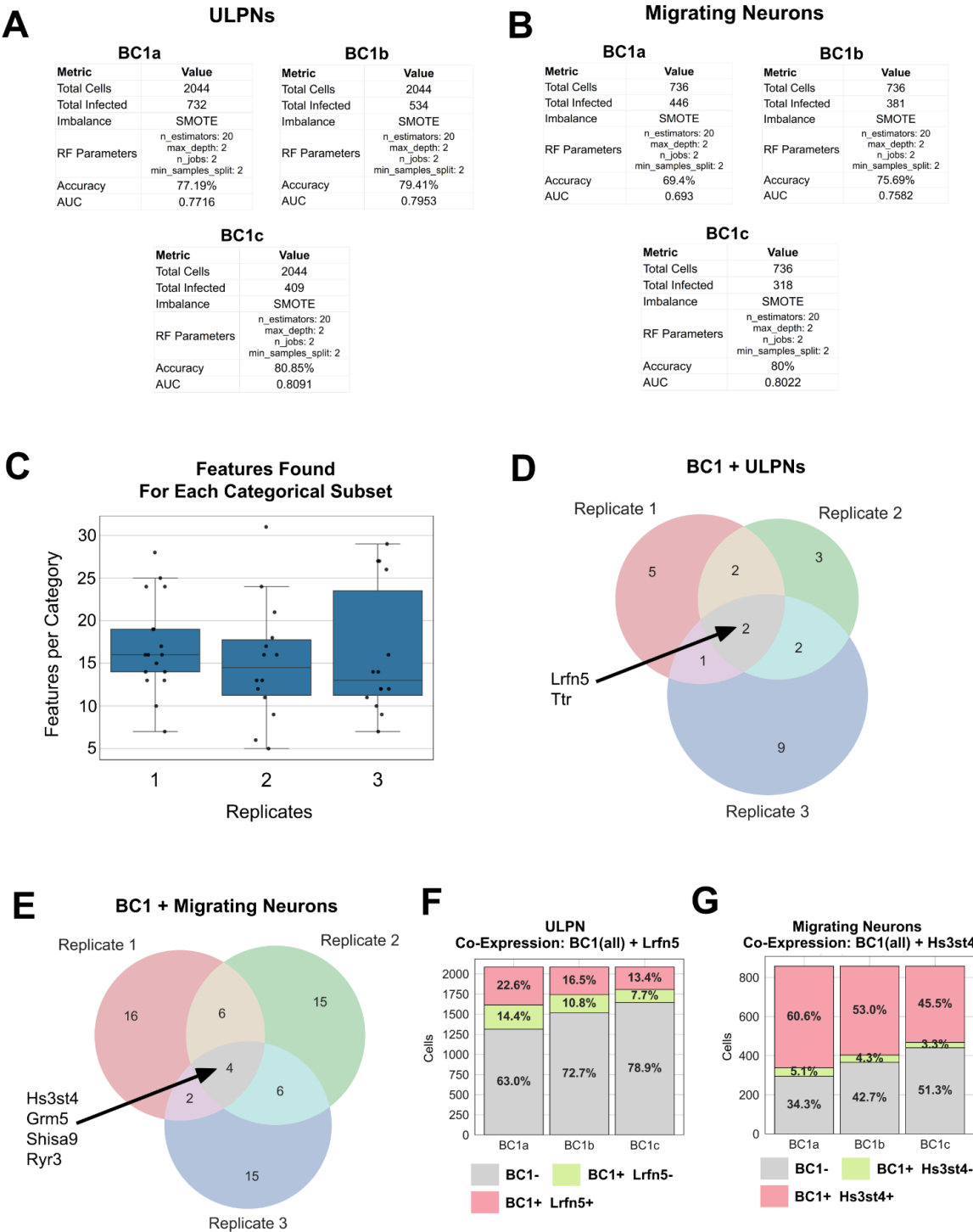
(B) A simplistic TAP build example with various parameters instantiated with the TAP object.

(C) Left: Runtime (seconds) of building a TAP with single-cell library sizes from 2,000 to 20,000 using one category (Cell Type). Right: Memory usage profiles in Mb for the same single-cell library sizes ( $n=5$  trials for each dataset size). Benchmarks were completed on a laptop with the following specifications: 40Gb RAM, 12th Gen Intel® Core™ i7-1255U  $\times$  12, and Ubuntu 24.04.

Supplemental 2



Supplemental 3



Supplemental 3: In-Vivo Replicate Details

(A & B) Metrics displayed in the TAP UI for each individual replicate selection of BC1 + ULPNs and BC1 + Migrating Neurons including: Total cells, training class count, imbalance method, model

parameters, accuracy and AUC for each replicate as its own classification.

(C) Boxplots of total features found for each categorical subset possible in **Figure 3E**.

(D) Venn Diagram illustrating the intersection of important features (score >0) in each replicate for the BC1 + ULPN Selection.

(E) Venn Diagram illustrates the intersection of important features (score >0) in each replicate for the BC1 + Migrating Neurons Selection.

(F) Coexpression barchart of all BC1 barcodes and Lrfn5. Stacked bars represent BC1- condition (gray), BC1+ Lrfn5- (green), and BC1+ Lrfn5+ (red).

(G) Coexpression barchart of all BC1 barcodes and Hs3st4. Stacked bars represent BC1- condition (gray), BC1+ Hs3st4- (green), and BC1+ Hs3st4+ (red).

Supplemental 4

A

Total Viral UMIs + Glutamatergic

| Metric              | Value                                                                 |
|---------------------|-----------------------------------------------------------------------|
| Total Cells         | 1848                                                                  |
| Total Infected (>0) | 1848                                                                  |
| Infection Calling   | Gaussian                                                              |
| RF Parameters       | n_estimators: 20<br>max_depth: 2<br>n_jobs: 2<br>min_samples_split: 2 |
| MSE                 | 62587                                                                 |

| Metric              | Value                                                                 |
|---------------------|-----------------------------------------------------------------------|
| Total Cells         | 2414                                                                  |
| Total Infected (>0) | 2414                                                                  |
| Infection Calling   | Gaussian                                                              |
| RF Parameters       | n_estimators: 20<br>max_depth: 2<br>n_jobs: 2<br>min_samples_split: 2 |
| MSE                 | 23366                                                                 |

| Metric              | Value                                                                 |
|---------------------|-----------------------------------------------------------------------|
| Total Cells         | 3762                                                                  |
| Total Infected (>0) | 3572                                                                  |
| Infection Calling   | Gaussian                                                              |
| RF Parameters       | n_estimators: 20<br>max_depth: 2<br>n_jobs: 2<br>min_samples_split: 2 |
| MSE                 | 4538                                                                  |

Total Viral UMIs + Interneurons

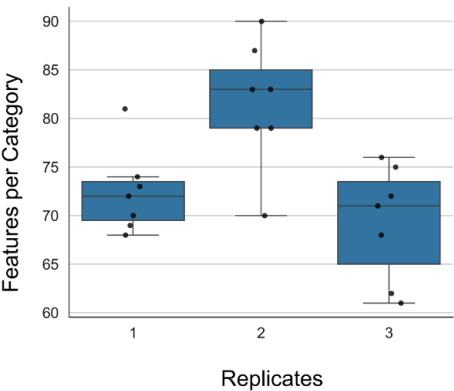
| Metric              | Value                                                                 |
|---------------------|-----------------------------------------------------------------------|
| Total Cells         | 1059                                                                  |
| Total Infected      | 1059                                                                  |
| Total Infected (>0) | Gaussian                                                              |
| RF Parameters       | n_estimators: 20<br>max_depth: 2<br>n_jobs: 2<br>min_samples_split: 2 |
| MSE                 | 1059                                                                  |

| Metric              | Value                                                                 |
|---------------------|-----------------------------------------------------------------------|
| Total Cells         | 391                                                                   |
| Total Infected (>0) | 391                                                                   |
| Infection Calling   | Gaussian                                                              |
| RF Parameters       | n_estimators: 20<br>max_depth: 2<br>n_jobs: 2<br>min_samples_split: 2 |
| MSE                 | 22261                                                                 |

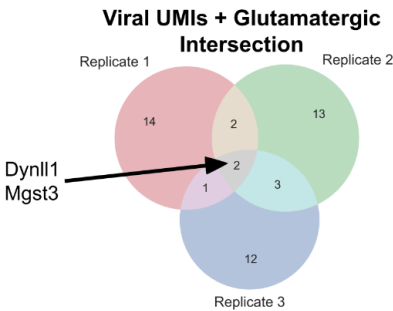
| Metric              | Value                                                                 |
|---------------------|-----------------------------------------------------------------------|
| Total Cells         | 615                                                                   |
| Total Infected (>0) | 578                                                                   |
| Infection Calling   | Gaussian                                                              |
| RF Parameters       | n_estimators: 20<br>max_depth: 2<br>n_jobs: 2<br>min_samples_split: 2 |
| MSE                 | 5276                                                                  |

B

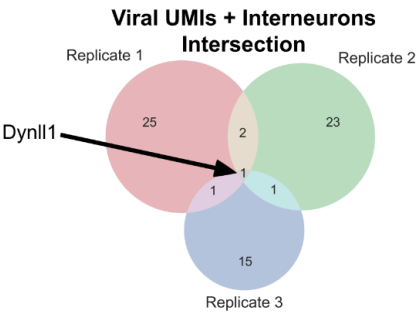
Features Found For Each Categorical Subset



C



D



#### **Supplemental 4: Rabies SADB19ΔG Replicate Details**

(A) Metrics displayed in the TAP UI for each individual replicate selection of Total Viral UMIs + Glutamatergic (Left) and Total Viral UMIs + Interneurons (Right) including: Total cells, training class count, imbalance method, model parameters, accuracy and AUC for each replicate as its own classification.

(B) Boxplots of total features found for each categorical subset possible in **Figure 4C**.

(C) Venn Diagram illustrates the intersection of important features (score >0) in each replicate for the Total Viral UMIs + Glutamatergic Selection. Showing on DYNLL1 at the intersection of all 3 replicates.

(D) Venn Diagram illustrates the intersection of important features (score >0) in each replicate for the Total Viral UMIs + Interneurons Selection. Showing on DYNLL1 and MGST3 at the intersection of all 3 replicates.