Scaling up reproducible research for single cell transcriptomics using MetaNeighbor (Procedure 2)

Procedure 2: Assessing cell type replicability against a pre-trained reference taxonomy

Procedure 2 demonstrates how to assess cell types of a newly annotated dataset against a reference cell type taxonomy. Here we consider the cell type taxonomy established by the Brain Initiative Cell Census Network (BICCN) in the mouse primary motor cortex. The BICCN taxonomy was defined across a compendium of datasets sampling across multiple modalities (transcriptomics and epigenomics), it constitutes one of the richest neuronal resources currently available. When matching against a reference taxonomy, we assume that the reference is of higher resolution than the query dataset, i.e. the query dataset samples the same set or a subset of cells compared to the reference.

Pre-train a reference MetaNeighbor model (1-5 minutes)

1. Start by loading an already merged SCE object containing the BICCN dataset. The full code for generating the dataset is available on Github, the dataset can be downloaded directly on FigShare.

```
library(SingleCellExperiment)

biccn_data = readRDS("full_biccn_hvg.rds")
dim(biccn_data)
```

```
## [1]    319 482712
```

```
colnames(colData(biccn_data))
```

```
##  [1] "sample_id"            "cluster_id"           "cluster_label"
##  [4] "subclass_label"       "class_label"          "cluster_color"
##  [7] "size"                 "passed_qc"            "joint_cluster_id"
## [10] "joint_cluster_label"  "joint_cluster_color"  "joint_subclass_id"
## [13] "joint_subclass_label" "joint_subclass_color" "joint_class_id"
## [16] "joint_class_label"    "joint_class_color"    "joint_cl"
## [19] "joint_cluster_size"   "joint_tree_order"     "study_id"
```

```
table(biccn_data$study_id)
```

```
##
##  scCv2  scCv3   scSS  snCv2 snCv3M snCv3Z   snSS
## 122641  71183   6288  76525 159738  40166   6171
```

The BICCN data contains 7 datasets totaling 482,712 cells. There are multiple sets of cell type labels depending on resolution (class, subclass, cluster) or type of labels (independent labels or labels defined from joint clustering). Note that, to reduce memory usage, we have already computed and restricted the dataset to a set of 319 highly variable genes.

2. Create pre-trained models with the `trainModel` function, which has identical parameters as the `MetaNeighborUS` function used in Protocol 1. Here, we choose to focus on two sets of cell types:

subclasses from the joint clustering (medium resolution, e.g., Vip interneurons, L2/3 IT excitatory neurons), and clusters from the joint clustering (high resolution, e.g., Chandelier cells). Create and store pre-trained models at the subclass level, then at the cluster level:

```
library(MetaNeighbor)

pretrained_model = MetaNeighbor::trainModel(
    var_genes = rownames(biccn_data), dat = biccn_data,
    study_id = biccn_data$study_id, cell_type = biccn_data$joint_subclass_label
)
write.table(pretrained_model, "pretrained_biccn_subclasses.txt")

pretrained_model = MetaNeighbor::trainModel(
    var_genes = rownames(biccn_data), dat = biccn_data,
    study_id = biccn_data$study_id, cell_type = biccn_data$joint_cluster_label
)
write.table(pretrained_model, "pretrained_biccn_clusters.txt")
```

For simplicity of use, we store the pretrained models to file using the `write.table` function.

The remainder of the procedure is independent and can be run in a new R session.

Compare annotations to pre-trained taxonomy (1 minute)

3. Start by loading the query dataset (Tasic 2016, neurons from mouse primary visual cortex, available in the scRNAseq package) and the pre-trained subclass and cluster taxonomies:

```
library(scRNAseq)
tasic = TasicBrainData(ensembl = FALSE)
tasic$study_id = "tasic"
biccn_subclasses = read.table("pretrained_biccn_subclasses.txt", check.names = FALSE)
biccn_clusters = read.table("pretrained_biccn_clusters.txt", check.names = FALSE)
```

We add a "study_id" column to the Tasic metadata, as this information will be needed later by MetaNeighbor. Note the "check.names = FALSE" argument when reading a pre-trained model, which is required to preserve the correct formatting of MetaNeighbor cell type names.

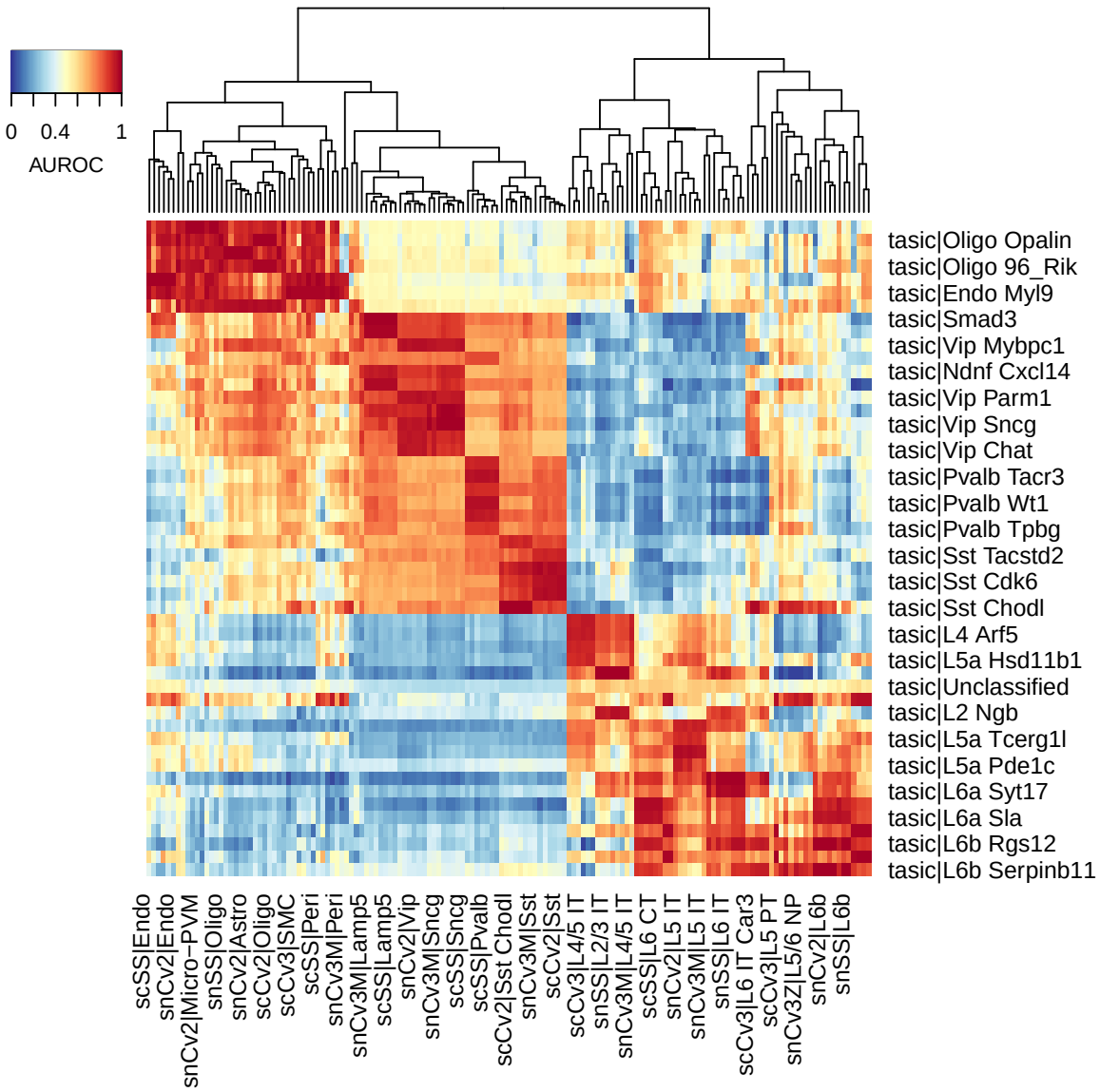
4. To run MetaNeighbor, we use the `MetaNeighborUS` function but, compared to Procedure 1, we provide a pre-trained model instead of a set of highly variable genes (which are already contained in the pre-trained model). Start by checking if Tasic cell types are consistent with the BICCN subclass resolution:

```
library(MetaNeighbor)

aurocs = MetaNeighborUS(
  trained_model = biccn_subclasses, dat = tasic,
  study_id = tasic$study_id, cell_type = tasic$primary_type,
  fast_version = TRUE
)
```

5. Visualize AUROCs as a rectangular heatmap, with the reference taxonomy cell types as columns and query cell types as rows:

```
plotHeatmapPretrained(aurocs)
```

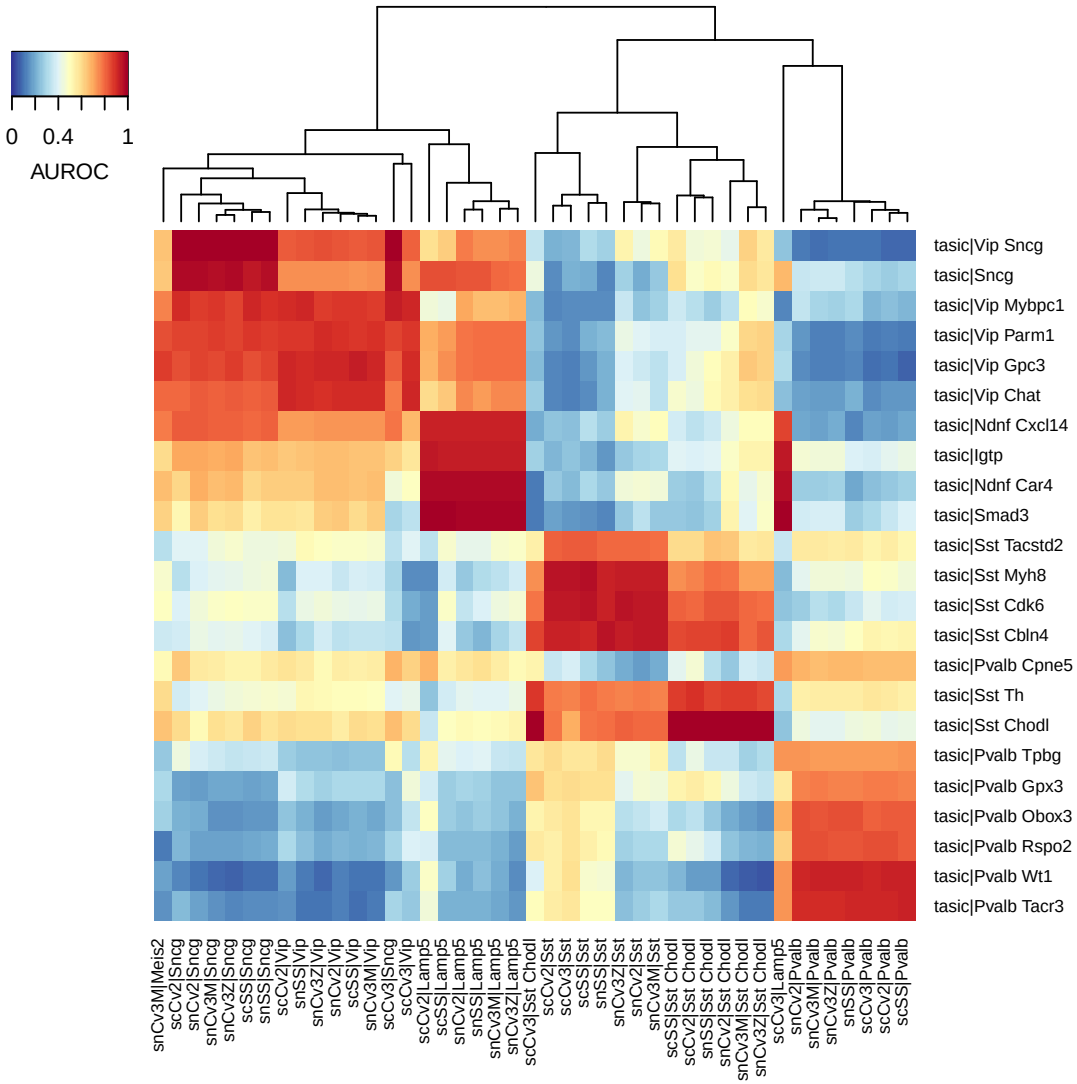


As in Procedure 1, we start by looking for evidence of global structure in the dataset. Here we recognize 3 red blocks, which correspond to non-neurons (top left), inhibitory neurons (middle) and excitatory neurons (bottom right). The presence of sub-blocks inside the 3 global blocks suggest that cell types can be matched more finely. For example, inside the inhibitory block, we can recognize sub-blocks corresponding to CGE-derived interneurons (Vip, Sncg and Lamp5 in the BICCN taxonomy) and MGE-derived interneurons (Pvalb and Sst in the BICCN taxonomy).

6. Refine AUROCs by focusing on inhibitory neurons using the `splitTrainClusters` and `splitTestClusters` utility functions to select the relevant cell types:

```
gabaergic_tasic = splitTestClusters(aurocs, k = 4)[[2]]
gabaergic_biccn = splitTrainClusters(aurocs[gabaergic_tasic,], k = 4)[[4]]
keep_cell = makeClusterName(tasic$study_id, tasic$primary_type) %in% gabaergic_tasic
tasic_subdata = tasic[, keep_cell]
aurocs = MetaNeighborUS(
  trained_model = biccn_subclasses[, gabaergic_biccn],
  dat = tasic_subdata, study_id = tasic_subdata$study_id,
  cell_type = tasic_subdata$primary_type, fast_version = TRUE
```

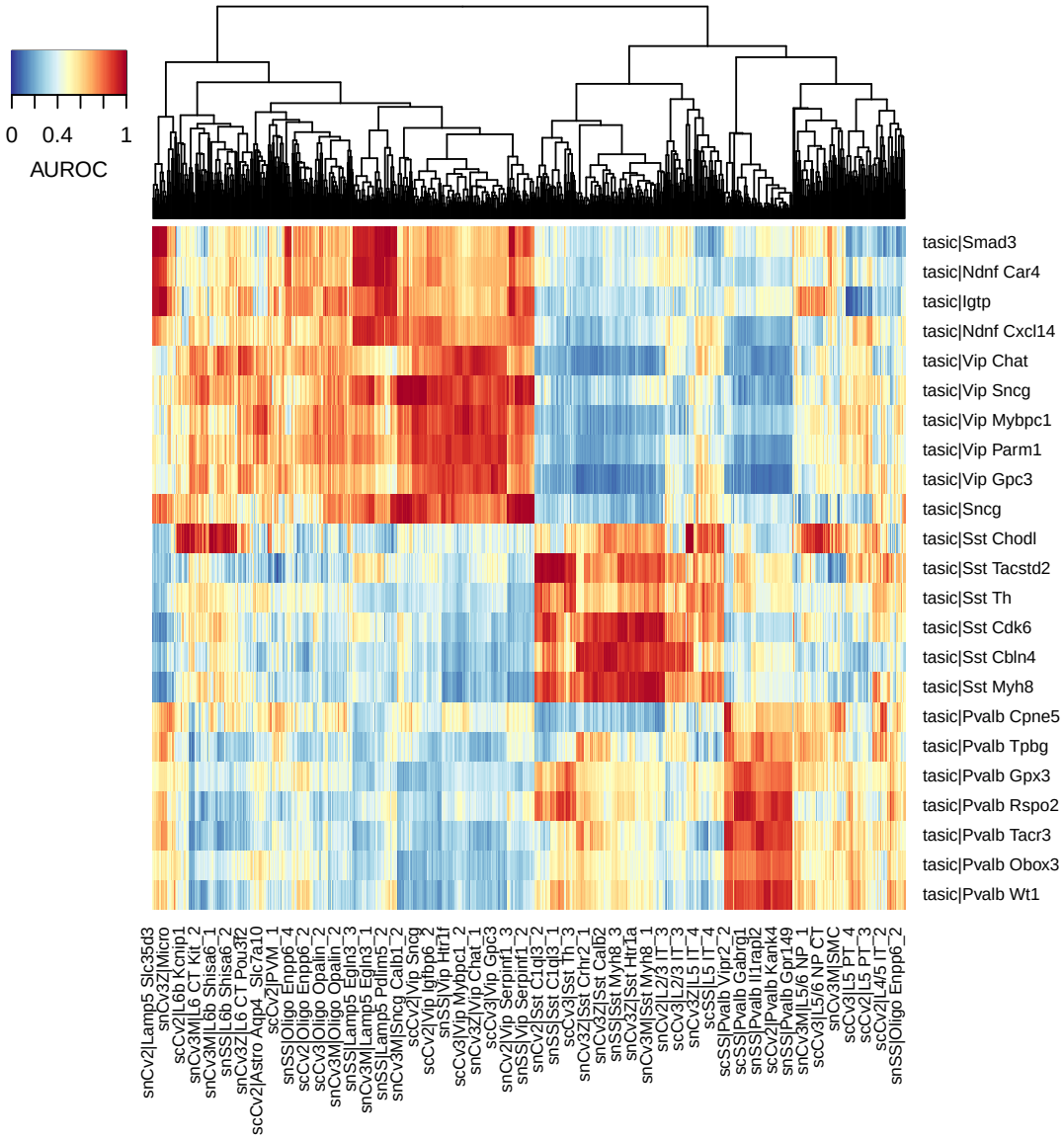
```
)
plotHeatmapPretrained(aurocs, cex = 0.7)
```



The heatmap suggests that there is a broad agreement at the subclass level between the BICCN MOp taxonomy and the Tasic 2016 dataset. For example, the Ndnf subtypes, Igtp and Smad3 cell types from the Tasic dataset match with the BICCN Lamp5 subclass.

7. The previous heatmaps suggest that all Tasic cell types can be matched with one BICCN subclass. We now go one step further and ask whether inhibitory cell types correspond to one of the BICCN clusters. Compute and visualize cell type similarity:

```
aurocs = MetaNeighborUS(trained_model = biccn_clusters,
                        dat = tasic_subdata,
                        study_id = tasic_subdata$study_id,
                        cell_type = tasic_subdata$primary_type,
                        fast_version = TRUE)
plotHeatmapPretrained(aurocs, cex = 0.7)
```



Here the heatmap is difficult to interpret due to the large number of BICCN cell types (output omitted here). Instead, investigate the top hits for each query cell type directly:

```
head(sort(aurocs["tasic|Sst Chodl",], decreasing = TRUE), 10)
```

```
##  scCv2|Sst Chodl  scCv3|Sst Chodl   scSS|Sst Chodl  snCv2|Sst Chodl
##        1.0000000        1.0000000        1.0000000        1.0000000
## snCv3M|Sst Chodl snCv3Z|Sst Chodl   snSS|Sst Chodl   scCv3|L6b Ror1
##        1.0000000        1.0000000        1.0000000        0.9960366
##    scSS|L6b Ror1  snCv3M|L6b Ror1
##        0.9947832        0.9944783
```

```
head(sort(aurocs["tasic|Pvalb Cpne5",], decreasing = TRUE), 10)
```

```
##  snCv2|Pvalb Vipr2_2  scCv2|Pvalb Vipr2_2   scSS|Pvalb Vipr2_2
##            0.9564926            0.9563014            0.9534328
## snCv3Z|Pvalb Vipr2_2   snSS|Pvalb Vipr2_2  scCv3|Pvalb Vipr2_2
##            0.9392809            0.9375598            0.9297189
```

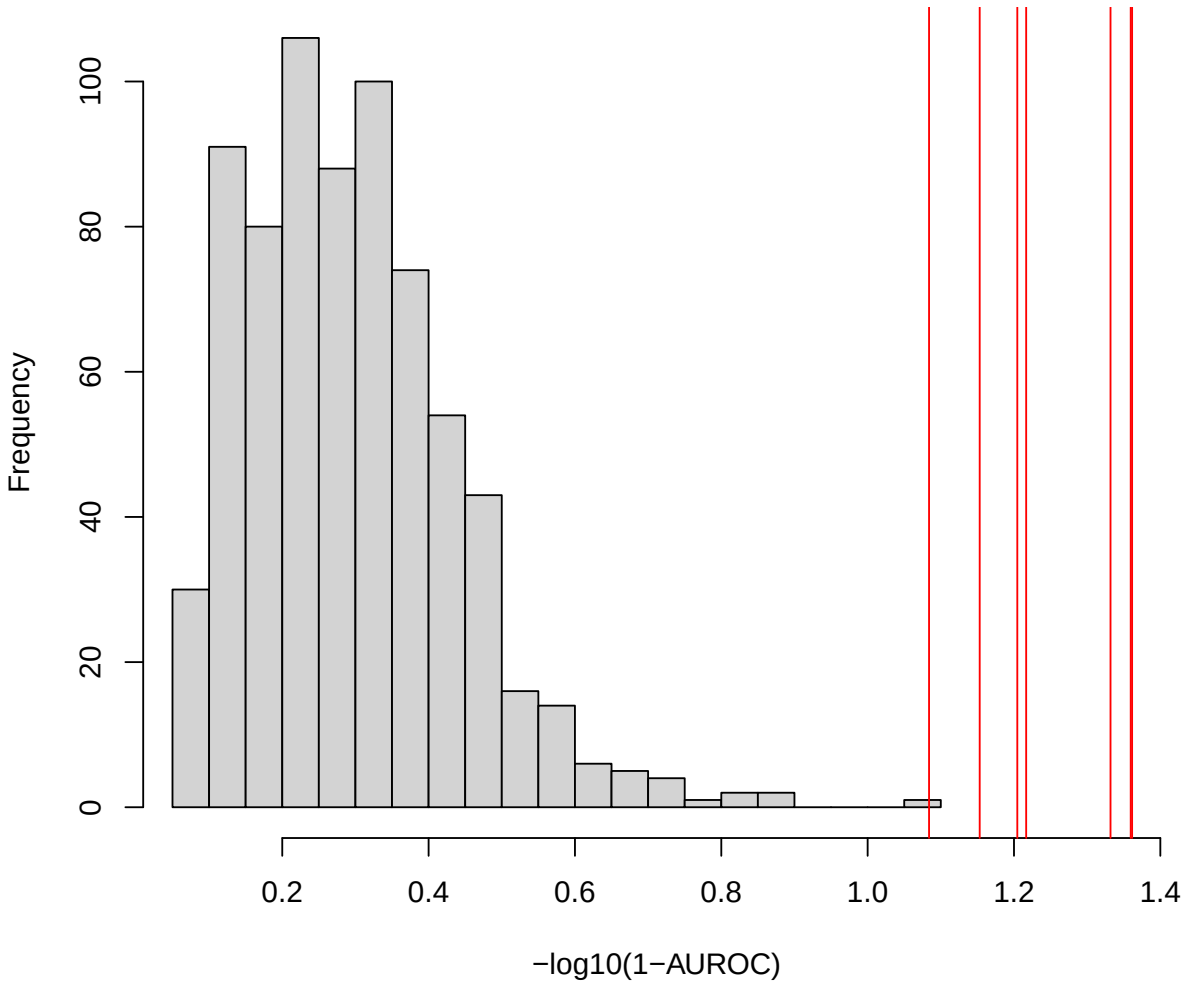
```
##     snCv3Z|L4/5 IT_2 snCv3M|Pvalb Vipr2_2      scCv2|L4/5 IT_2
##            0.9177663            0.9175751            0.8719640
##      snCv2|L4/5 IT_2
##            0.8676611
```

We note two properties of matching against a pre-trained reference. First, replicable cell types have a clear top match in each of the reference dataset. Sst Chodl (long-projecting interneurons) match to similarly named clusters in the BICCN with an AUROC > 0.9999, Pvalb Cpne5 (Chandelier cells) match with the Pvalb Vipr2_2 cluster with AUROC > 0.93. Second, we have to be beware of false positives. For example, Sst Chodl secondarily matches with the L6b Ror1 cell types with AUROC > 0.98, an excitatory cell type only distantly related with long-projecting interneurons. When we use the pre-trained model, we only compute AUROCs with the BICCN data as the reference data, so we cannot identify reciprocal hits. If we had been able to use "Tasic|Sst Chodl" as the reference cluster, its votes would have gone heavily in favor of the BICCN's Sst Chodl, making L6b Ror1 a low AUROC match on average. Because of the low dimensionality of gene expression space, we expect false positive hits to occur just by chance (e.g., cell types reusing similar pathways) when a cell type is missing in the query dataset. Here L6b Ror1 (an excitatory type) had no natural match with the Tasic inhibitory cell types and voted for its closest match, long-projecting interneurons.

There are three alternatives to separate true hits from false positive hits. First, if a cell type is highly replicable, it will have a clear top matching cluster in the reference dataset. Second, if the query dataset is known to be a particular subset of the reference dataset (e.g., inhibitory neurons, as is the case here), we recommend restricting the reference taxonomy to that subset. Third, if the first two solutions don't yield clear results or cannot be performed, it is possible to go back to reciprocal testing by using the full BICCN dataset instead of the pre-trained reference.

8. We illustrate the first solution in the case of Chandelier cells. Visualize the strength of the best hits by running the following:

```
chandelier_hits = aurocs["tasic|Pvalb Cpne5",]
is_chandelier = getCellType(names(chandelier_hits)) == "Pvalb Vipr2_2"
hist(-log10(1-chandelier_hits[!is_chandelier]), breaks = 20,
     xlab = "-log10(1-AUROC)", xlim = range(-log10(1-chandelier_hits)),
     main = "AUROC for Pvalb Cpne5 - Pvalb Vipr2_2 hits")
abline(v = -log10(1-chandelier_hits[is_chandelier]), col = "red")
```



To illustrate AUROC differences, we chose a logarithmic scaling to reflect that AUROC values do not scale linearly: when AUROCs are close to 1, a difference of 0.05 is substantial. Here, the best matching BICCN cluster ("Pvalb Vipr2_2") is order of magnitudes better than other clusters, suggesting very strong replicability.

9. The second solution to avoid false positive hits is to subset the reference to cell types that reflect the composition of the query datasets. Since we are looking at inhibitory neurons, restrict the BICCN taxonomy to inhibitory cell types, which names all start with "Pvalb", "Sst", "Lamp5", "Vip" or "Sncg":

```
is_gaba = grepl("^(Pvalb|Sst|Lamp5|Vip|Sncg)", getCellType(colnames(biccn_clusters)))
biccn_gaba = biccn_clusters[, is_gaba]
aurocs = MetaNeighborUS(trained_model = biccn_gaba,
                        dat = tasic_subdata,
                        study_id = tasic_subdata$study_id,
                        cell_type = tasic_subdata$primary_type,
                        fast_version = TRUE)
head(sort(aurocs["tasic|Sst Chodl",], decreasing = TRUE), 10)
```

```
##  scCv2|Sst Chodl  scCv3|Sst Chodl   scSS|Sst Chodl  snCv2|Sst Chodl
##        1.0000000        1.0000000        1.0000000        1.0000000
## snCv3M|Sst Chodl snCv3Z|Sst Chodl   snSS|Sst Chodl   snCv2|Sst Th_3
##        1.0000000        1.0000000        1.0000000        0.8965108
##  snCv3M|Sst Th_3 snCv3M|Sst Pappa
##        0.8839431        0.8721883
```

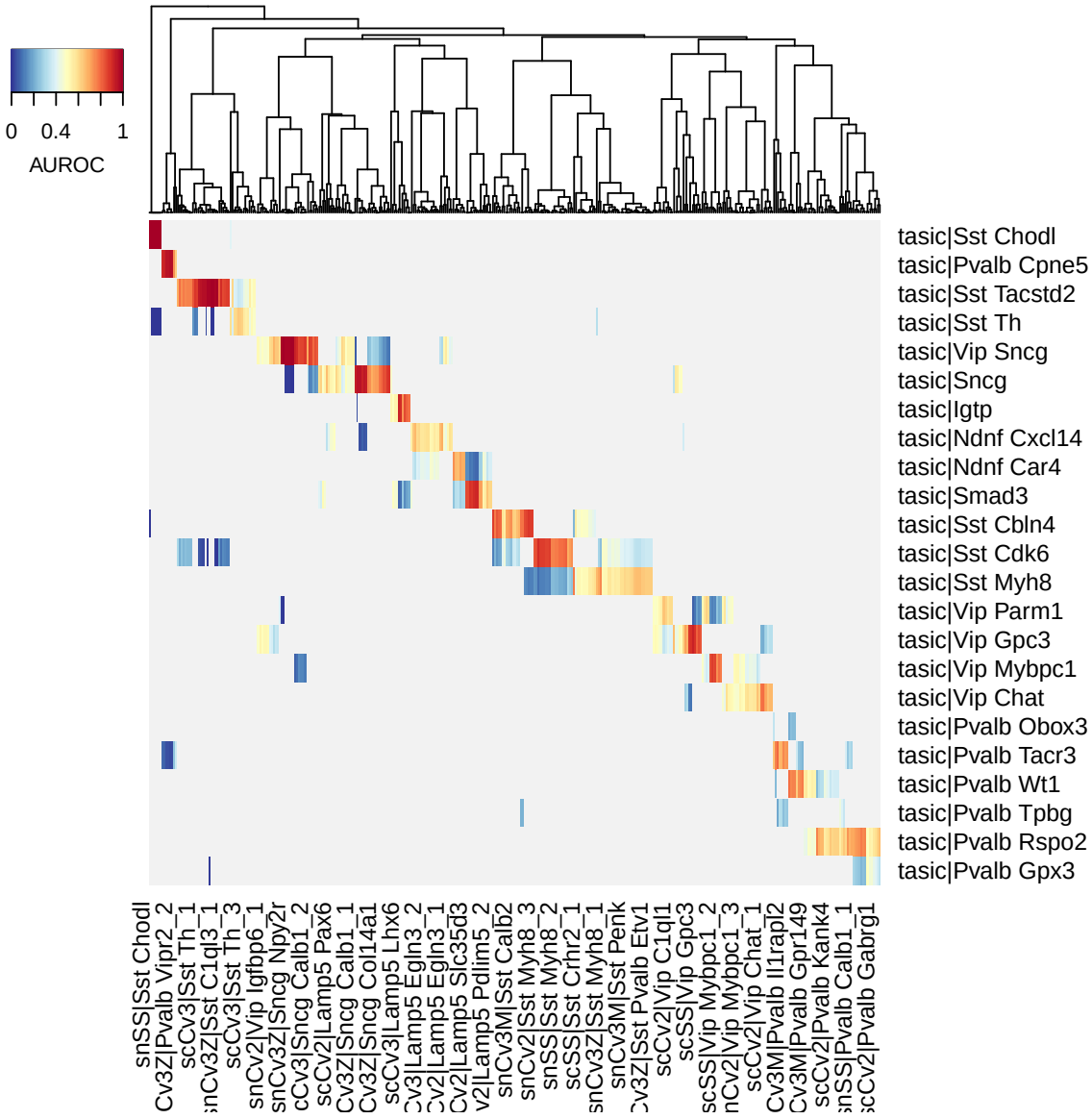
```
head(sort(aurocs["tasic|Pvalb Cpne5",], decreasing = TRUE), 10)
```

```
## snCv3Z|Pvalb Vipr2_2 snCv3M|Pvalb Vipr2_2  snCv2|Pvalb Vipr2_2
##            0.9960796            0.9959839            0.9939759
##   snSS|Pvalb Vipr2_2   scSS|Pvalb Vipr2_2  scCv2|Pvalb Vipr2_2
##            0.9939759            0.9895774            0.9893861
##  scCv3|Pvalb Vipr2_2 snCv3M|Pvalb Vipr2_1      scSS|Lamp5 Lhx6
##            0.9640467            0.9212086            0.8676611
##   scCv3|Sncg Slc17a8
##            0.8668962
```

Again, we note that there is a significant gap between the best hit and the secondary hit, but now secondary hits are closely related cell types (Sst subtype for Sst Chodl, secondary Chandelier cell type Pvalb Vipr2_1 for Pvalb Cpne5).

10. To obtain a more stringent mapping between the query cell types and reference cell types, compute one-vs-best AUROC, which will automatically match the best hit against the best secondary hit:

```
best_hits = MetaNeighborUS(trained_model = biccn_gaba,
                           dat = tasic_subdata,
                           study_id = tasic_subdata$study_id,
                           cell_type = tasic_subdata$primary_type,
                           one_vs_best = TRUE,
                           fast_version = TRUE)
plotHeatmapPretrained(best_hits)
```



Now the hit structure is much sparser, which helps identify 1:1 and 1:n hits. The heatmap suggests that most Tasic cell types match with one or several BICCN clusters. Inspect the top hits for 3 cell types from the Tasic dataset:

```
head(sort(best_hits["tasic|Sst Chodl",], decreasing = TRUE), 10)
```

```
##  scCv2|Sst Chodl  scCv3|Sst Chodl   scSS|Sst Chodl  snCv2|Sst Chodl
##        1.0000000        1.0000000        1.0000000        1.0000000
## snCv3M|Sst Chodl snCv3Z|Sst Chodl   snSS|Sst Chodl    snSS|Sst Th_2
##        1.0000000        1.0000000        1.0000000        0.4094994
```

```
head(sort(best_hits["tasic|Pvalb Cpne5",], decreasing = TRUE), 10)
```

```
## snCv3M|Pvalb Vipr2_2 snCv3Z|Pvalb Vipr2_2   snSS|Pvalb Vipr2_2
##            0.9698189            0.9678068            0.9547284
##  snCv2|Pvalb Vipr2_2   scSS|Pvalb Vipr2_2  scCv2|Pvalb Vipr2_2
##            0.9527163            0.9245473            0.9164990
##  scCv3|Pvalb Vipr2_2 snCv3M|Pvalb Vipr2_1
```

```
##           0.7444668           0.6348089
```

```
head(sort(best_hits["tasic|Sst Tacstd2",], decreasing = TRUE), 10)
```

```
##  scCv2|Sst C1ql3_1  snCv2|Sst C1ql3_1 snCv3Z|Sst C1ql3_1 snCv3M|Sst C1ql3_1
##          0.9962406          0.9924812          0.9924812          0.9887218
##  scCv3|Sst C1ql3_1   scSS|Sst C1ql3_1  scCv3|Sst C1ql3_2   scSS|Sst C1ql3_2
##          0.9852608          0.9812030          0.9661654          0.9661654
##   snSS|Sst C1ql3_1  scCv2|Sst C1ql3_2
##          0.9624060          0.9586466
```

Using this more stringent assessment, we confirm that Sst Chodl strongly replicates inside the BICCN (one-vs-best AUROC ~ 1, best secondary hit $= 0.41$), same for Pvalb Cpne5 (one-vs-best AUROC > 0.74, best secondary hit $= 0.63$), while for example Sst Tacstd2 corresponds to multiple BICCN subtypes (including Sst C1ql3_1, Sst C1ql3_2, AUROC > 0.95).

Pre-training a MetaNeighbor model thus provides a rigorous, fast and simple way to query a large reference dataset and obtain quantitative estimations of the replicability of newly annotated clusters.