

1 **HieRFIT: Hierarchical Random Forest for Information Transfer**

2 Yasin Kaymaz¹, Florian Ganglberger², Ming Tang¹, Francesc Fernandez-Albert³, Nathan

3 Lawless³, Timothy Sackton¹

4 ¹ Informatics Group, Harvard University, Cambridge, MA, USA.

5 ² VRVis Research Center, Vienna, Austria.

6 ³ Global Computational Biology & Digital Sciences, Boehringer Ingelheim Pharma GmbH & Co KG,
7 Biberach an der Riss, DE.

8 **Abstract**

9 The emergence of single-cell RNA sequencing (scRNA-seq) has led to an explosion in novel
10 methods to study biological variation among individual cells, and to classify cells into functional and
11 biologically meaningful categories. Here, we present a new cell type projection tool, HieRFIT
12 (**H**ierarchical **R**andom **F**orest for **I**nformation **T**ransfer), based on hierarchical random forests.
13 HieRFIT uses *a priori* information about cell type relationships to improve classification accuracy,
14 taking as input a hierarchical tree structure representing the class relationships, along with the
15 reference data. We use an ensemble approach combining multiple random forest models, organized
16 in a hierarchical decision tree structure. We show that our hierarchical classification approach
17 improves accuracy and reduces incorrect predictions especially for inter-dataset tasks which reflect
18 real life applications. We use a scoring scheme that adjusts probability distributions for candidate
19 class labels and resolves uncertainties while avoiding the assignment of cells to incorrect types by
20 labeling cells at internal nodes of the hierarchy when necessary. Using HieRFIT, we re-analyzed
21 publicly available scRNA-seq datasets showing its effectiveness in cell type cross-projections with
22 inter/intra-species examples. HieRFIT is implemented as an R package and it is available at
23 (<https://github.com/yasinkaymaz/HieRFIT/releases/tag/v1.0.0>)

24

25 **Corresponding Author:** Timothy Sackton

26 **Introduction**

27 Single-cell RNA-seq (scRNA-seq) technology has provided an unparalleled picture of the
28 cell-to-cell complexity of biology in multicellular organisms. As technological improvements have
29 allowed increasingly large studies, comprehensive cell atlas experiments have revealed
30 unprecedented cell-to-cell heterogeneity and molecular dynamism of cell types across both human
31 and model organisms (Cao et al., 2017, Rosenberg et al., 2018). Single-cell genomics have enabled
32 tracing developmental lineages of early embryonic cells and building transcriptional landscapes of
33 organogenesis at single-cell resolution, and uncovering novel rare cell populations (Cao et al., 2019,
34 Tabula Muris et al., 2018).

35 As single-cell experiments grow in size and scope, the computational challenges associated
36 with analyzing and interpreting these data are also growing. In particular, identifying cell types
37 present in a sequenced population is critically important for enabling biological insight. Widely
38 prevalent single cell analyses protocols incorporate unsupervised clustering methods as a key step
39 in this process. For example, k-means, hierarchical clustering, KNN (k-nearest neighbor) or SNN
40 (shared-nearest-neighbor) graphs, and Louvain community detection are all used in a variety of
41 different packages, such as SC3 (Kiselev et al., 2017) and Seurat (Butler et al., 2018). Unsupervised
42 clustering methods attempt to identify a consistent and biologically meaningful set of cell types or
43 cell states in an experiment, usually via a projection of high dimensional data. These approaches
44 have identified numerous novel subtypes (Aevermann et al., 2018, Plasschaert et al., 2018, Suo et
45 al., 2018), although complexities of parameter optimization, number of available cells, and intrinsic
46 noise of single-cell data can pose challenges (Kiselev et al., 2019, Tang et al., 2020).

47 However, unsupervised clustering approaches do not provide any rapid or automated way
48 of defining cluster identities, which is often done by manually checking marker gene expression. In
49 addition to being cumbersome, manual annotation depends on the robustness of a handful of a
50 priori marker genes. When cell types are highly similar to each other transcriptomically, manual

51 annotation may be prone to human error as reliable and obvious marker genes may not exist
52 (Lähnemann et al., 2020). An alternative to unsupervised clustering is to use the rich information
53 from larger atlas projects, and focus on information transfer to new studies (Wilbrey-Clark et al.,
54 2020). While potentially faster and more accurate for cell type annotation than unsupervised
55 clustering, especially for small-scale studies, integration and accurate information transfer between
56 existing atlas datasets play a critical role. Supervised machine learning methods, using large cell
57 atlas datasets as training data, provide a potential approach to automate information transfer for
58 faster and accurate projections (Petegrosso et al., 2020).

59 A number of supervised classification methods have been developed, including
60 singleCellNet (Tan and Cahan, 2019), ACTINN (Ma and Pellegrini, 2019), Garnett (Pliner et al.,
61 2019), with different strengths and limitations. These methods differ in various aspects such as
62 feature selection, for instance, singleCellNet trains its models with random forest after extracting a
63 set of feature pairs from the reference data while ACTINN uses neural networks that automatically
64 chooses the features. Garnett, on the other hand, relies only on a set of cell type specific marker
65 genes as input independent of a reference dataset. Although the majority of these developed
66 methods are flat classifiers, hierarchical classification has also been implemented in the single-cell
67 context with CHETAH (de Kanter et al., 2019), and scClassify (Lin et al., 2019), which allowed
68 intermediate class assignments, although their outputs provided limited insight into actual cell
69 types.

70 Despite the rapid proliferation of cell-type assignment methods, a number of limitations still
71 exist with current approaches. Many existing methods work best when the reference training data
72 is composed of a few well-represented cell types, and when the query data contains a few or no
73 novel types (Abdelaal et al., 2019a). However, an ideal classification should be able to handle many
74 candidate cell classes, potentially hundreds, and not rely on a minimum input threshold of query

75 data, as some single-cell protocols produce low-throughput data in which rare cell types are
76 represented with only a few cells (Campbell et al., 2017). In addition, handling complex
77 classification tasks by conventional methods usually involves either assigning to a cell type with
78 low confidence or the best case is declaring them as ‘undetermined’. However, this approach
79 underestimates the potentially informative biological signal which is often challenging to harvest
80 and valuable to resolve experimental questions. Furthermore, considering cell types as discrete
81 entities with clear boundaries is far from ideal as, in reality, many cells are in transitioning
82 intermediate stages, which makes classification more compelling (Macaulay et al., 2016). Thus,
83 alternative approaches that benefit from hierarchical consideration of cell types are required to
84 eliminate these issues in the single-cell identity detection.

85 Here, we propose a new hierarchical classification approach, HieRFIT, which uses a
86 hierarchical tree structure of reference cell clusters, allowing custom defined intermediate classes
87 (internal nodes) that have biological meaning. Using this hierarchical model, we both improve
88 HieRFIT’s ability to provide accurate cell type classification, and allow cells that cannot be
89 accurately classified to be assigned to the best supported internal node. We implemented our
90 approach as an R package and tested against various classification tasks.

91 **Methods**

92 **Constructing the cell type decision tree with class hierarchies**

93 A key input component of HierFIT is the cell type hierarchical tree. We define this hierarchy
94 as a tree, τ , which is a subtype of directed acyclic graphs, where each cell type or cell class is
95 represented as a node v , and the connections between the nodes are edges, E . Nodes can only have
96 a single parent, but can have multiple child nodes. The nodes in the tree, τ , are also asymmetric
97 (each child node cannot be a parent of its own parent), and the tree itself is transitive (each node is
98 also a child node of its parent's ancestral node). From this cell type tree, we can define an ancestral
99 hierarchy. Let A represent a set of all ancestral nodes of a given node and Y represent the set of
100 class labels of all nodes. Then, $A_j = \{v_j, v_{j+1}, v_{j+2}, \dots, v_k\}$ is the set of nodes comprising the ancestral
101 path for node v_j reaching up to the root node v_k and $Y_j = \{y_1, y_2, \dots, y_n\}$ is the class label set for
102 children of node v_j . We define the terminal nodes with no children as leaves. To define this
103 hierarchical tree for a given reference datasets, the user can input a cell type table in a tab delimited
104 format with each row designates a leaf cell type from the reference dataset and columns represent
105 the intermediate cell types to be used as internal nodes (**Supplementary Table 1**). HierFIT can
106 also create a *de novo* tree out of cell type distances based on their averaged gene expressions using
107 hierarchical clustering if an input tree is not provided.

108 **Feature selection from reference data and local classifier training**

109 Feature selection is performed for each local classifier (internal node) separately. Let M be
110 the normalized expression matrix to be used as a training data, which is composed of genes G and
111 samples (cells) X accompanied by a set of class types Y . In addition to the existing class types, an
112 'OutGroup' class that represents the other cell types is also added to the set, Y . 'OutGroup' class
113 sample size is limited to a maximum 500 cells (same as other classes, and can be altered by user)

114 for all nodes and is formed by randomly selecting cells from classes that are not present for the
115 local classifier. HieRFIT selects a set of separate features, $f_j \in G$, for every internal node, $v_j \in v$,
116 using the corresponding subset data $m_j \subset M$. Genes with very limited variation across cell types, m_j
117 ($\sigma^2 < 0.01$), are pre-filtered in order to eliminate non-informative features. After standardizing the
118 expression by centering at the mean and scaling by the standard deviation, HieRFIT computes
119 eigenvectors of the data with principal component analysis using the '*prcomp*' function from the R
120 stats package. To define features, HieRFIT first selects principal components (PCs) that usefully
121 separate class labels Y_j , by computing a t-test on the component scores of each PC and selecting PCs
122 with $P < 0.05$ (following Bonferroni correction). To turn these informative PCs into highly variable
123 feature sets, HieRFIT chooses the top 2000 variables (genes) that are most correlated with their
124 eigen vectors based on absolute component loading values (number of top genes selected can be
125 changed by user). The number of genes selected from each component is proportional to variance
126 explained by the PC. Further, wilcoxon rank sum test between the class labels is applied to further
127 select (by default) 200 differentially expressed genes (based on adjusted p-values) as features to be
128 used in local classifier training.

129 HieRFIT constructs a reference classifier with multiple local classifiers, one for each of the
130 internal nodes on the hierarchical tree. Local classifiers are created using a random forest algorithm
131 implemented with the R package Caret, with the features selected separately for each node using
132 the procedure described above, and with 500 trees (by default). The training sample set of each
133 local classifier, m_j , corresponds to the cells from reference data with cell type labels matching the
134 class labels of the node's children, Y_j . The array of local classifiers is stored as an S4 object (in R) in
135 the hierarchical organization to be used for projecting cell types on a query data.

136

137 **Sigmoid calibration and noise injection**

138 The random forest classifier produces as output a vector of votes for each class, one from
139 each tree. However, these vote distributions are not equivalent to class probabilities, therefore,
140 they need to be transformed with a calibration function before they can be used as probabilities.
141 HieRFIT implements Platt scaling (Platt, 1999) for this purpose: as the final step of creating a
142 HieRFIT model, we construct a sigmoid function on reference data with class labels using a
143 multinomial logistic regression implemented in the 'nnet' R package. This sigmoid function allows
144 the conversion of class votes as the unprocessed output of random forest classifier to class
145 probabilities. In order to provide a certain level of flexibility against dropout events in scRNA-seq,
146 we also implemented a noise injection step prior to generating the sigmoid function (Zur et al.,
147 2009). Noise injection occurs by setting expression values of a subset of randomly selected feature
148 sets of each local classifier (by default 10% of all features) to zero.

149 **Asymmetric entropy-based certainty measurement**

150 In order to convert class probabilities into class assignments, allowing for the possibility
151 that some cells cannot be accurately assigned, we implemented a certainty function per candidate
152 class. We used asymmetric entropy measurement (Marcellin et al., 2006) in our certainty function
153 as follows;

154 Let p_i denote probability of class $y_i \in Y_j$ at the node v_j , then, the asymmetric entropy as a
155 measure of uncertainty is

$$156 \quad h(p_i) = \frac{p_i (1 - p_i)}{(1 - 2w_i) p_i + w_i^2}$$

157 where w_i is probability of y_i at which maximum uncertainty is achieved. Note that h is equal to
158 quadratic entropy of Gini when $w_i = 0.5$ in binary class modalities. From h , we then derived a

159 function called ‘certainty function’, U , which contains an additional coefficient, λ , to assign
160 directionality as follows;

161
$$U = \lambda \cdot [1 - h(p_i)] \quad \text{where} \quad \begin{cases} \lambda = 1 & \text{if } p_i \geq w_i \\ \lambda = -1 & \text{if } p_i < w_i \end{cases}$$

162 Certainty scores center at zero when $p_i = w_i$ and range between -1 and 1 representing maximum
163 certainties about unrelatedness and relatedness to the class, respectively. In order to obtain a set of
164 empirical probability centroids, $W = \{w_1, w_2, \dots, w_i, \dots, w_n\}$, for each class, HieRFIT randomizes the
165 feature set f_j of $m_j \subset M$ for the corresponding node with random permutations and calculates
166 expected probabilities of each class as the mean across iterations.

167 **Scoring scheme and decision rule for class projection**

168 HieRFIT then scores each in a “top-down” manner, which refers to taking all ancestral node
169 scores and their metrics into account beginning from root node. In order to project class labels from
170 the HieRFIT model created using the reference dataset and a cell type tree, the first step is to obtain
171 an array of certainty scores from all local classifiers for all class types. Let $x \in X$ be a cell in a query
172 dataset. In order to determine class type of x , HieRFIT generates an array of path certainty scores
173 $U_{ij}(x)$ which is calculated using classification certainty scores for every node on the ancestral path,
174 where i represents the class types of node j by traversing the tree and following the ancestral path
175 reaching to the root v_k as follows;

176
$$U_{ij}(x) = \sum_{v_j \in A}^k U_j(x) - U_j^{sib,out}(x)$$

177 where $U_{ij}(x)$ is the path certainty score of cell $x \in X$ for class i in the node v_j , and $U_j^{sib,out}$ is the
178 sum of all siblings and outgroup certainty scores for the classifier node v_j .

179 During the score aggregation for decision, our rule for assigning class labels is

$$180 \quad \text{Class}(x) = \begin{cases} \operatorname{argmax}_{y_i \in Y} U_{ij}(x) & \text{if } U_{ij}(x) > \alpha \\ \text{'Undetermined'} & \text{if } U_x = \emptyset \end{cases}$$

181 HierFIT assigns the maximum scoring class label out of all candidate classes that pass a certainty
182 threshold, α . If none of the nodes passes, “Undetermined” is returned as a class label. The certainty
183 threshold is set to 0.05 by default, but can be changed by the user.

184 **Performance evaluations**

185 For intra-dataset performance evaluation tasks, we used 5-fold cross validation, in other
186 words, training models with 80% and testing them on 20% of data. As the evaluation metric, we
187 calculated precision, recall, and F-measure averaged across iterations of cross validation. For inter-
188 dataset evaluation tasks, in which training and test data originate from two separate datasets, we
189 relied on the concordance between prior and predicted cell types of query datasets. We excluded
190 intermediate cell type, ‘undetermined’, and multi-class assignments in metric calculations. In
191 addition, given that HierFIT uses a non-mandatory leaf node prediction approach and can return
192 intermediate class labels, we also accounted for intermediate cell type assignments in performance
193 evaluation. Therefore, we utilized precision, recall, and F-measure calculations modified for
194 hierarchical classifications (Kiritchenko et al., 2005). For these, let A_y be a set of all ancestral labels
195 for the predicted class label y and A_θ be the set of all ancestral labels for the true class θ_x of test
196 sample x , then hierarchical precision (hP) and recall (hR) are

$$197 \quad hP = \frac{\sum_{x \in X} |A_y \cap A_\theta|}{\sum_{x \in X} |A_y|} \quad hR = \frac{\sum_{x \in X} |A_y \cap A_\theta|}{\sum_{x \in X} |A_\theta|}$$

198 where $|A_y \cap A_\theta|$ is the number of intersecting nodes between ancestors of predicted and true class
199 labels. Then, the hierarchical F-measure (hF) is

$$200 \quad hF_\beta = \frac{(\beta^2 + 1) \cdot hP \cdot hR}{\beta^2 \cdot hP + hR}, \quad \text{where } \beta = 1$$

201 **Datasets**

202 We analyzed PBMC scRNA-seq data from 10X Genomics with 2,700 single-cells by following
203 standard processing workflow as instructed on Seurat online tutorials
204 (https://satijalab.org/seurat/v3.1/pbmc3k_tutorial.html) (Butler et al., 2018). The main steps of
205 this analysis were data quality control and normalization, identifying variable genes, data scaling,
206 dimension reduction, clustering, finding differentially expressed genes, and assigning cell type
207 identities to clusters based on known cell markers. We used another public PBMC scRNA-seq
208 dataset with 68K cells (Zheng 68K) as one of the reference datasets to generate a HierFIT model
209 (Zheng et al., 2017b). We followed the same analysis steps in the publication (and code in GitHub
210 <https://github.com/10XGenomics/single-cell-3prime-paper>). We relabeled the cell types for easier
211 interpretation.

212 To evaluate the prediction performances with various data types, we selected several
213 published single-cell datasets with available class types and expression data (**Supplemental Table**
214 **2**). We used the same cross validation folds of these datasets previously generated for
215 benchmarking and performance evaluations through intra and inter-dataset challenges (Abdelaal et
216 al., 2019b). These datasets originate from 10 separate scRNA-seq studies, some with multiple levels
217 of cell type annotations.

218 **Results**

219 **Overview of the algorithm**

220 We approached the cell type classification task as a hierarchical decision problem with a set
221 of predefined class relations. Our assumption of relationship between sub-classes and upper level
222 classes does not necessarily have to reflect biologically defined developmental trajectories but
223 rather represents organization of broader categories for cell type identities. HieRFIT uses multiple
224 local random forest classifiers, organized in a higher-level hierarchical decision tree, to split the
225 complex tasks into smaller and simpler ones. Here, we give a brief overview of the method, which is
226 described more in details in the Methods. Given a reference expression matrix, HieRFIT first
227 extracts the most informative principle components (PCs), which distinguish reference class types.
228 Then, it selects a set of genes from those components as predictors based on their correlations with
229 eigenvectors. Using the predictor set, it trains one classifier with corresponding subset data for
230 each parent node on a user defined hierarchical tree and builds a reference classifier. In order to
231 accurately project information from reference dataset on new experiments, we also implemented a
232 scoring scheme for assigning class labels in a non-mandatory leaf node prediction manner, which
233 allows us to provide intermediate cell types with broader context when data fails to provide enough
234 resolution for more specific cell types. The uncertainty function that we derived utilizes the
235 empirically learned background probability distribution and helps to determine whether observed
236 probability is informative for inferring class types. Our certainty based scoring scheme properly
237 finds the most likely ancestral path on the hierarchical cell type tree, which also provides additional
238 confidence about identity of query cells.

239

240

241 **Hierarchical model construction and its algorithm**

242 HieRFIT takes reference datasets with cell type labels along with a hierarchy of
243 corresponding reference cell types as prior information. The hierarchical tree of cell type can be
244 customized by the user with proper intermediate types. If the user lacks such prior information,
245 HieRFIT can generate the hierarchy *de novo* by computing the distances between the reference cell
246 types based on mean transcriptome expressions. To construct the reference HieRFIT model
247 (HierMod), we implement a six step protocol that is repeated for every internal node on the
248 hierarchical tree (**Figure 1A**). For each node a local classifier is generated using a random forest
249 classification algorithm. To prepare the reference expression data for model training, the first step
250 is to extract a cell data matrix that corresponds to the node. Training data is relabeled by bundling
251 the grandchildren under the node's children labels and adding an outgroup class as the representer
252 of other classes. A principal component analysis using the relabeled data provides the components
253 that are highly variable across cells and we select the components whose loadings significantly
254 separates cells with shared type from the others. The selected significant components allow us to
255 reduce the total number of genes to a highly variable gene set among which we select the final
256 feature set following the Wilcoxon rank sum test. This final feature set is used in the training of the
257 local classifier with the cell type labels (or relabels). All local classifiers are stored as an array of
258 models which are organized in accordance with the input tree hierarchy.

259 **Path certainty score computation and class selection**

260 Classical measures used in decision trees such as Shannon's entropy or quadratic entropy of
261 Gini are not suited well for real life imbalanced data with their symmetry assumptions for
262 equiprobability distributions among classes (Zighed et al., 2010). Therefore, we created an
263 alternative certainty function derived from asymmetric entropy (see methods). The main stage of
264 assigning a reference cell type to a query cell is to compute the array of certainty (U) values for each

265 internal and terminal node (**Figure 1B**). In order to obtain such a metric, HieRFIT takes the gene
266 expression array of the query cell and the reference model (HierMod) as inputs. Along with the
267 gene expression array of the query cell, a randomized (shuffled 1000 times) expression array for
268 the same query is generated. Class votes from local classifiers are obtained for both observed and
269 shuffled expression arrays, simultaneously. Normalized class votes gathered from the local
270 classifiers are converted to class probabilities with logistic regression (aka sigmoid calibration)
271 function. The observed expression array is used to acquire adjusted class probabilities from each
272 local classifier, while the shuffled expression array is used to determine class centroids. These
273 centroids are used as certainties of the classes for the query with the observed class probabilities
274 using the certainty function (see Methods for details).

275 After computing the certainty values of each node for the query, we calculate the path
276 certainty scores for all candidate nodes by adding up the certainty values along the ancestral path
277 and subtracting all outgroup and sibling nodes certainties (**Figure 1C**). This path score defines the
278 final value for the nodes to be considered as candidate classes. The decision stage simply consisted
279 of choosing the maximum scoring node among the ones whose score exceeds a certain threshold
280 (α). In cases where no class exceeds the threshold, HieRFIT returns an “Undetermined”. This
281 decision scheme permits internal nodes to be cell types of the query as well as leaf nodes that are
282 constrained with input the reference data in the first place.

283 **PBMC cell type classification with HieRFIT**

284 For demonstration purposes, we generated a reference model using the 68K PBMC single-
285 cell dataset from Zheng et al (Zheng et al., 2017b). We created an example hierarchical tree which
286 organized the reference cell types into two main groups, myeloid lineage and lymphoid lineage,
287 along with the hematopoietic stem cells (HSC) using the input tree file in **Supplemental Table 1**
288 (**Figure 2A**). Two main groups branched into further intermediate groups and general cell types.

289 The terminal nodes comprise reference cell types from the PBMC data. We used this custom made
290 hierarchical tree to create a HierFIT model with maximum 500 cells per cell type in the training
291 process.

292 To test our hypothesis that our hierarchical classification approach provides more accurate
293 and meaningful results as compared to conventional way of cell type identification, we used a toy
294 dataset, another 3K PBMC, as the query (10X Genomics, [https://support.10xgenomics.com/single-](https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc3k)
295 [cell-gene-expression/datasets/1.1.0/pbmc3k](https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc3k)). This publicly available dataset has been generated
296 by 10X Genomics and processed through the Seurat pipeline. We followed the same guideline as in
297 the Seurat tutorial to identify cell types with no prior information. This manual type annotation
298 involves several commonly accepted processes, such as finding the variable genes, dimension
299 reduction, and clustering of cells. Finally, determining the cell types involves manually checking the
300 marker gene expression that are differentially expressed in clusters against the rest of the groups.
301 This process results annotation of 3K PBMC data with cell types: 'B cells', 'Megakaryocytes', 'CD14+
302 Monocytes', 'CD16+ Monocytes', 'Dendritic cells', 'NK cells', 'CD8 T cells', 'CD4 T Naive' and 'Memory
303 cells' (**Figure 2B**, left). Then, we tested the hierMod we created with the 68K PBMC data by
304 projecting the reference cell types on the same 3K PBMC dataset. HierFIT projections labeled the
305 cell in the query with leaf node labels as well as intermediate cell type defined in the model
306 hierarchical tree above (**Figure 2B**, right). HierFIT projections demonstrated a significant
307 concordance with Seurat cell types for the distinct cell types, such as 'B cells', 'Megakaryocytes', 'NK
308 cells', and 'Dendritic cells' (although these cells' HierFIT projections were "Classical DCs" rather
309 than the parent node "Dendritic cells" on the tree). Seurat 'CD8 T cells' were labeled extensively
310 with subtypes "CD8 T GZMK+" and "CD8 T Cytotoxic" as well as their parent node "CD8 T cells",
311 partially (**Figure 2C**). "CD4 T Memory" cells were labeled mainly as "CD4 T Memory" and "CD4 T
312 Reg" in addition to the parent node "CD4 T cells". "CD4 T Naive" cell group, on the other hand,
313 received labels from almost all CD4 T sub-levels cell types and intermediate types such as "CD4 T

314 cells” or even “T cells” as higher nodes on the tree. Interestingly, a group of cells within naïve CD4 T
315 cells received CD8 T cell labels, especially “CD8 T Naïve Cytotoxic”. Similarly, a subgroup of “CD14+
316 Monocytes” were labeled as “CD16+ Monocytes”, while all “CD16+ Monocytes” were correctly
317 labeled by HierFIT. A small group of cells from the CD14+ cells were labeled as a parent node
318 “Monocytes”.

319 **HierFIT classifications are concordant with marker gene expressions**

320 To investigate the discordance between Seurat annotations and HierFIT projections for
321 some of the cell groups, we further explored the marker gene expressions and their distribution. As
322 the heatmap of the confusion matrix demonstrates, 7 out of 9 cell types were labeled with cell types
323 by HierFIT with more than 80% concordance (**Figure 3A**). Two of the cell types, “CD14+
324 Monocytes” and “CD4 T Naïve” received classification labels that resulted in 67.7% and 73.5%
325 concordance, respectively. For the “CD4 T Naïve” cell types, we examined the expression
326 distribution of the CD8 T cell markers, CD8A and CD8B, as well as major CD4 markers for naïve
327 cells, IL7R and CCR7 (**Figure 3B**).

328 Within this group, the subset of cells which are predicted by HierFIT to be CD8 T cells or its
329 subtypes expressed CD8A and CD8B at high levels, suggesting these cells are properly assigned to
330 the CD8 subtype (**Figure 3B**, upper panel - violin plots). The co-expression of the two markers also
331 clearly showed that the significant majority of these cells in fact expressed at least one of these
332 markers or both at the same time (**Figure 3B**, upper panel - UMAP panel with co-expression
333 projections). This observation supports the accuracy of HierFIT projections that, in fact, these cells
334 are a class of CD8 T cells rather than CD4 T cells. On the other hand, the group of cells that were
335 concordantly labeled as CD4 T cells or its subtypes carried the proper CD4 T naïve marker
336 expressions, IL7R and CCR7, in line with their projected cell types (**Figure 3B**, lower panel – violin
337 plots and UMAP projections).

338 We also investigated another group of cells with discordant predictions in “CD14+
339 Monocytes”. Of these monocytes, 27.7% were predicted as “CD16+ Monocytes”. When we examined
340 the marker gene expression in cells classified as “CD14+ Monocytes” by Seurat, we observed that a
341 significant portion of them expressed CD16 (FCGR3A) monocyte marker at high levels (**Figure 3C**,
342 upper panel). On the other hand, HieRFIT classification of these cells demonstrated a clearer
343 separation between these two highly similar subtypes of monocytes while preserving the major
344 monocyte marker expression in all cells even in cells predicted with the label of the parent node,
345 “Monocytes” (**Figure 3C**, lower panel).

346 **Comparative performance evaluation with intra-dataset tests**

347 We evaluated the performance of HieRFIT on a large number of different datasets, with
348 varying complexity, technology, and size. These include human and mouse pancreas datasets
349 (Baron et al., 2016, Muraro et al., 2016, Segerstolpe et al., 2016, Xin et al., 2016), human PBMC
350 (Zheng et al., 2017a), human lung cancer cell lines (Tian et al., 2019), mouse cortex and nervous
351 system (Tasic et al., 2018, Zeisel et al., 2018) as well as whole mouse datasets from Tabula Muris
352 consortium (2018) (**Supplemental Table 2**). We also compared its performance against other cell
353 type classification that use supervised machine learning approaches to create a predictive model
354 based on the training data. The first benchmarking was based on intra-dataset evaluations with 5-
355 fold cross validation. Some of the datasets with multi-level cell type annotations were treated
356 separately as different datasets. We calculated the mean-F1 score of each classification tool as the
357 overall performance averaged across each cell class in the datasets. To obtain a fair benchmarking,
358 we included only leaf node predictions of HieRFIT and excluded the intermediate node
359 classifications in the F1-score computations.

360 We compared the performance of HieRFIT against 21 classification approaches with various
361 modes using 17 unique tools. Based on the median value of the mean-F1 scores from test datasets,

362 HieRFIT demonstrates better performance than 16 of them (**Supplemental Figure 1**). LDA,
363 ACTINN, SingleR, SVM, singleCellNet, and SVM with rejection demonstrate comparable performance
364 against HieRFIT (**Figure 4A**, heatmap). Both SVM and SVM with rejection option perform better
365 HieRFIT on most datasets except two of them. Out of 18 mean-F1 scores of datasets, ACTINN is
366 better on 12 datasets, LDA is 11, singleCellNet 10, and singleR is better on only 3 datasets compared
367 to HieRFIT. SingleR and singleCellNet fails to complete the tasks on the complex datasets with large
368 number of cell types, such as Zeisel (237) and AMB (92), TM (55), and Zheng datasets. 5 out of these
369 6 classification approaches lack an important feature, a rejection option. HieRFIT, along with LDA
370 (with rejection), scClassify, and CHETAH, returned low levels of ‘unlabeled’ predictions while SVM
371 (with rejection), scmap (both ‘cell’ or ‘cluster’ modes), Cell-BLAST, and scID classifications
372 contained high level of ‘unlabeled’ results (**Figure 4A**, boxplot). Almost all of the classification tools
373 perform the worst on Zheng PBMC (11 cell types) dataset, likely due to its intrinsic complexity.

374 We further explored the performance of HieRFIT in depth by comparing it to other two
375 tools, scClassify and CHETAH, with similar hierarchical classification approaches to ours and with
376 the most commonly used software, Seurat. We computed the hierarchical precision, recall, and F-1
377 score, which takes the intermediate cell type predictions into account when computing the
378 performance metric. To be fair to the other tools, we used the same hierarchical tree that HieRFIT
379 used in the computation of the hierarchical metrics for the other tools. We obtained the results for
380 hierarchical precision, recall, and F-1 score measurements from the 18 intra-datasets through 5-
381 fold cross-validations.

382 HieRFIT and the other three classification tools demonstrate high levels of hierarchical
383 precision in all datasets, >91%, except ‘Zheng’ datasets (**Figure 4B**, upper panel). However,
384 scClassify fails to return the results for AMB (92) and Seurat fails to identify enough significant
385 anchors for “CellBench (CEL-Seq2)” dataset. On the other hand, HieRFIT, returns class predictions

386 with consistently high recall rates (> 89%) for all datasets while scClassify and CHETAH showed
387 significantly lower recalls especially on tasks with complex datasets with large number of cell types
388 (**Figure 4B**, middle panel). As the performance metric that takes precision and recall into account,
389 hierarchical F-1 score clearly demonstrates that HieRFIT performs at consistent levels and
390 comparable to Seurat classifications (**Figure 4B**, lower panel).

391 To better evaluate the HieRFIT results in the hierarchical classification context, we
392 categorized the projected cell types based on their positions on the reference tree (**Figure 4C**).
393 These categories reflect the level of prediction accuracy relative to the hierarchical relationship
394 defined as cell type similarities. These categories are as follows: The projection cell type is
395 categorized as ‘Correct node’ if it is same as the true cell type (prior), as ‘Correct parent’ if it is
396 parent of true cell type, as ‘Correct ancestral node’ if it is on the ancestral path (excluding parent
397 node) of true type, as ‘Incorrect sibling’ if it is a sibling of true type, and as ‘Incorrect clade’ is if it is
398 any other node with an unshared parent as true label. Using these schemes, we checked the
399 distributions of categorized HieRFIT projections for each intra-dataset task (**Figure 4D**). HieRFIT
400 returns a large proportion of correct leaf nodes for the majority of datasets. Even for the complex
401 datasets, such as TM (55), AMB (92), and Zeisel (237), the correct leaf node rates are 95%, 85%,
402 and 75%, respectively, while Zheng PBMC dataset (11 cell type) results in inferior profile due to its
403 complexity with high rates of incorrect sibling and clade assignments. The rates of assignments
404 from other categories are relatively lesser simply due to the intra-dataset tasks using part of the
405 same data to test the performance.

406 **Robustness against various scRNA-seq methods (PBMC bench)**

407 To evaluate the performance of HieRFIT on inter-dataset tasks in which the reference model
408 is built on a dataset completely different from the test set, we utilized another public data collection
409 generated for PBMC from two individuals (Ding et al., 2019). This type of inter-dataset tasks

410 provide more realistic results as they reflect real life usage better. PBMC1 and PBMC2 samples were
411 split into multiple subsets and sequenced with 8 different versions of single-cell RNA-seq methods.
412 In the experiment, we trained the model using data from one method and tested it on datasets
413 generated using other methods (and on the second of the sample pairs in case same methods). As
414 hierarchical precision, recall, and F1-score distributions on all of the combinations, HierFIT
415 performs consistently well on almost all tasks with average 85% rates (**Figure 5**). On the other
416 hand, Seurat and CHETAH, sacrifices extensive precision and recall rates, respectively, on many
417 tests while scClassify performs slightly better than those two. However, HierFIT outperforms all
418 with the highest hF1-score in average across multiple tasks. Although HierFIT's worst performance
419 appears to be the model generated with inDrop data when tested on CEL-Seq data as 64% hF-score,
420 it is still comparably better than its contenders. Overall, these results show that HierFIT exhibits
421 robustness against various batch effects due to different scRNA-seq methods and performs better
422 than other tools in various aspects.

423 **Discussion**

424 Defining cell types is a fundamental and complex challenge in single-cell biology, which is
425 becoming increasingly difficult as the diversity of single-cell experiments increases. In this study,
426 we attempted to address one of the challenges in the developing field of scRNA-seq with an
427 alternative perspective. Hierarchical classification, as opposed to common flat classifiers, is
428 currently generating more interest in the community because it takes the cell type relationships
429 into account in addition to providing more insight into intermediate cell types (Wu and Wu, 2020).
430 The hierarchical approach has been used in many other fields including medical sciences
431 (Dimitrovski et al., 2011).

432 In this work, we hypothesized that hierarchical organization of cell types and class
433 relationships will provide more accurate decisions compared to flat classification approaches. We,
434 then, implemented our approach as a user-friendly R package and evaluated its performance with
435 commonly used public datasets. We demonstrated HieRFIT's better classification of PBMC cell
436 types, even in low abundances, in concordance with their marker gene expression profiles as
437 opposed to manual annotations by widely used software, Seurat. In addition, the performance
438 evaluations against other available single-cell classification tools and machine learning algorithms
439 showed that HieRFIT provided the most reasonably accurate results. HieRFIT's performance stayed
440 stable across various types of datasets produced with different methods while other tools
441 sometimes showed diminished accuracy, in particular, inter-dataset tasks. With its 'divide and
442 conquer' approach, HieRFIT was able handle very complex tasks with a large number of cell types
443 and total cells without any issue.

444 Furthermore, HieRFIT showed consistently better performance on classification challenges
445 against two other tools, CHETAH and scClassify, which have similar hierarchical classification
446 approaches. Both CHETAH and scClassify learn tree topologies directly from reference data as

447 CHETAH builds a binary tree of cell types using average linkage distances based on their Spearman
448 correlations while scClassify uses hierarchical ordered partitioning and collapsing hybrid
449 (HOPACH) algorithm for tree construction which allows multi-children nodes. However, both tools
450 define the intermediate cell types with labels that are hard to interpret. HieRFIT on the other hand
451 covers both approaches by providing users an option, in addition to ability to create a de novo tree,
452 to define a tree containing intermediate nodes with meaningful cell labels as opposed to other tools.

453 We implemented the 'local classifier per parent node' (LCPN) approach in HieRFIT as
454 opposed to the global classifier approach that takes the entire tree topology into a single model.
455 Hierarchical classification implemented in LCPN attitude has been reported to have better accuracy
456 as compared to flat classifiers (Gauch et al., 2009, Jin et al., 2008, Xiao et al., 2007). In addition,
457 using various combinations of different classification algorithms as local classifiers has previously
458 been reported (Secker et al., 2007). One important feature of HieRFIT originates from its 'non-
459 mandatory leaf node prediction' based decision scheme which allows intermediate nodes to be
460 assigned as well. Our decision rule is based on choosing the best scoring node on the tree.
461 Alternative decision approaches have been proposed, e.g. 'sequential boolean decision rule' which
462 chooses child nodes, starting from root, until reaching to a leaf node (Bryant, 1992). However, this
463 approach might be prone to error propagation more than other top-down approaches. The
464 challenge is to properly combine local classifiers so that their unbiased outputs can be used for the
465 decision making process. It is commonly known that machine learning based classifiers are prone
466 to imbalanced class sizes in addition to other intrinsic biases such as batch effect. To account for
467 these, we utilized a certainty function derived from asymmetric entropy which provided precise
468 confidence estimations about class assignment. The path certainty metric accumulates higher
469 scores when correct cell types are picked along the ancestral path while their siblings and out-
470 groups behave as antagonists. Thus, accumulated confidence allows better decision regardless of
471 the complexity of data or tree topology.

472 As all other computational approaches, HieRFIT also has several limitations. First of all, it
473 requires a reference data with properly annotated cell types. Although relying on reference data
474 and its cell types can introduce biases due to inconsistencies in annotations, HieRFIT's ensemble
475 based classifiers, random forest, can compensate for subtle fluctuations. Reference based
476 classification approaches usually miss the opportunity to discover novel cell types due to their
477 dependency on prior information. Another limitation of reliance on reference data is that some cell
478 types are represented with low numbers of cells. However, with the fast development of new
479 methods, single-cell based atlas projects provide exponentially increasing datasets. Secondly,
480 HieRFIT relies on a user provided tree, a predefined class hierarchy, and assumes that the tree
481 topology reflects biological cell type relationships with their underlying gene expression profiles in
482 the reference data. To prevent senseless results, users must be cautious about providing a tree
483 topology for classification purposes with HieRFIT. If a user skips to provide a cell type tree, creating
484 a class hierarchy by learning from data (e.g. by hierarchical clustering) can also be limited since
485 similarity driven hierarchy is prone to data specific artifacts and over-fitting.

486 In this study, we proposed to utilize hierarchical relationships between cell types to better
487 harvest biological information and provide more insight about the cell type identities. HieRFIT
488 provides stable and accurate cell type classification of single-cell RNA-seq data with hierarchical
489 manner. It will contribute to the field not only by providing a new perspective and faster cell type
490 projections from larger atlas projects but also allowing cross comparisons across various datasets
491 effectively.

492 **Acknowledgements**

493 Special thanks to Aaron Kitzmiller, Hansaim Lim, and Varenka Rodriguez Diblasi.

494 **Software availability**

495 HieRFIT is available as an R package through GitHub

496 (<https://github.com/yasinkaymaz/HieRFIT>).

497

498 References

- 499 2018. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature*, 562,
500 367-372.
- 501 ABDELAAL, T., MICHIELSEN, L., CATS, D., HOOGDUIJN, D., MEI, H., REINDERS, M. J. T. &
502 MAHFOUZ, A. 2019a. A comparison of automatic cell identification methods for
503 single-cell RNA sequencing data. *Genome Biology*, 20, 194.
- 504 ABDELAAL, T., MICHIELSEN, L., CATS, D., HOOGDUIJN, D., MEI, H., REINDERS, M. J. T. &
505 MAHFOUZ, A. 2019b. A comparison of automatic cell identification methods for
506 single-cell RNA sequencing data. *Genome Biol.*, 20, 194.
- 507 AEVERMANN, B. D., NOVOTNY, M., BAKKEN, T., MILLER, J. A., DIEHL, A. D., OSUMI-
508 SUTHERLAND, D., LASKEN, R. S., LEIN, E. S. & SCHEUERMANN, R. H. 2018. Cell type
509 discovery using single-cell transcriptomics: implications for ontological
510 representation. *Human Molecular Genetics*, 27, R40-R47.
- 511 BARON, M., VERES, A., WOLOCK, SAMUEL L., FAUST, AUBREY L., GAUJOUX, R., VETERE, A.,
512 RYU, JENNIFER H., WAGNER, BRIDGET K., SHEN-ORR, SHAI S., KLEIN, ALLON M.,
513 MELTON, DOUGLAS A. & YANAI, I. 2016. A Single-Cell Transcriptomic Map of the
514 Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure. *Cell*
515 *Systems*, 3, 346-360.e4.
- 516 BRYANT, R. E. 1992. Symbolic Boolean manipulation with ordered binary-decision
517 diagrams. *ACM Comput. Surv.*, 24, 293-318.
- 518 BUTLER, A., HOFFMAN, P., SMIBERT, P., PAPALEXI, E. & SATIJA, R. 2018. Integrating single-
519 cell transcriptomic data across different conditions, technologies, and species. *Nat.*
520 *Biotechnol.*
- 521 CAMPBELL, J. N., MACOSKO, E. Z., FENSELAU, H., PERS, T. H., LYUBETSKAYA, A., TENEN, D.,
522 GOLDMAN, M., VERSTEGEN, A. M. J., RESCH, J. M., MCCARROLL, S. A., ROSEN, E. D.,
523 LOWELL, B. B. & TSAI, L. T. 2017. A molecular census of arcuate hypothalamus and
524 median eminence cell types. *Nat. Neurosci.*, 20, 484-496.
- 525 CAO, J., PACKER, J. S., RAMANI, V., CUSANOVICH, D. A., HUYNH, C., DAZA, R., QIU, X., LEE, C.,
526 FURLAN, S. N., STEEMERS, F. J., ADEY, A., WATERSTON, R. H., TRAPNELL, C. &
527 SHENDURE, J. 2017. Comprehensive single-cell transcriptional profiling of a
528 multicellular organism. *Science*, 357, 661-667.
- 529 CAO, J., SPIELMANN, M., QIU, X., HUANG, X., IBRAHIM, D. M., HILL, A. J., ZHANG, F.,
530 MUNDLOS, S., CHRISTIANSEN, L., STEEMERS, F. J., TRAPNELL, C. & SHENDURE, J.
531 2019. The single-cell transcriptional landscape of mammalian organogenesis.
532 *Nature*, 1.
- 533 DE KANTER, J. K., LIJNZAAD, P., CANDELLI, T., MARGARITIS, T. & HOLSTEGE, F. C. P. 2019.
534 CHETAH: a selective, hierarchical cell type identification method for single-cell RNA
535 sequencing. *Nucleic Acids Research*, 47, e95-e95.
- 536 DIMITROVSKI, I., KOCEV, D., LOSKOVSKA, S. & DŽEROSKI, S. 2011. Hierarchical annotation
537 of medical images. *Pattern Recognit.*, 44, 2436-2449.
- 538 DING, J., ADICONIS, X., SIMMONS, S. K., KOWALCZYK, M. S., HESSION, C. C., MARJANOVIC, N.
539 D., HUGHES, T. K., WADSWORTH, M. H., BURKS, T., NGUYEN, L. T., KWON, J. Y. H.,
540 BARAK, B., GE, W., KEDAIGLE, A. J., CARROLL, S., LI, S., HACOEN, N., ROZENBLATT-

- 541 ROSEN, O., SHALEK, A. K., VILLANI, A.-C., REGEV, A. & LEVIN, J. Z. 2019. Systematic
542 comparative analysis of single cell RNA-sequencing methods. *bioRxiv*, 632216.
- 543 GAUCH, S., CHANDRAMOULI, A. & RANGANATHAN, S. 2009. Training a hierarchical
544 classifier using inter document relationships. *J. Am. Soc. Inf. Sci.*, 60, 47-58.
- 545 JIN, B., MULLER, B., ZHAI, C. & LU, X. 2008. Multi-label literature classification based on the
546 Gene Ontology graph. *BMC Bioinformatics*, 9, 525.
- 547 KIRITCHENKO, S., MATWIN, S. & FAZEL FAMILI, A. 2005. Functional annotation of genes
548 using hierarchical text categorization. in *Proc. of the BioLINK SIG: Linking Literature,*
549 *Information and Knowledge for Biology (held at ISMB-05).*
- 550 KISELEV, V. Y., ANDREWS, T. S. & HEMBERG, M. 2019. Challenges in unsupervised
551 clustering of single-cell RNA-seq data. *Nature Reviews Genetics*, 20, 273-282.
- 552 KISELEV, V. Y., KIRSCHNER, K., SCHAUB, M. T., ANDREWS, T., YIU, A., CHANDRA, T.,
553 NATARAJAN, K. N., REIK, W., BARAHONA, M., GREEN, A. R. & HEMBERG, M. 2017.
554 SC3: consensus clustering of single-cell RNA-seq data. *Nature Methods*, 14, 483-486.
- 555 LÄHNEMANN, D., KÖSTER, J., SZCZUREK, E., MCCARTHY, D. J., HICKS, S. C., ROBINSON, M. D.,
556 VALLEJOS, C. A., CAMPBELL, K. R., BEERENWINKEL, N., MAHFOUZ, A., PINELLO, L.,
557 SKUMS, P., STAMATAKIS, A., ATTOLINI, C. S.-O., APARICIO, S., BAAIJENS, J.,
558 BALVERT, M., BARBANSON, B. D., CAPPUCCIO, A., CORLEONE, G., DUTILH, B. E.,
559 FLORESCU, M., GURYEV, V., HOLMER, R., JAHN, K., LOBO, T. J., KEIZER, E. M., KHATRI,
560 I., KIELBASA, S. M., KORBEL, J. O., KOZLOV, A. M., KUO, T.-H., LELIEVELDT, B. P. F.,
561 MANDOIU, I. I., MARIONI, J. C., MARSCHALL, T., MÖLDER, F., NIKNEJAD, A.,
562 RACZKOWSKI, L., REINDERS, M., RIDDER, J. D., SALIBA, A.-E., SOMARAKIS, A.,
563 STEGLE, O., THEIS, F. J., YANG, H., ZELIKOVSKY, A., MCHARDY, A. C., RAPHAEL, B. J.,
564 SHAH, S. P. & SCHÖNHUTH, A. 2020. Eleven grand challenges in single-cell data
565 science. *Genome Biol.*, 21, 31.
- 566 LIN, Y., CAO, Y., KIM, H. J., SALIM, A., SPEED, T. P., LIN, D., YANG, P. & YANG, J. Y. H. 2019.
567 scClassify: hierarchical classification of cells. *bioRxiv*, 776948.
- 568 MA, F. & PELLEGRINI, M. 2019. ACTINN: automated identification of cell types in single cell
569 RNA sequencing. *Bioinformatics*, 36, 533-538.
- 570 MACAULAY, I. C., SVENSSON, V., LABALETTE, C., FERREIRA, L., HAMEY, F., VOET, T.,
571 TEICHMANN, S. A. & CVEJIC, A. 2016. Single-Cell RNA-Sequencing Reveals a
572 Continuous Spectrum of Differentiation in Hematopoietic Cells. *Cell Rep.*, 14, 966-
573 977.
- 574 MARCELLIN, S., ZIGHED, D. A. & RITSCHARD, G. 2006. An asymmetric entropy measure for
575 decision trees.
- 576 MURARO, M. J., DHARMADHIKARI, G., GRÜN, D., GROEN, N., DIELEN, T., JANSEN, E., VAN
577 GURP, L., ENGELSE, M. A., CARLOTTI, F., DE KONING, E. J. & VAN OUDENAARDEN, A.
578 2016. A Single-Cell Transcriptome Atlas of the Human Pancreas. *Cell Syst*, 3, 385-
579 394.e3.
- 580 PETEGROSSO, R., LI, Z. & KUANG, R. 2020. Machine learning and statistical methods for
581 clustering single-cell RNA-sequencing data. *Brief Bioinform*, 21, 1209-1223.
- 582 PLASSCHAERT, L. W., ŽILIONIS, R., CHOO-WING, R., SAVOVA, V., KNEHR, J., ROMA, G.,
583 KLEIN, A. M. & JAFFE, A. B. 2018. A single-cell atlas of the airway epithelium reveals
584 the CFTR-rich pulmonary ionocyte. *Nature*, 560, 377-381.
- 585 PLATT, J. C. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to
586 Regularized Likelihood Methods. *ADVANCES IN LARGE MARGIN CLASSIFIERS*.

- 587 PLINER, H. A., SHENDURE, J. & TRAPNELL, C. 2019. Supervised classification enables rapid
588 annotation of cell atlases. *Nature Methods*, 16, 983-986.
- 589 ROSENBERG, A. B., ROCO, C. M., MUSCAT, R. A., KUCHINA, A., SAMPLE, P., YAO, Z.,
590 GRAYBUCK, L. T., PEELER, D. J., MUKHERJEE, S., CHEN, W., PUN, S. H., SELLERS, D. L.,
591 TASIC, B. & SEELIG, G. 2018. Single-cell profiling of the developing mouse brain and
592 spinal cord with split-pool barcoding. *Science*, 360, 176-182.
- 593 SECKER, A. D., DAVIES, M. N., FREITAS, A. A., TIMMIS, J., MENDAO, M. & FLOWER, D. R.
594 2007. An experimental comparison of classification algorithms for hierarchical
595 prediction of protein function. *Expert Update (Magazine of the British Computer*
596 *Society's Specialist Group on AI)*, 9, 17-22.
- 597 SEGERSTOLPE, Å., PALASANTZA, A., ELIASSON, P., ANDERSSON, E. M., ANDRÉASSON, A. C.,
598 SUN, X., PICELLI, S., SABIRSH, A., CLAUSEN, M., BJURSELL, M. K., SMITH, D. M.,
599 KASPER, M., ÄMMÄLÄ, C. & SANDBERG, R. 2016. Single-Cell Transcriptome Profiling
600 of Human Pancreatic Islets in Health and Type 2 Diabetes. *Cell Metab*, 24, 593-607.
- 601 SUO, S., ZHU, Q., SAADATPOUR, A., FEI, L., GUO, G. & YUAN, G.-C. 2018. Revealing the Critical
602 Regulators of Cell Identity in the Mouse Cell Atlas. *Cell Reports*, 25, 1436-1445.e3.
- 603 TABULA MURIS, C., OVERALL, C., LOGISTICAL, C., ORGAN, C., PROCESSING, LIBRARY, P.,
604 SEQUENCING, COMPUTATIONAL DATA, A., CELL TYPE, A., WRITING, G.,
605 SUPPLEMENTAL TEXT WRITING, G. & PRINCIPAL, I. 2018. Single-cell
606 transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature*, 562, 367-372.
- 607 TAN, Y. & CAHAN, P. 2019. SingleCellNet: A Computational Tool to Classify Single Cell RNA-
608 Seq Data Across Platforms and Across Species. *Cell Systems*, 9, 207-213.e2.
- 609 TANG, M., KAYMAZ, Y., LOGEMAN, B., EICHHORN, S., LIANG, Z. S., DULAC, C. & SACKTON, T.
610 B. 2020. Evaluating single-cell cluster stability using the Jaccard similarity index.
611 *bioRxiv*.
- 612 TASIC, B., YAO, Z., GRAYBUCK, L. T., SMITH, K. A., NGUYEN, T. N., BERTAGNOLLI, D., GOLDY,
613 J., GARREN, E., ECONOMO, M. N., VISWANATHAN, S., PENN, O., BAKKEN, T., MENON,
614 V., MILLER, J., FONG, O., HIROKAWA, K. E., LATHIA, K., RIMORIN, C., TIEU, M.,
615 LARSEN, R., CASPER, T., BARKAN, E., KROLL, M., PARRY, S., SHAPOVALOVA, N. V.,
616 HIRSCHSTEIN, D., PENDERGRAFT, J., SULLIVAN, H. A., KIM, T. K., SZAFER, A., DEE, N.,
617 GROBLEWSKI, P., WICKERSHAM, I., CETIN, A., HARRIS, J. A., LEVI, B. P., SUNKIN, S.
618 M., MADISEN, L., DAIGLE, T. L., LOOGER, L., BERNARD, A., PHILLIPS, J., LEIN, E.,
619 HAWRYLYCZ, M., SVOBODA, K., JONES, A. R., KOCH, C. & ZENG, H. 2018. Shared and
620 distinct transcriptomic cell types across neocortical areas. *Nature*, 563, 72-78.
- 621 TIAN, L., DONG, X., FREYTAG, S., KA, L. C., SU, S., JALALABADI, A., AMANN-ZALCENSTEIN, D.,
622 WEBER, T. S., SEIDI, A., JABBARI, J. S., NAIK, S. H. & RITCHIE, M. E. 2019.
623 Benchmarking single cell RNA-sequencing analysis pipelines using mixture control
624 experiments. *Nat Methods*, 16, 479-487.
- 625 WILBREY-CLARK, A., ROBERTS, K. & TEICHMANN, S. A. 2020. Cell Atlas technologies and
626 insights into tissue architecture. *Biochemical Journal*, 477, 1427-1442.
- 627 WU, Z. & WU, H. 2020. Accounting for cell type hierarchy in evaluating single cell RNA-seq
628 clustering. *Genome Biol*, 21, 123.
- 629 XIAO, Z., DELLANDREA, E., DOU, W. & CHEN, L. 2007. Hierarchical classification of
630 emotional speech. *IEEE Trans. Multimedia*.

631 XIN, Y., KIM, J., OKAMOTO, H., NI, M., WEI, Y., ADLER, C., MURPHY, A. J., YANCOPOULOS, G.
632 D., LIN, C. & GROMADA, J. 2016. RNA Sequencing of Single Human Islet Cells Reveals
633 Type 2 Diabetes Genes. *Cell Metab*, 24, 608-615.

634 ZEISEL, A., HOCHGERNER, H., LÖNNERBERG, P., JOHNSON, A., MEMIC, F., VAN DER ZWAN,
635 J., HÄRING, M., BRAUN, E., BORM, L. E., LA MANNO, G., CODELUPPI, S., FURLAN, A.,
636 LEE, K., SKENE, N., HARRIS, K. D., HJERLING-LEFFLER, J., ARENAS, E., ERNFORS, P.,
637 MARKLUND, U. & LINNARSSON, S. 2018. Molecular Architecture of the Mouse
638 Nervous System. *Cell*, 174, 999-1014.e22.

639 ZHENG, G. X., TERRY, J. M., BELGRADER, P., RYVKIN, P., BENT, Z. W., WILSON, R., ZIRALDO,
640 S. B., WHEELER, T. D., MCDERMOTT, G. P., ZHU, J., GREGORY, M. T., SHUGA, J.,
641 MONTESCLAROS, L., UNDERWOOD, J. G., MASQUELIER, D. A., NISHIMURA, S. Y.,
642 SCHNALL-LEVIN, M., WYATT, P. W., HINDSON, C. M., BHARADWAJ, R., WONG, A.,
643 NESS, K. D., BEPPU, L. W., DEEG, H. J., MCFARLAND, C., LOEB, K. R., VALENTE, W. J.,
644 ERICSON, N. G., STEVENS, E. A., RADICH, J. P., MIKKELSEN, T. S., HINDSON, B. J. &
645 BIELAS, J. H. 2017a. Massively parallel digital transcriptional profiling of single cells.
646 *Nat Commun*, 8, 14049.

647 ZHENG, G. X. Y., TERRY, J. M., BELGRADER, P., RYVKIN, P., BENT, Z. W., WILSON, R.,
648 ZIRALDO, S. B., WHEELER, T. D., MCDERMOTT, G. P., ZHU, J., GREGORY, M. T., SHUGA,
649 J., MONTESCLAROS, L., UNDERWOOD, J. G., MASQUELIER, D. A., NISHIMURA, S. Y.,
650 SCHNALL-LEVIN, M., WYATT, P. W., HINDSON, C. M., BHARADWAJ, R., WONG, A.,
651 NESS, K. D., BEPPU, L. W., DEEG, H. J., MCFARLAND, C., LOEB, K. R., VALENTE, W. J.,
652 ERICSON, N. G., STEVENS, E. A., RADICH, J. P., MIKKELSEN, T. S., HINDSON, B. J. &
653 BIELAS, J. H. 2017b. Massively parallel digital transcriptional profiling of single cells.
654 *Nat. Commun.*, 8, 14049.

655 ZIGHED, D. A., RITSCHARD, G. & MARCELLIN, S. 2010. Asymmetric and Sample Size
656 Sensitive Entropy Measures for Supervised Learning. *In: RAS, Z. W. & TSAY, L.-S.*
657 *(eds.) Advances in Intelligent Information Systems*. Berlin, Heidelberg: Springer
658 Berlin Heidelberg.

659 ZUR, R. M., JIANG, Y., PESCE, L. L. & DRUKKER, K. 2009. Noise injection for training artificial
660 neural networks: a comparison with weight decay and early stopping. *Medical*
661 *physics*, 36, 4810-4818.

662

663

664 **Figure Legends:**

665 **Figure 1. HieRFIT workflow overview. Reference model generation and query** 666 **projections.**

667 Overview of HieRFIT reference model generation and prediction of a query cell class. **A)** Main
668 process starts with obtaining a tree as a user input or creating from the data. The steps for
669 generating the reference model with a hierarchical tree: **1.** Pick an internal node i (i.e. node “B”) on
670 the tree, **2.** Re-group its children nodes and create an outgroup node for it, **3.** Extract the input
671 expression data based on new group labels for the node, **4.** Perform Principal Component Analysis
672 and pick the components that separate the class labels for variable feature selection, **5.** perform
673 Wilcoxon Rank sum test to determine differentially expressed features, **6.** Train a local classifier
674 (Random forest) with the group labels and the expression matrix with selected features. Repeat the
675 process until all node classifiers are constructed. **B)** Query of a test cell and certainty calculations.
676 Given an array of feature expressions of the query cell, the first step is to compute the certainty
677 array (U) for the candidate classes. Votes are collected from each node i (i.e. node “B”) for both
678 observed query data and its shuffled data separately. Votes are converted to probabilities using
679 sigmoid calibration with multinomial logistic regression. Using the probability centroids (w_i) as the
680 outcome of the randomized array and the observed probabilities (p_i), compute the certainty value
681 of each class of the node (i.e the certainty of class “E” is 0.24). Repeat the process for every class of
682 all internal nodes. **C)** Determining the cell type/class of a query cell. Step 1: Path certainty scores of
683 each candidate class are computed using the certainty values of nodes for the given query by
684 traversing the tree. The sum of certainty values of outgroup and sibling nodes along the path (nodes
685 in gray) are summed and subtracted from the sum of Certainty values of nodes on the path (nodes
686 in green). Step 2: As the final step, scores are evaluated and the maximum scoring class is returned
687 as the outcome. If none of the classes passes the threshold, α , “Undetermined” is returned.

688 **Figure 2. Demonstration of HierFIT usage on a PBMC dataset.**

689 **A)** The cell type tree used in HierFIT reference model with 68K-PBMC data. **B)** The UMAP
690 representation of 3K-PBMC data from 10X Genomics. Cells are colored with cell types which were
691 identified through Seurat clustering and marker expressions (left), cells are colored with HierFIT
692 reference cell types along with intermediate types specified in the tree file (right). **C)** Alluvial
693 diagram demonstrating the cross comparisons of HierFIT projections with the Seurat cell type
694 labels. Each line connecting the two vertical black columns (left bar: prior labels, right bar:
695 projections) represent a cell and are colored based on its HierFIT projection type. Annotations with
696 less than 1%, 'HSC' and 'Monocyte progenitor' were not shown.

697 **Figure 3. Concordance analysis of Seurat and HierFIT classifications with gene**
698 **expression of cells**

699 **A)** The heatmap representation of the confusion matrix that summarized the projection results of
700 the 3K-PBMC query data with percent distribution among the tree node labels. **B)** Violin plots of
701 CD8A and CD8B genes and their co-expression values projected on UMAP representation of cells
702 classified as "Naïve CD4 T cells" by Seurat while HierFIT predicts them as CD8 cells or its subtypes
703 (upper panel). Similarly, violin plots and co-expression values of IL7R and CCR7 genes projected on
704 UMAP representation of cells predicted as CD4 T cells or its subtypes by HierFIT in concordance
705 with Seurat (lower panel). **C)** Normalized expression distribution of three marker genes, "LYZ" and
706 "FCGR3A (CD16)", markers of "monocytes" and subset "CD16 monocytes", respectively, among the
707 cells classified as "CD14+ Monocytes" by Seurat (upper panel). Similar violin plots for expression
708 distribution of the same set of cells grouped based on HierFIT projections (lower panel).

709

710 **Figure 4. Performance on various types of datasets. Robustness against batch biases.**

711 Performance results on various types of datasets and comparative benchmarking against other cell
712 type classification tools. **A)** A heatmap representing the mean F1 scores of each test dataset for the
713 classification tools. The number of cell types of each dataset is shown below the columns. Failed
714 tests without a score are grayed-out. Percent unlabeled data distribution from each test data is
715 shown with an adjacent box plot for each tool. Asterisk (*): Classification tools with rejection
716 option. **B)** Hierarchical precision (red), recall (cyan), and F-score (green) metrics for the tools
717 HieRFIT, scClassify, CHETAH, and Seurat. **C)** Various categories of projected cell types by HieRFIT
718 based on their position on tree relative to the prior label. In addition to categories in the table,
719 “Correct children” or “Correct grandchildren” categories are also possible in case of a correct sub-
720 level type assignment. Bar plot summarizes the distribution of these categories for HieRFIT outputs
721 among all test datasets above. **D)** Stacked-bar plot summarizes the distribution of projection
722 categories for HieRFIT outputs among all test datasets above.

723 **Figure 5. Robustness against batch biases.**

724 Hierarchical precision, recall, and F1-score values of HieRFIT and 3 tools for comparing the
725 performances in various batches with inter-dataset tests. At each iteration, a dataset from paired
726 PBMCs produced with a scRNA method was used to generate the reference model and tested on the
727 second pair of the PBMCs.

728

729

730

731

732 **Supplemental Tables**

733 **Supplemental Table 1:** An example tab-separated cell type table to be used as an input for tree
734 construction and creating a reference model (referencing 68K PBMC dataset). Each row specifies all
735 ancestral/intermediate cell types of each reference cell type (leaves at the end of rows).

736 **Supplemental Table 2:** Datasets used in the benchmarking analysis.

737

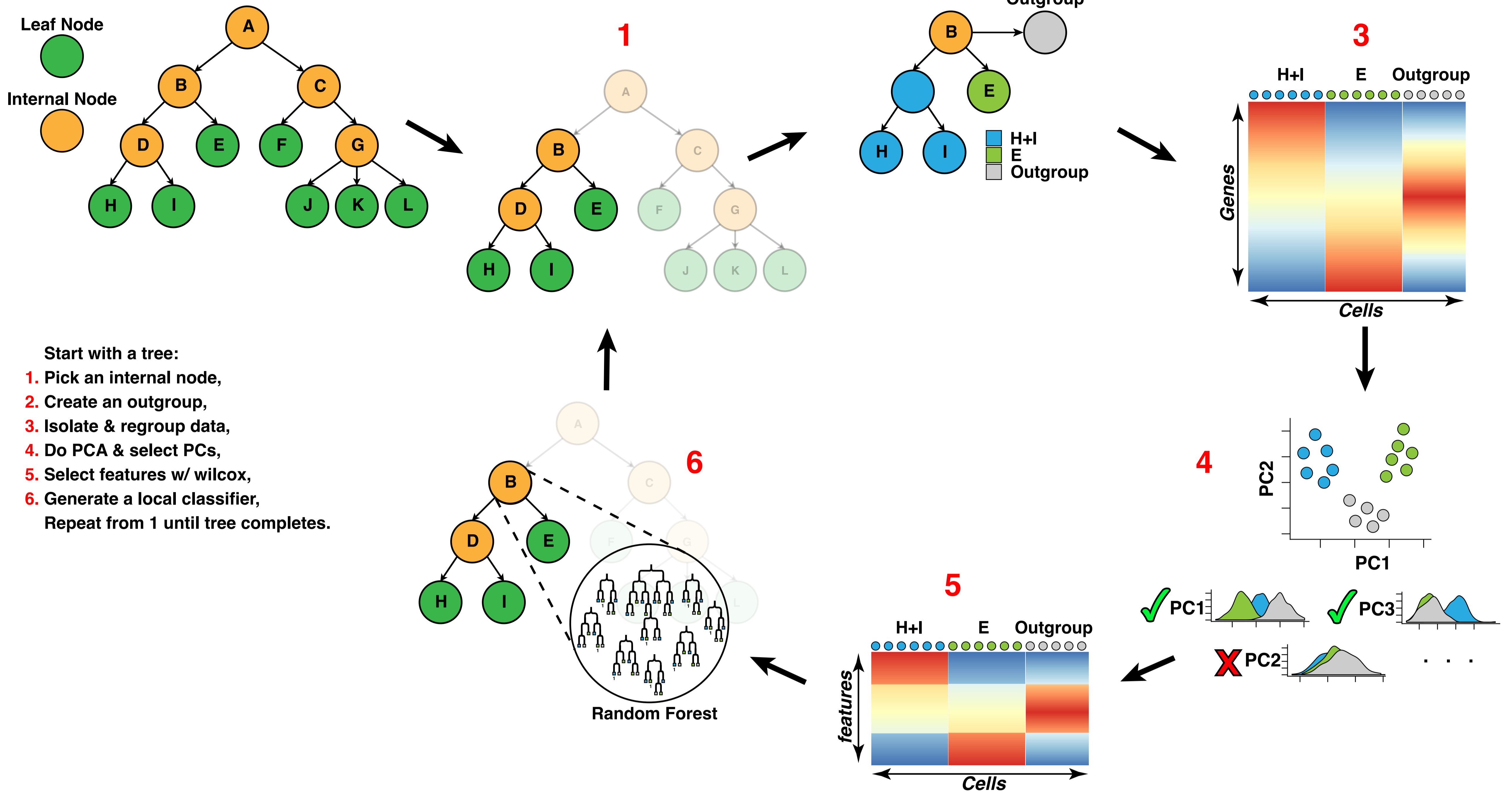
738 **Supplemental Figures**

739 **Supplemental Figure 1.** Boxplot for mean F1 scores distribution of each classification tool as an
740 outcome of 18 datasets with 5-fold cross-validation tests (upper plot). Percent unlabeled data
741 distribution from each test data (middle plot). A heatmap showing the Mean F1 scores of each test
742 dataset for the classification tools (lower panel). Failed tests without a score are grayed-out.

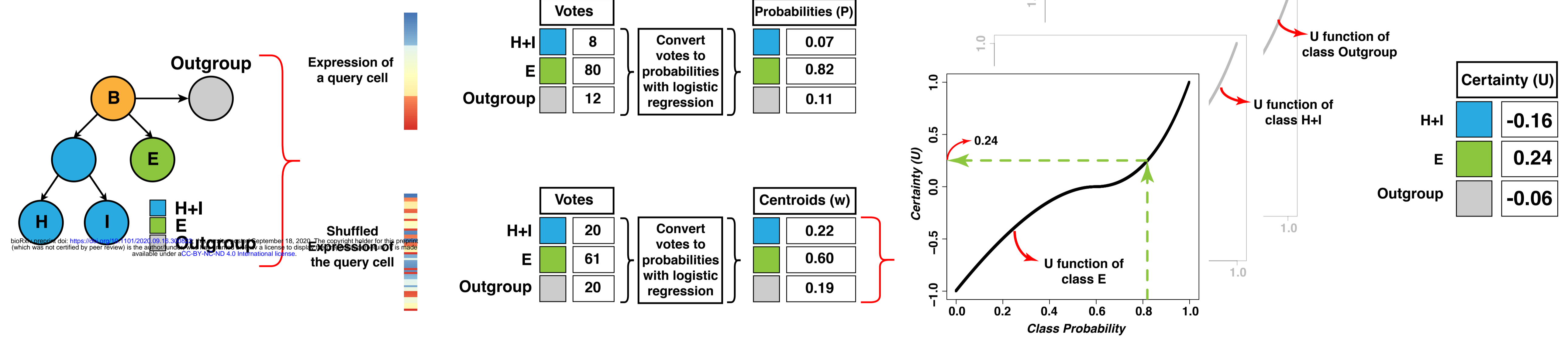
743

Figure 1

A



B



C

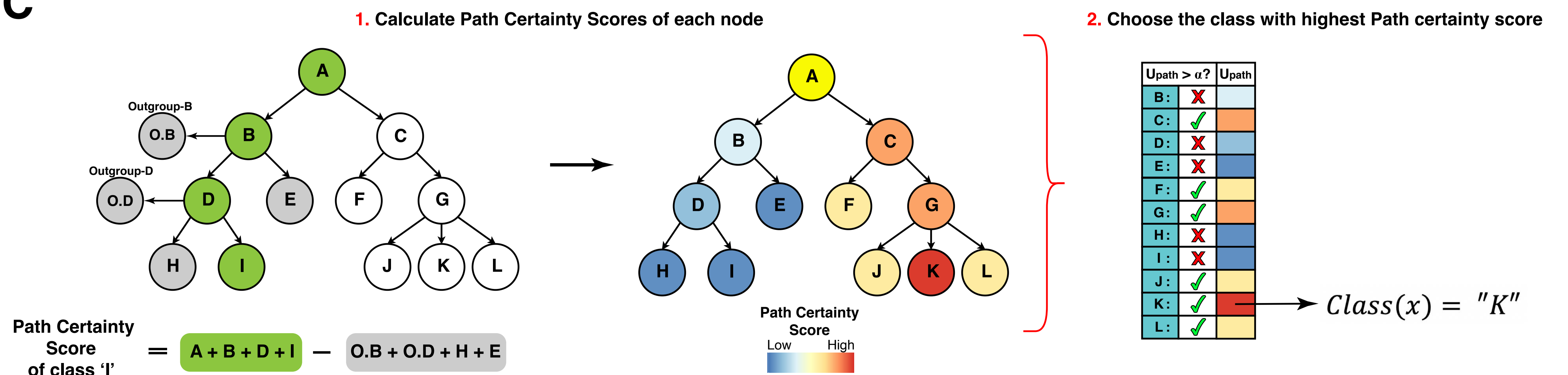
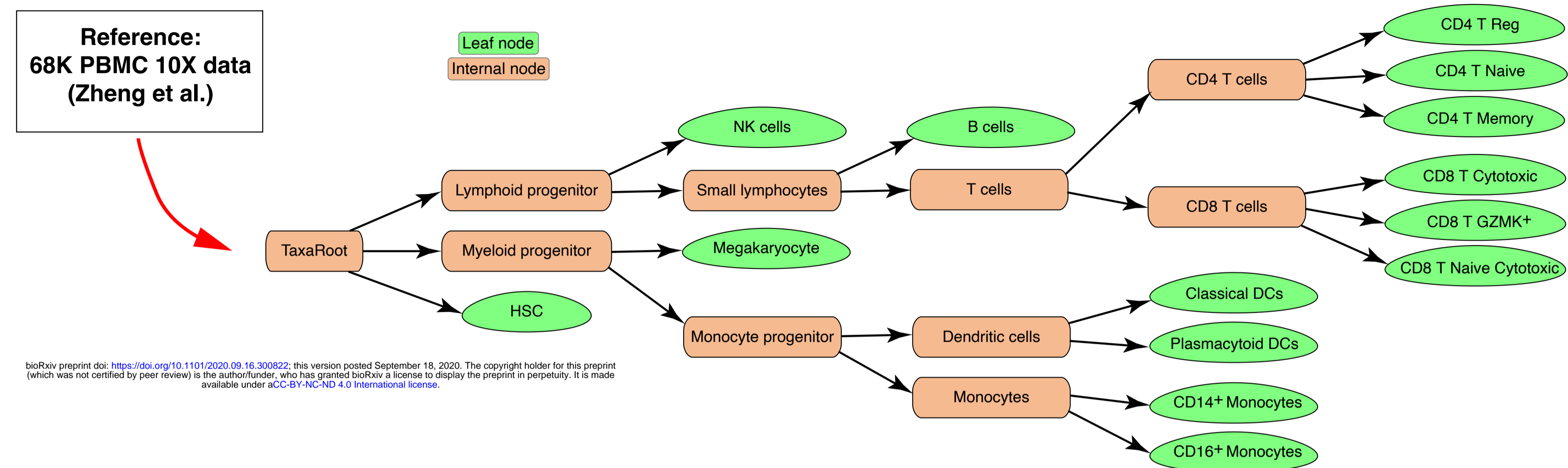


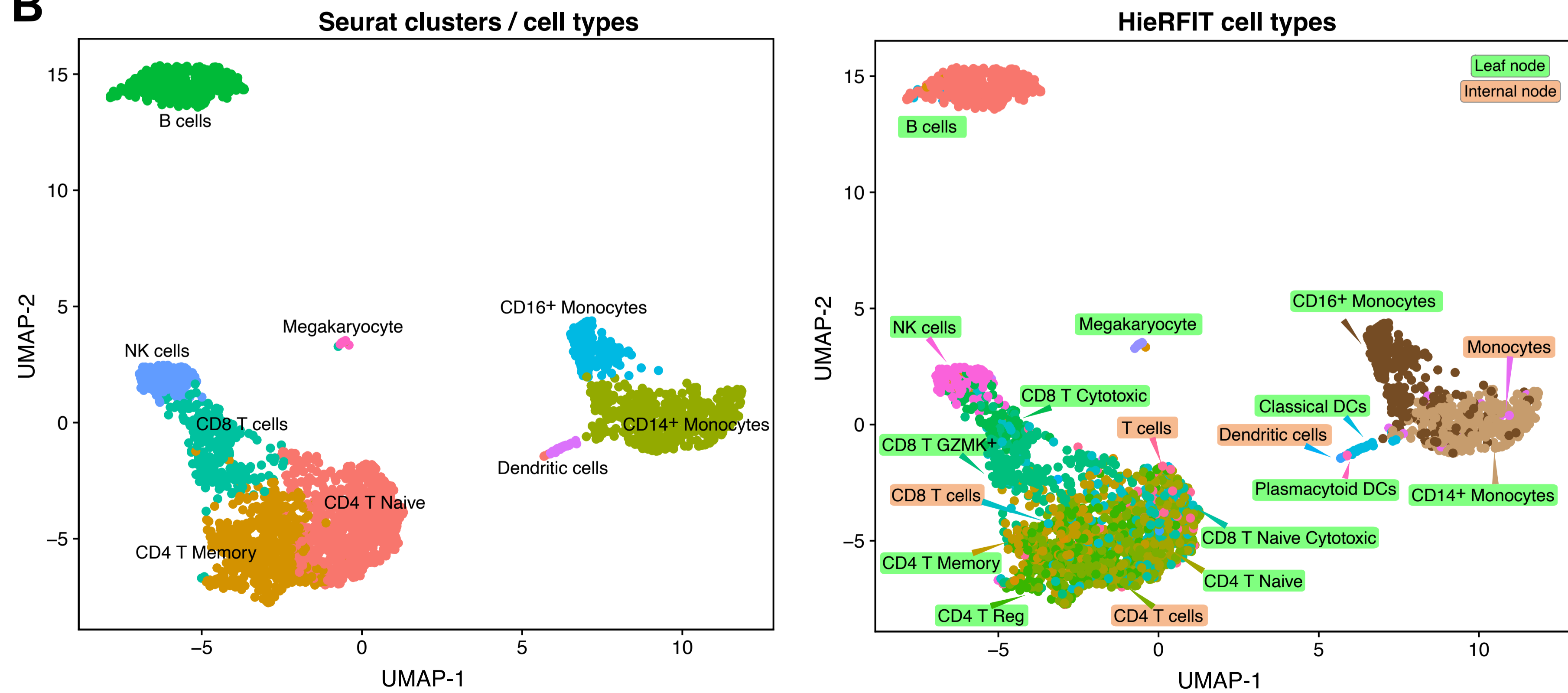
Figure 2

A



bioRxiv preprint doi: <https://doi.org/10.1101/2020.09.16.300822>; this version posted September 18, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

B



C

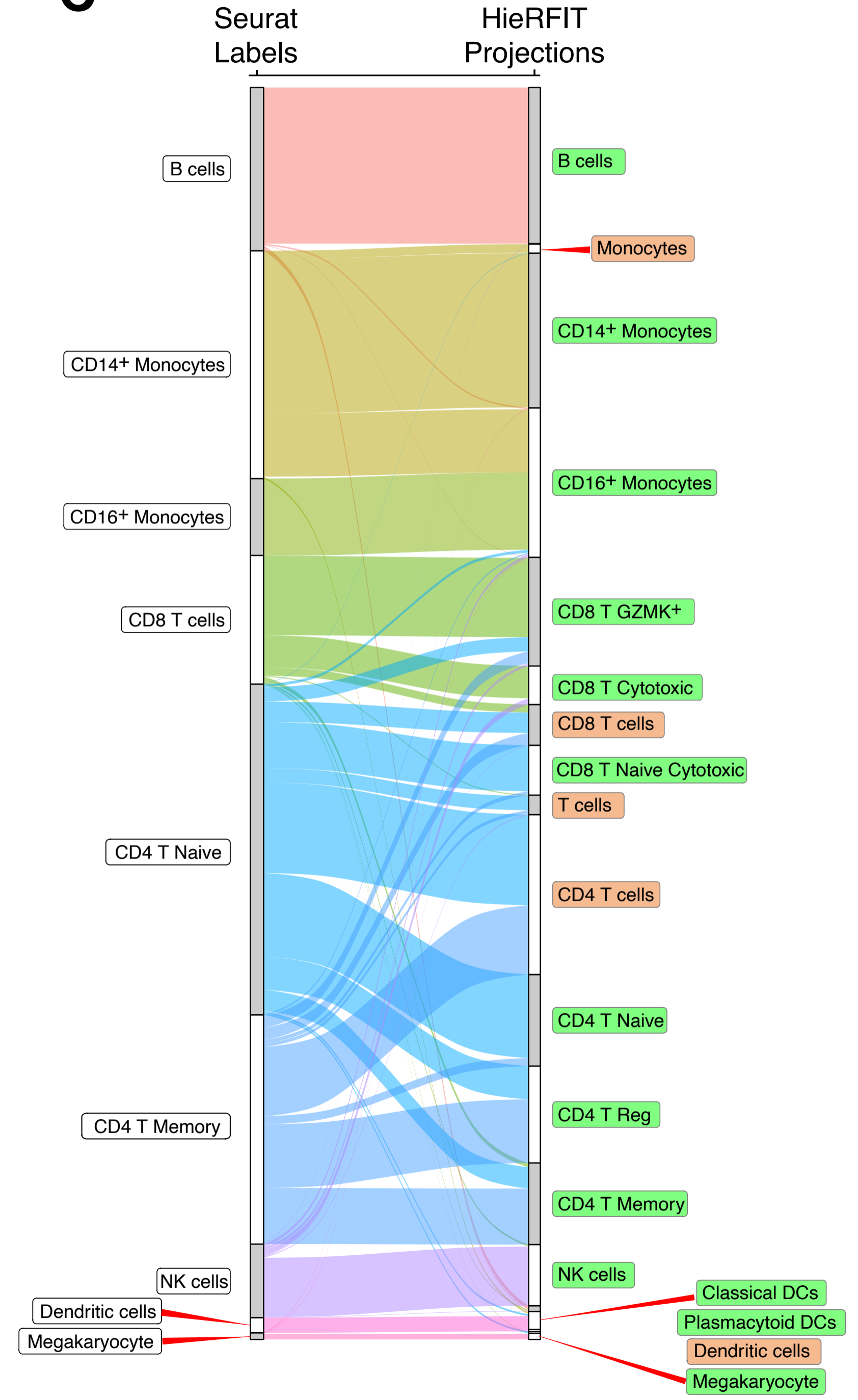
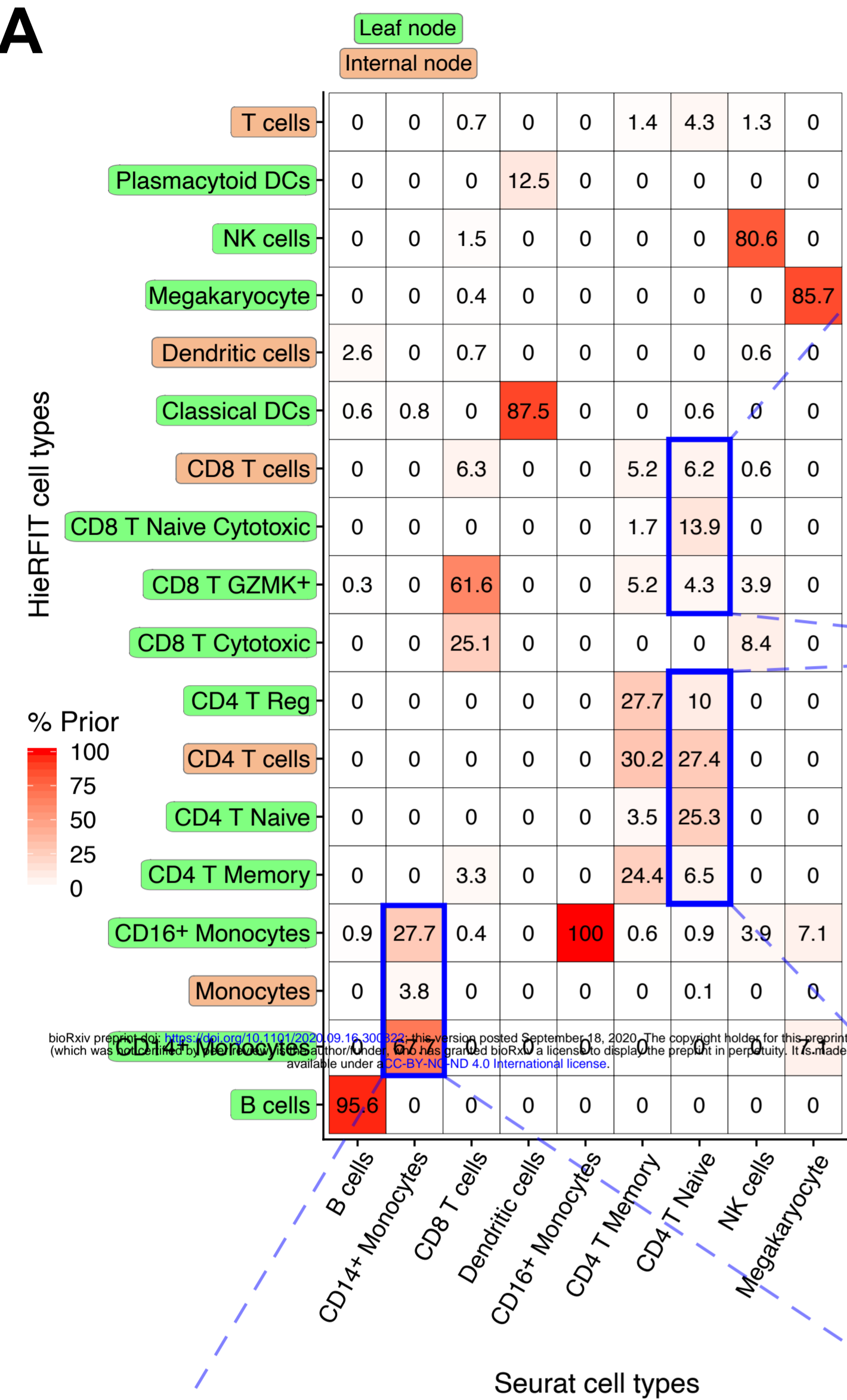
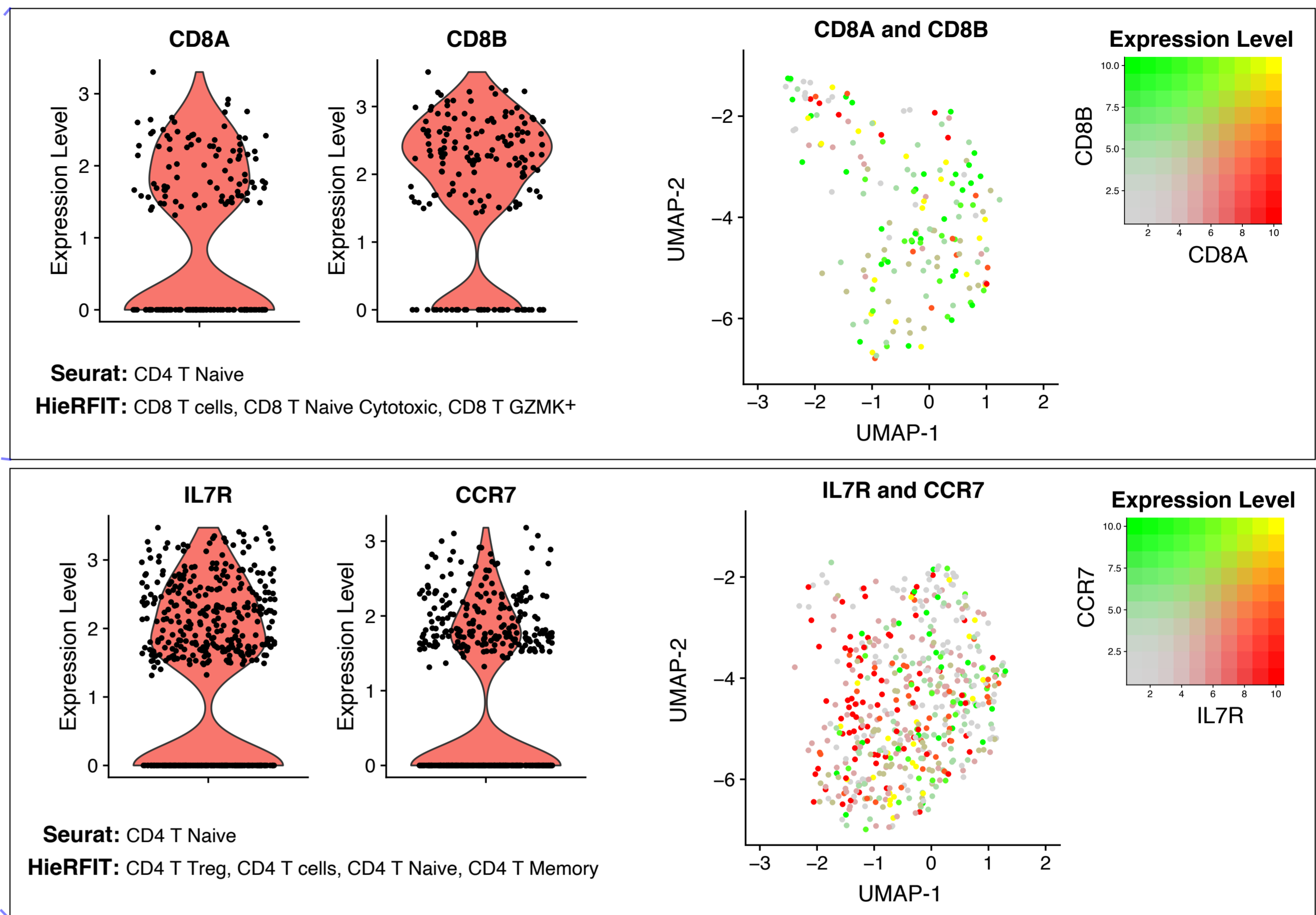


Figure 3

A



B



C

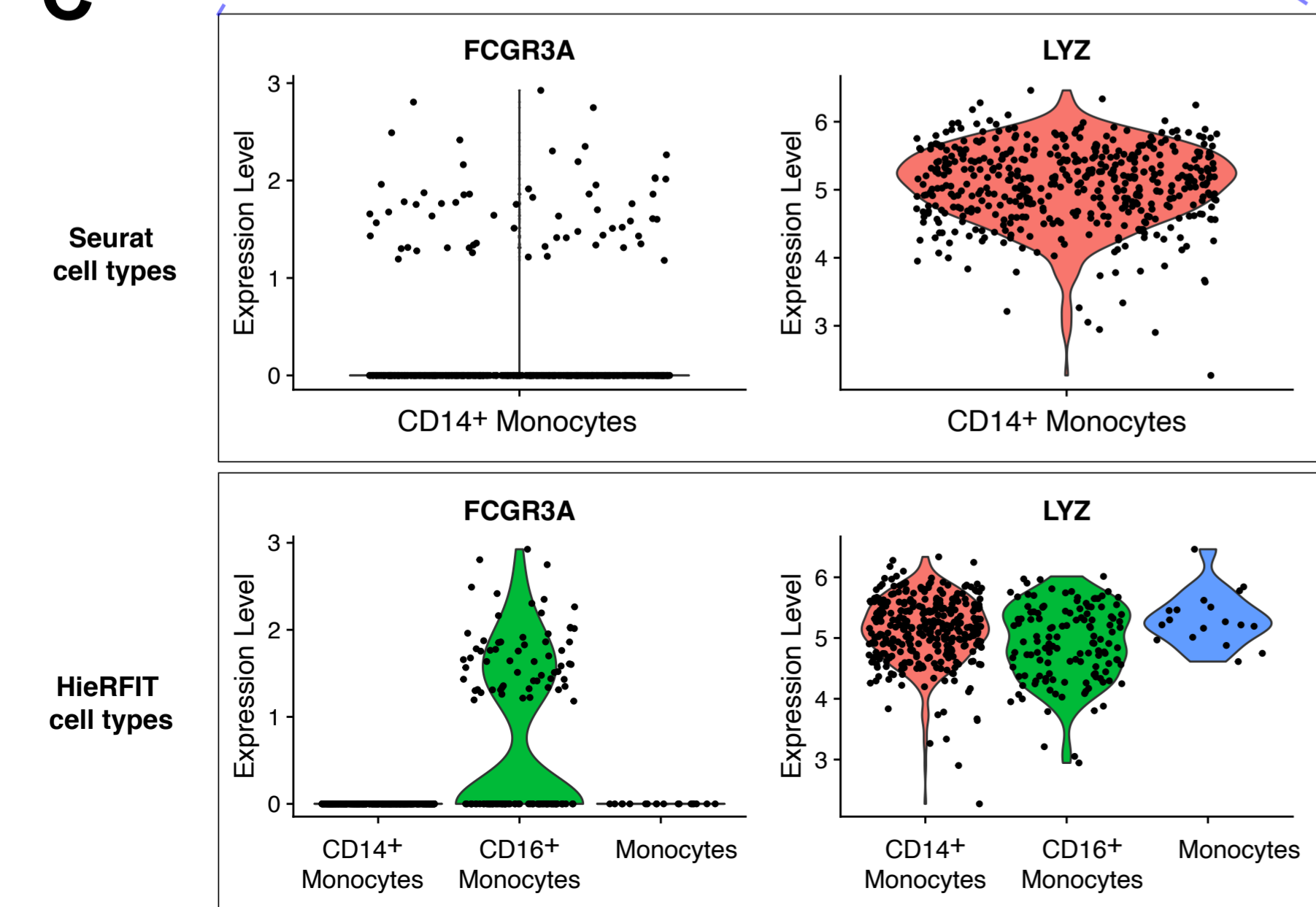
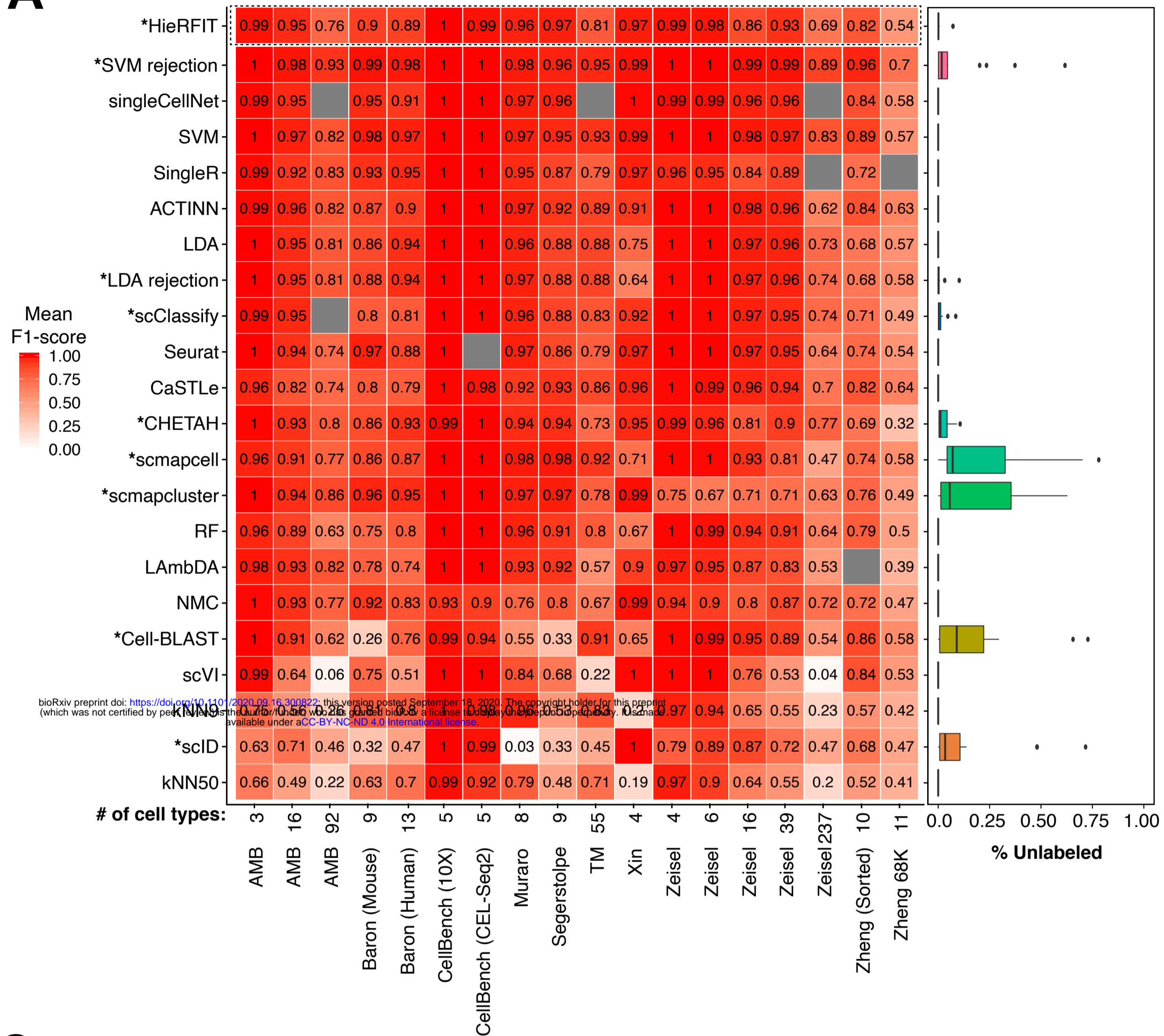
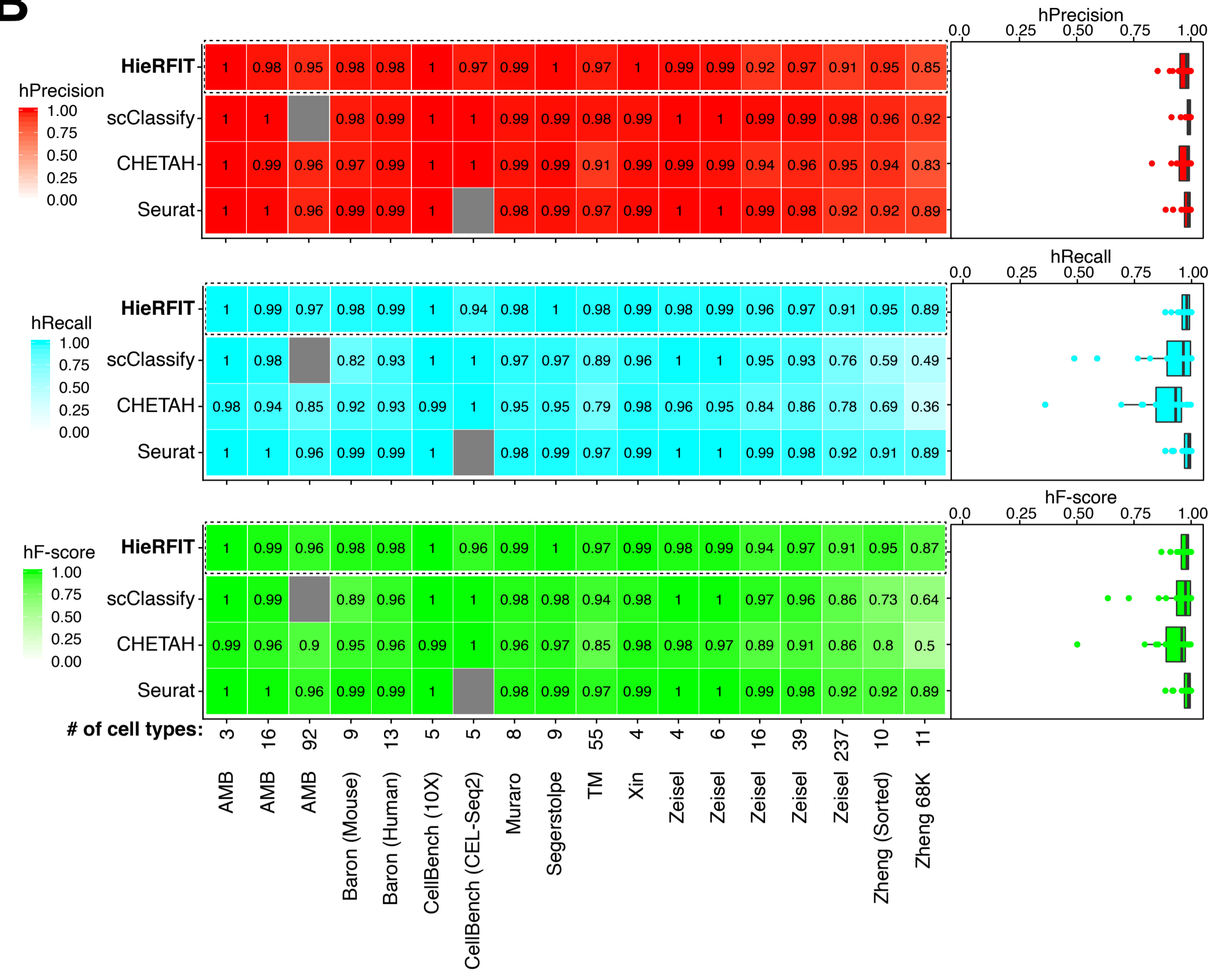


Figure 4

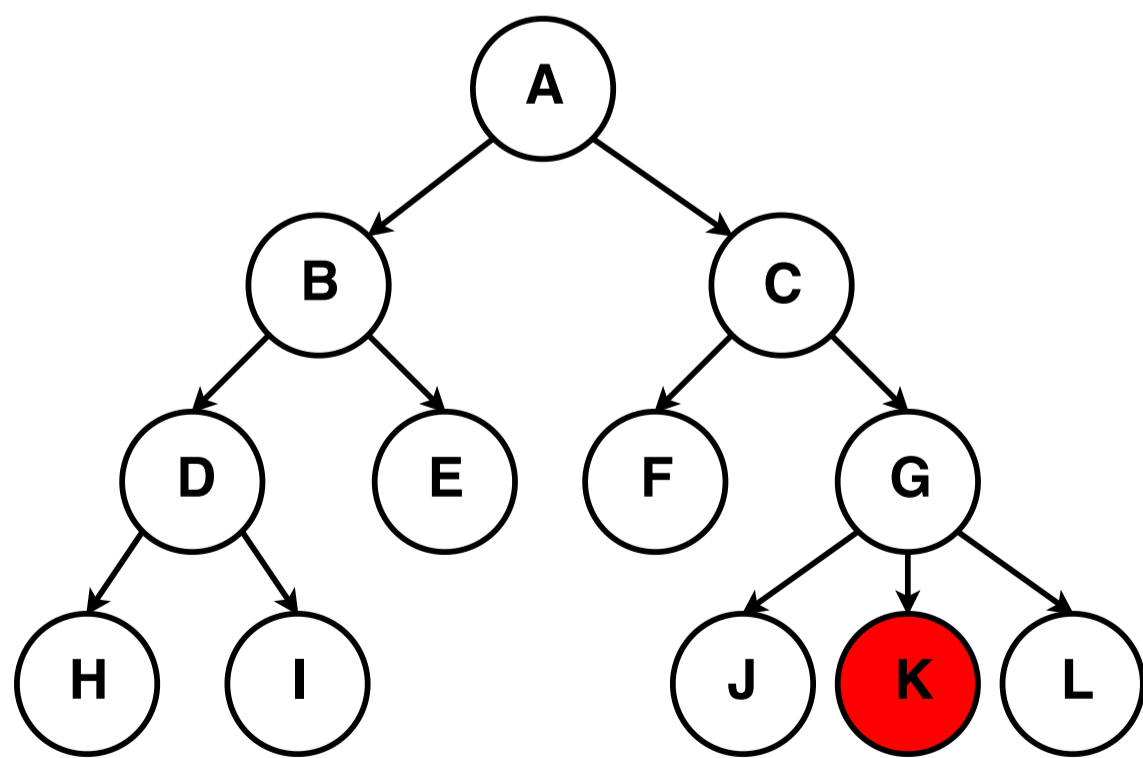
A



B



C



Prior	Projected	Categories
K	K	Correct node
	G	Correct parent node
	C	Correct ancestral node
	L	Incorrect node sibling
	B, D, E, H, I, F	Incorrect clade

D

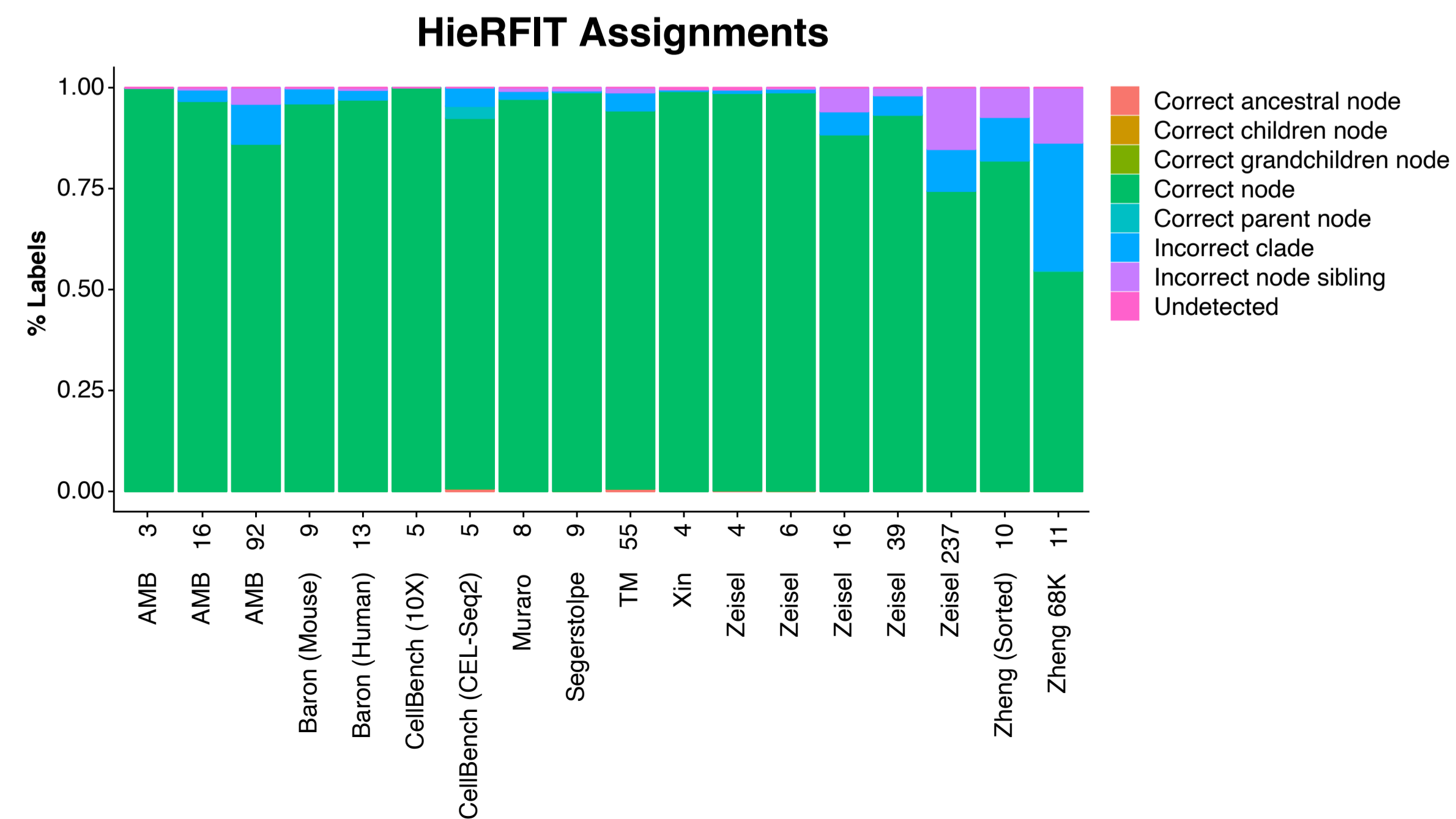


Figure 5

bioRxiv preprint doi: <https://doi.org/10.1101/2020.09.16.300822>; this version posted September 18, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

