# Single-cell reference mapping to construct and extend cell type hierarchies

Lieke Michielsen[1,2,3,+], Mohammad Lotfollahi[4,+], Daniel Strobl[4,5], Lisa Sikkema[4,6], Marcel J.T. Reinders[1,2,3], Fabian J. Theis[4,6,7,*], Ahmed Mahfouz[1,2,3,*]

[1] Department of Human Genetics, Leiden University Medical Center, Einthovenweg 20, 2333ZC, Leiden, The Netherlands
[2] Leiden Computational Biology Center, Leiden University Medical Center, Einthovenweg 20, 2333ZC, Leiden, The Netherlands
[3] Delft Bioinformatics Lab, Delft University of Technology, Van Mourik Broekmanweg 6, 2628XE, Delft, The Netherlands
[4] Institute of Computational Biology, Helmholtz Zentrum München, Munich, Germany
[5] Institute of Clinical Chemistry and Pathobiochemistry, TUM School of Medicine, Technical University of Munich, 81675 Munich, Germany
[6] TUM School of Life Sciences Weihenstephan, Technical University of Munich, Germany
[7] Department of Mathematics, Technical University of Munich, Munich, Germany

[+] These authors contributed equally
* Correspondence to: fabian.theis@helmholtz-muenchen.de & a.mahfouz@lumc.nl

## Abstract

Single-cell genomics is now producing an ever-increasing amount of datasets that, when integrated, could provide large-scale reference atlases of tissue in health as well as disease. Such large-scale atlases not only increase the scale and generalizability of analyses but also enable combining the knowledge generated by individual studies. Specifically, individual studies often differ in terms of cell annotation terminology and depth, with different groups specializing in different cell type compartments, often using distinct terminology. Understanding how these distinct sets of annotations are related and complement each other would mark a major step towards a consensus-based cell type annotation that reflects the latest knowledge in the field. Whereas recent computational techniques, referred to as “reference mapping” methods, facilitate the usage and expansion of existing reference atlases by mapping new datasets (i.e. queries) onto an atlas; a systematic approach towards harmonizing dataset-specific cell type terminology and annotation depth is still lacking. Here, we present “*treeArches*”, a framework to automatically build and extend reference atlases while enriching them with an updatable hierarchy of cell type annotations across different datasets. We demonstrate various use cases for treeArches, from automatically resolving relations between reference and query cell types to identifying unseen cell types not present in the reference, such as disease-associated cell states. We envision *treeArches* enabling data-driven construction of consensus, atlas-level cell type hierarchies, as well as facilitating efficient usage of reference atlases.

## Introduction

Single-cell sequencing technologies have revolutionized our understanding of human health. Hereto, large single-cell datasets - referred to as “reference atlases” - have been built to characterize the cellular heterogeneity of whole organs. The concept of a “reference” suggests it should help analyze and interpret new datasets (called “query”). This, however, is complicated by batch effects between reference and query, limited computational resources, and data privacy and sharing. Recently, we along with others developed computational approaches known as ”reference mapping” methods to address these challenges [1–4].

Reference mapping methods have changed the traditional paradigm of manual and time-consuming cell annotation and novel cell type discovery. Recent efforts to build organ- and body-scale cell atlases in collaborative efforts within consortia such as the human cell atlas (HCA) have leveraged reference mapping to make their resources public for individual users and labs, annotate new datasets, and identify novel cell types by disease to healthy references [5–11]. As a result, users can contextualize their datasets within these references to identify novel cell types from healthy to disease-specific variations and thus enabling the discovery of disease-affected cell types that can be prioritized for treatment design. However, identifying novel unseen cell types requires expert knowledge and additional data analysis. Recent approaches leverage uncertainty-aware classification [2,12] or distance-based approaches [4] combined with a core reference mapping method to flag such unseen cell types. Still, the relation of the flagged cell type is not associated with a specific cell type in the reference such as being a subtype or altered state of a specific cell type or a completely novel cell type not in the reference.

Additionally, in the presence of annotated query data, updating the existing atlas while harmonizing its labels with the query remains an open question due to different labeling schemes, hierarchical relations between the cell types [13], and the presence of study-specific cell types as stated before. Due to the large individual variability, and a large number of diseases, new cell types keep being discovered. To create or extend a reference atlas, we would ideally leverage information from multiple scRNA-seq datasets and harmonize their cell annotations. This, however, is not as easy as it seems since all datasets are annotated at a different resolution. Furthermore, it is difficult to match cell types based on the names only. Databases such as the Cell Ontology try to overcome this problem, but a complete naming convention is still missing [13]. Learning the relation between cell types could even be used to improve the Cell Ontology database.

To address these challenges, we propose a framework building upon recent approaches for reference mapping called single-cell architectural surgery (scArches) [1] and single-cell Hierarchical Progressive Learning (scHPL) [14] to progressively build and update a reference atlas and corresponding hierarchical classifier (treeArches, see **Methods**). Our approach allows users to build a reference atlas using existing integration methods supported by scArches (e.g. scVI, scANVI, totalVI, and all others described in [15]). Next, we can use scHPL to augment this reference atlas by learning the relations between cell types to construct a cell type hierarchy. Afterwards, query data, which can be either annotated or unannotated, can be mapped to the reference. If the query is annotated, the query cells can expand the newly updated tree by highlighting potential novel cell types and their

relationship with other cell types in the reference. Otherwise, the created reference can be used to annotate the query cells and identify new unseen cell types in the query. We show that treeArches can be used to create a reference atlas and corresponding cell type hierarchy from scratch, update an existing reference atlas and the hierarchy by finding novel relations between cell types, and leverage a reference atlas to transfer labels to a new dataset.

## Results

### treeArches enables efficient building and updating of hierarchically annotated reference atlases

treeArches consists of two main steps, which are removing the batch effects between datasets and matching the annotated cell types to construct a cell type hierarchy. Assume that we start with multiple labeled datasets called reference datasets in the following. During the first step, we leverage existing neural network-based reference building models (e.g. sc(AN)VI [15] or scGen [16]) which have been shown to be top performers in recent data benchmarking efforts [17] and compatible with scArches to construct a latent space. Next, we use scHPL to construct the cell type hierarchy (Fig 1A). For each dataset, we train a classifier in the learned latent space and cross-predict the labels of the other dataset(s). Using the confusion matrices, we automatically match the cell types to create a hierarchy. This hierarchy also represents a hierarchical classifier where every node represents a cell type of one or more of the datasets. Afterwards, we can map new query datasets to the learned latent space using architectural surgery, a transfer learning approach to map query datasets to references, implemented by scArches (Fig 1B). Architectural surgery brings the advantage that the count matrices of the reference datasets are not needed anymore for querying the model - instead, we only use the pre-trained neural network architecture. The query datasets can either be labeled or unlabeled. In the case of a labeled dataset, we match the cell types from the query to the reference and again update the hierarchy we had learned on the reference datasets. In the case of an unlabeled query, we annotate the cells using the learned hierarchy.

When matching the cell types or predicting labels of a query dataset, identifying new cell types that are not present in the reference is important. This is only possible when biological variation is preserved when mapping the datasets to the latent space and when the classifier in scHPL recognizes unseen cells that are not present in the tree, i.e. scHPL rejects these cells and identifies them as a new cell type. Within scHPL, a cell is rejected if it meets one of the following criteria: 1) if the posterior probability of the classifier is lower than a threshold which means the predicted label is ambiguous, 2) if the distance between a cell and its closest neighbors is too big, 3) if the reconstruction error after PCA is too high which means the query cell is too different from the reference cell types. These three thresholds are automatically set based on the distribution of the data (see **Methods**)

### treeArches detects new cell types in PBMC datasets

We showcase the method with a relatively straightforward example where we build a cell type hierarchy using one bone marrow and three PBMC datasets [18–21] (Table S1). We consider three datasets as the reference (Freytag, Oetjen, and Sun), and one as the query (10X). Since both scArches and scHPL are invariant to a different order of the datasets,

treeArches will be invariant as well [1,14]. The names of the cell types in the datasets are harmonized already, only not all cell types exist in all datasets (Table S2). Nine out of sixteen cell types are present in all four datasets, while four cell types are specific to the Oetjen dataset. The challenge is thus to match the cell types that exist in multiple datasets correctly and to add other cell types as new nodes to the tree. For this second challenge, it is important to note that these new cell types are not forced to be aligned with other existing cell types during the integration step and that the classifier used by scHPL contains a good rejection option during the matching step.

We remove the batch effects from the reference datasets using scVI [22] and match the cell types in the learned latent space (see **Methods**) (Fig 1C-D, Fig S1). When building the hierarchy, we start with the Freytag dataset and progressively add the Oetjen and Sun datasets. The constructed tree looks almost completely as we would expect: most cell types are matched (e.g. the CD4+ T cells from all three datasets) and cell types that are only found in one dataset are added as new cell types to the tree (e.g. the CD10+ B cells and erythrocytes).

The megakaryocyte progenitor cells from the three datasets, however, do not match. The Freytag and Sun datasets are PBMC datasets and the Oetjen dataset is a bone marrow dataset. Looking at the expression of marker genes and the location of the megakaryocyte progenitor cells in the UMAP embedding supports our claim that the cell types from Sun and Freytag should not match Oetjen in the hierarchy (Fig S2). Based on marker gene expression, the MK progenitor cells in the Oetjen dataset should be annotated as early erythrocytes and the MK progenitor cells in the Freytag and Sun dataset as platelets. These cell types of Freytag and Sun are very small (14 and 16 cells respectively), which makes it difficult to train the classifier and explains why these two are not matched.

Next, we align the query dataset to the latent space of the reference datasets using scArches and update the learned hierarchy with the new cell types (Fig 1E-F). For this step, only the trained model and reference latent space are needed. Again, almost all cell types (10 out of 12) are added to the correct node in the tree. Two mistakes are that the plasma cells and the megakaryocyte progenitors are added to the tree as new cell types. These cell types contain 21 and 18 cells respectively, which makes them difficult to match compared to the other cell types in the query dataset which contain more than 1000 cells on average.

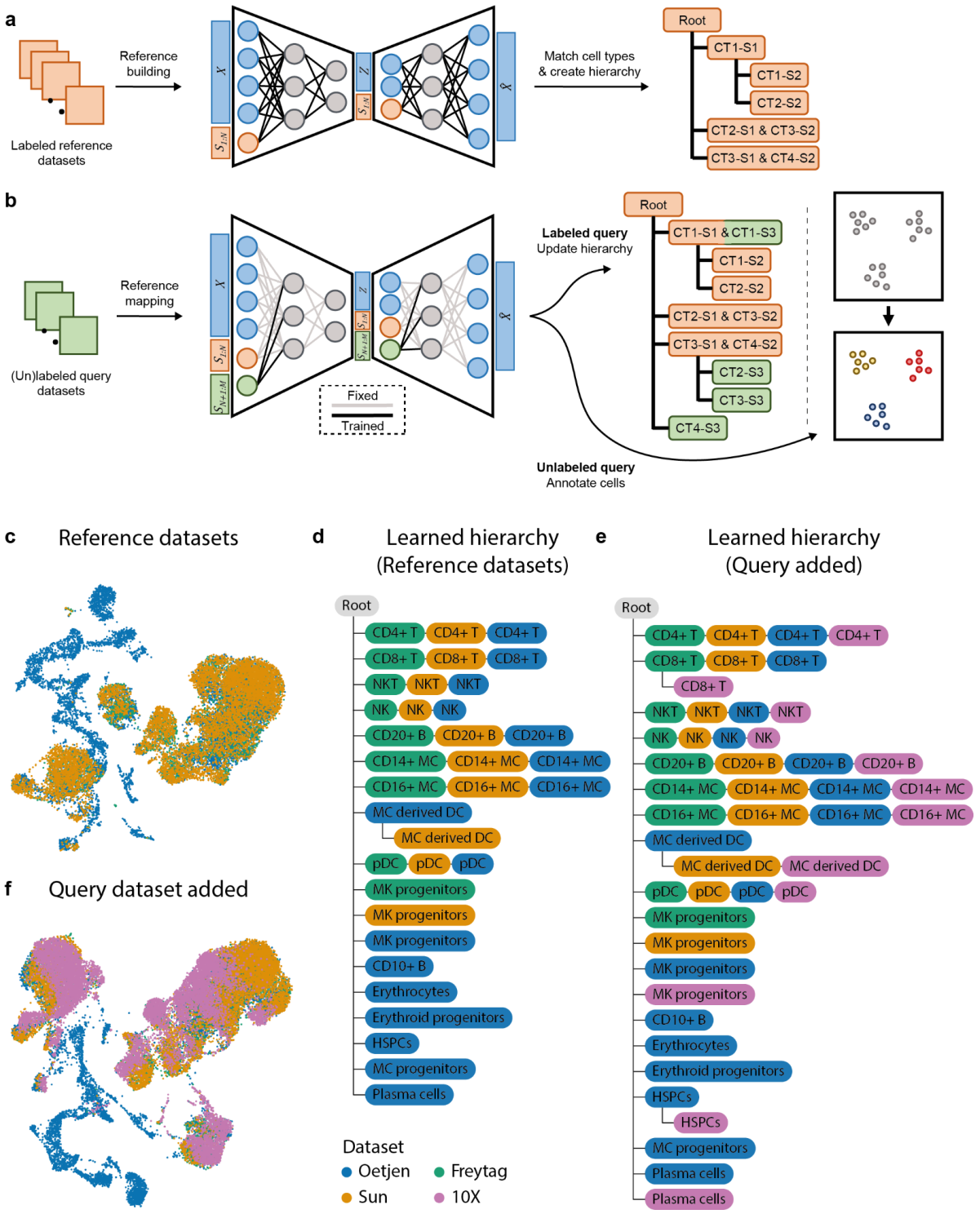


**Figure 1. A schematic version of treeArches and an example using PBMC and bone marrow datasets. (a)** Pre-training of a latent representation using labeled public reference datasets. After integration, a cell type hierarchy is created by matching the cell types of the different datasets. Here, for instance, cell types (CT) 1 and 2 from study (S) 2 are subtypes of CT1 from S1. **(b)** (Un)labeled query datasets can be added to the latent representation by applying architectural surgery. After integration, the cell type hierarchy is updated with labeled query datasets. Unlabeled query datasets can be annotated using the learned hierarchy. **(c)** UMAP embedding showing the integrated latent space of the three reference datasets. **(d)** Cell type hierarchy tree learned from the three reference datasets. **(e)** Updated hierarchy after the 10X dataset was added. **(f)** UMAP embedding showing the integrated latent space of the reference and query datasets.

**Increasing the resolution of the human lung cell atlas using treeArches**

The human lung cell atlas (HLCA) is a carefully constructed reference atlas for the human respiratory system [6]. Sikkema et al. aligned 14 datasets, harmonized the annotations, and constructed a cell type hierarchy consisting of 5 levels (Fig 2A, Fig S3). Furthermore, they used scArches to align multiple datasets to this reference atlas. Since the cell type hierarchy for the reference is well-defined, we can omit the reference building step and leverage treeArches to update the reference hierarchy using one of the query datasets (Meyer) [23]. Using scHPL, we matched the cell types of the Meyer dataset to the cell types from the reference (Fig S4). In the updated hierarchy, we see in general that most cell types from the query dataset match a cell type from the reference dataset as expected. For some parts of the hierarchy, we can even increase the resolution. Suppose we zoom in on the blood vessel branch in the tree, for instance. In that case, we see that the pulmonary and systemic endothelial vascular arterial cell types from the query both match 'EC arterial' from the reference (Fig 2B).

For some parts of the tree, e.g. the airway epithelium secretory cells, the matches are not what we would expect based on the names (Fig 2C). The secretory goblet cells from the query dataset match not only the goblet but also the club cells from the reference and the secretory club cells match the transitional club-AT2 cells. This cell type has only been recently discovered, which could explain why it is missing from the original Meyer annotations [24–26]. Based on the expression of marker genes, we can conclude that the match between the transitional club-AT2 and secretory club cells is a correct match (Fig 2D). The expression of the marker genes in the other cell types, however, is ambiguous and it is hard to determine what is the correct match.

Furthermore, we see that there are sixteen cell types from the query added to the root node of the tree as a new cell type (Fig S4). Of these cell types, most of them, e.g. chondrocytes, erythrocytes, Schwann cells, and B plasmablasts, are indeed not in the reference atlas. For some, such as some macrophage subtypes that are seen as new, it is a bit more difficult to determine whether they are new or whether they should match one of the macrophage subtypes in the tree. The 'Macro CHIT1' cells from the Meyer dataset, for instance, form a relatively big cell type of 1570 cells and are still seen as new. We visualized the expression of *CHIT1*, the gene this cell type was named after, and the marker genes that were used to annotate the cells in the reference data (Fig S5). This shows that the Macro CHIT1 cell type is the only cell type that expresses *CHIT1.* Furthermore, the marker gene profile of the other cell types does not correspond to the profile of the Macro CHIT1 cells, which indicates that this cell type was indeed rejected correctly.

Twelve out of 77 cell types, however, are missing from the tree, which means that it was impossible to match these cell types with a cell type from the reference. Due to many-to-many matches between the reference and query cell types, it is sometimes unclear where a cell type should be added to the tree. Especially, when the boundary between cell types forms a gradient, it can be quite subjective where to put the threshold. If this threshold is different in each dataset or if cells are wrongly annotated in general, this can cause impossible matching scenarios. Here, we notice that this mainly happens with some immune and stromal subtypes. The B cells and plasma cells from the reference and Meyer dataset, for instance, could not be matched automatically, which is caused by the plasma cells in the Meyer dataset that are partially misannotated (Fig S6).

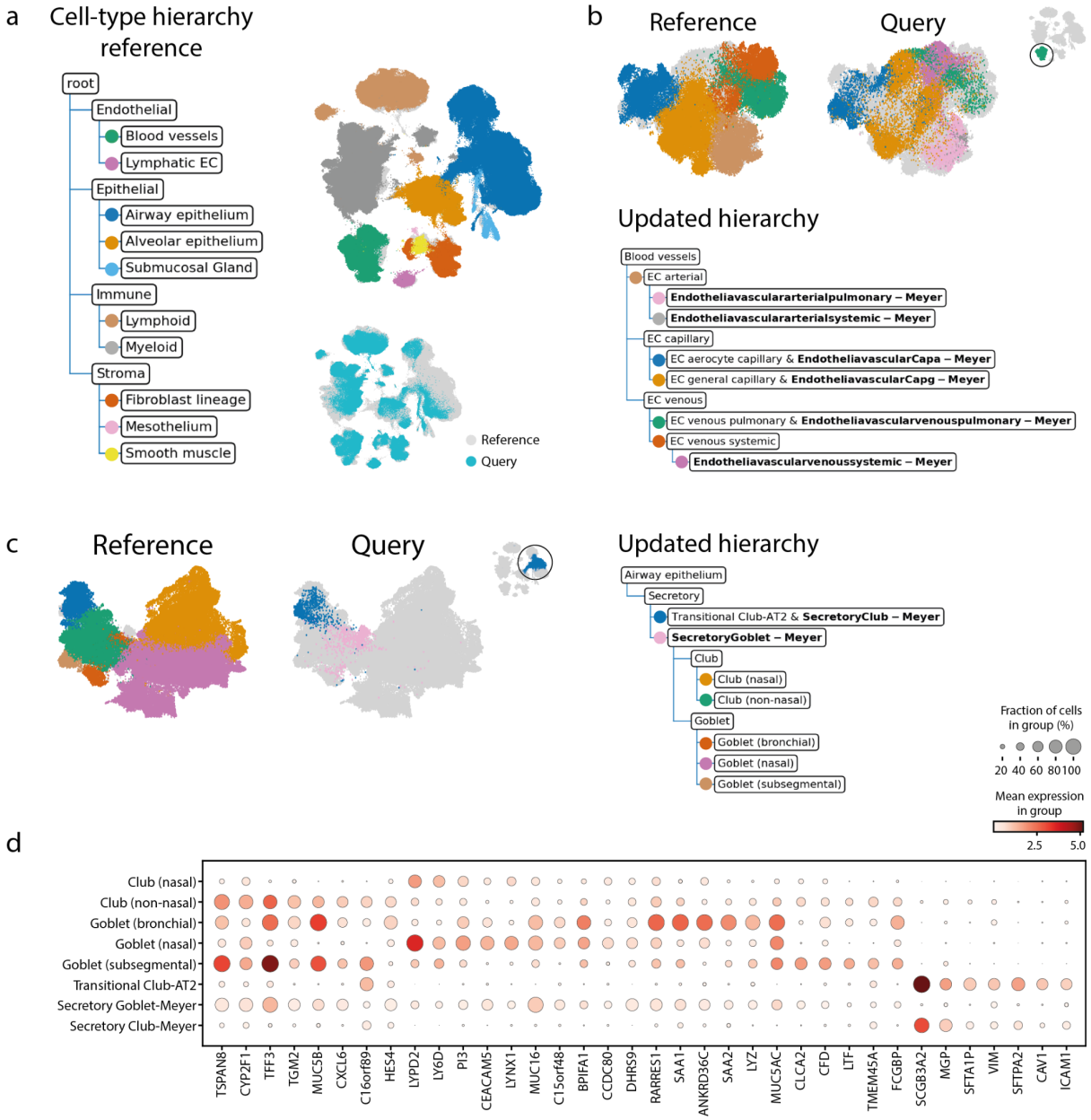


**Figure 2. Updated hierarchy when adding Meyer to the reference atlas. (a)** Cell type hierarchy corresponding to the reference atlas (only the first two levels are shown). Each node represents a cell type in the reference atlas instead of a cell type in a separate dataset of the reference atlas. The UMAP embedding shows the aligned reference and query dataset. The cells in the reference dataset are colored according to their level 2 annotation. **(b-c)** Updated hierarchy zoomed in on the blood vessels and airway epithelium secretory cells respectively. The UMAP embeddings are colored according to their finest resolution. **(d)** Expression of marker genes for club and goblet cells in the reference and query cell types.

### treeArches identifies unseen disease-associated cell types in the query data

Next, we show how we can use treeArches to detect previously unseen cell types in idiopathic pulmonary fibrosis (IPF) samples [27]. This dataset was mapped on the HLCA with scArches (Fig 3A-C). Ideally, we would use scHPL to update the hierarchy with the new cell types. A downside of the original annotations, however, is that the resolution is very low. Cells are, for instance, only annotated as endothelial cells. Therefore, we used scHPL to

predict the labels of the IPF data and compare those predictions to the original annotations (Fig 3D). In the predictions, we see some interesting differences between the IPF and healthy cells.

For the IPF cells, many macrophages and epithelial cells are rejected, while almost none for the healthy cells. Furthermore, most healthy Col1+ cells are predicted to be alveolar fibroblasts, while the diseased Col1+ are mainly SM-activated stress response cells. In all datasets, however, we notice confusion between the B cells and dendritic cells. Based on marker gene expression, the cells originally annotated as B cells and dendritic cells are more likely to be plasmablasts and B cells respectively (Fig S7). The cells originally annotated as dendritic cells also overlap in the UMAP with the lymphoid lineage mainly instead of the myeloid lineage (Fig 3A-B).

Next, we annotated the cells at a higher resolution (see **Methods**) and used these annotations to update the hierarchy (Fig S8). In the updated hierarchy, the healthy and IPF transitioning epithelial cells, a cell type not present in the reference atlas, are correctly added as a new cell type. As expected, we also see some differences in how the healthy and IPF cell types are added to the tree. IPF alveolar macrophage proliferating cells, for instance, are seen as new, while the healthy cells are a perfect match with the same cell type in the hierarchy. For other IPF macrophage cell types, however, this is not the case even though many cells were rejected previously. Comparing the new annotations with the previously obtained predictions and the matches in the hierarchy, we notice that there are still many macrophages rejected (Fig 3E). For most IPF cell types, however, only a subset of the cells is rejected. For instance, for the IPF monocyte-derived macrophages (Md-M), 486 cells are rejected and 750 are predicted to be Md-M (reference). Therefore, the two cell types are still matched. Comparing the two IPF 'subtypes' of Md-M, the top differentially expressed gene is *SPP1*. Monocytes and macrophages expressing *SPP1* are known to be a hallmark of IPF pathogenesis [28,29]. The rejected Md-M cells are the only group of cells expressing *SPP1* (Fig 3F). Combining the confusion matrices with the created hierarchy, these diseased subtypes are thus easily found either directly as the proliferating cells or by looking at the rejected cells of a matched cluster.

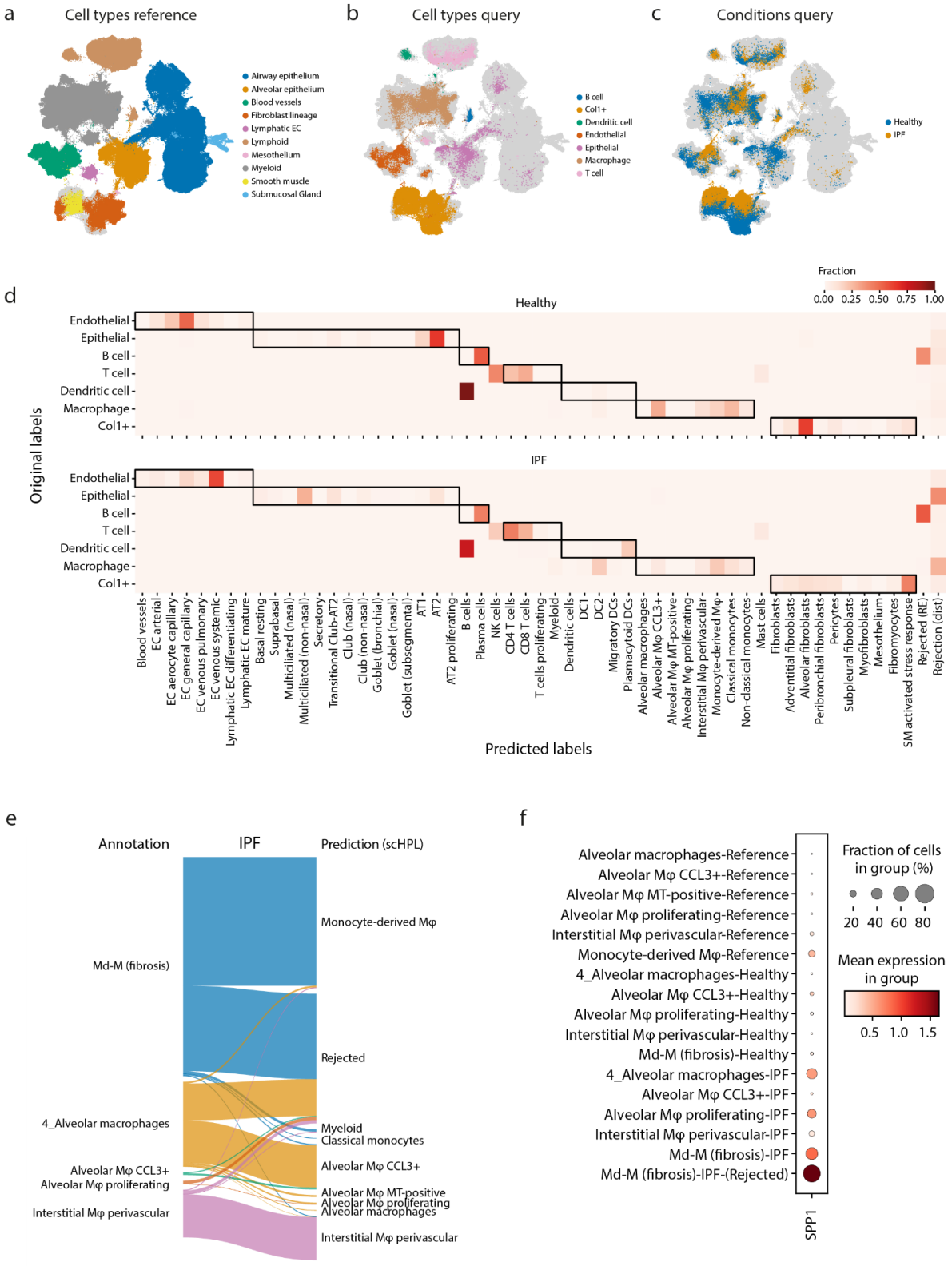


**Figure 3. Identifying diseased cells in IPF data. (a-c)** UMAPs show the HLCA and IPF datasets after alignment. The cells are colored according to their cell type or condition. **(d)** Heatmap showing the predicted labels by scHPL and original labels. The dark boundaries indicate the hierarchy of the reference tree. **(e)** Sankey diagram showing the new annotations and predictions for the macrophages for the IPF condition. **(f)** Expression of *SPP1* in the different cell types of the reference and query datasets.

**treeArches can correctly map cell types across species**

Next, we show how treeArches can be applied to map the relationship between cell types of different species. We construct a cell type hierarchy for the motor cortex of the brain using human, mouse, and marmoset data (Table S3) [30]. We integrate the reference datasets, mouse and marmoset, using scVI and construct the cell type hierarchy for the GABAergic neurons using scHPL (Fig 4A-B). Here, we only focus on this subset to make the results less cluttered. Almost all cell types are a perfect match, except for 'Meis2' and 'Sncg'. In the latent space, the Meis2 cell types from mouse and marmoset also show no overlap, and both cell types were defined using different marker genes (Fig S9A-B). Furthermore, Bakken et al. also didn't find a match between these two [30]. It is less clear why the Sncg cell types (559 and 960 cells in mouse and marmoset respectively) do not match. Even though the cell types are aligned in the UMAP embedding as expected and the marker genes correspond quite well, the cells are rejected based on distance (Fig S9C-D). This means that the cells are still too separated in the latent space. Next, we align the human dataset to the reference using architectural surgery and add the human cell type to the reference hierarchy (Fig 4B-C). Here, the constructed hierarchy looks like what we would expect based on the names of the cell types.

All previous results were obtained using the default parameters (number of neighbors = 50, see **Methods**), which turned out to be relatively robust (Fig S10). The main difference is whether a match is found between the Sncg cell types. When increasing the number of neighbors, this match is correctly found.

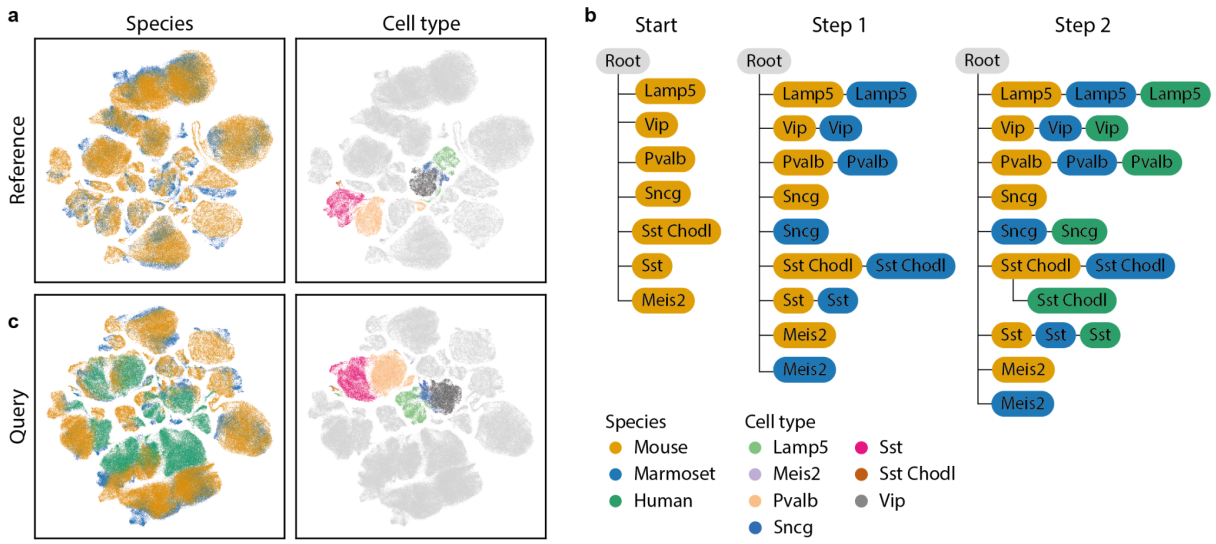


**Figure 4. Results motor cortex across species. (a)** UMAP embedding of the integrated reference datasets. **(b)** Learned hierarchy when combining mouse and marmoset (step 1) and after adding human (step 2). The color of each node represents the dataset(s) that the cell type originates from. **(c)** UMAP embedding after architectural surgery with the human dataset.

## Discussion

In this study, we present treeArches, a method to create and extend a reference atlas and the corresponding cell type hierarchies. treeArches builds on scArches, which allows users to easily map new query datasets to the latent space learned from the reference datasets using architectural surgery. Architectural surgery has the advantage that the reference datasets are not needed anymore for the mapping and that the latent space corresponding to the reference datasets does not change. This last point is especially important for scHPL, which then allows users to match the cell types of multiple labeled datasets to build a cell type hierarchy. If the latent space of all datasets would be altered when a new dataset is added, we would have to restart the construction of the tree completely.

We have shown three different situations where treeArches can be applied: building a reference atlas from scratch, extending an existing reference atlas to add new cell types or increase the resolution, or using an existing reference atlas to label cells in a new dataset. By using the HLCA data, we show an example of how treeArches can be used to extend a hierarchy or to label cells in a new dataset. The HLCA reference atlas consists of 16 datasets with a well-defined cell type hierarchy. We show that treeArches can be used to extend this hierarchy. For instance, by increasing the resolution of some branches of the tree, but also by adding new cell types. We could also detect diseased cell types in the IPF datasets.

Whether building or extending a reference atlas or labeling new cells, it is essential that we can detect new cell types, such as disease-specific cell types. To do so, it is important that during the mapping, the cell types are not forced to align; the biological variation should be preserved. Furthermore, during the classification, there should be a correctly working rejection option (i.e. cells are recognized to belong to a new unseen class). Here, we showed that this indeed works in all tested scenarios.

Due to the extended rejection options, however, it is difficult to match small cell types (less than 50 cells). We adapted the kNN classifier from scHPL such that the number of neighbors automatically decreases when there is a small cell type in the training data, but apparently this is not enough in all cases. The number of neighbors used is a trade-off between the ability to learn a representation for small cell types and a generalizable representation for the big cell types.

treeArches relies on the original annotations to extend the cell type hierarchy. This can be a problem in two different situations. If the annotations are missing or at a too low resolution, it is impossible to extend the atlas. This was the case with the original annotations of the IPF dataset. It can also be that there are annotations at a high resolution, but that these are (partially) incorrect. Especially when there is no clear boundary between cell types, but it looks more like a gradient, experts might disagree on where to put the threshold. Inconsistencies like this might result in a hierarchy that looks erroneous at first sight. In those cases, however, treeArches can still be more useful than expected. A cell type hierarchy that looks different than expected, is usually a sign that something is wrong with the original annotations. Certain parts of the dataset, e.g. the cell types that could not be added to the tree or caused confusion, can then be reannotated. Furthermore, the tree can still be adapted afterwards. Examples of this are the goblet and club cells in the HLCA and the

megakaryocyte progenitor cells in the PBMC datasets. The learned hierarchy is a good starting point. Based on marker gene expression or expert knowledge, cell types can also be added to the tree, removed from the tree, or rewired.

Our proposed method builds upon existing data integration methods. Thus, it naturally inherits both advantages and issues linked to existing models. As previously reported [1,2], the choice of the reference building algorithm and reference atlas itself can influence the quality of reference mapping. Therefore, in scenarios where the query dataset is strikingly different from the reference, the integrated query will still contain batch effects leading to inaccurate estimation of hierarchies in treeArches. This erroneous modeling results in weak label transfer results and thus identifies many overlapping cell types between query and reference as a new cell type only present in the query. We advise users to choose a comprehensive reference atlas and extensively benchmark and screen various data integration methods for an optimal reference representation [17],

In summary, we present treeArches, a method that can be used to combine multiple labeled datasets to create or extend a reference atlas and the corresponding cell type hierarchy. This way we provide users with an easy-to-use pipeline to map new datasets to a current reference atlas, match cell types across multiple labeled datasets, and consistently label cells in new datasets. With the increasing availability of reference atlases, we envision treeArches facilitating the usage of reference atlases allowing users to automatically analyze their datasets from label transfer to the automatic identification of novel cell states in the query data. In conclusion, treeArches will enable a data-driven path towards consensus-based cell type annotation of (human) tissues and will significantly speed up the building and annotation of atlases.

## Methods

### treeArches

treeArches is a framework built around scArches (version 0.5.3) [1] and scHPL (version 1.0.1) [14] to build and update a reference atlas and corresponding cell type hierarchy using multiple labeled datasets. The first step is to create a reference atlas by aligning multiple labeled datasets. This is done using any of the existing integration methods supported by scArches. scArches is a deep learning framework leveraging conditional variational autoencoders such as scVI, and scANVI [15,31,32]. Next, we use the constructed latent space to match the cell types of the reference datasets and build the corresponding cell type hierarchy using scHPL [14].
Afterwards, query datasets can be aligned to the reference atlas using scArches. Query datasets can be either labeled or unlabeled. In the case of a labeled dataset, we can extend the cell type hierarchy by matching the cell types of the query dataset to the reference. If the query dataset is unlabeled, we can use the hierarchy of the reference to transfer cell annotations to the query cells.

We enhanced the original version of scHPL by adding the option to use a *k*-nearest neighbor (kNN) classifier. The dimensionality of the latent space learned by scArches is relatively low (varying between 10 and 30 dimensions). We noticed that the linear SVM originally implemented doesn't perform well, since the cell types are not linearly separable anymore. Therefore, it is better to use scHPL with the kNN classifier in this case. In contrast to the

linear SVM, we train a multiclass classifier for every parent node instead of a binary classifier for every child node [14]. During training, we set the default number of neighbors to 50. However, when there are cell types in the dataset that consist of less than 50 cells, this is not ideal. Therefore, we added an extra option (*dynamic_neighbors*) to automatically decrease *k* to the size of the smallest cell type across the direct child nodes. Since the tree consists of multiple classifiers, it can thus be that they all use a different number of neighbors because of this option. For the kNN classifier itself, we implemented alternatives using either the FAISS library [33] or the scikit-learn library [34]. The FAISS implementation is faster than the scikit-learn library but is only available on Linux.

### Detecting new or diseased cell types

We have implemented three methods to detect new or diseased cell types: 1) a threshold on the posterior probability, 2) a threshold on the reconstruction error, and 3) a threshold on the distance between query and reference. The first two options were already implemented in the previous version of scHPL. The default threshold for the first option is 0.5. The threshold for the second rejection option is determined using a nested cross-validation loop. It is the median reconstruction error that gives a certain amount of false negatives on the test folds (default = 0.5%). The third option rejects cells whose distance to the predicted class is too big. The threshold for rejection is determined by calculating the neighbors for all cells in the training set, averaging the distance across the neighbors, and taking the 99th percentile.

## Datasets

### PBMC datasets

The dataset was obtained from the recent data integration benchmark [17]. The data contains bone marrow samples from Oetjen et al. [18] and also PBMC samples that were obtained from 10x Genomics https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_v3, Freytag et al. and Sun et al. [19,20], the original url and the preprocessing and annotation details can be found in Luecken et al. [17]. Marker genes specific to early erythrocytes and platelets were downloaded from Azimuth [3].

### Brain datasets

We used datasets from the primary motor cortex of three species: human, mouse, and marmoset [30]. We downloaded the datasets from the Cytosplore comparison viewer. In these datasets, genes were already matched based on one-to-one homologs. For the analysis, we only kept these one-to-one matches (15,860 genes in total). We selected 2,000 highly variable genes based on the reference datasets (mouse and marmoset) and used those counts as input for treeArches. The datasets are annotated at three different resolutions: Class, Subclass, and RNA_cluster. The class level contains three broad brain cell types: GABAergic neurons, glutamatergic neurons, and non-neuronal cells. At the subclass level, the cells are annotated at a higher resolution (5-10 subclasses per class). The RNA_cluster level contains the highest resolution. Here, we will use the subclass level to match the cell types. Marker genes used for visualization were chosen based on Supplementary Tables 5 and 6 from the original paper [30].

### Human Lung Cell Atlas

The human lung cell atlas (HLCA) is a carefully constructed reference atlas for the human respiratory system [6]. Sikkema et al. aligned 14 datasets, harmonized the annotations, and

built a cell type hierarchy consisting of 5 levels. When matching the cell types, we used the latent space generated in their original paper (downloaded from https://zenodo.org/record/6337966#.YqmGIidBx3g). When updating the hierarchy with the IPF data, we removed the cell types smaller than 10 cells. Marker genes were downloaded from the lung reference v2 from Azimuth [3,6].
We annotated the fibrosis-specific cell types in greater detail by subclustering the cell types of interest (macrophages, epithelial cells, myofibroblasts and identifying the subtypes by marker gene expression. We identified transitioning/basaloid epithelial cells by KRT5/KRT17 expression, inflammatory monocyte-derived macrophages by SPP1 expression, and myofibroblasts by the expression of CTHRC1.

### Code availability

treeArches is part of the scArches repository (https://github.com/theislab/scarches). The code for scHPL as a standalone package can be found here: https://github.com/lcmmichielsen/scHPL. All code to reproduce the results and figures can be found at the reproducibility GitHub: https://github.com/lcmmichielsen/treeArches-reproducibility

## Data availability

PBMC count data:
https://drive.google.com/uc?id=1Vh6RpYkusbGIZQC8GMFe3OKVDk5PWEpC
Brain count data: https://doi.org/10.5281/zenodo.6786357
PBMC + brain latent space: https://doi.org/10.5281/zenodo.6786357
HLCA latent space: https://zenodo.org/record/6337966#.YqmGIidBx3g

## Acknowledgments

ML acknowledge useful feedback from Malte Luecken regarding the experiments. We also appreciate Sergey Rebakov's contributions in merging the code within scArches code base.

### Author contributions

ML conceived the project with contributions from LM and AM. ML, LM, AM designed the experiments. LM performed the experiments. DS helped with the analysis of the IPF data. LS provided feedback essential to HLCA analysis. LM and ML wrote the manuscript. AM, MJTR, FJT supervised and approved the manuscript.

### Competing interest

F.J.T. consults for Immunai Inc., Singularity Bio B.V., CytoReason Ltd, and Omniscope Ltd, and has ownership interest in Dermagnostix GmbH and Cellarity.

## Funding

This research was supported by an NWO Gravitation project: BRAINSCAPES: A Roadmap from Neurogenetics to Neurobiology (NWO: 024.004.012) and by grant number 2019-002438 from the Chan Zuckerberg Foundation, by the European Union's Horizon 2020 research and innovation programme under grant agreement No 874656, by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI [ZT-I-PF-5-01] and by the

Bavarian Ministry of Science and the Arts in the framework of the Bavarian Research Association “ForInter” (Interaction of human brain cells).

## Supplementary Material

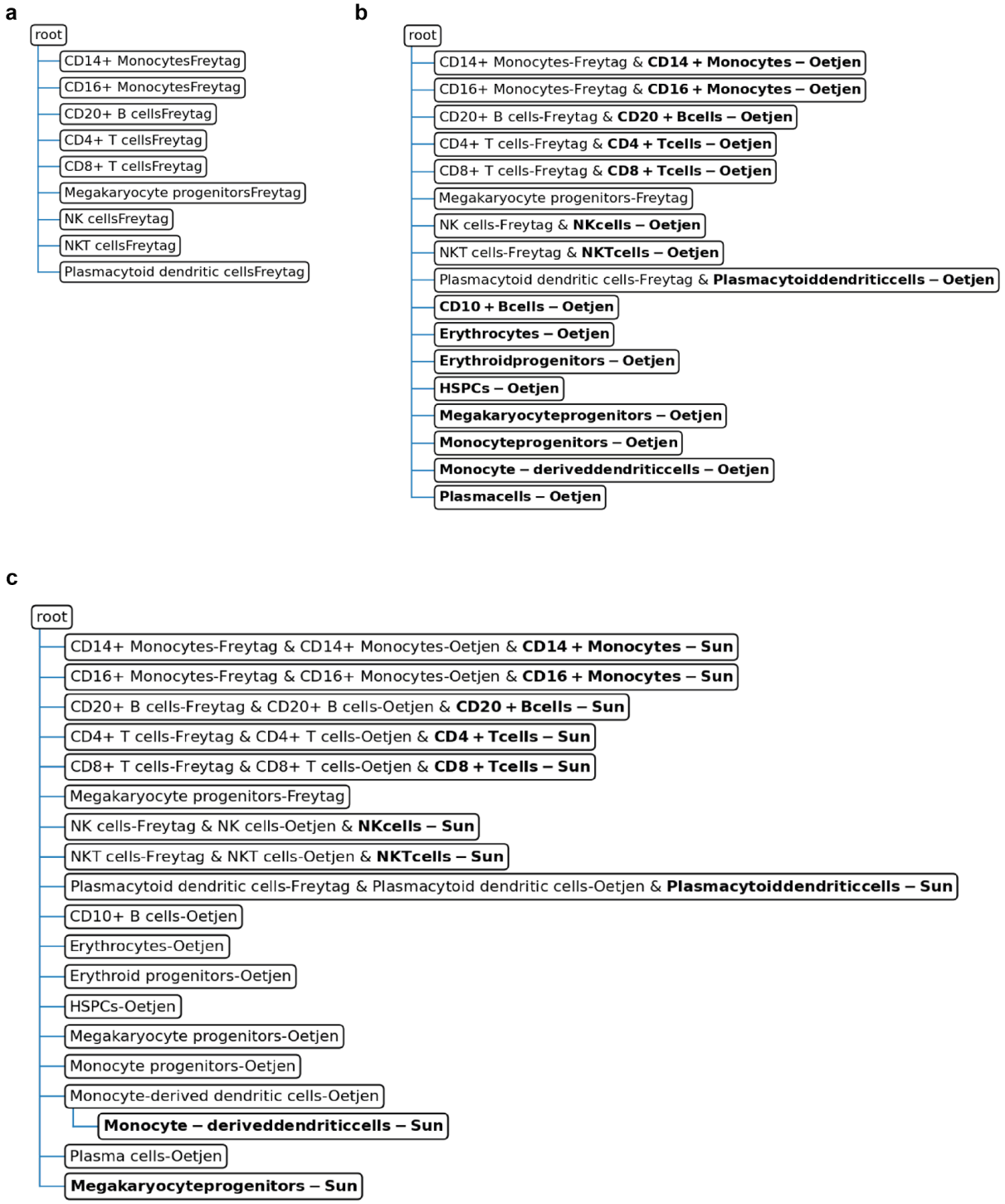


**Figure S1:** Intermediate step when creating the cell type hierarchy for the reference PBMC datasets. a) Starting tree, which is a flat tree containing only the cell types of the Freytag dataset, b) Oetjen dataset added, c) Sun dataset added

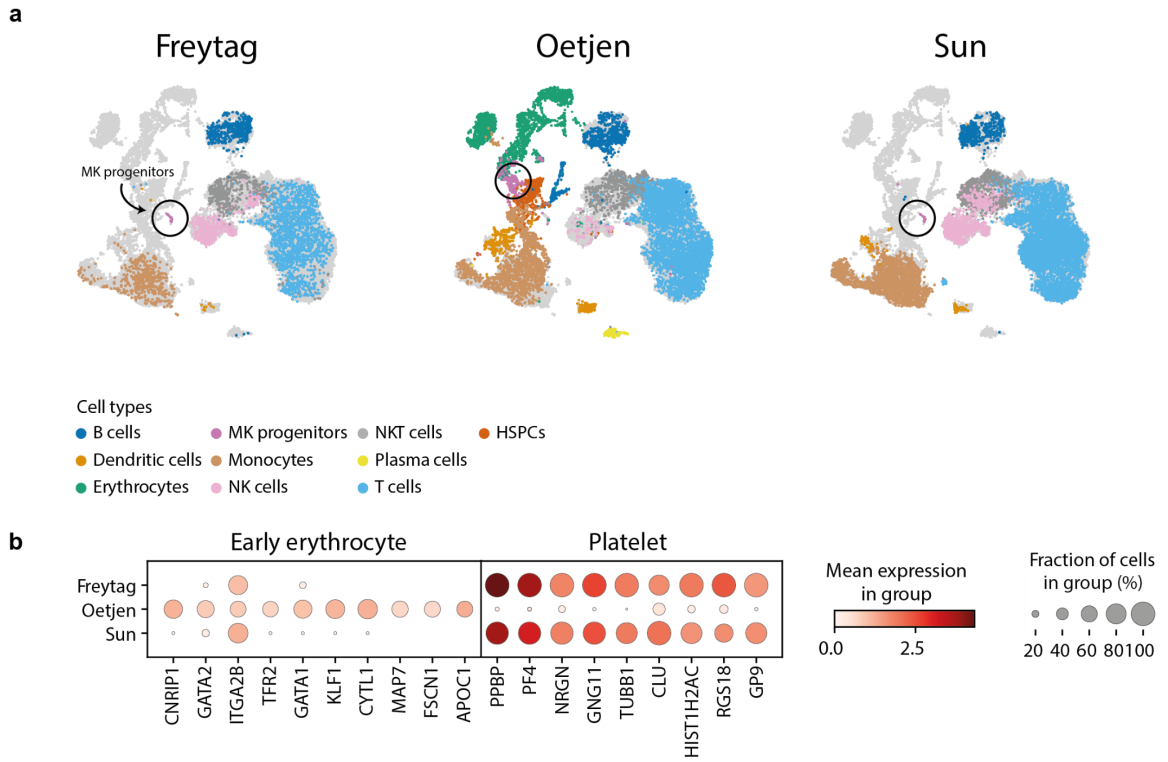


**Figure S2:** a) UMAP embedding showing the different cell types in the Freytag, Oetjen, and Sun dataset. Megakaryocyte (MK) progenitor cells of the Freytag and Sun dataset are at a different location compared to the Oetjen dataset. b) Marker gene expression for early erythrocytes and platelets in the three different datasets.

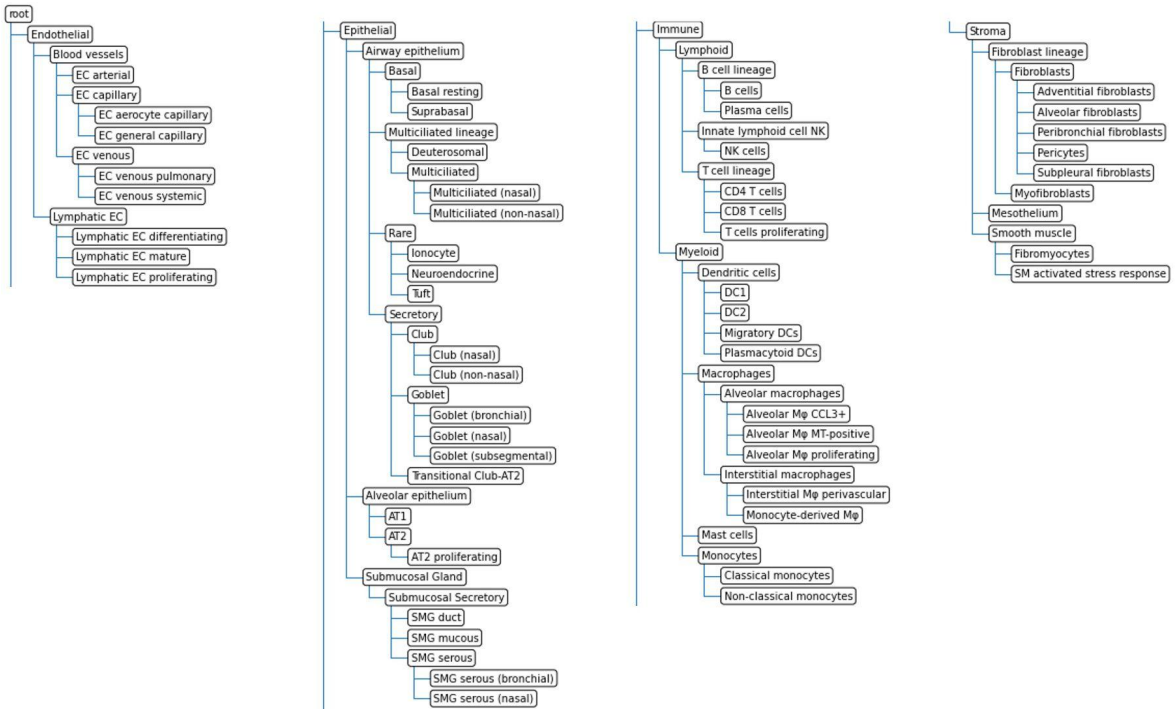


**Figure S3**: Cell type hierarchy constructed for the reference atlas [6].

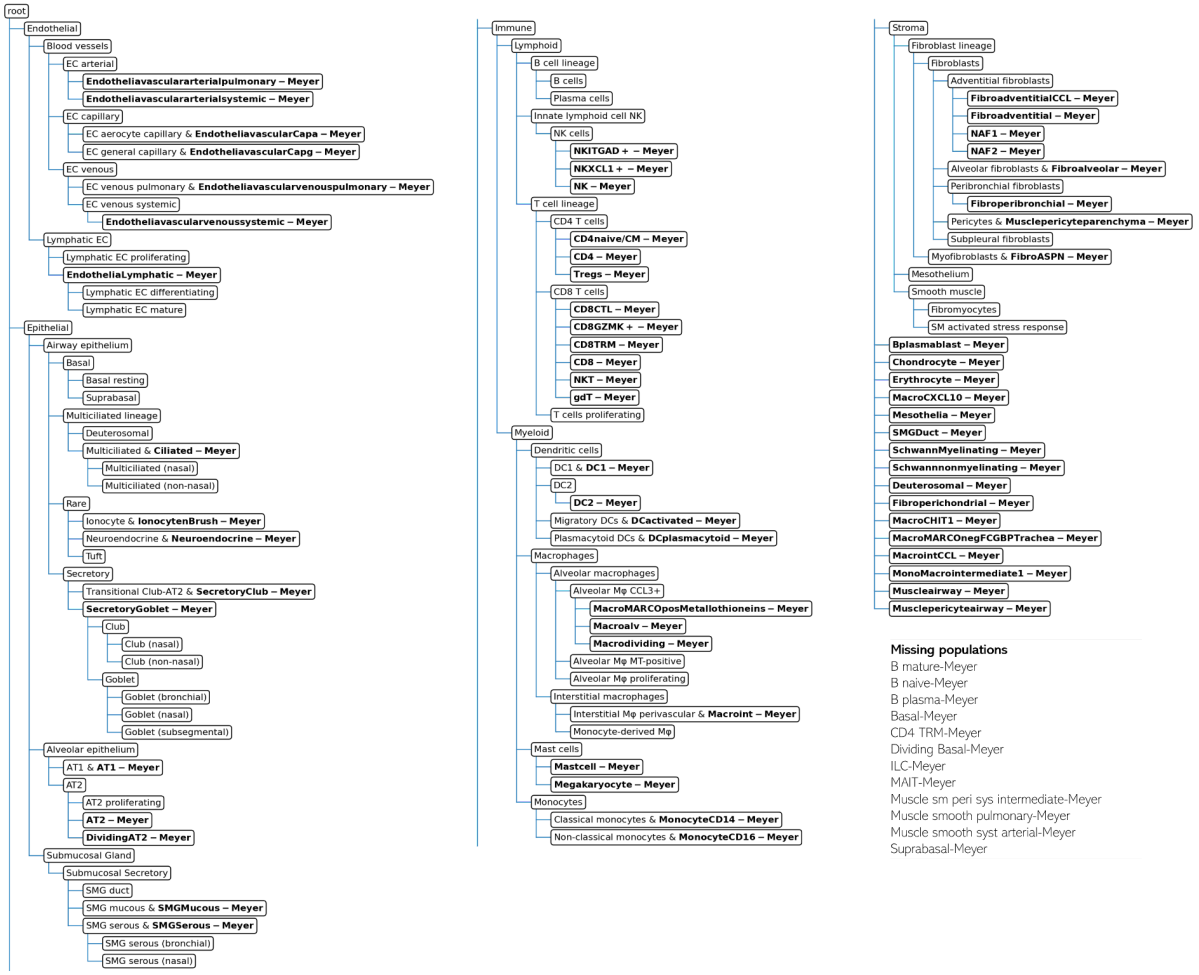


**Figure S4.** Updated cell type hierarchy learned by adding a query dataset (Meyer dataset) to the reference tree.

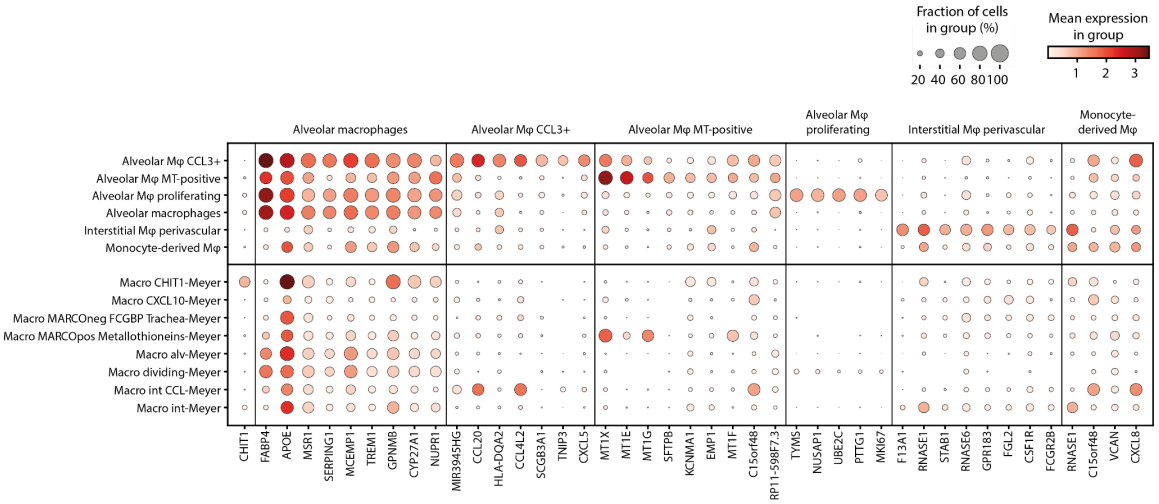


**Figure S5.** Marker gene expression for macrophage cell types in the reference datasets and Meyer dataset. The first column shows the expression of *CHIT1*, a gene used to annotate the Macro CHIT1 cells in the Meyer dataset. The rest of the genes are grouped according to the cell type in the reference atlas they were used as a marker for.

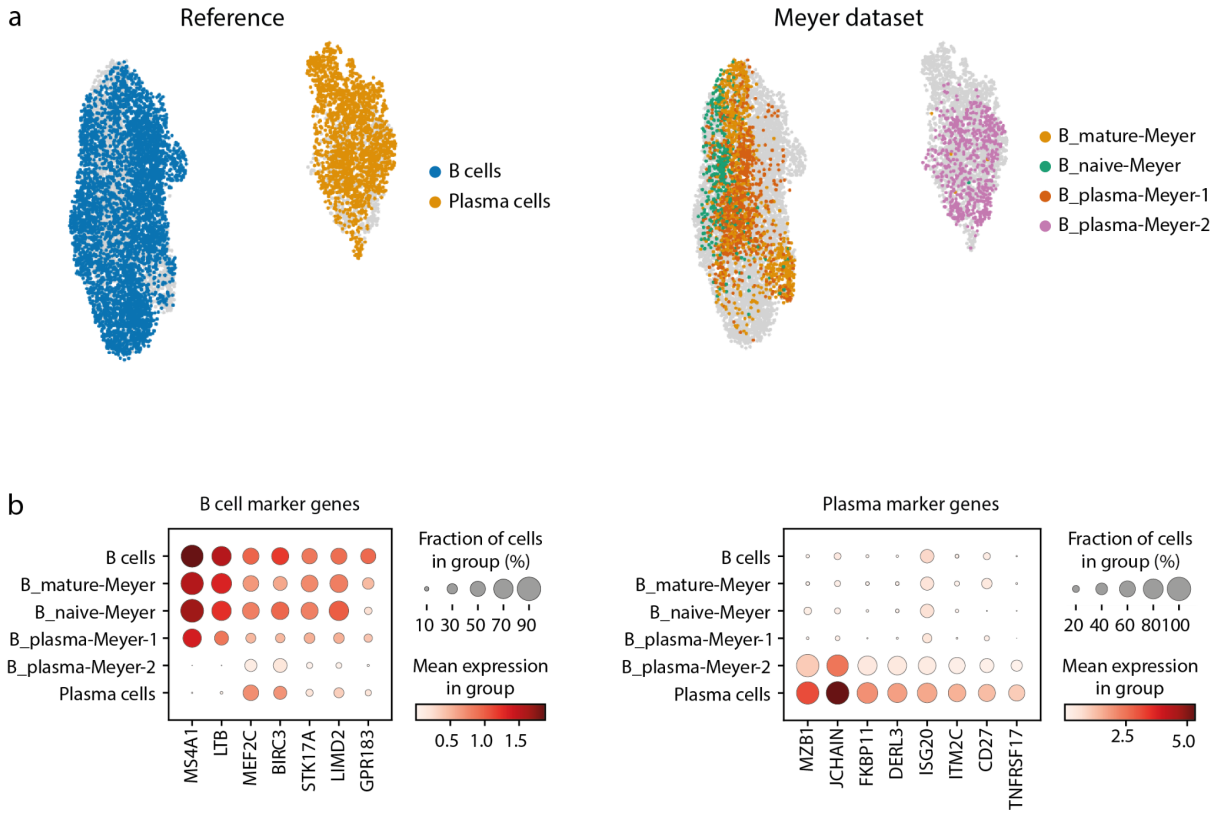


**Figure S6.** a) UMAPs showing the B cells and plasma cells in the reference and Meyer dataset. We split the plasma cells in the Meyer dataset into two groups. The first group overlaps with the reference B cells and the second group overlaps with the reference plasma cells. b) B cell and plasma cell marker gene expression in the reference and Meyer cell types. Plasma-1 from Meyer shows B cell marker gene expression, while Plasma-2 from Meyer shows plasma marker gene expression.

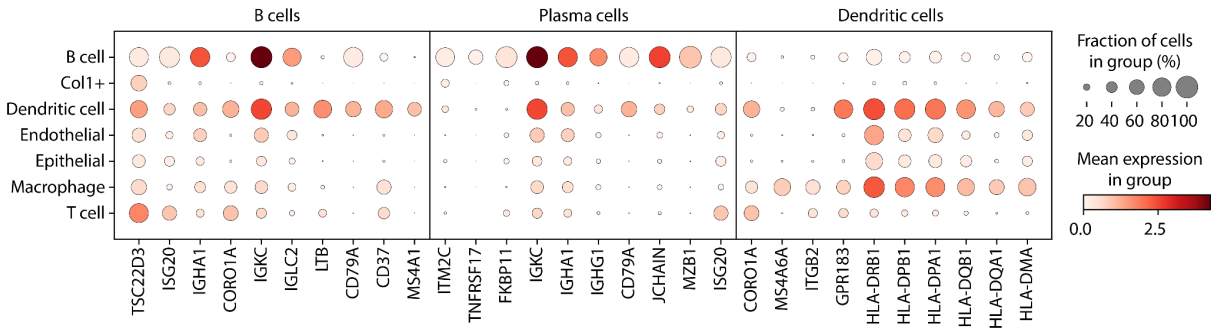


**Figure S7.** Expression of marker genes for B cells, plasma cell, and dendritic cells in the cell types in the IPF dataset.

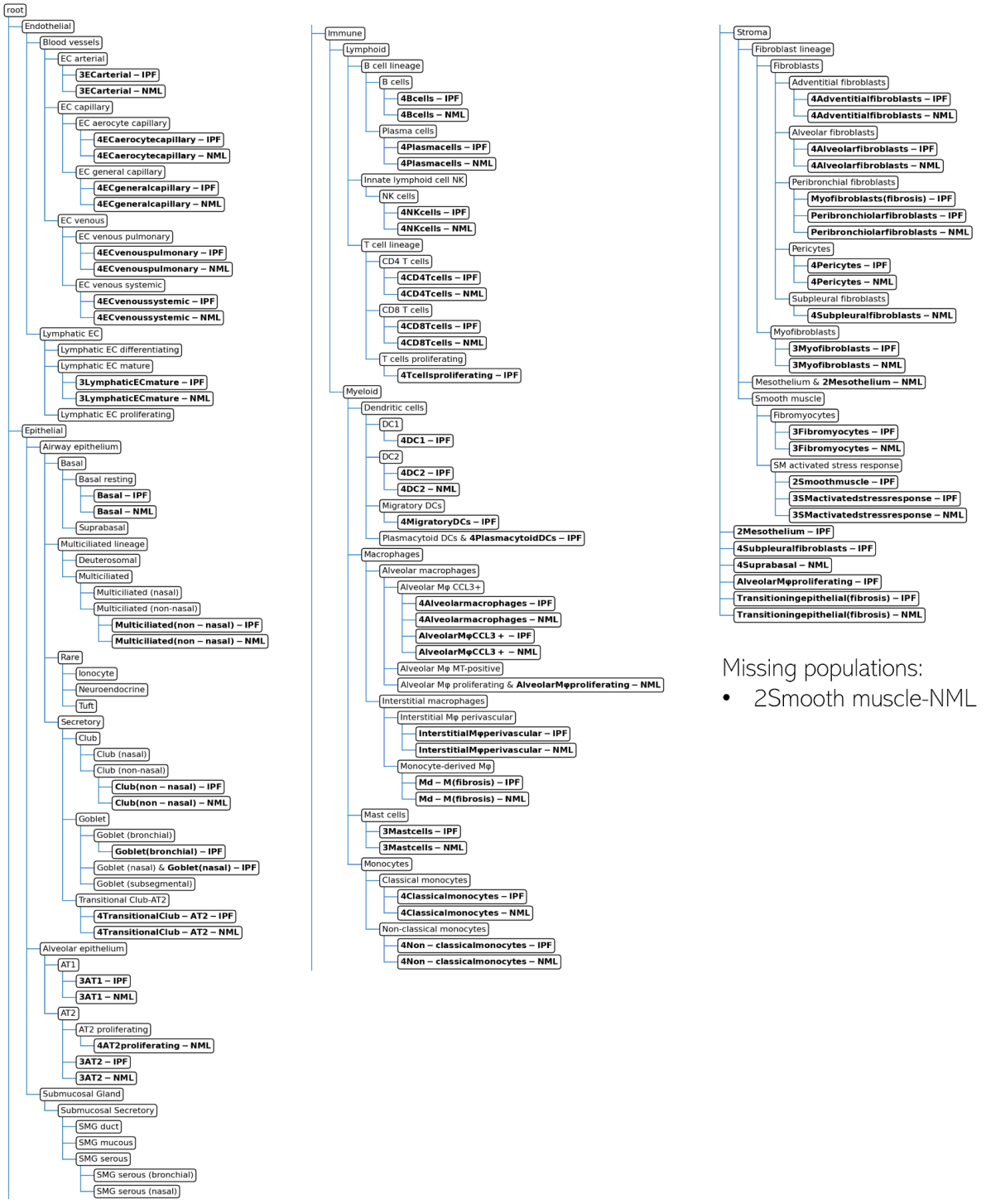


**Figure S8.** Updated hierarchy of the HLCA after adding the IPF dataset (IPF condition and normal condition).

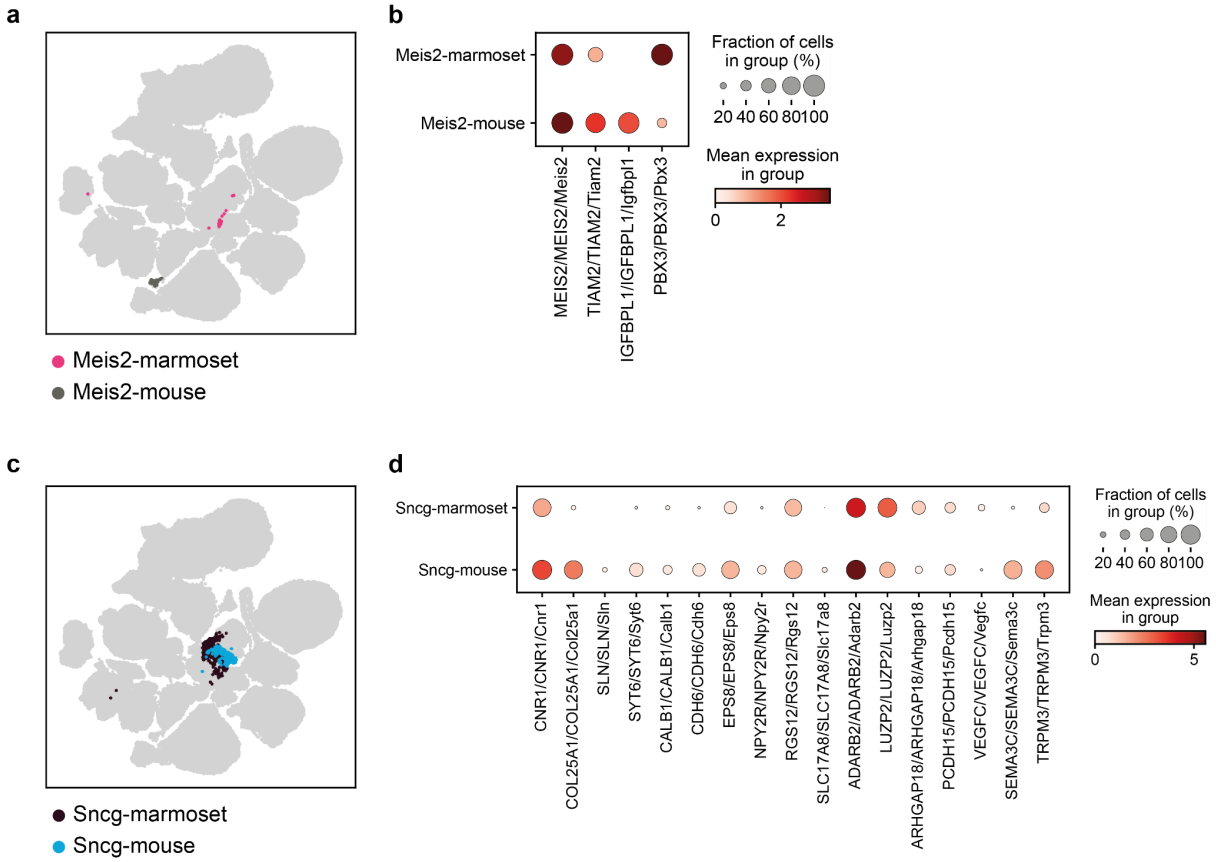


**Figure S9:** a) and c) UMAP embedding showing the integrated latent space of the reference datasets (mouse and human). The Meis2 and Sncg cell types are highlighted respectively. b) and d) Marker gene expression for the Meis2 and Sncg cell types respectively. The three gene names shown are the human/marmoset/mouse gene names.

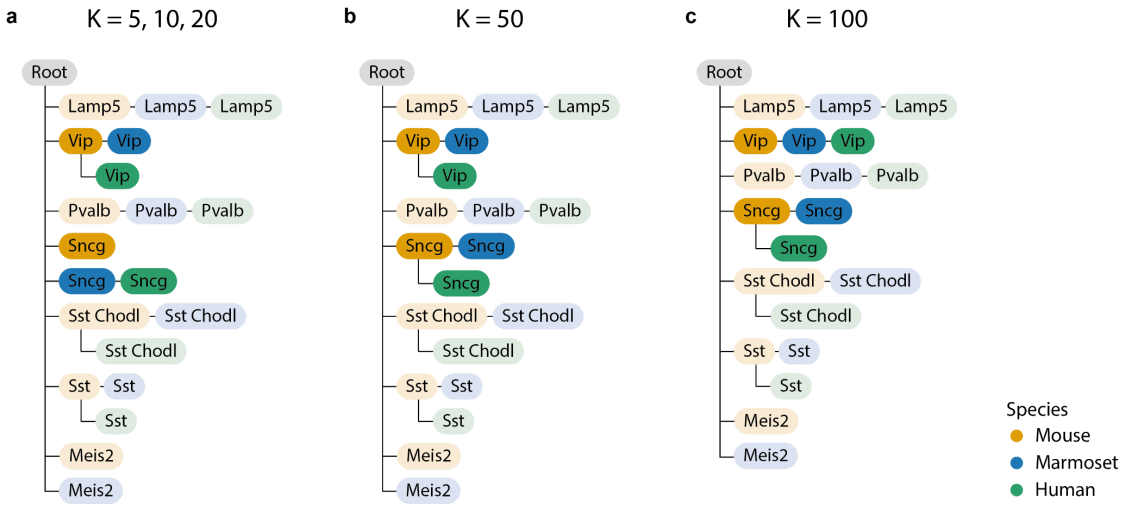


**Figure S10:** Influence of the number of neighbors (K) on the learned hierarchy. The nodes are colored according to the species they come from. The links between most nodes are robust and do not change when the number of neighbors varies. The differences between the trees are highlighted using brighter colors.

**Table S1:** Information on PBMC datasets used in this study

| Dataset (add reference) | Tissue | No. of Samples | No. of cells | No. of genes | No. of cell types | Protocol | Reference or query |
|---|---|---|---|---|---|---|---|
| Oetjen [18] | Bone marrow | 3 | 9581 | 12303 | 16 | 10X v2 | Reference |
| Sun [19] | PBMC | 4 | 8829 | 12303 | 10 | 10X | Reference |
| Freytag [20] | PBMC | 1 | 3347 | 12303 | 9 | 10X v2 | Reference |
| 10X [21] | PBMC | 1 | 10727 | 12303 | 12 | 10X v3 | Query |

**Table S2:** Overview of cell types in the PBMC datasets

| cell type | Oetjen | Sun | Freytag | 10X |
|---|---|---|---|---|
| CD4+ T | 2524 | 4312 | 1238 | 2937 |
| CD8+ T | 985 | 578 | 270 | 350 |
| NKT | 608 | 649 | 432 | 1056 |
| NK | 89 | 973 | 476 | 756 |
| CD20+ B | 491 | 409 | 427 | 1546 |
| CD10+ B | 207 | | | |
| CD14+ MC | 997 | 1501 | 452 | 3388 |
| CD16+ MC | 165 | 271 | 25 | 364 |
| MC derived DC | 214 | 82 | | 182 |
| pDC | 133 | 40 | 11 | 81 |
| MK progenitor | 219 | 14 | 16 | 21 |
| Erythrocytes | 1502 | | | |
| Erythroid prog. | 463 | | | |
| HSPC | 445 | | | 28 |
| MC progenitor | 428 | | | |
| Plasma cell | 111 | | | 18 |

**Table S3:** Information on brain datasets used in this study

| Species | No. of cells | No. of genes | No. of cell types (class/subclass/ RNA_cluster) | Protocol |
|---|---|---|---|---|
| Mouse | 159,739 | 27,439 | 3/23/116 | 10X v3 |
| Marmoset | 69,279 | 27,466 | 3/22/94 | 10X v3 |
| Human | 76,621 | 32,991 | 3/20/127 | 10X v3 |

**References**

1. Lotfollahi M, Naghipourfar M, Luecken MD, Khajavi M, Büttner M, Wagenstetter M, et al. Mapping single-cell data to reference atlases by transfer learning. Nat Biotechnol [Internet]. 2022;40:121–30. Available from: http://dx.doi.org/10.1038/s41587-021-01001-7

2. Lotfollahi M, Rybakov S, Hrovatin K, Hediyeh-zadeh S, Talavera-López C, Misharin AV, et al. Biologically informed deep learning to infer gene program activity in single cells [Internet]. bioRxiv. 2022 [cited 2022 May 4]. p. 2022.02.05.479217. Available from: https://www.biorxiv.org/content/10.1101/2022.02.05.479217v1

3. Hao Y, Hao S, Andersen-Nissen E, Mauck WM, Zheng S, Butler A, et al. Integrated analysis of multimodal single-cell data. Cell [Internet]. Elsevier; 2021;0. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0092867421005833

4. Kang JB, Nathan A, Weinand K, Zhang F, Millard N, Rumker L, et al. Efficient and precise single-cell reference atlas mapping with Symphony. Nat Commun [Internet]. 2021;12:5890. Available from: http://dx.doi.org/10.1038/s41467-021-25957-x

5. Suo C, Dann E, Goh I, Jardine L, Kleshchevnikov V, Park J-E, et al. Mapping the developing human immune system across organs. Science [Internet]. 2022;eabo0510. Available from: http://dx.doi.org/10.1126/science.abo0510

6. Sikkema L, Strobl D, Zappia L, Madissoon E, Markov NS, Zaragosi L, et al. An integrated cell atlas of the human lung in health and disease [Internet]. bioRxiv. 2022 [cited 2022 May 10]. p. 2022.03.10.483747. Available from: https://www.biorxiv.org/content/10.1101/2022.03.10.483747v1

7. Tabula Sapiens Consortium*, Jones RC, Karkanias J, Krasnow MA, Pisco AO, Quake SR, et al. The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. Science [Internet]. 2022;376:eabl4896. Available from: http://dx.doi.org/10.1126/science.abl4896

8. Osorio D, McGrail DJ, Sahni N, Stephen Yi S. Drug combination prioritization for cancer treatment using single-cell RNA-seq based transfer learning [Internet]. bioRxiv. 2022 [cited 2022 May 17]. p. 2022.04.06.487357. Available from: https://www.biorxiv.org/content/10.1101/2022.04.06.487357v1

9. Swamy VS, Fufa TD, Hufnagel RB, McGaughey DM. Building the mega single-cell transcriptome ocular meta-atlas. Gigascience [Internet]. 2021;10. Available from: http://dx.doi.org/10.1093/gigascience/giab061

10. Bharat A, Querrey M, Markov NS, Kim S, Kurihara C, Garza-Castillon R, et al. Lung transplantation for patients with severe COVID-19. Sci Transl Med [Internet]. 2020;12. Available from: http://dx.doi.org/10.1126/scitranslmed.abe4282

11. Wang M, Zadeh S, Pizzolla A, Thia K, Gyorki DE, McArthur GA, et al. Characterization of the treatment-naive immune microenvironment in melanoma with BRAF mutation. J Immunother Cancer [Internet]. 2022;10. Available from: http://dx.doi.org/10.1136/jitc-2021-004095

12. Brbić M, Zitnik M, Wang S, Pisco AO, Altman RB, Darmanis S, et al. MARS: discovering novel cell types across heterogeneous single-cell experiments. Nat Methods [Internet]. 2020;17:1200–6. Available from: http://dx.doi.org/10.1038/s41592-020-00979-3

13. Jupp S, Burdett T, Leroy C, Parkinson HE. A new Ontology Lookup Service at

EMBL-EBI. ceur-ws.org [Internet]. 2015; Available from: http://ceur-ws.org/Vol-1546/paper_29.pdf

14. Michielsen L, Reinders MJT, Mahfouz A. Hierarchical progressive learning of cell identities in single-cell data. Nat Commun [Internet]. Springer Science and Business Media LLC; 2021;12:1–12. Available from: https://doi.org/10.1038/s41467-021-23196-8

15. Gayoso A, Lopez R, Xing G, Boyeau P, Valiollah Pour Amiri V, Hong J, et al. A Python library for probabilistic analysis of single-cell omics data. Nat Biotechnol [Internet]. 2022;40:163–6. Available from: http://dx.doi.org/10.1038/s41587-021-01206-w

16. Lotfollahi M, Wolf FA, Theis FJ. scGen predicts single-cell perturbation responses. Nat Methods [Internet]. 8/2019 [cited 2020 May 4];16:715–21. Available from: http://www.nature.com/articles/s41592-019-0494-8

17. Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Mueller MF, et al. Benchmarking atlas-level data integration in single-cell genomics [Internet]. Available from: http://dx.doi.org/10.1101/2020.05.22.111161

18. Oetjen KA, Lindblad KE, Goswami M, Gui G, Dagur PK, Lai C, et al. Human bone marrow assessment by single-cell RNA sequencing, mass cytometry, and flow cytometry. JCI Insight [Internet]. 2018;3. Available from: http://dx.doi.org/10.1172/jci.insight.124928

19. Sun Z, Chen L, Xin H, Jiang Y, Huang Q, Cillo AR, et al. A Bayesian mixture model for clustering droplet-based single-cell transcriptomic data from population studies. Nat Commun [Internet]. 2019;10:1649. Available from: http://dx.doi.org/10.1038/s41467-019-09639-3

20. Freytag S, Tian L, Lönnstedt I, Ng M, Bahlo M. Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data. F1000Res [Internet]. 2018;7:1297. Available from: http://dx.doi.org/10.12688/f1000research.15809.2

21. Genomics 10x. 10x Datasets Single Cell Gene Expression [Internet]. 2018. Available from: https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_v3

22. Gayoso A, Lopez R, Xing G, Boyeau P, Wu K, Jayasuriya M, et al. scvi-tools: a library for deep probabilistic analysis of single-cell omics data [Internet]. bioRxiv. 2021 [cited 2022 Jan 14]. p. 2021.04.28.441833. Available from: https://www.biorxiv.org/content/10.1101/2021.04.28.441833v1

23. Madissoon E, Oliver AJ, Kleshchevnikov V, Wilbrey-Clark A, Polanski K, Orsi AR, et al. A spatial multi-omics atlas of the human lung reveals a novel immune cell survival niche [Internet]. bioRxiv. 2021 [cited 2022 May 10]. p. 2021.11.26.470108. Available from: https://www.biorxiv.org/content/10.1101/2021.11.26.470108v1

24. Basil MC, Cardenas-Diaz FL, Kathiriya JJ, Morley MP, Carl J, Brumwell AN, et al. Human distal airways contain a multipotent secretory cell that can regenerate alveoli. Nature [Internet]. 2022;604:120–6. Available from: http://dx.doi.org/10.1038/s41586-022-04552-0

25. Kadur Lakshminarasimha Murthy P, Sontake V, Tata A, Kobayashi Y, Macadlo L, Okuda K, et al. Human distal lung maps and lineage hierarchies reveal a bipotent progenitor. Nature [Internet]. 2022;604:111–9. Available from: http://dx.doi.org/10.1038/s41586-022-04541-3

26. Rustam S, Hu Y, Mahjour SB, Randell SH, Rendeiro AF, Ravichandran H, et al. A Unique Cellular Organization of Human Distal Airways and Its Disarray in Chronic Obstructive

Pulmonary Disease [Internet]. bioRxiv. 2022 [cited 2022 Jul 7]. p. 2022.03.16.484543. Available from: https://www.biorxiv.org/content/10.1101/2022.03.16.484543v3

27. Tsukui T, Sun K-H, Wetter JB, Wilson-Kanamori JR, Hazelwood LA, Henderson NC, et al. Collagen-producing lung cell atlas identifies multiple subsets with distinct localization and relevance to fibrosis. Nat Commun [Internet]. 2020;11:1920. Available from: http://dx.doi.org/10.1038/s41467-020-15647-5

28. Morse C, Tabib T, Sembrat J, Buschur KL, Bittar HT, Valenzi E, et al. Proliferating SPP1/MERTK-expressing macrophages in idiopathic pulmonary fibrosis. Eur Respir J [Internet]. 2019;54. Available from: http://dx.doi.org/10.1183/13993003.02441-2018

29. Karman J, Wang J, Bodea C, Cao S, Levesque MC. Lung gene expression and single cell analyses reveal two subsets of idiopathic pulmonary fibrosis (IPF) patients associated with different pathogenic mechanisms. PLoS One [Internet]. 2021;16:e0248889. Available from: http://dx.doi.org/10.1371/journal.pone.0248889

30. Bakken TE, Jorstad NL, Hu Q, Lake BB, Tian W, Kalmbach BE, et al. Comparative cellular analysis of motor cortex in human, marmoset and mouse. Nature [Internet]. Nature Publishing Group; 2021;598:111–9. Available from: https://www.nature.com/articles/s41586-021-03465-8

31. Sohn K, Lee H, Yan X. Learning Structured Output Representation using Deep Conditional Generative Models. In: Cortes C, Lawrence N, Lee D, Sugiyama M, Garnett R, editors. Advances in Neural Information Processing Systems [Internet]. Curran Associates, Inc.; 2015. Available from: https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf

32. Lotfollahi M, Naghipourfar M, Theis FJ, Wolf FA. Conditional out-of-distribution generation for unpaired data using transfer VAE. Bioinformatics [Internet]. 2020;36:i610–7. Available from: https://doi.org/10.1093/bioinformatics/btaa800

33. Johnson J, Douze M, Jégou H. Billion-Scale Similarity Search with GPUs. IEEE Transactions on Big Data [Internet]. 2021;7:535–47. Available from: http://dx.doi.org/10.1109/TBDATA.2019.2921572

34. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python [Internet]. 2011 p. 2825–30. Available from: http://scikit-learn.sourceforge.net.