Supplementary Note 1

When matching the cell populations from two datasets, we distinguish five options: simple, multiple columns, multiple rows, complex, and impossible. When describing the different scenarios within these options, we sometimes make a distinction between leaf nodes and internal nodes. Here, it is important to remember that only *T1* can have internal nodes since this is the tree that is updated. *T2* is always a flat classification tree, so only consists of the root node and leaf nodes.

Simple

In this scenario, we find a unique match between a cell population, P_i, from dataset 1 and a cell population, P_j, from dataset 2. As as consequence, $X_{j,i}$ will be 1 or 2 and the rest of row *j* and column *i* in *X* are zero. Within this scenario, there are three different options:

1. Both cell populations are leaf or internal nodes. This indicates a perfect match. The tree is not updated, but the labels of P_j are renamed to P_i (Figure S4A). This is the same scenario as the 'perfect match' scenario described in the main text.
2. P_i is a leaf or internal node, but P_j is the root node of *T2*. This indicates that P_i is missing in dataset 2. The node, however, is already in the tree, so it is not updated (Figure S4B).
3. P_i is the root of *T1*, but the P_j is a leaf node. This indicates that P_j is missing in dataset 1. The cell population is thus also not in the tree yet, so we will add it as a child to the root (Figure S4C). This is the same scenario as the 'new population' scenario described in the main text.

Multiple rows

In this scenario, a cell population, P_i, from dataset 1 matches multiple populations from dataset 2. In *X* there will be multiple non-zero values in column *i.* Here, we distinguish two different scenarios:

1. P_i matches only cell populations from dataset 2 that are leaf node. We consider the cell populations from dataset 2 subpopulations of P_i, so we add them as descendants to P_i (Figure S5A). This is the same scenario as the 'splitting nodes' scenario described in the main text.
2. The root node of *T2* is also involved. We simple ignore this node and for the rest do the same as above (Figure S5B-C).

Multiple columns

This scenario is quite similar to the multiple rows scenario. Here, however, multiples populations from dataset 1 match one cell population, P_j, of dataset 2. In *X* there will be multiple non-zero values in row *j.* This scenario is a little more complex since the populations from dataset 1 does not have to be leaf nodes or the root node, but there can also be internal nodes in this tree. Here, we distinguish three different scenarios:

1. The root node of *T1* and *T2* are not involved, so multiple cell populations, which can be leaf or internal nodes, from dataset 1 match P_j. We consider the cell populations from dataset 1 subpopulations of P_j, so we need to add P_j as a parent node to these cell populations. (Figure S6A). This is same scenario as the 'merging nodes' scenario described in the main text. It could be, however, that this node already exists in this tree. (Figure S6B). If this is the case, we have a perfect match between a node from tree 1 and tree 2, so we do not have to update the tree, but we only have to update the labels of P_j.
2. Besides leaf or internal nodes, the root of *T1* is involved. This indicates that P_j is 'bigger' than the cell populations from dataset 1 as part of it is unlabeled. Therefore, we add P_j as a

descendant to the root of *T1*. Next, we rewire the involved cell populations from dataset 1 such that they become descendants of P_j (Figure S6C).

3. The root node of *T2* is involved. This indicates that multiple cell populations from dataset 1 are missing in dataset 2. These nodes, however, are already in the tree, so the tree can remain the same (Figure S6D).

Complex

The scenarios described above were all relatively easy. A cell population from one dataset matches either one or multiple cell populations from another. It could also happen, however, that multiple cell populations from dataset 1 match multiple cell populations from dataset 2 (Figure S7). As a consequence, there will a certain place $X_{j,l}$ which is either 1 or 2 and there are two or more non-zero values in the corresponding row *j* and column *i.* Here, we distinguish three different scenarios:

1. The root node of *T1* is involved. We just assume that the boundary should be adjusted and this is automatically done, so we remove this `1' from the table (Figure S7A). If the situation is still complex after the one is removed, we continue to scenario 2 or 3. If not, we treat it as a multiple rows problem as explained above.
2. The root node of *T2* is involved. Again, we just assume that the boundary should be adjusted, so we remove this `1' from the table (Figure S7B). If the situation is still complex after the one is removed, we continue to scenario 3. If not, we treat it as a multiple columns problem as explained above.
3. Multiple leaf/internal nodes of dataset 1 are involved and multiple leaf nodes of dataset 2. We can only solve this if the 'complex' cell population, P_i, of dataset 1 is not a leaf node. Otherwise we are dealing with an impossible scenario which is described below. If the complex node is an internal node, we attach the involved cell populations of dataset 2 as descendants to the complex node (splitting scenario) and attach the involved cell populations of dataset 1, except for P_i, to P_j (Figure S7C).

Impossible

Sometimes, it could be impossible to match the labels from two datasets. Something could have gone wrong during the clustering, e.g. a population 1 and 2 from dataset 1 match population A from dataset 2, but population 2 also matches population C from dataset 2 (Figure S8A). Here, population A and C should be merged into population 2, but population A should also be split into population 1 and 2. Population 2, however, cannot be added to the tree twice.

It could also be that dataset 2 contains labels at a different resolution, e.g. that population B is a subpopulation of population A (Figure S8B). This is not what we assumed and thus impossible to match.

Both scenarios occur when a leaf node from dataset 1 is at a crossing of multiple rows and multiple columns (i.e. a complex situation). An extra difficulty is that there are thus multiple situations that could explain this. All of these situation are not what we desired and thus we call it impossible and do nothing.

Supplementary Figures

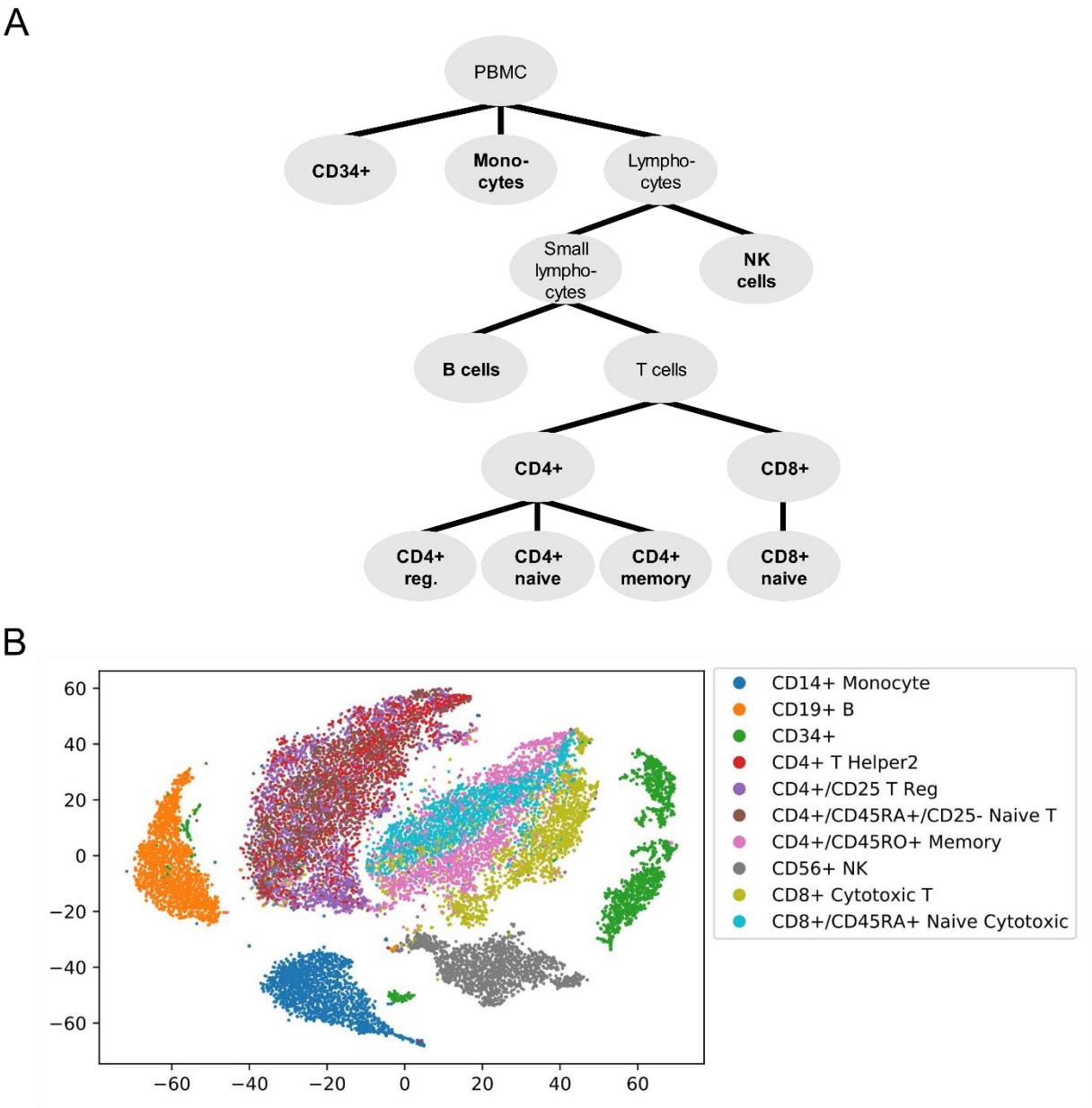


Figure S1 (A) Classification tree for the PBMC-FACS dataset. Bold names indicate cell populations that exist in our dataset. (B) t-SNE plot of the PBMC-FACS data

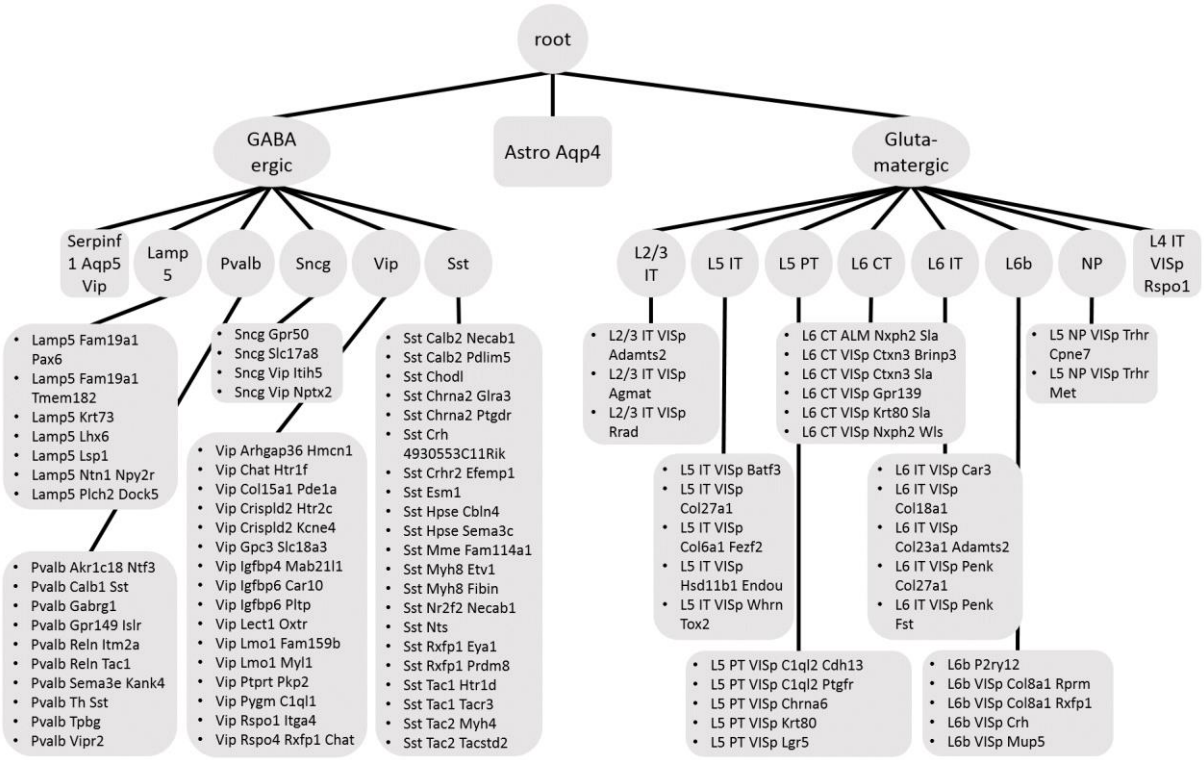


Figure S2 Classification tree of the AMB dataset. The circular and rectangle nodes indicate internal and leaf nodes respectively. If multiple leaf nodes are descendants of the same internal node, they are placed in the same rectangle.

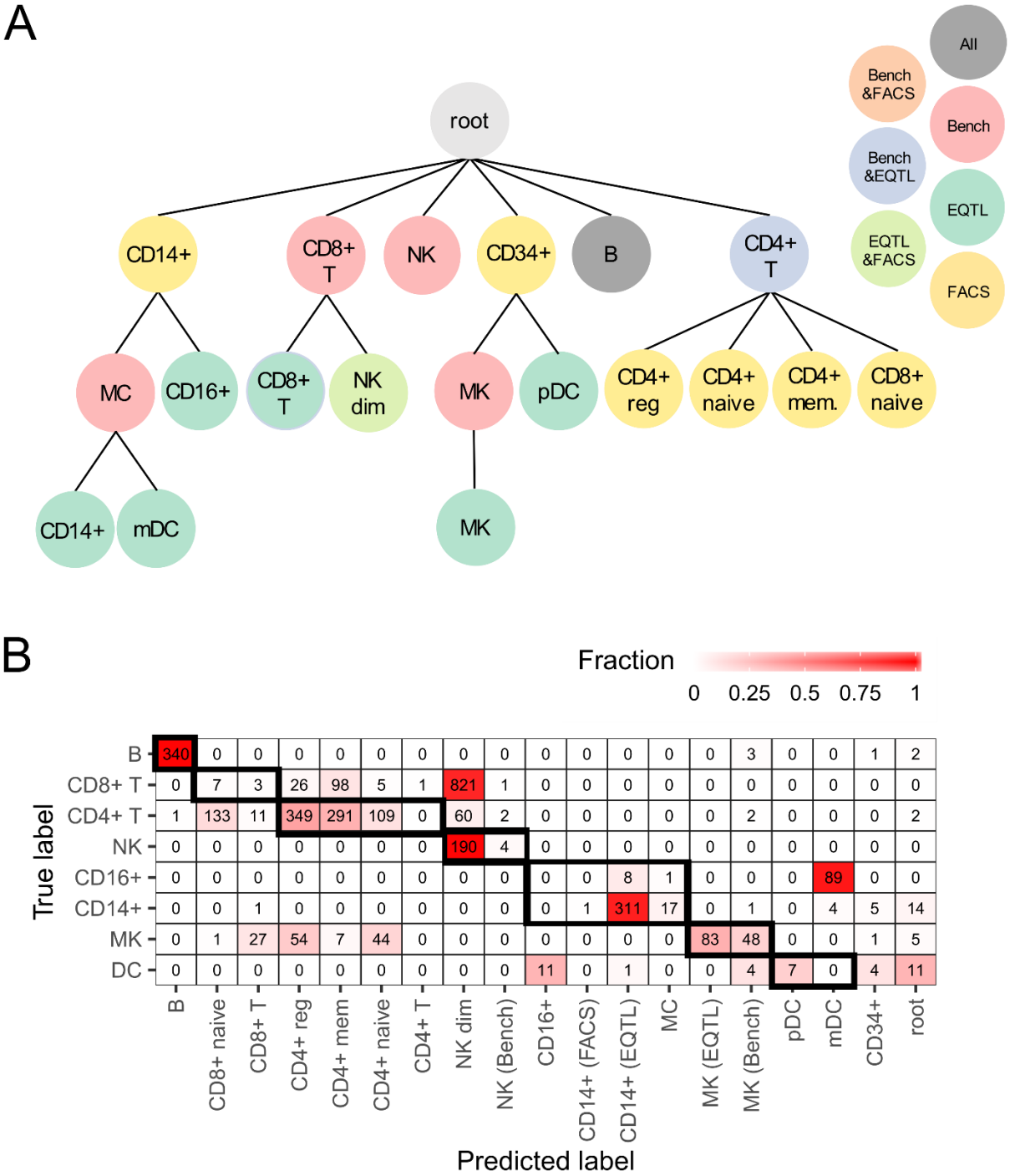
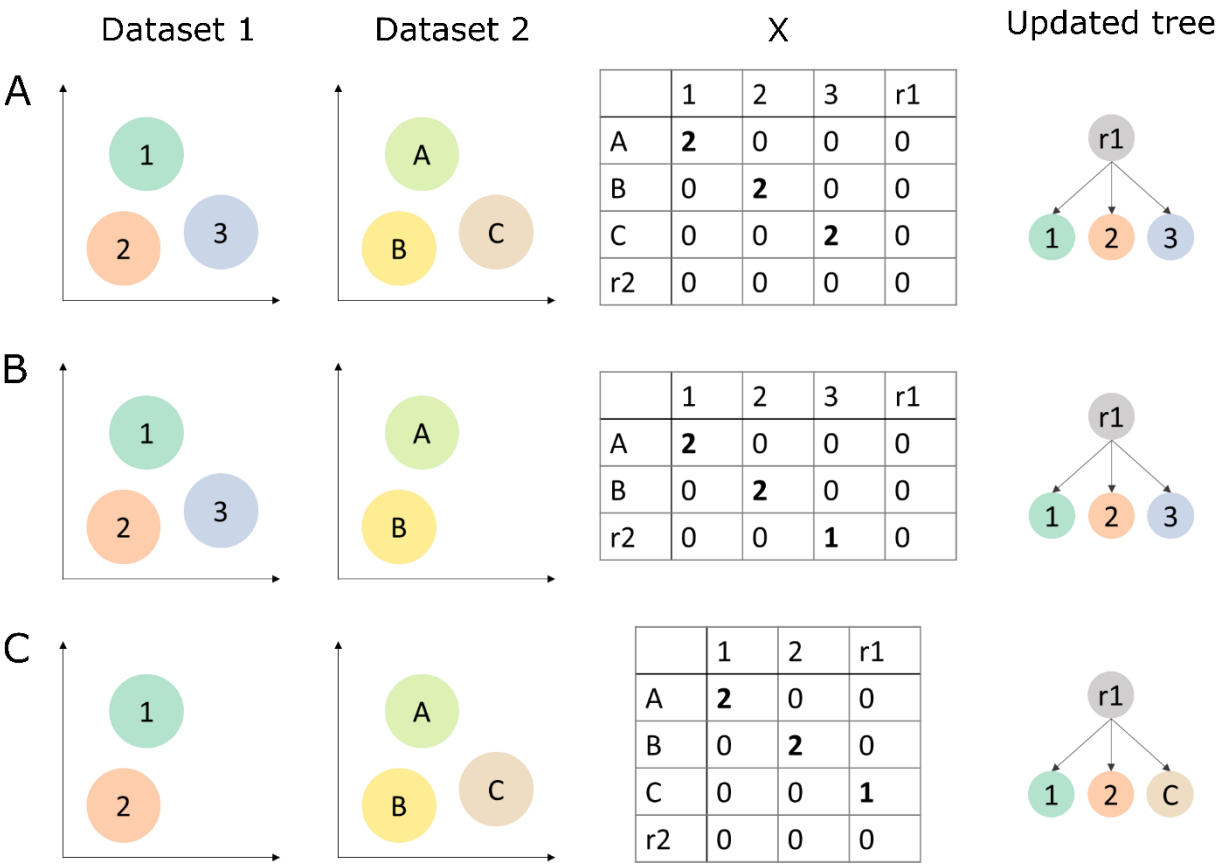


Figure S3 (A) Constructed classification tree when using a one-class SVM during the inter-dataset experiment. The color of a node represents the dataset(s) of the cell population. If a color refers to multiple dataset, this indicates that the populations from these datasets had a perfect match. (B) Confusion matrix when using the constructed classification tree to predict the labels of PBMC-Bench10Xv3.

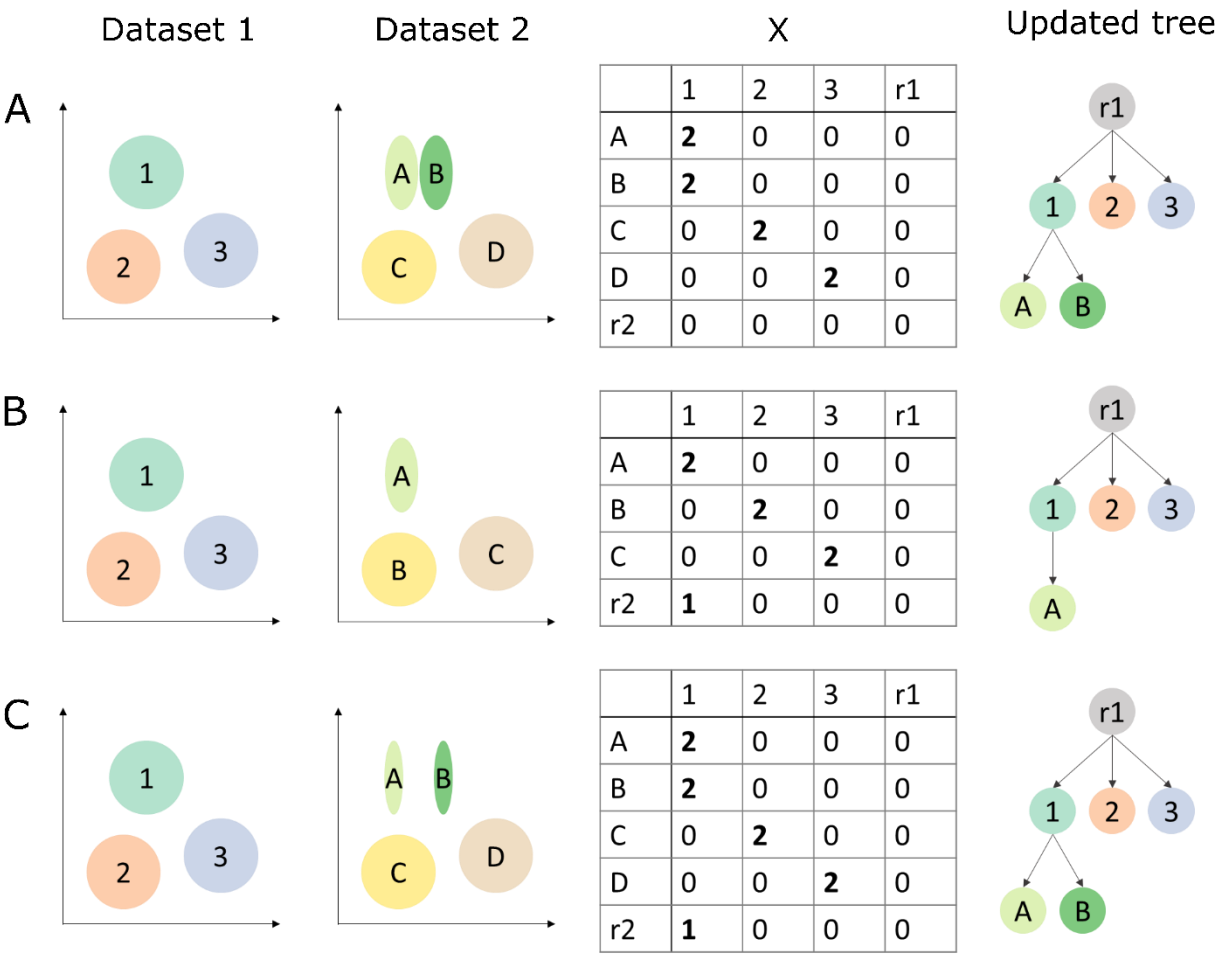


	1	2	3	r1
A	**2**	0	0	0
B	0	**2**	0	0
C	0	0	**2**	0
r2	0	0	0	0

	1	2	3	r1
A	**2**	0	0	0
B	0	**2**	0	0
r2	0	0	**1**	0

	1	2	r1
A	**2**	0	0
B	0	**2**	0
C	0	0	**1**
r2	0	0	0

Figure S4 Schematic examples of the simple scenarios. For each scenario, we show what the cell populations in the two datasets could look like, X and the updated tree.

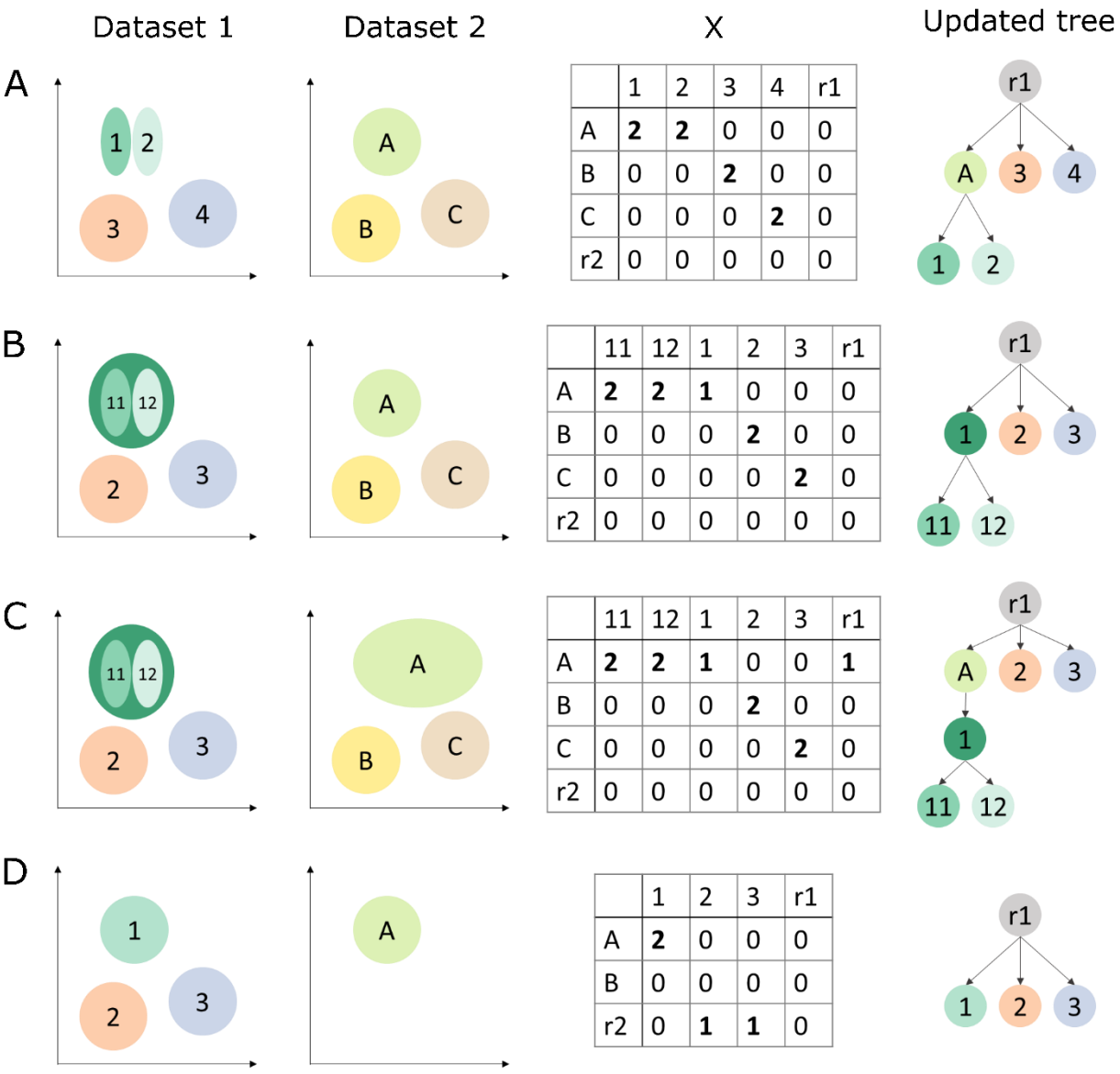


	1	2	3	r1
A	**2**	0	0	0
B	**2**	0	0	0
C	0	**2**	0	0
D	0	0	**2**	0
r2	0	0	0	0

	1	2	3	r1
A	**2**	0	0	0
B	0	**2**	0	0
C	0	0	**2**	0
r2	**1**	0	0	0

	1	2	3	r1
A	**2**	0	0	0
B	**2**	0	0	0
C	0	**2**	0	0
D	0	0	**2**	0
r2	**1**	0	0	0

Figure S5 Schematic examples of the colsums scenarios. For each scenario, we show what the cell populations in the two datasets could look like, X and the updated tree.



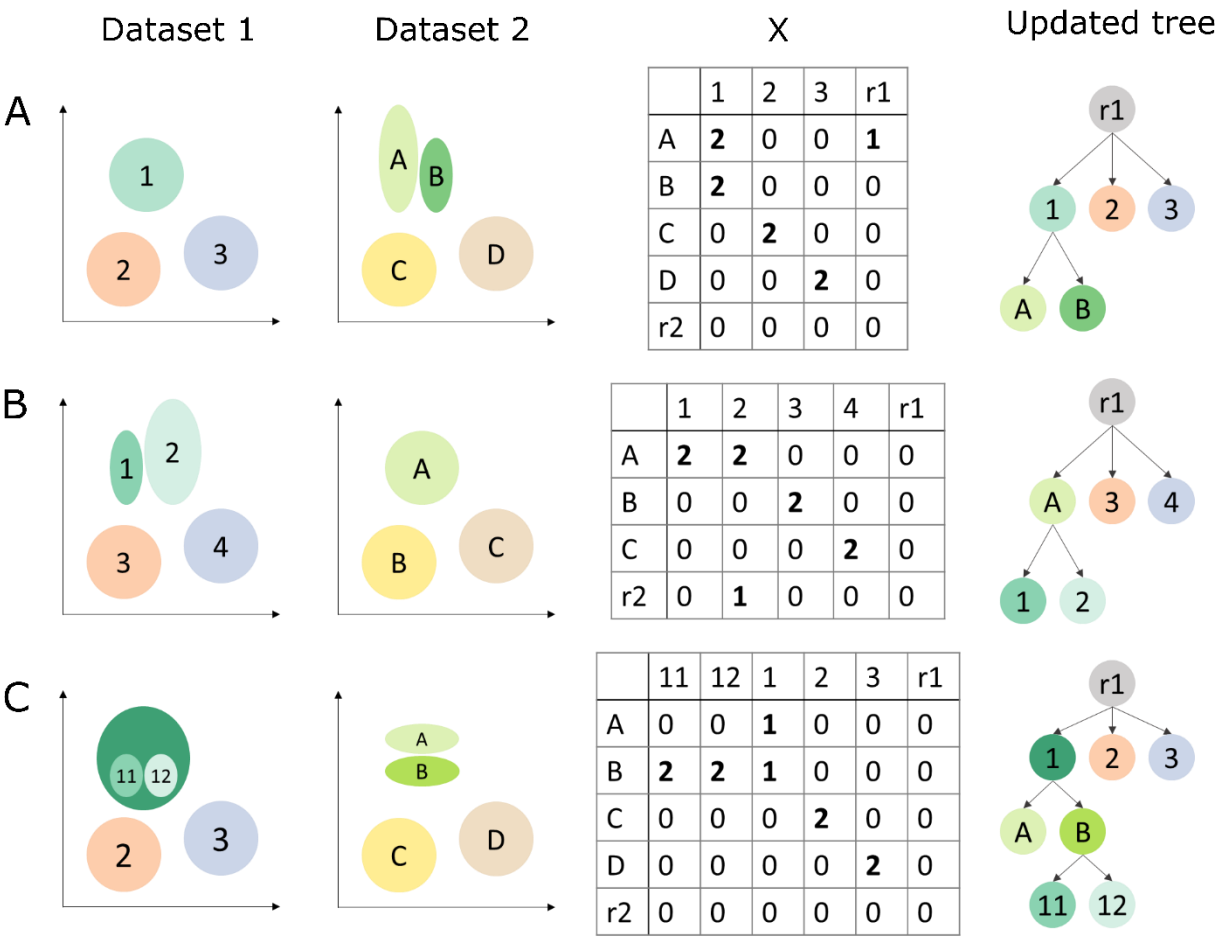
	1	2	3	4	r1
A	**2**	**2**	0	0	0
B	0	0	**2**	0	0
C	0	0	0	**2**	0
r2	0	0	0	0	0

	11	12	1	2	3	r1
A	**2**	**2**	**1**	0	0	0
B	0	0	0	**2**	0	0
C	0	0	0	0	**2**	0
r2	0	0	0	0	0	0

	11	12	1	2	3	r1
A	**2**	**2**	**1**	0	0	**1**
B	0	0	0	**2**	0	0
C	0	0	0	0	**2**	0
r2	0	0	0	0	0	0

	1	2	3	r1
A	**2**	0	0	0
B	0	0	0	0
r2	0	**1**	**1**	0

Figure S6 Schematic examples of the rowsums scenarios. For each scenario, we show what the cell populations in the two datasets could look like, X and the updated tree.

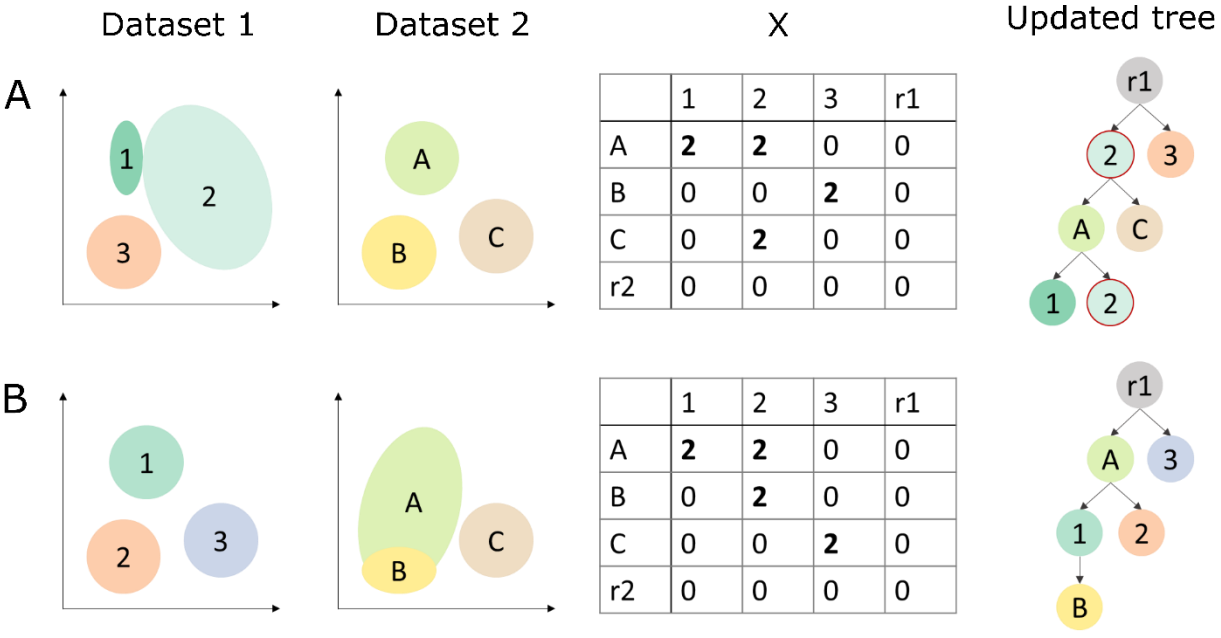


	1	2	3	r1
A	**2**	0	0	**1**
B	**2**	0	0	0
C	0	**2**	0	0
D	0	0	**2**	0
r2	0	0	0	0

	1	2	3	4	r1
A	**2**	**2**	0	0	0
B	0	0	**2**	0	0
C	0	0	0	**2**	0
r2	0	**1**	0	0	0

	11	12	1	2	3	r1
A	0	0	**1**	0	0	0
B	**2**	**2**	**1**	0	0	0
C	0	0	0	**2**	0	0
D	0	0	0	0	**2**	0
r2	0	0	0	0	0	0

Figure S7 Schematic examples of the complex scenarios. For each scenario, we show what the cell populations in the two datasets could look like, X and the updated tree.



	1	2	3	r1
A	**2**	**2**	0	0
B	0	0	**2**	0
C	0	**2**	0	0
r2	0	0	0	0

	1	2	3	r1
A	**2**	**2**	0	0
B	0	**2**	0	0
C	0	0	**2**	0
r2	0	0	0	0

Figure S8 Schematic examples of the impossible scenarios. For each scenario, we show what the cell populations in the two datasets could look like, X and why the updated tree is not possible.

Supplementary tables

Table S1 Labels of the simulated dataset when testing tree construction

Original label	Label Batch 1	Label Batch 2	Label Batch 3
Group1	Group12	Group1	Group1
Group2	Group12	Group2	Group2
Group3	Group3	Group3	Group3
Group4	Group456	Group4	Group4
Group5	Group456	Group56	Group5
Group6	Group456	Group56	Group6

Table S2 Labels of PBMC-FACS dataset when testing tree construction

Original label	Label Batch 1	Label Batch 2	Label Batch 3
CD14+ Monocytes	CD14+ Monocytes	CD14+ Monocytes	CD14+ Monocytes
CD19+ B-cells	CD19+ B-cells	CD19+ B-cells	CD19+ B-cells
CD34+ cells	CD34+ cells	CD34+ cells	CD34+ cells
CD56+ NK cells	CD56+ NK cells	CD56+ NK cells	CD56+ NK cells
CD4+ T-cells	T-cells	CD4+ T-cells	-
CD4+/CD25+ reg. T-cells	T-cells	-	CD4+/CD25+ reg. T-cells
CD4+/CD45RA+/CD25- naive T-cells	T-cells	-	CD4+/CD45RA+/CD25- naive T-cells
CD4+/CD45RO+ mem. T-cells	T-cells	-	CD4+/CD45RO+ mem. T-cells
CD8+ T-cells	T-cells	CD8+ T-cells	-
CD8+/CD45RA+ naïve T-cells	T-cells	-	CD8+/CD45RA+ naïve T-cells

Table S3 Confusion matrix of the linear SVM on the PBMC data. Here, the linear SVM was trained using the predefined hematopoietic tree.

	CD34+	CD14+ MONOCYTES	CD56+ NK CELLS	CD19+ B CELLS	CD4+ T CELLS	CD4+/CD25 T REG	CD4+/CD45RA+/ CD25- NAIVE T	CD4+/CD45RO+ MEMORY	CD8+ CYTOTOXIC T	CD8+/CD45RA+ NAIVE CYTOTOXIC	T CELLS	SMALL LYMPHO-CYTES	LYMPHO-CYTES	ROOT
CD34+	1964	1	1	26	0	0	0	0	6	0	0	0	0	2
CD14+ MONOCYTES	0	1979	1	1	2	15	0	1	0	0	0	0	0	1
CD56+ NK CELLS	2	0	1990	0	0	0	0	1	6	0	0	0	0	1
CD19+ B CELLS	0	0	0	1999	0	0	0	1	0	0	0	0	0	0
CD4+ T CELLS	0	1	0	0	191	1012	790	3	0	2	0	0	0	1
CD4+/CD25 T REG	0	0	0	0	113	1707	175	4	0	0	0	0	0	1
CD4+/CD45RA+/ CD25- NAIVE T	0	1	0	1	94	140	1751	1	1	9	0	0	0	2
CD4+/ CD45RO+ MEMORY	3	0	0	0	1	33	14	1890	32	26	0	0	0	1
CD8+ CYTOTOXIC T	0	0	0	0	1	5	0	44	1885	65	0	0	0	0
CD8+/CD45RA+ NAIVE CYTOTOXIC	0	0	0	1	3	0	13	36	31	1916	0	0	0	0

Table S4 Confusion matrix of the linear SVM on the PBMC data. Here, the linear SVM was trained on the altered tree. The CD4+ memory T-cells are a subpopulation of CD8+ T-cells now.

	CD34+	CD14+ MONOCYTES	CD56+ NK CELLS	CD19+ B CELLS	CD4+ T CELLS	CD4+/CD25 T REG	CD4+/CD45RA+/ CD25- NAIVE T	CD4+/CD45RO+ MEMORY	CD8+ CYTOTOXIC T	CD8+/CD45RA+ NAIVE CYTOTOXIC	T CELLS	SMALL LYMPHO-CYTES	LYMPHO-CYTES	ROOT
CD34+	1964	1	1	26	0	0	0	2	0	4	0	0	0	2
CD14+ MONOCYTES	0	1979	1	1	0	17	0	1	0	0	0	0	0	1
CD56+ NK CELLS	2	0	1990	0	0	0	0	7	0	0	0	0	0	1
CD19+ B CELLS	0	0	0	1999	0	0	0	0	0	1	0	0	0	0
CD4+ T CELLS	0	1	0	0	0	1118	875	5	0	0	0	0	0	1
CD4+/CD25 T REG	0	0	0	0	0	1797	200	2	0	0	0	0	0	1
CD4+/CD45RA+/ CD25- NAIVE T	0	1	0	1	0	133	1860	2	0	1	0	0	0	2
CD4+/ CD45RO+ MEMORY	3	0	0	0	0	4	0	1974	0	18	0	0	0	1
CD8+ CYTOTOXIC T	0	0	0	0	0	2	0	775	0	1223	0	0	0	0
CD8+/CD45RA+ NAIVE CYTOTOXIC	0	0	0	1	0	0	3	24	0	1972	0	0	0	0

Table S5 Labels of the simulated dataset when testing tree construction with missing cell populations

Original label	Label Batch 1	Label Batch 2	Label Batch 3
Group1	Group12	Group1	Group1
Group2	Group12	Group2	Group2
Group3	Group3	Group3	Group3
Group4	Group456	Group4	Group4
Group5	Group456	-	Group5
Group6	Group456	Group6	Group6