universität wien

Cell type determination of single cell transcriptomics via ML

Sebastian Bittner 11776808
Stephan Buchner 11805363
Philipp Trinh 11839949
Julian Zimmermann 12144285

"C. elegans connectome in anatomy model ." From *Math Insight*. http://mathinsight.org/image/c_elegans_connectome_anatomy_model

Überblick

1. Background / Intro
2. Data exploration
3. Clustering
4. Machine learning Modelle
5. Neural Networks
6. Mögliche Fortsetzung

universität wien

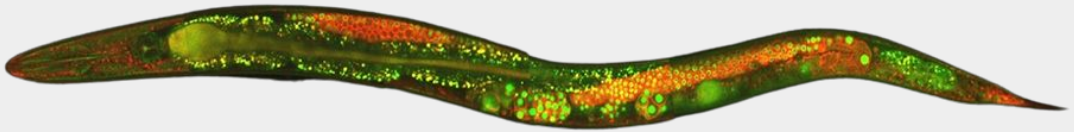
Background & Data preprocessing

Background

- Caenorhabditis elegans
  - Fadenwurm
  - 1 mm lang
  - Im Boden
  - Modellorganismus für viele Bereiche der Biologie

by Mark Leaver
https://hymanlab.org/hyman_lab/c-elegans/

- Single cell transcriptomics
  - Expression jeder Zelle für alle Gene
  - Gen-Expressions-Analyse

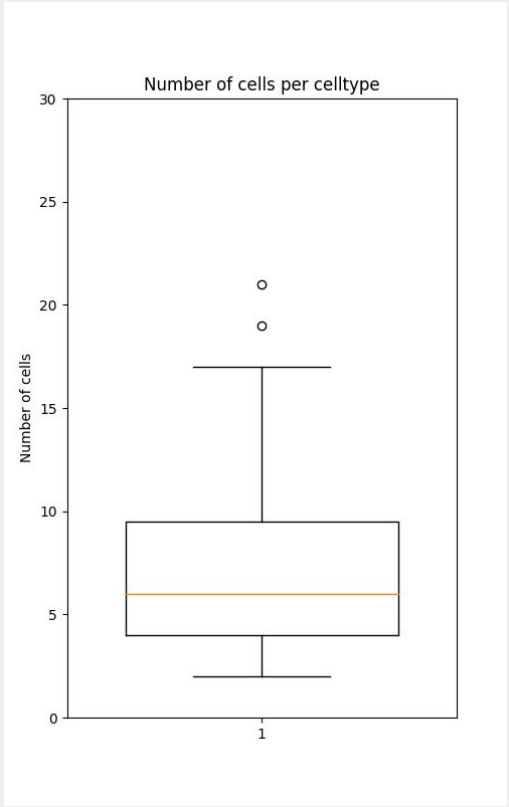


by Ian Chin-Sang,
Queens University, ON,
Canada

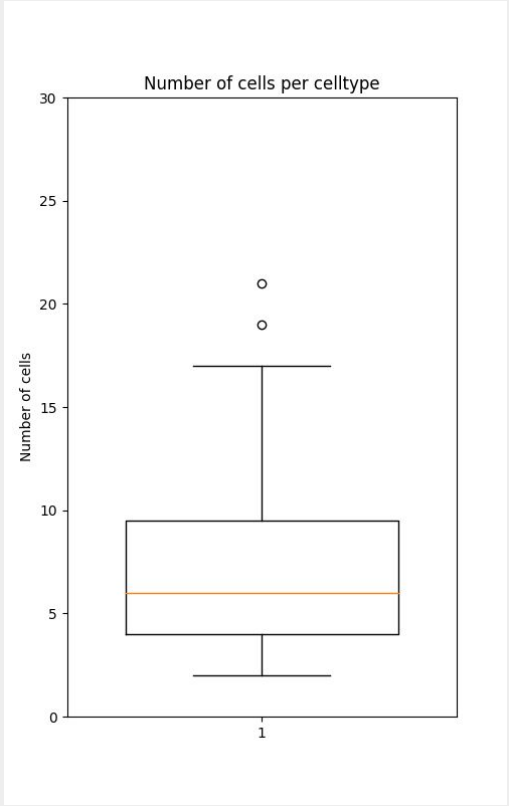
Data preprocessing

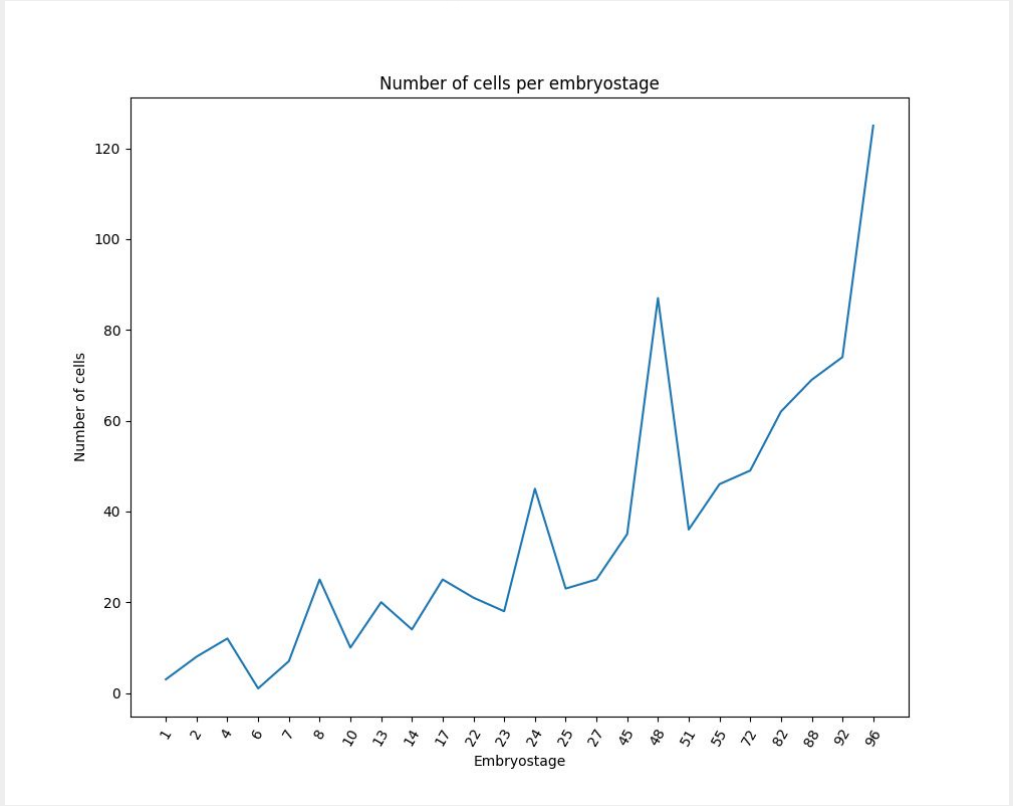
- Filtern via scanpy[2]:
  - Zellen: minimum 200 Gene exprimiert
  - Gene: minimum in 3 Zellen exprimiert
  - Normalisieren über Zellen
  - Variance stabilization: log1p
  - Selektion von den 2000 variabelsten Genen
- Batch effects[3]: z-transformation pro Embryo
- Selektion von bereits manuell annotierten Zellen[4]
- Expressionsmatrix: 840 Zellen x 2000 Gene

Data exploration



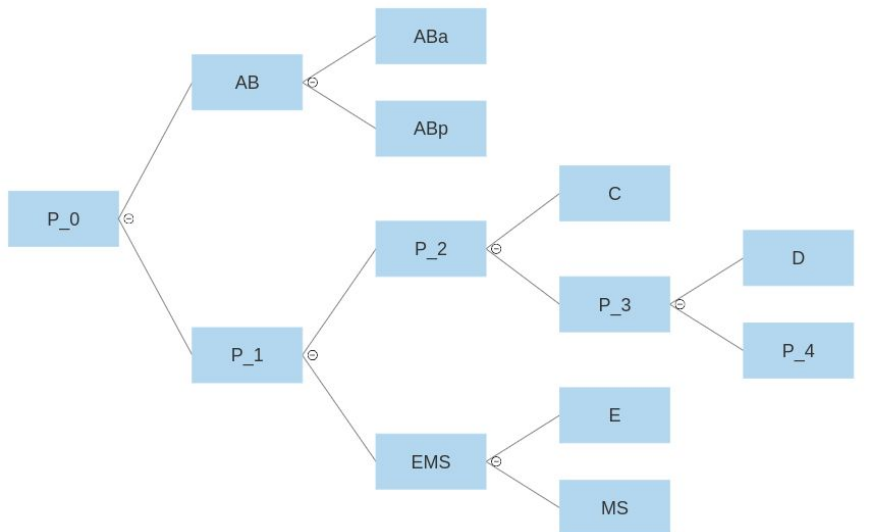
Data exploration

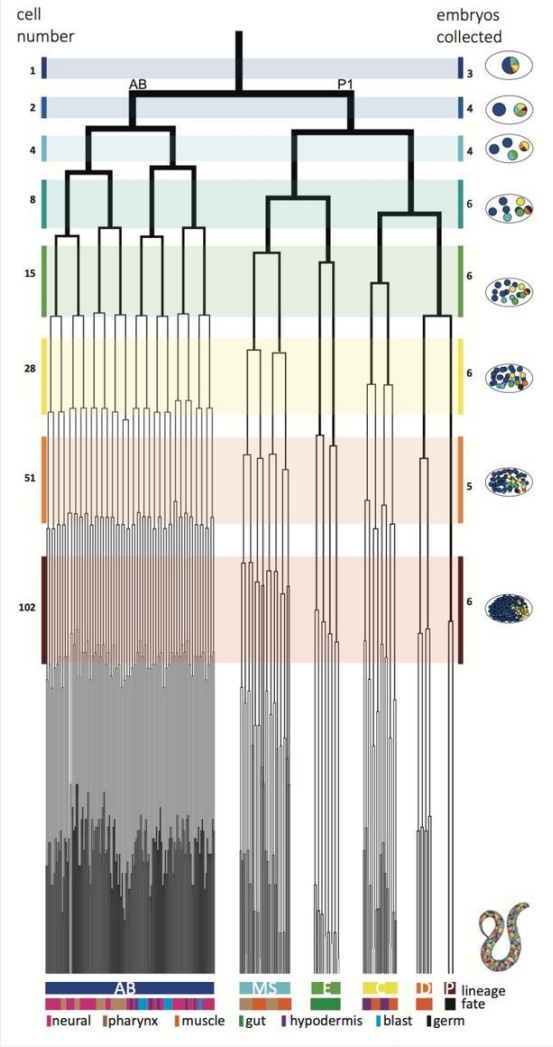




Data exploration II

- Equivalenzgruppen[4]:
- 119 Zelltypen
- Differenzierungsbaum[5]:





universität wien

Clustering

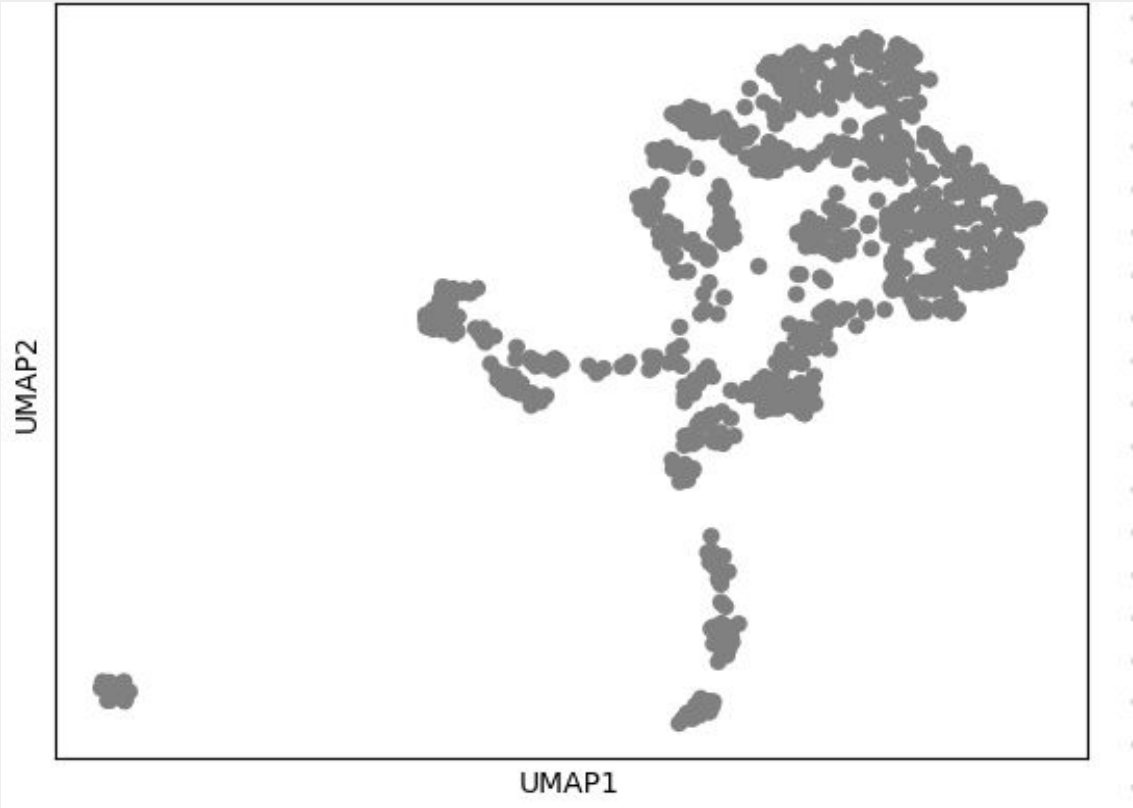
Clustering

UMAP[6]:

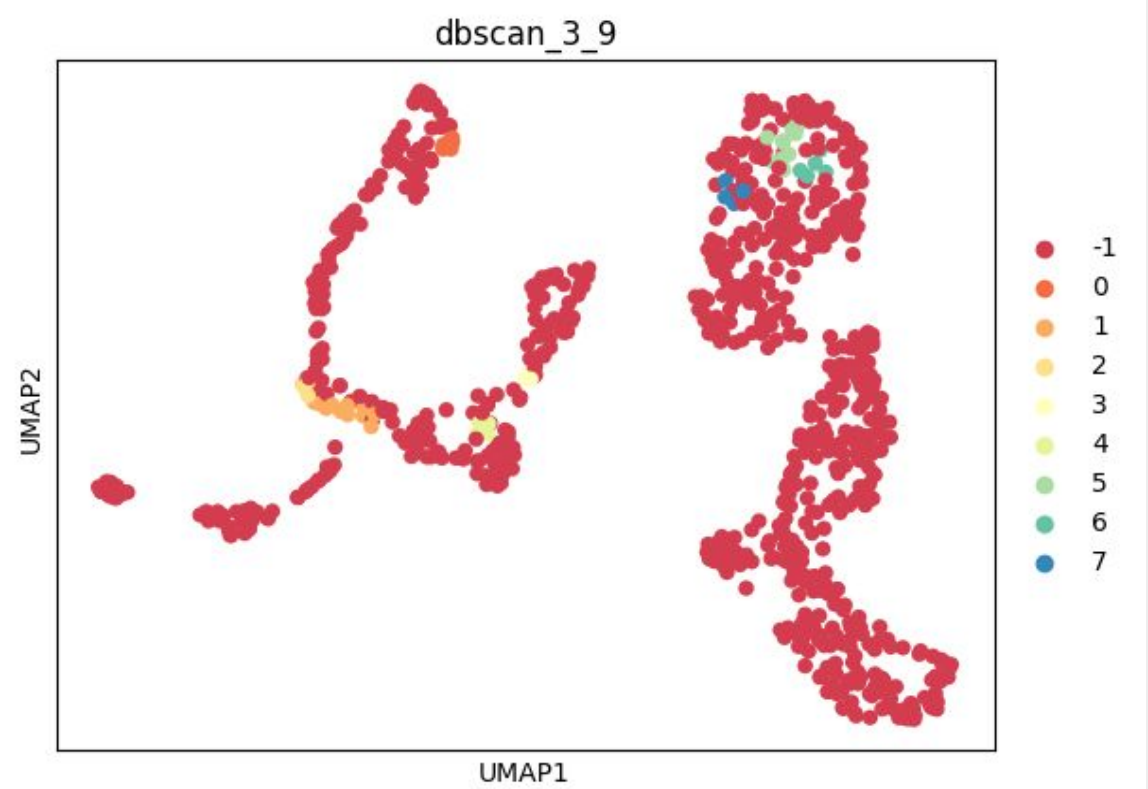
- Distance Measure zwischen Samples
- Lokale Struktur (k-nearest-neighbour -Graph) der Daten
- Reduktion der Raumdimension auf 2 bei Erhaltung der Distanzen

=> Methode, mehrdimensionale Datenpunkte zweidimensional darzustellen

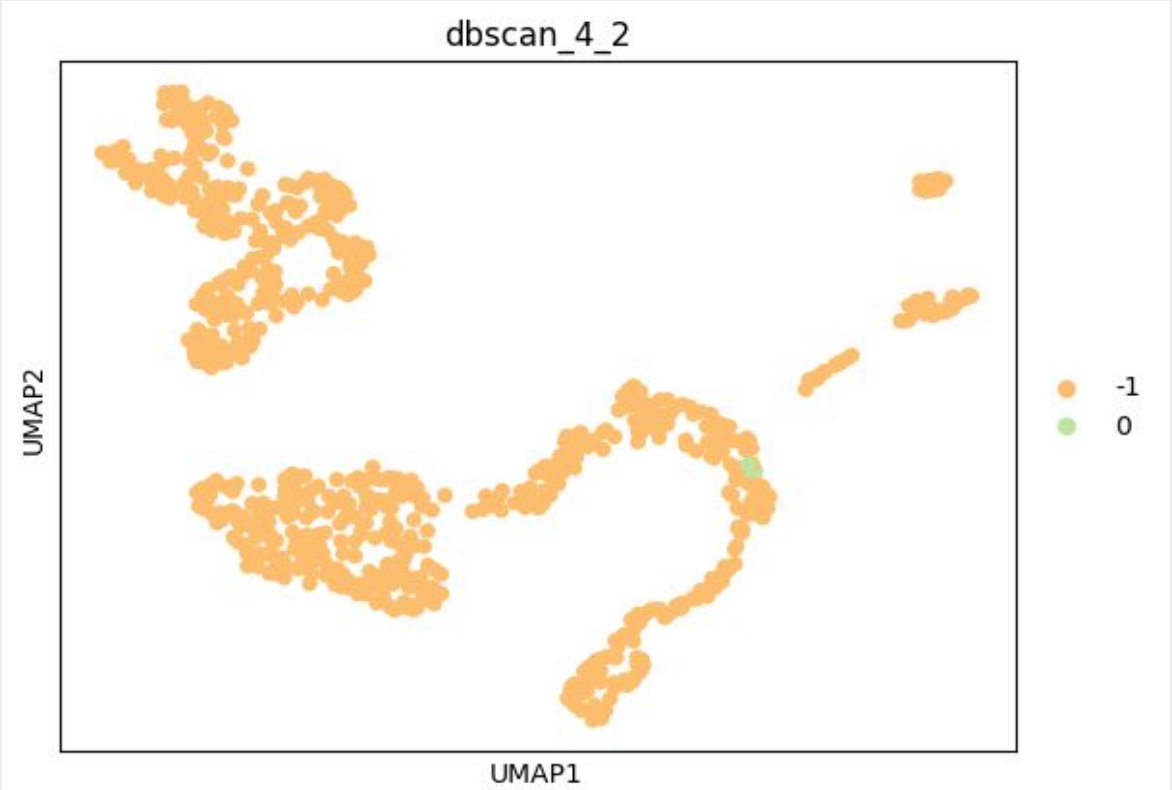
Clustering



Clustering



Clustering



Clustering

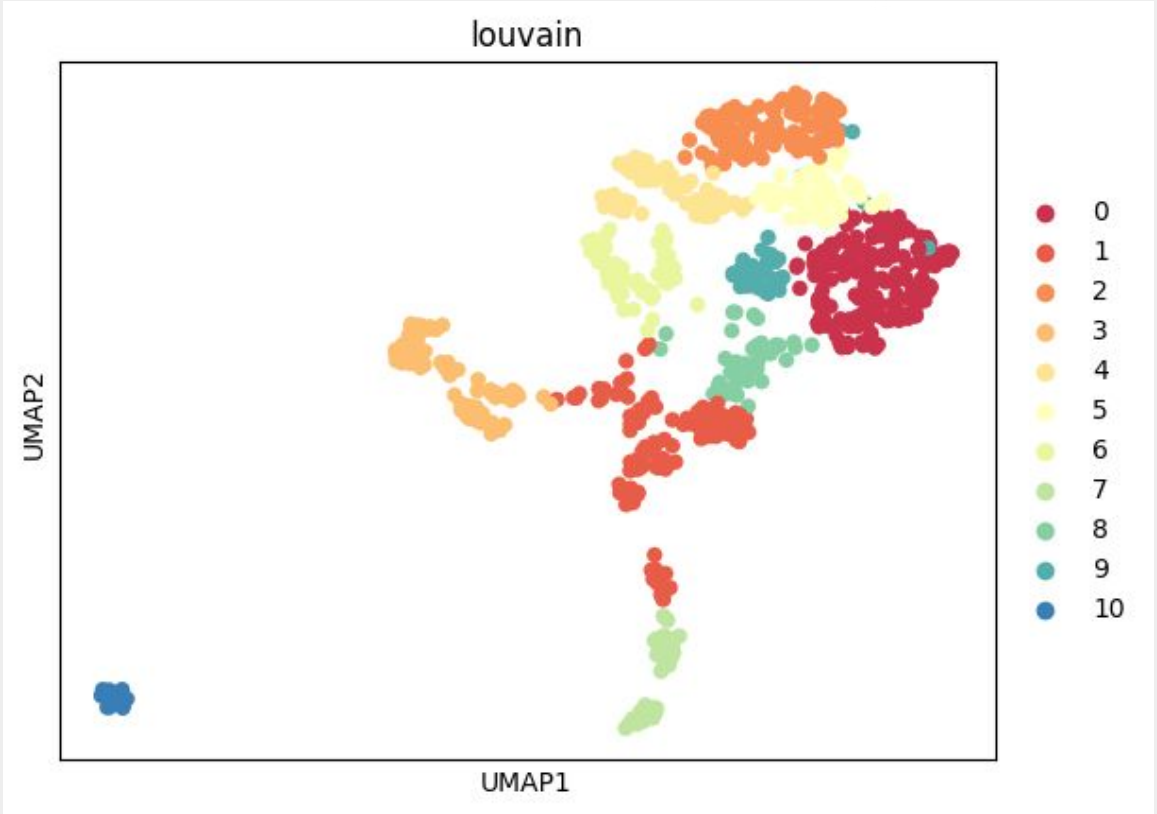
Louvain[7]:

- Distance Measure zwischen Samples
- k-nearest-neighbour-Graphen der Daten
- Zuweisung jedes Knotens zu einer “Gemeinschaft”
- Iterative Neuzuweisung basierend auf Clusterdichte (“Modularität”).

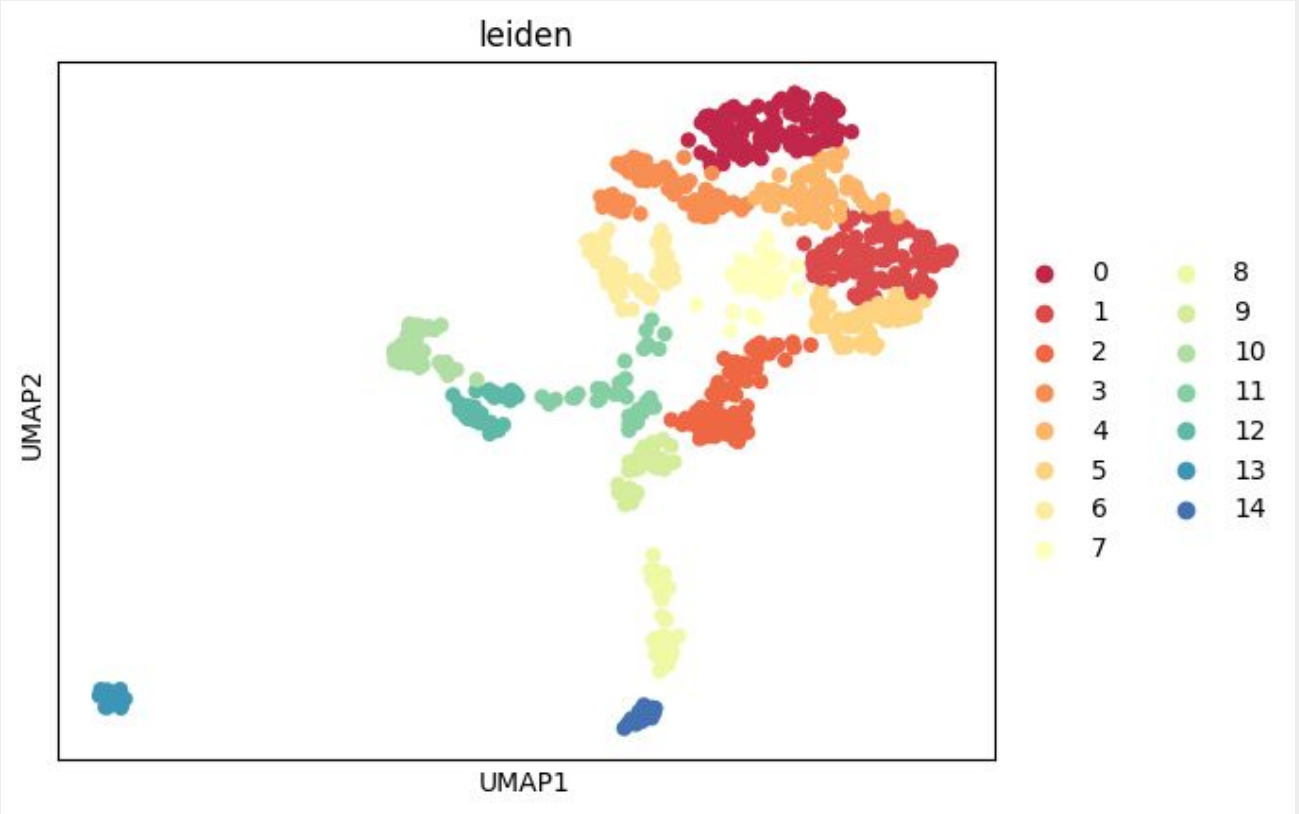
Leiden[7]: Verbesserte Version, robuster gegenüber Rauschen.

ScanPy: k-nearest-neighbour-Graphen basierend auf Genexpressionsdaten.

Clustering

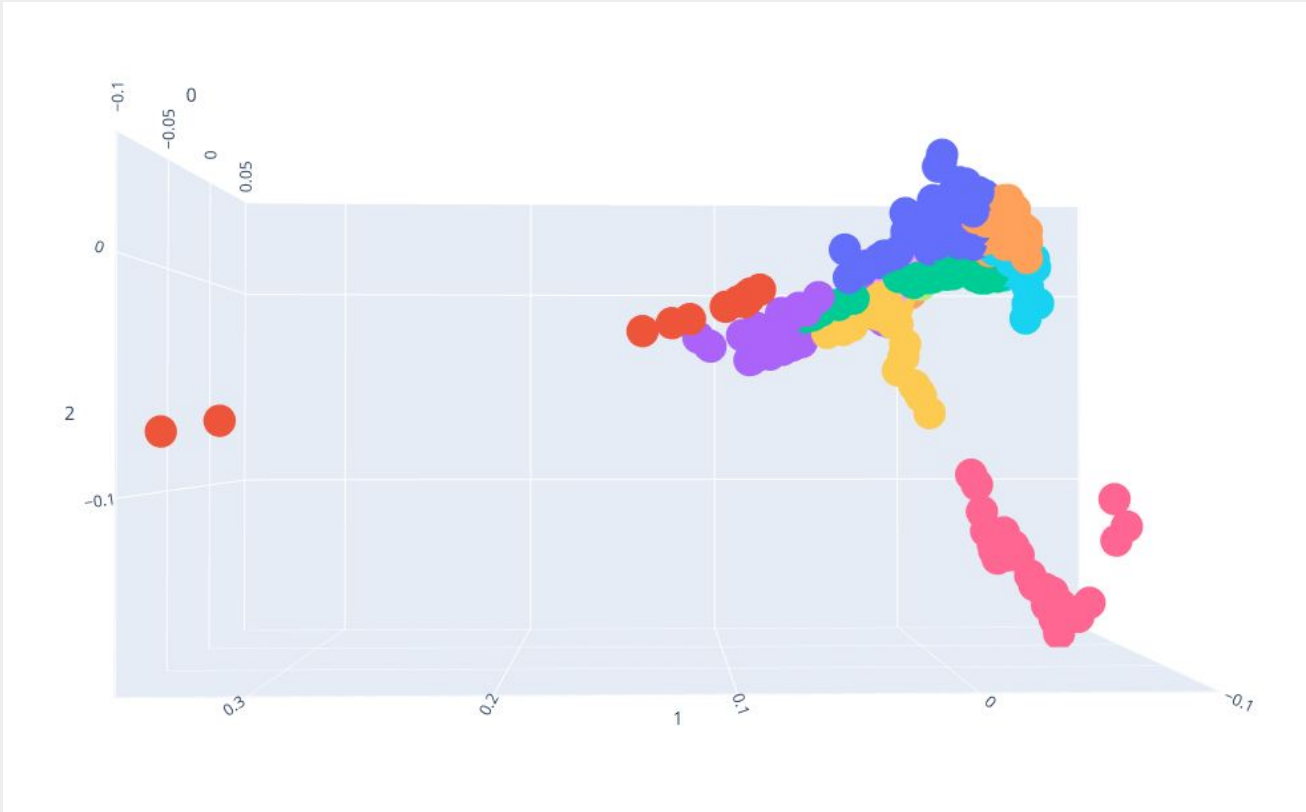


Clustering

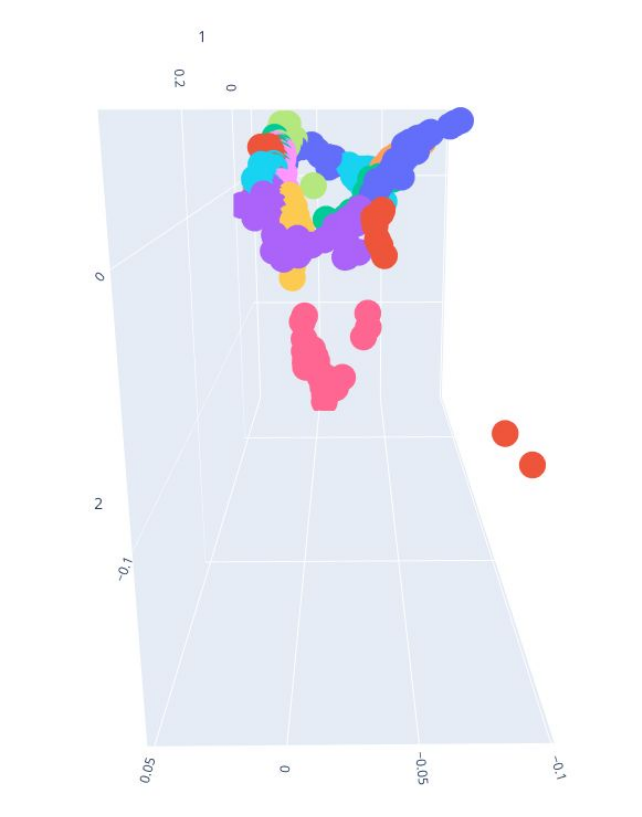


Clustering

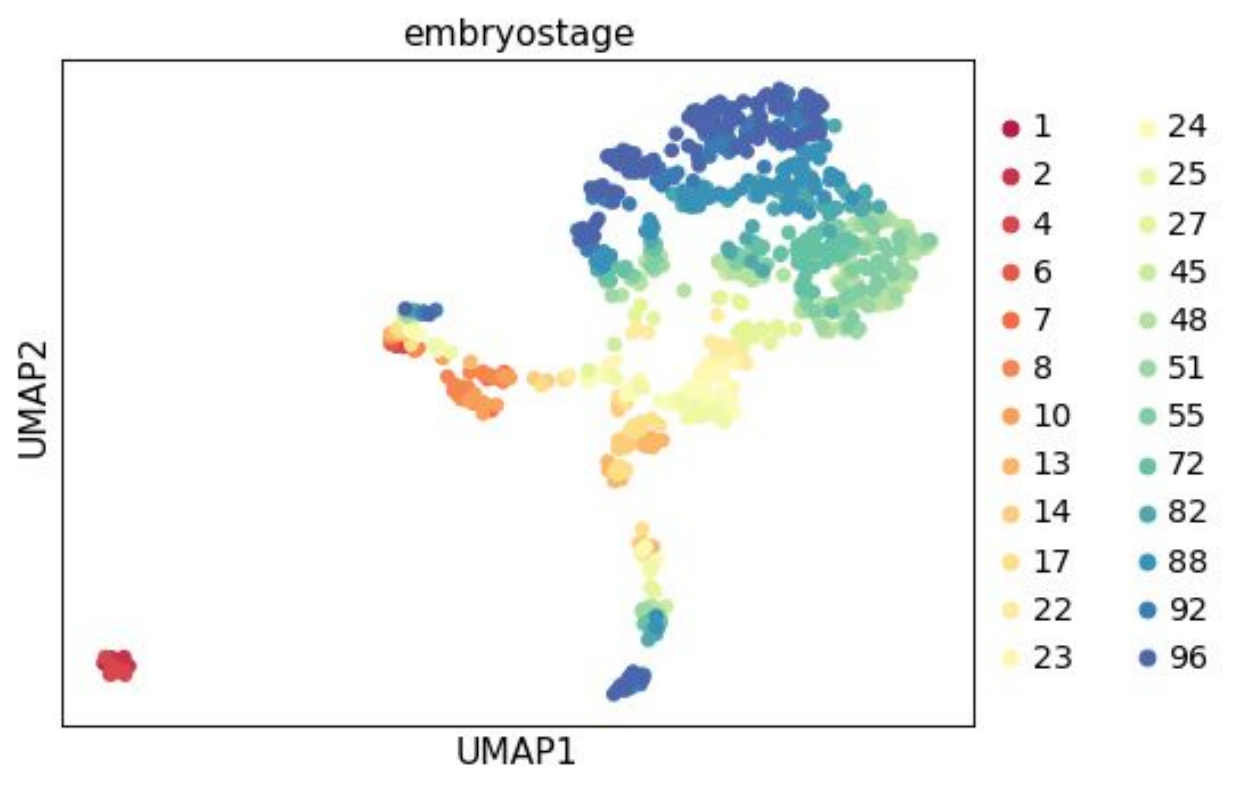
universität wien



Clustering

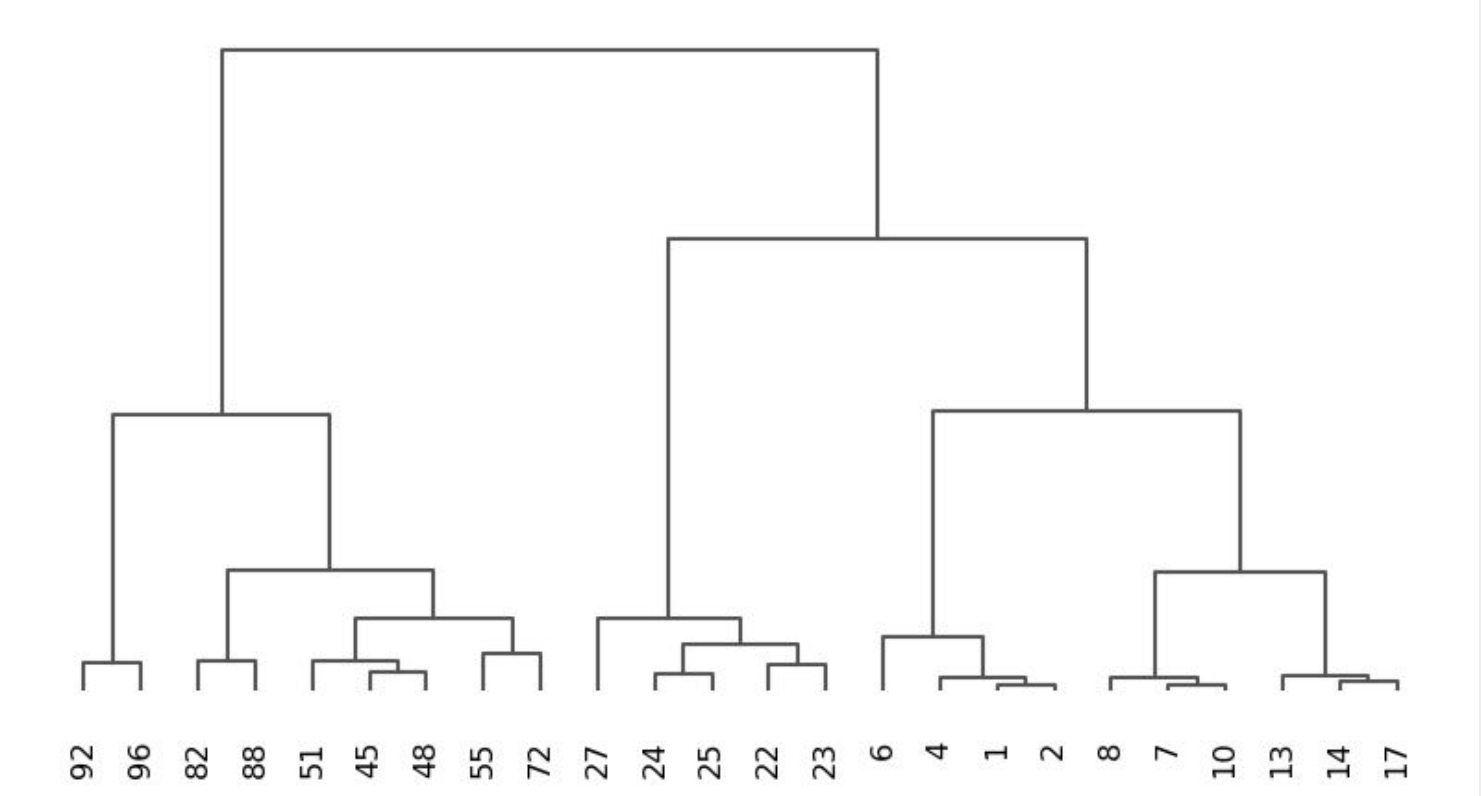


Clustering

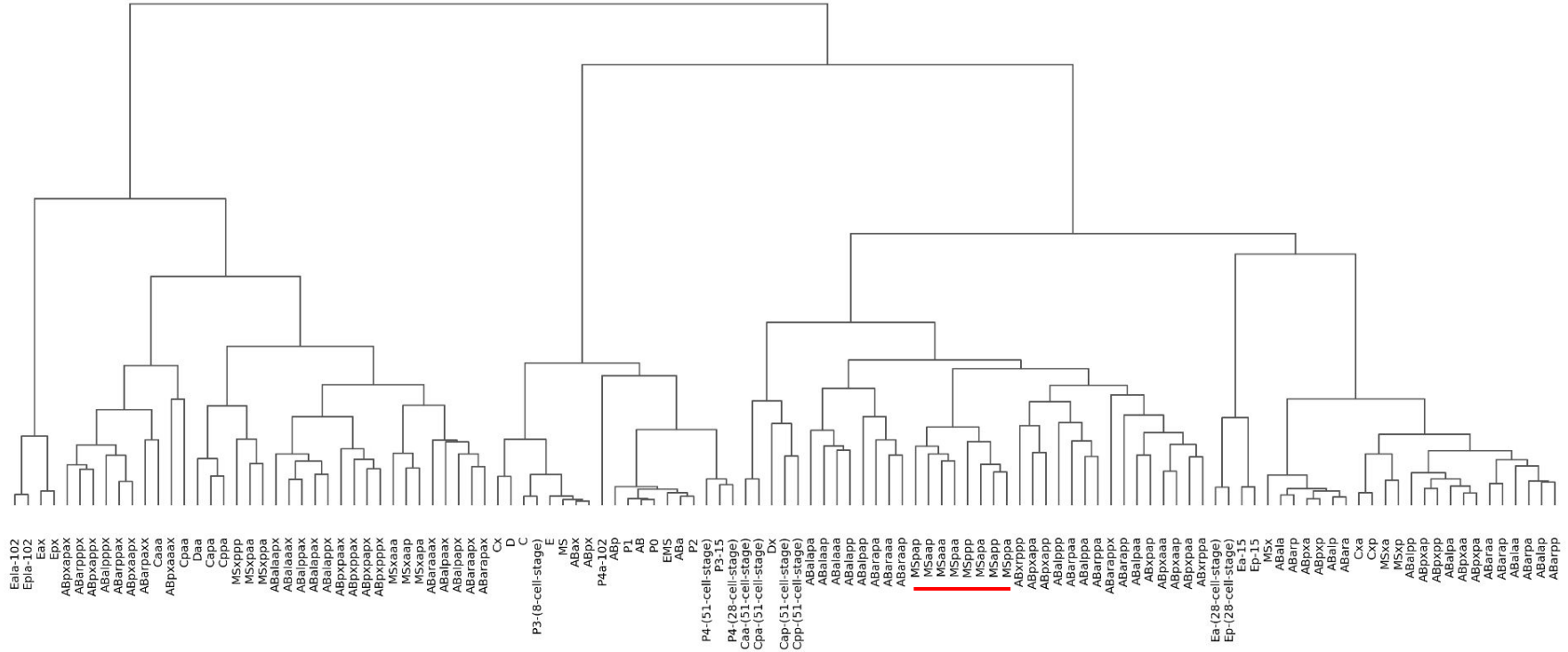

embryostage
UMAP2
UMAP1
1
2
4
6
7
8
10
13
14
17
22
23
24
25
27
45
48
51
55
72
82
88
92
96

Hierarchical Clustering

universität wien



Hierarchical Clustering



universität wien

Machine Learning Modelle

ML Modelle

via scikitlearn[8]:

1. SVM
2. Random forest
3. KNeighborsClassifier
4. Multinomial Logistic Regression
5. MLP Classifier
6. Neural Networks

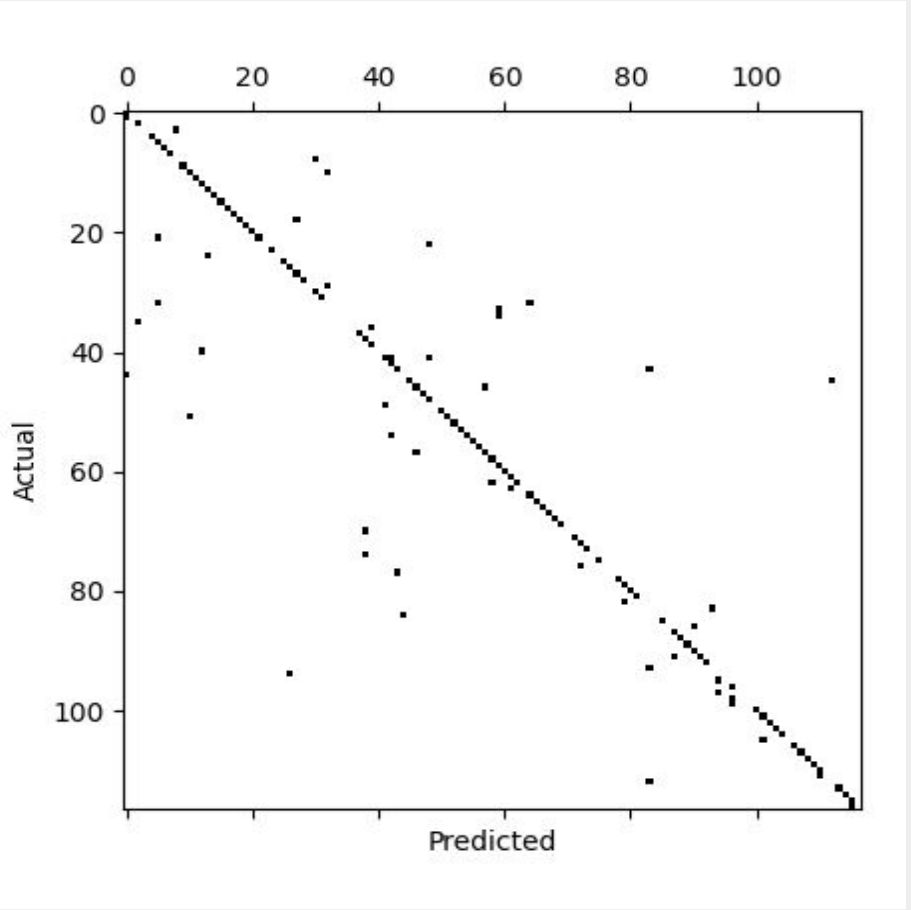
Support Vektor Maschine

1.

- 72 % true positives
- hyperparameters:

C	100
kernel	linear

tested param grid =
{"C": [0.1, 0.8, 4, 10, 30, 100],
"kernel": ["linear"]}



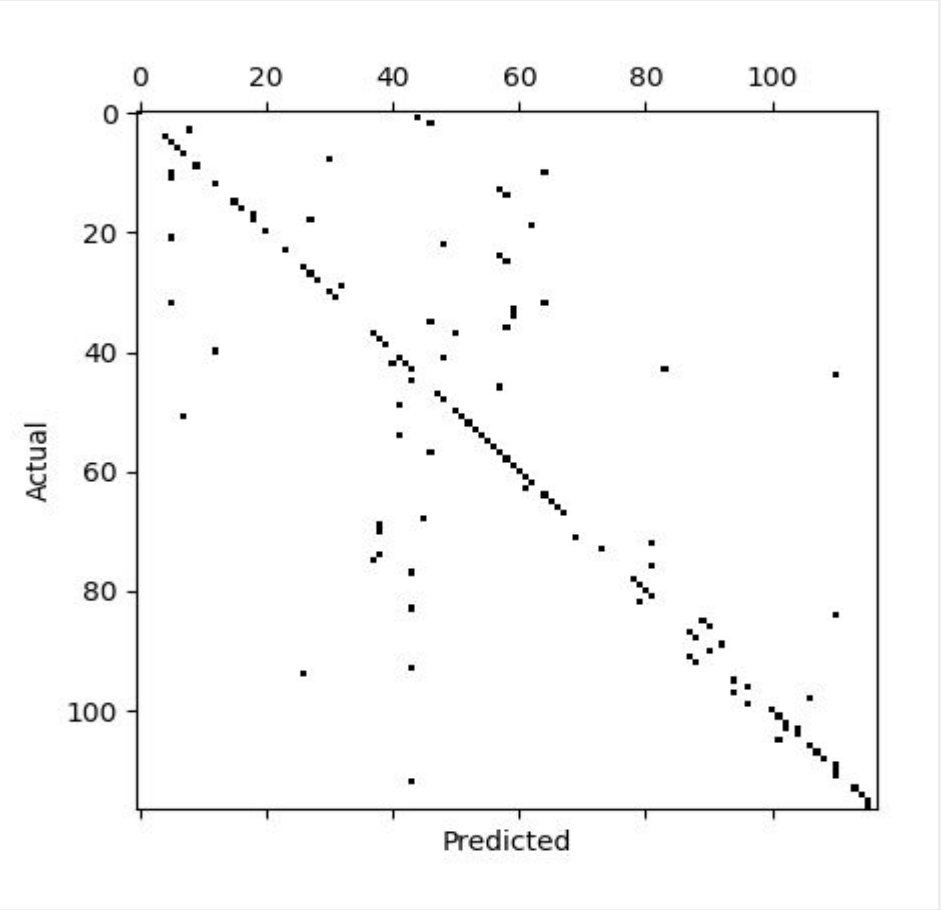
Random Forest

2.

- 58 % true positives
- hyperparameters:

n_estimators	1000
max_depth	None

tested param grid =
{"n_estimators": [10, 100, 1000],
"max_depth": [None, 10, 100]}



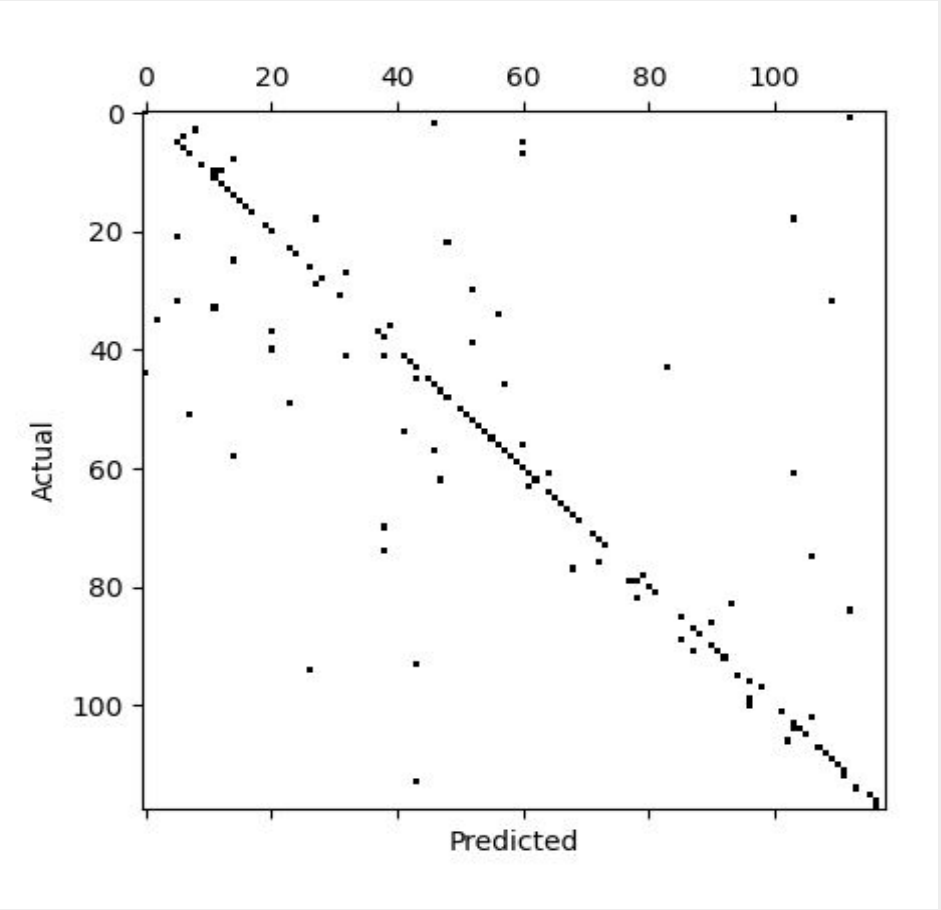
Kneighbors Classifier

3.

- 58.3 % true positives
- hyperparameters:

n_neighbors	1

tested param grid = {"n_neighbors": [1, 5, 10, 20, 50]}



Multinomial Logistic Regression

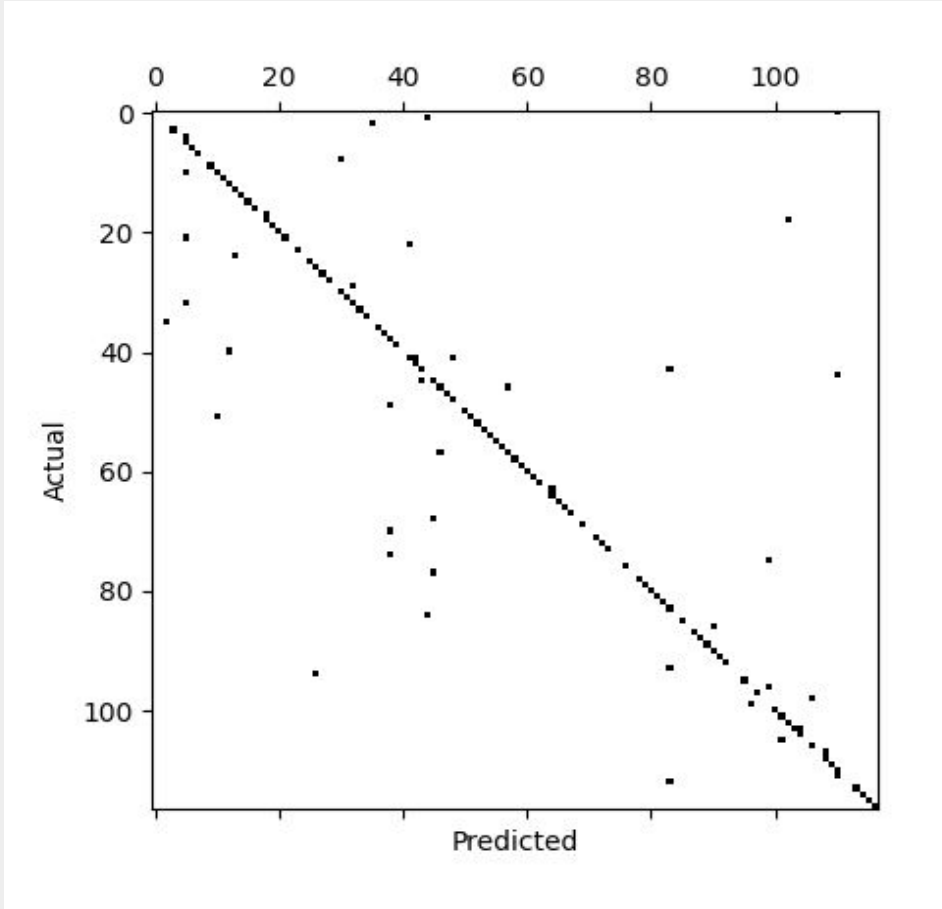
4.

- 75 % true positives
- hyperparameters:

C:	100
penalty	L1
solver	saga
multi_class	multinomial
max iteration	1000

tested param grid =

{'penalty': ['l1', 'l2'],
'solver': ['newton-cg', 'lbfgs', 'saga'],
'C': [0.1, 1, 10, 100]}
'multi_class': ['multinomial'],}



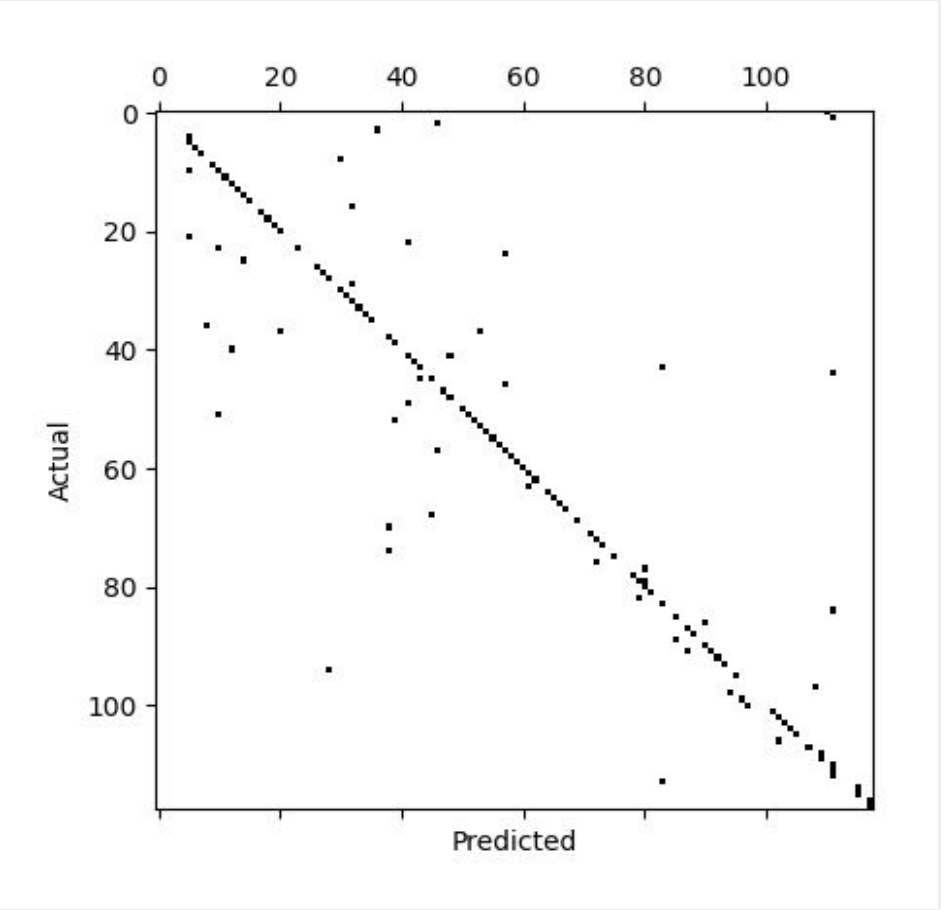
MLP Classifier

5.

- 68.45 % true positives
- hyperparameters:

activation:	relu
alpha:	1.00
hidden layer sizes:	(100)
max iteration:	1000

tested param grid =
{"hidden_layer_sizes": [(100,), (100, 100), (100, 100, 100)],
"activation": ["relu", "logistic", "tanh"],
'alpha': [0.001, 0.1, 0.01, 1]}



universität wien

Neuronale Netzwerke

6.

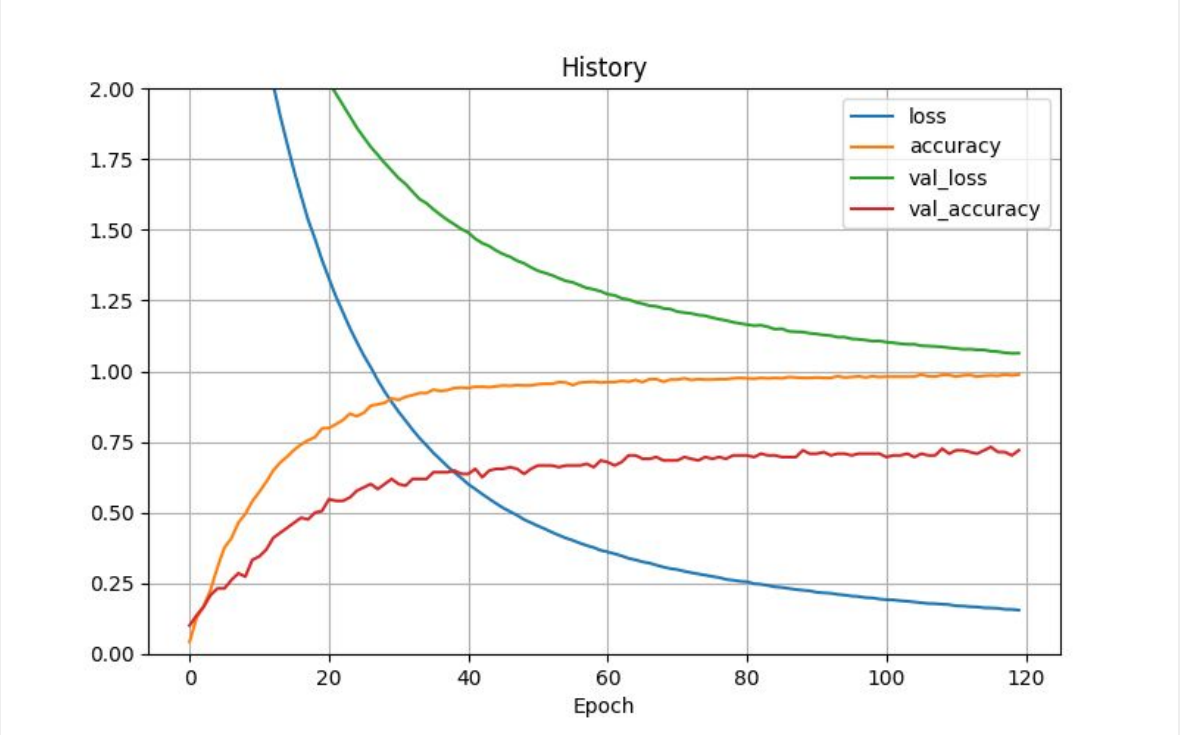
via tensorflow.keras[9]

Modell 1

2 Dense layers:	600, 800 Neuronen
Aktivierungsfunktion	tanh
1 Softmax	119 Neuronen (entsprechend den 119 Zelltypen)
optimizer	Standard “Sgd” von keras

Ergebnis Modell 1

Richtig erkannte Samples:
~68-75%

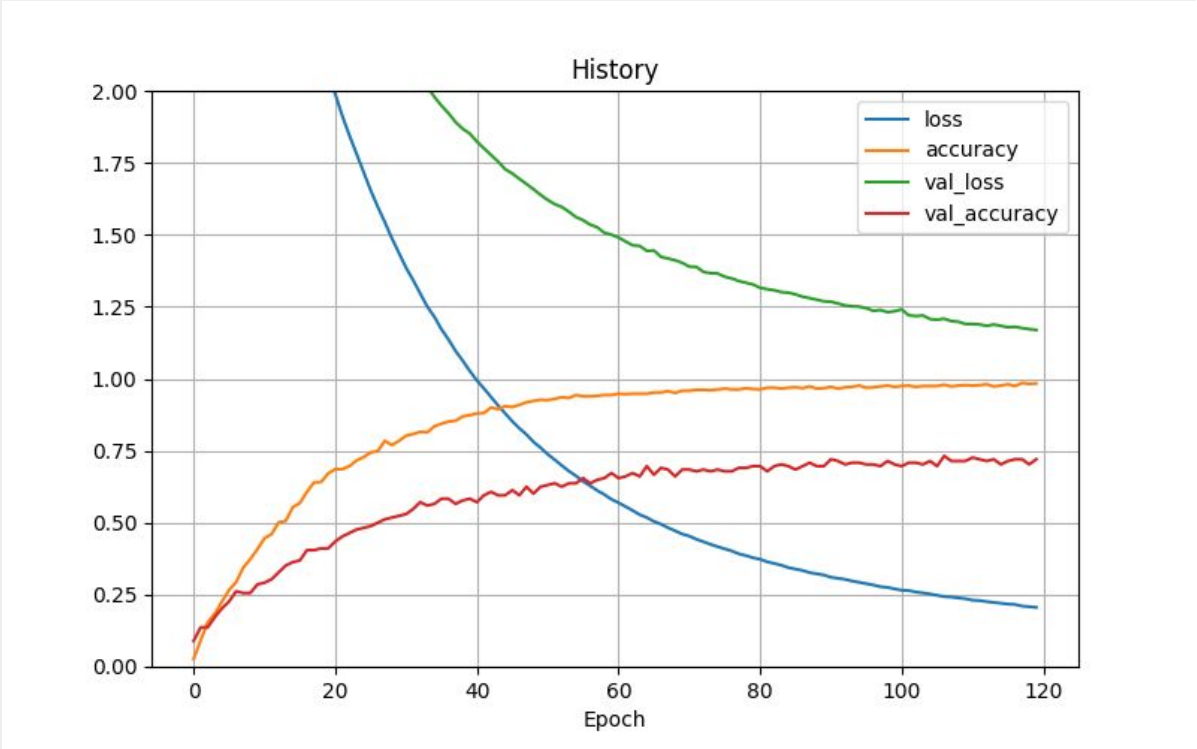


Modell 2

4 Dense layers	500,400,300,200 Neuronen
Aktivierungsfunktion	tanh
1 Softmax	119 Neuronen (entsprechend den 119 Zelltypen)
optimizer	Standard “Sgd” von keras

Ergebnis Modell 2

Richtig erkannte Samples:
~67-73%

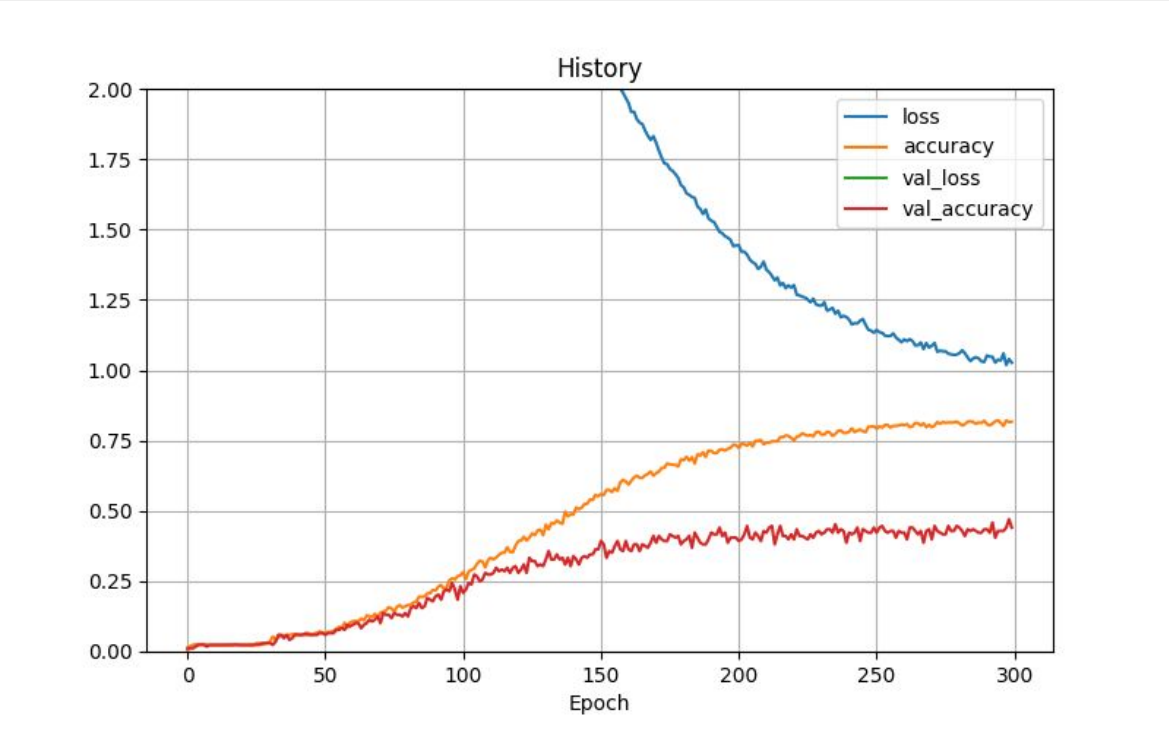


NN with Multihead attention

- Multihead Attention ist Basis für Transformer -> GPT[10]
- Hat drei inputs (query, key, value)
- Input is zu jedem der inputs jeweils durch einen dense layer verbunden
- Nachher noch ein dense und ein softmax layer
- Scheint nicht so geeignet zu sein für unsere Daten

Multihead Attention Model Ergebnis

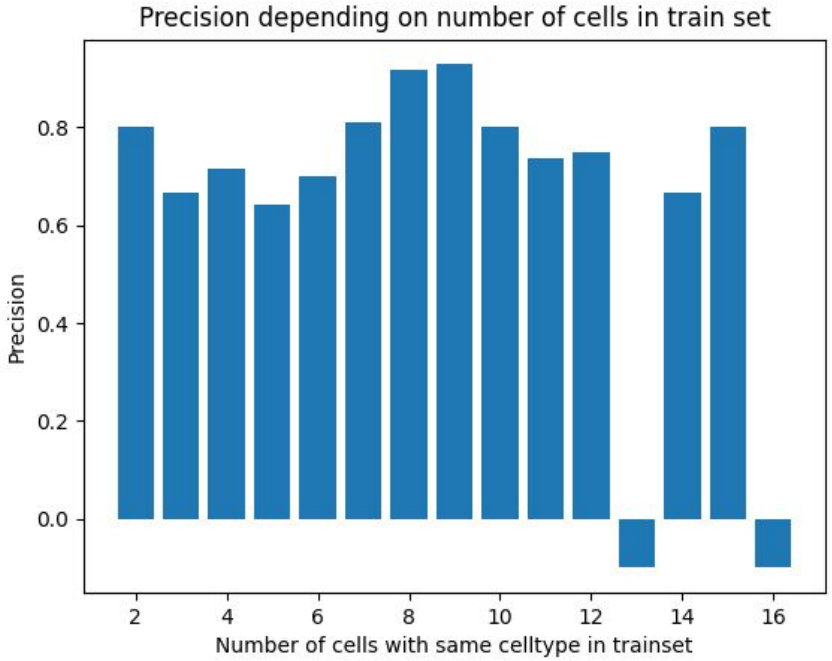
Richtig erkannte Samples:
~40-50%

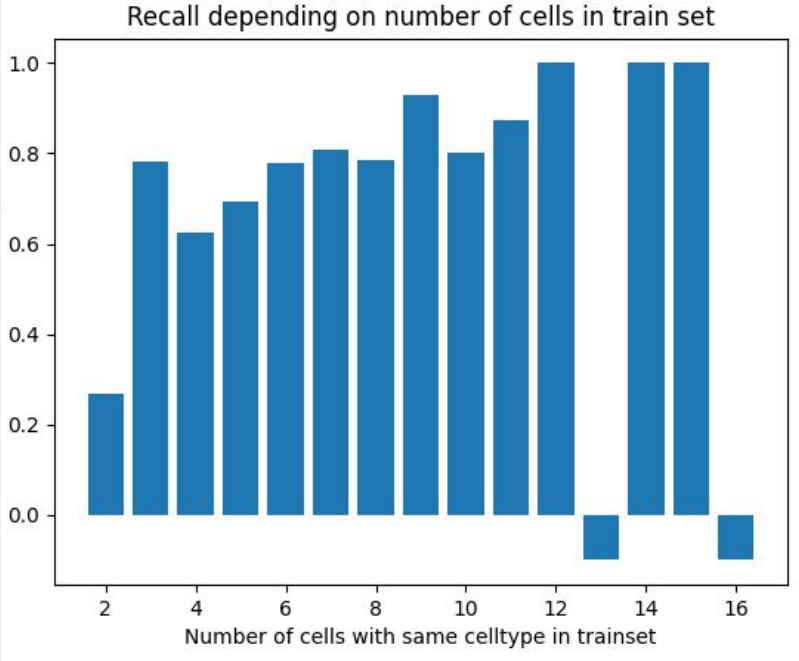


Auswirkung der Anzahl an Samples im Trainingsset

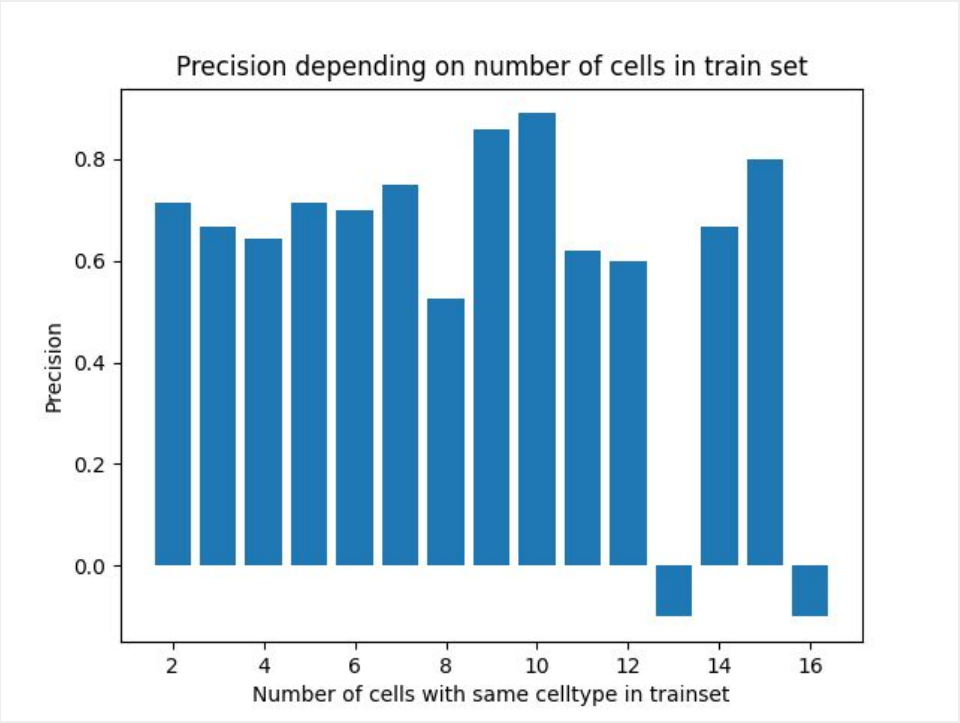
- Recall: Anzahl richtig erkannter Samples von Zelltypen, die n mal im Trainingsset vorkommen/Anzahl Samples der entsprechenden Zelltypen im Testset
- Precision: Anzahl richtig erkannter Samples von Zelltypen, die n mal im Trainingsset vorkommen/Anzahl Samples im Testset, die als der entsprechende Zelltyp klassifiziert wurden

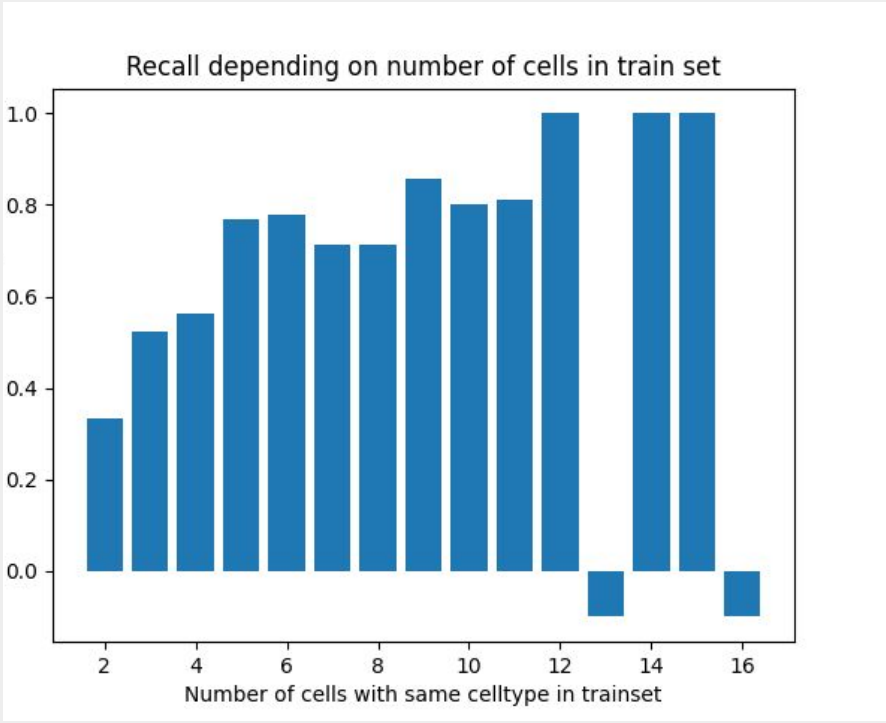
Modell 1


Precision depending on number of cells in train set
Precision
0.0
0.2
0.4
0.6
0.8
2
4
6
8
10
12
14
16
Number of cells with same celltype in trainset


Recall depending on number of cells in train set
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
12
14
16
Number of cells with same celltype in trainset

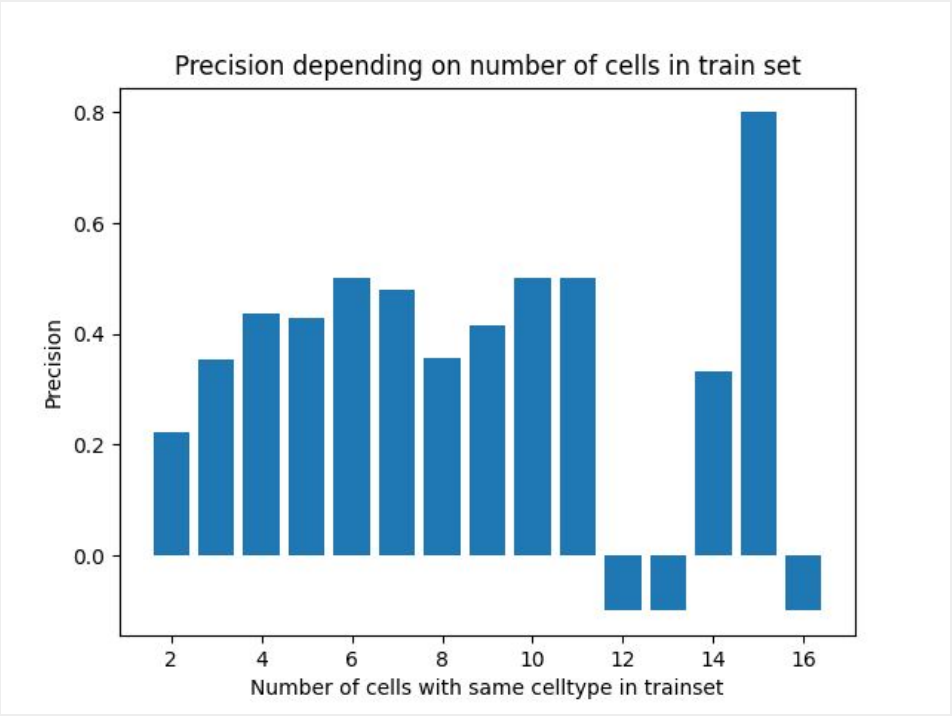
Modell 2

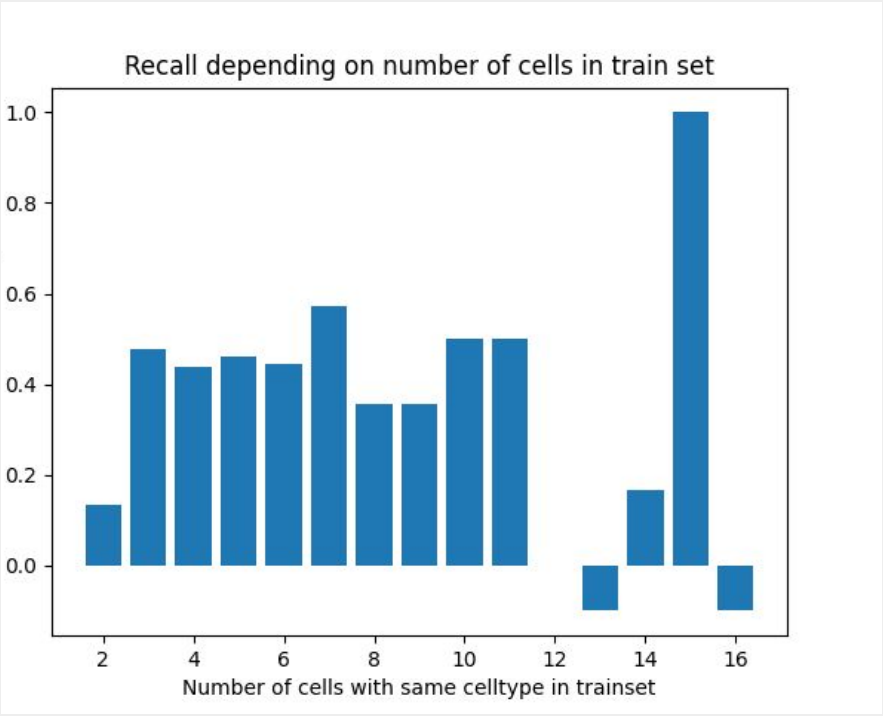

Precision depending on number of cells in train set
Precision
0.0
0.2
0.4
0.6
0.8
2
4
6
8
10
12
14
16
Number of cells with same celltype in trainset


Recall depending on number of cells in train set
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
12
14
16
Number of cells with same celltype in trainset

Multihead Attention

universität wien





Modellen-vergleich

universität wien

Modellname	SVM	Random Forest	Kneighbor Classifier	Multinomial Logistic Regression	MLP	Neural Networks
Test-genauigkeit	72 %	58 %	58.3 %	**75 %**	68.45 %	**75 %**

Mögliche Fortsetzung

- Semi-supervised learning
- Andere Dimension-Reduktionsmethoden
- Andere Resolutions für Clustering-Algorithmen
- Rohe Daten

Danke!

Quellenangabe

1 Le H, Peng B, Uy J, Carrillo D, Zhang Y, et al. (2022) Machine learning for cell type classification from single nucleus RNA sequencing data. PLOS ONE 17(9): e0275070. https://doi.org/10.1371/journal.pone.0275070

2 Wolf, F. A., Angerer, P., & Theis, F. J. (2018). SCANPY: large-scale single-cell gene expression data analysis. *Genome biology*, *19*(1), 15. https://doi.org/10.1186/s13059-017-1382-0

3 Tran, H.T.N., Ang, K.S., Chevrier, M. *et al.* A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biol* **21**, 12 (2020). https://doi.org/10.1186/s13059-019-1850-9

4 Cole, A. G., Hashimshony, T., Du, Z., & Yanai, I. (2023). *Gene regulatory patterning codes in early cell fate specification of the C. elegans embryo*. Cold Spring Harbor Laboratory. http://dx.doi.org/10.1101/2023.02.05.527193

5 Sulston, J. E., Schierenberg, E., White, J. G., & Thomson, J. N. (1983). The embryonic cell lineage of the nematode Caenorhabditis elegans. *Developmental biology*, *100*(1), 64–119. https://doi.org/10.1016/0012-1606(83)90201-4

6 McInnes, L., Healy, J., & Melville, J. (2018, February 9). *UMAP: Uniform manifold approximation and projection for dimension reduction*. arXiv.Org. https://arxiv.org/abs/1802.03426

7 Traag, V. A., Waltman, L., & van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, *9*(1), 1–12. https://doi.org/10.1038/s41598-019-41695-z

8 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2012, January 2). *Scikit-learn: Machine learning in python*. arXiv.Org. https://arxiv.org/abs/1201.0490

9 Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., … Zheng, X. (2016, May 27). TensorFlow: A system for large-scale machine learning. arXiv.Org. https://arxiv.org/abs/1605.08695

10 Ghojogh, B., & Ghodsi, A. (2020, December 20). *Attention mechanism, transformers, BERT, and GPT: Tutorial and survey*. Unknown. https://www.researchgate.net/publication/347623569_Attention_Mechanism_Transformers_BERT_and_GPT_Tutorial_and_Survey