UNIVERSITÄTS KLINIKUM FREIBURG

From dynamic modelling of disease trajectories to single-cell omics: State-of-the-art methods development in julia

Maren Hackenberg

Institute of Medical Biometry and Statistics, Faculty of Medicine and Medical Center – University of Freiburg, Germany

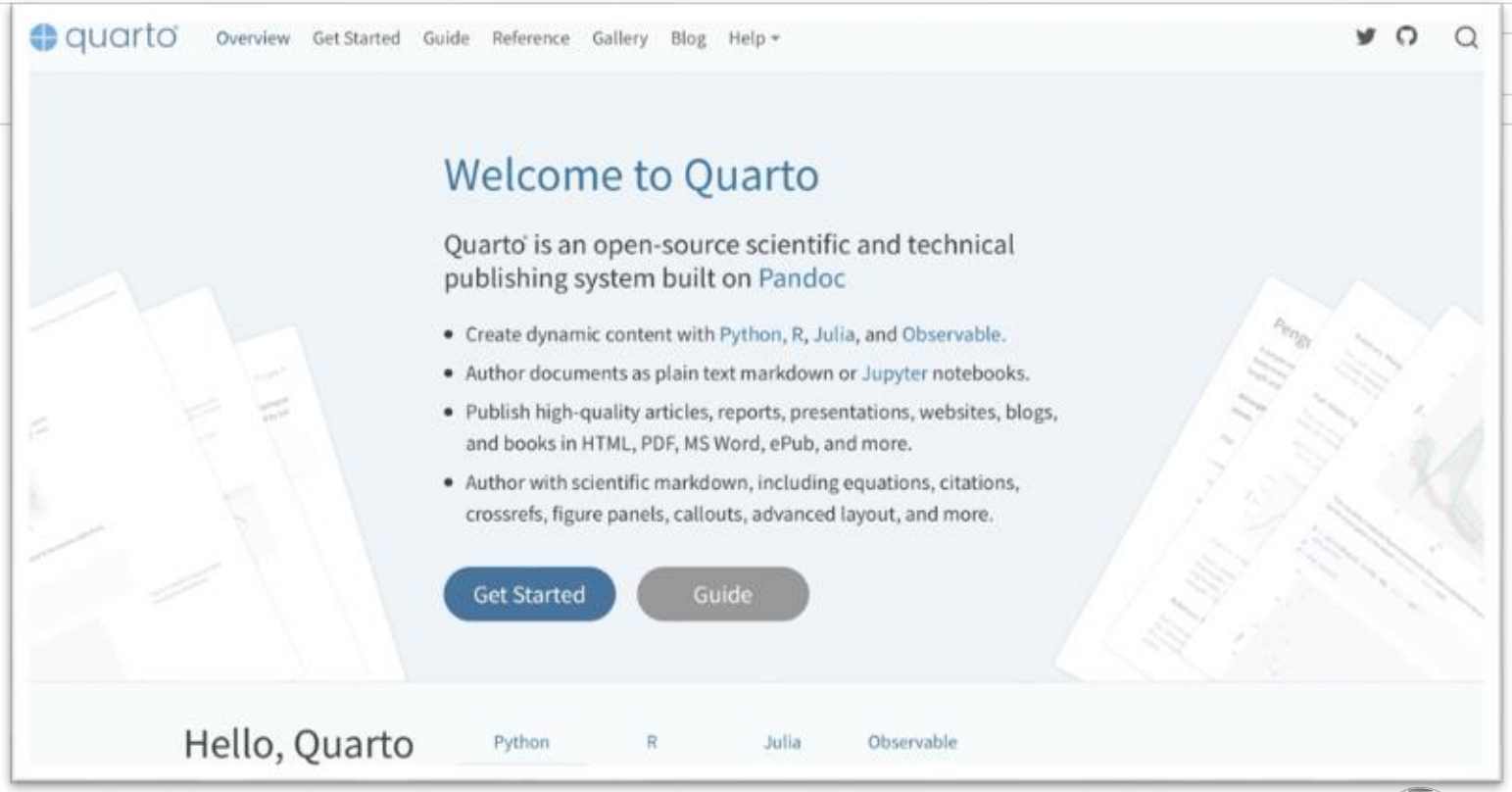
BMS-Aned Seminar – November 24, 2022

Jul 28 5 min

COMPANY NEWS AND EVENTS
OPEN SOURCE

Announcing Quarto, a new scientific and technical publishing system

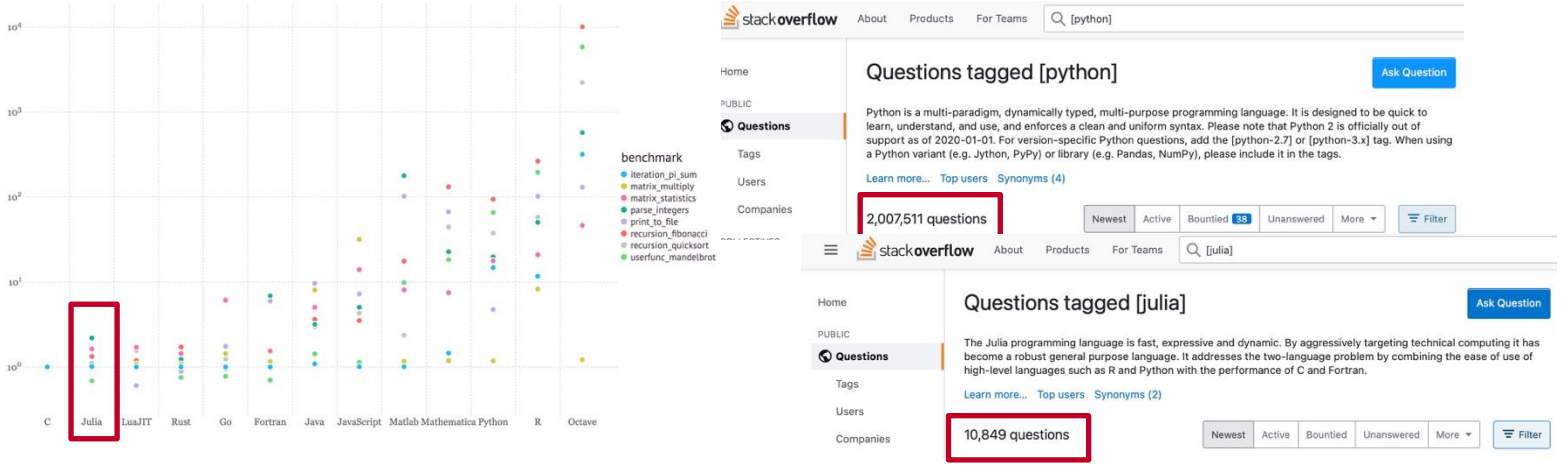
J.J. ALLAIRE


quarto
Overview
Get Started
Guide
Reference
Gallery
Blog
Help
Welcome to Quarto
Quarto is an open-source scientific and technical publishing system built on Pandoc
Create dynamic content with Python, R, Julia, and Observable.
Author documents as plain text markdown or Jupyter notebooks.
Publish high-quality articles, reports, presentations, websites, blogs, and books in HTML, PDF, MS Word, ePub, and more.
Author with scientific markdown, including equations, citations, crossrefs, figure panels, callouts, advanced layout, and more.
Get Started
Guide
Hello, Quarto
Python
R
Julia
Observable

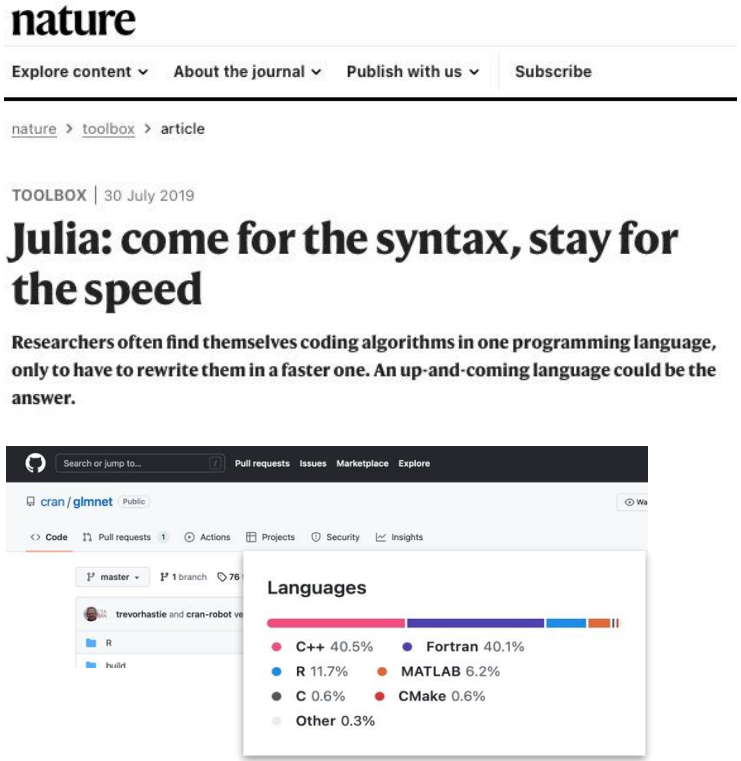
…so why Julia?

- Fast yet easy to learn with a syntax similar to R and Python
- Everything is pure Julia
- Easy interfaces to other languages

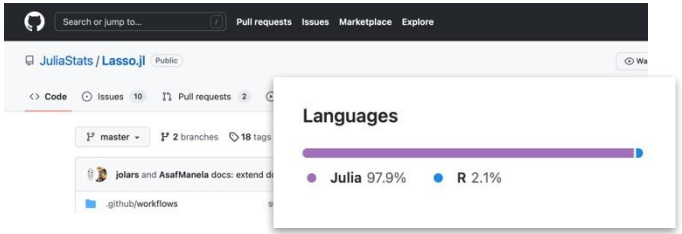
- Young language with a smaller user community
- Less training material
- Less mature package ecosystem



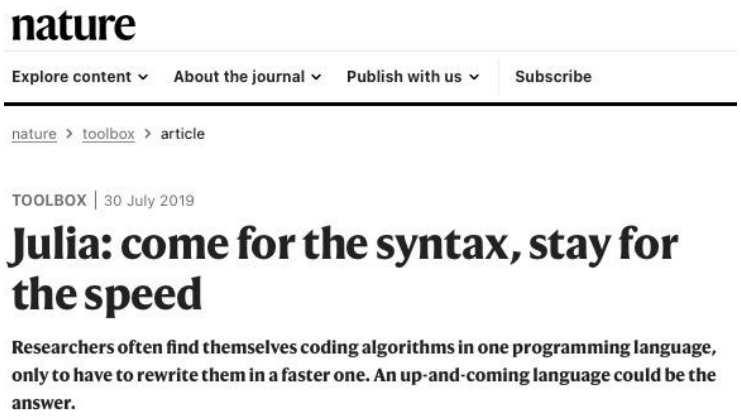
Julia doesn’t constrain your thinking



- Complex algorithms can be implemented in the language itself



Julia doesn’t constrain your thinking


nature

Explore content · About the journal · Publish with us · Subscribe

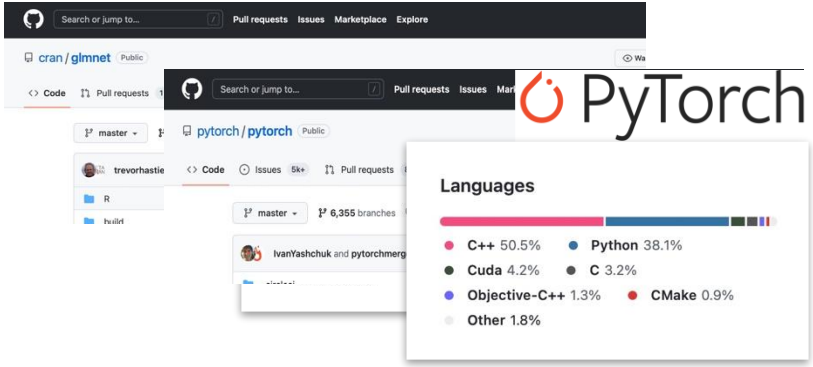
nature › toolbox › article

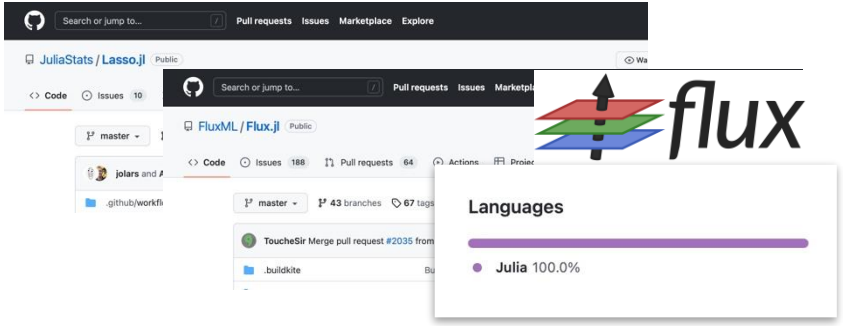
TOOLBOX | 30 July 2019

Julia: come for the syntax, stay for the speed

Researchers often find themselves coding algorithms in one programming language, only to have to rewrite them in a faster one. An up-and-coming language could be the answer.

- Complex algorithms can be implemented in the language itself
- Modification of existing libraries is easy, enabling faster prototyping and methods development
- No “legacy characteristics”





UNIVERSITÄTS KLINIKUM FREIBURG

Exemplary modelling challenge: Disease trajectories of patients with spinal muscular atrophy

SMARTCARE

Baseline characterisation

- age
- SMA subtype
- …

Different motor function tests over time

- RULM
- HFMSE
- …

Latent health status

Explicit model

$\frac{d}{dt}\mu(t)=?$

Subgroup-specific local models

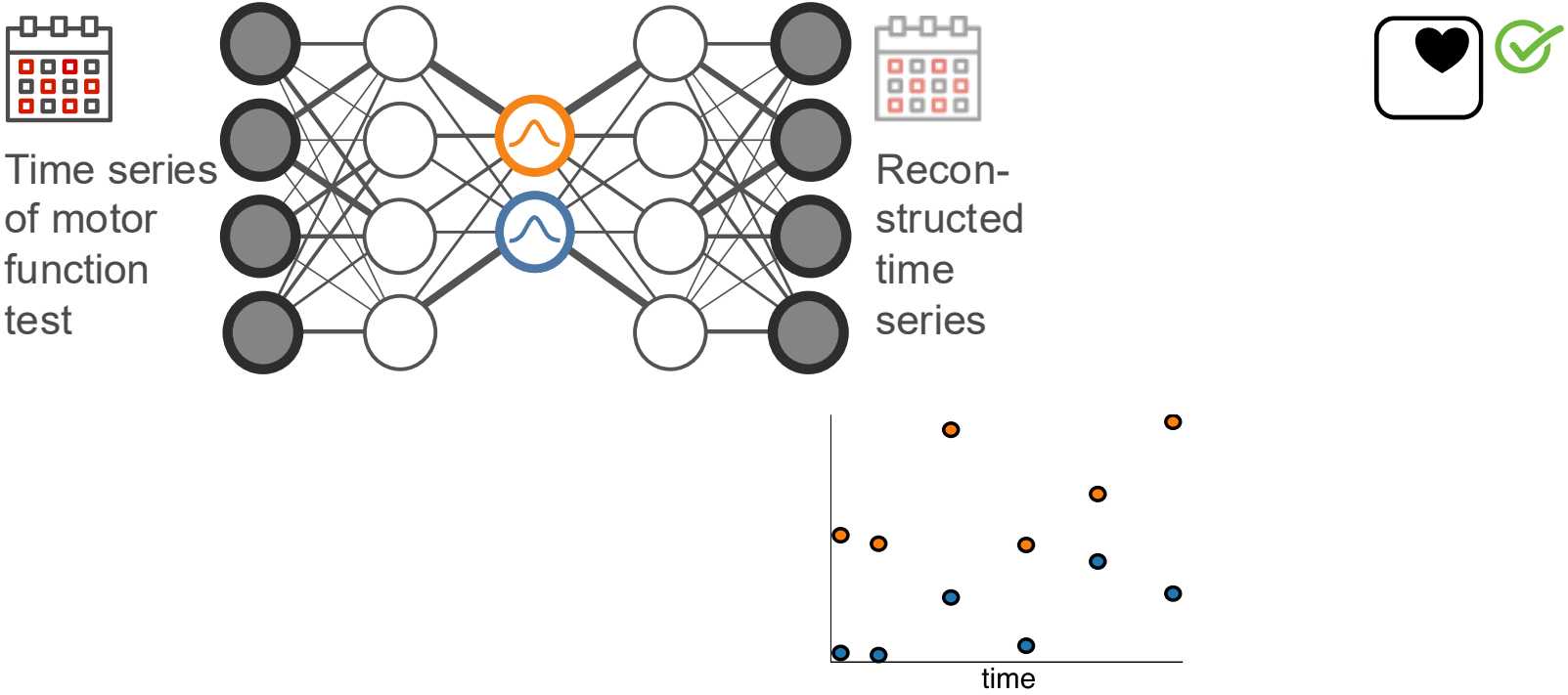
Hetero-geneity

Irregular time points

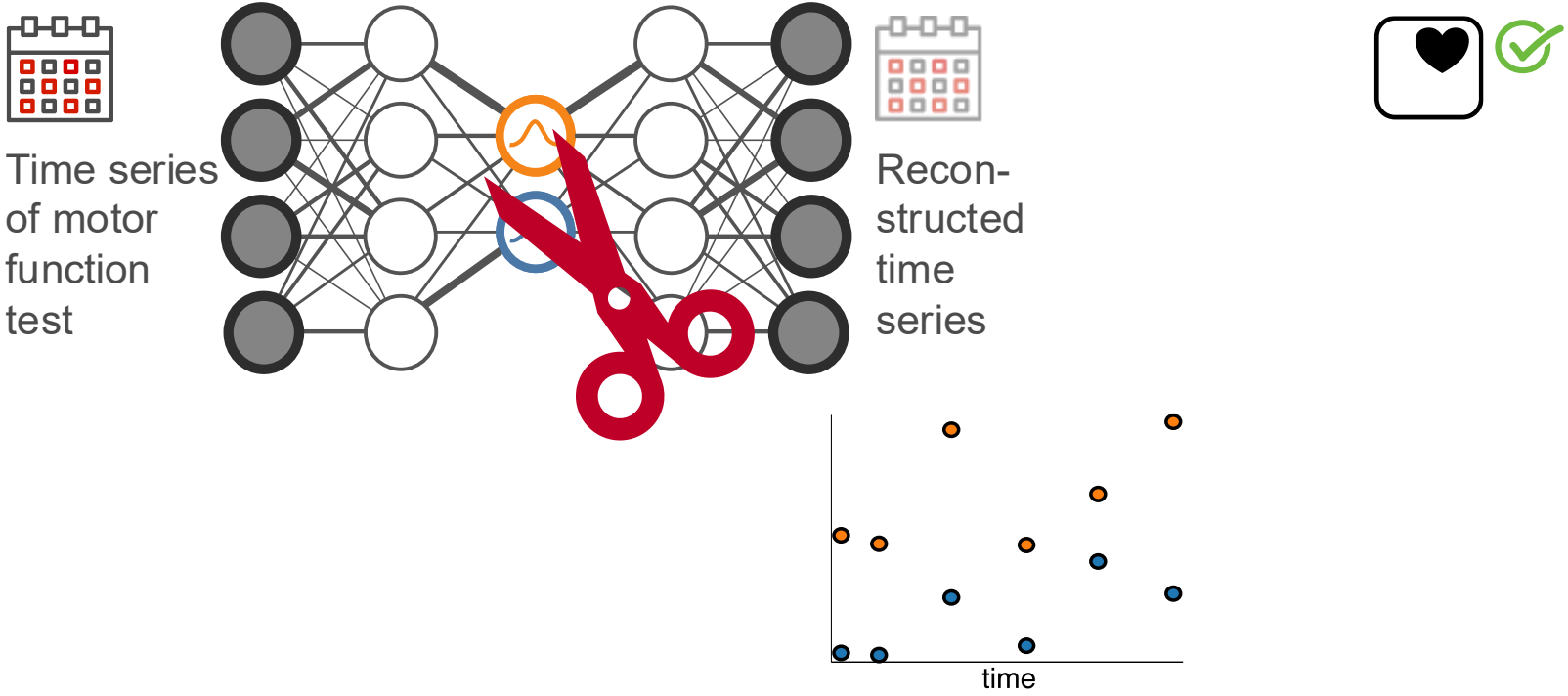
RULM
HFMSE

Different motor function tests

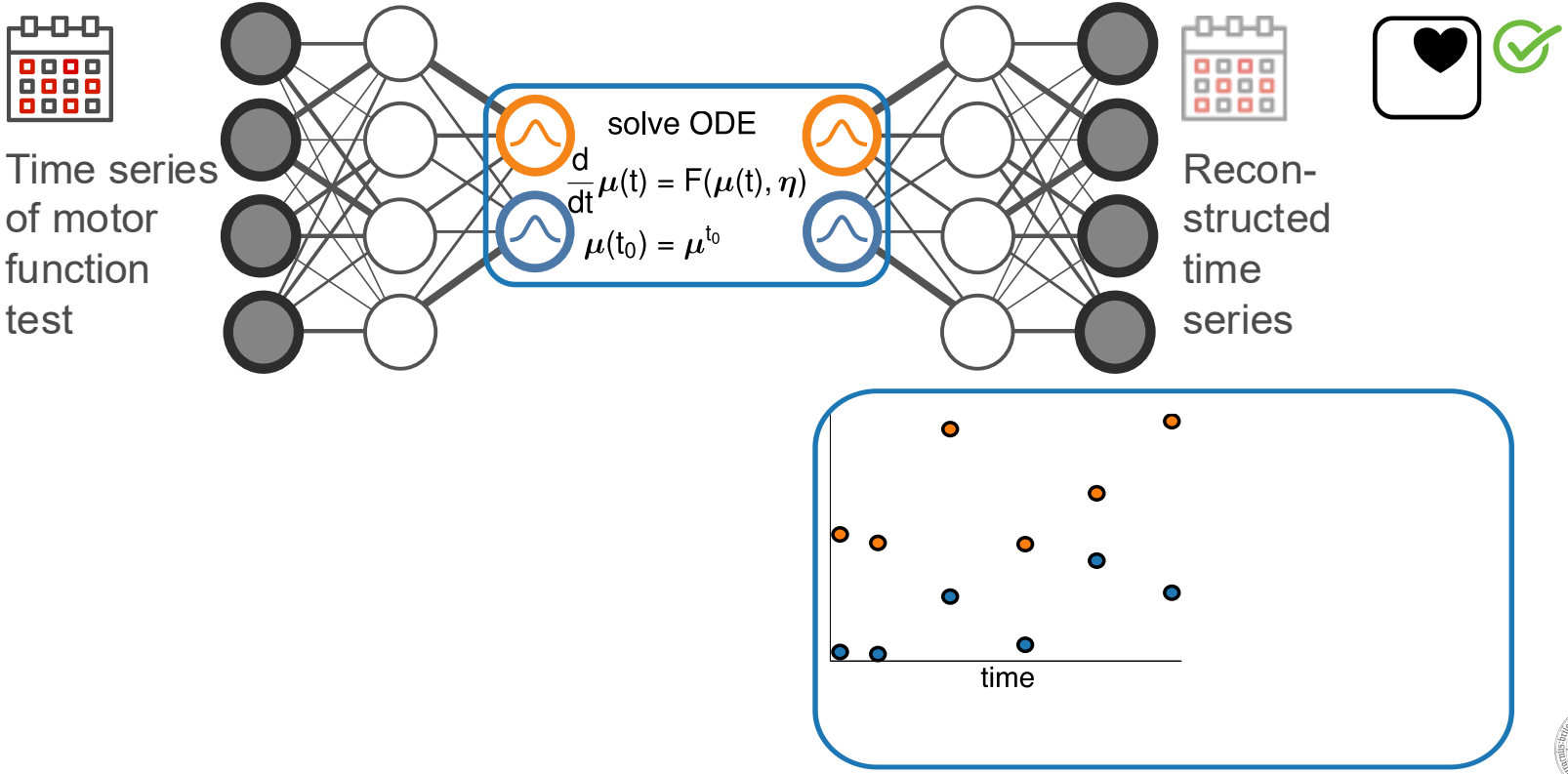
Describe individual SMA trajectories as ODEs in the latent space of a deep learning model



Describe individual SMA trajectories as ODEs in the latent space of a deep learning model



Describe individual SMA trajectories as ODEs in the latent space of a deep learning model



Differentiable programming for flexible model building

Machine Learning

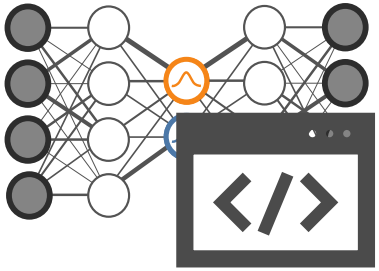
Scientific Computing

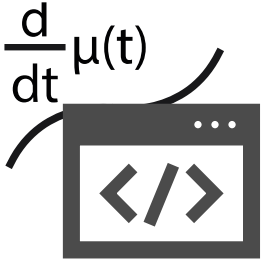
TensorFlow

PyTorch

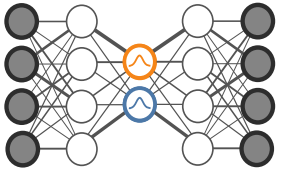
Zygote

write as a computer program

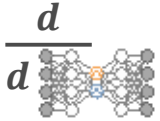




loss(



, $\frac{d}{dt}\mu(t)$)

$\frac{d}{d}$  loss

$\frac{d}{d\,\frac{d}{dt}\mu(t)}$ loss

optimise via automatic differentiation

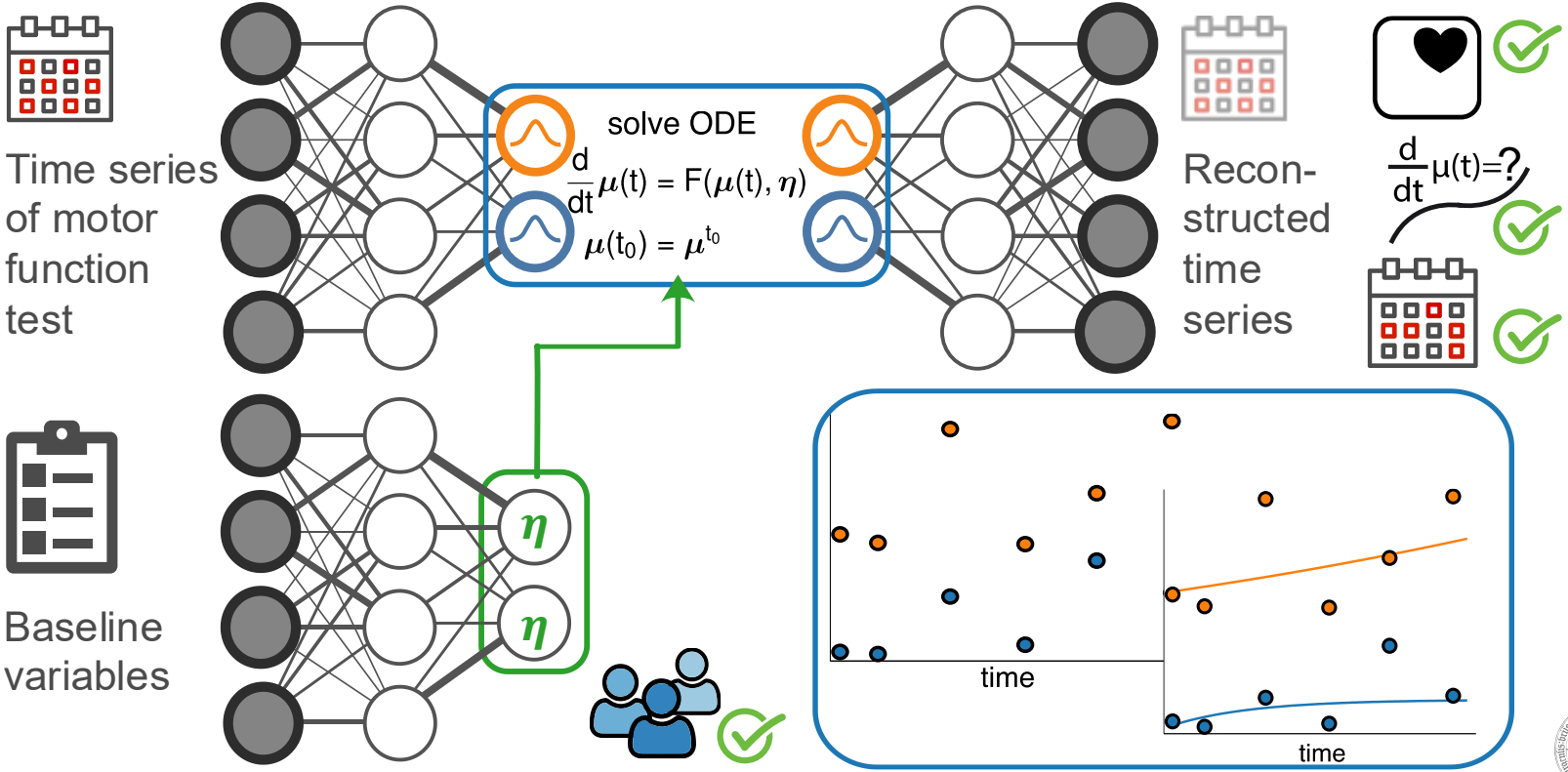
UNIVERSITÄTS KLINIKUM FREIBURG

Engineering trade-offs in automatic differentation tools

- Graph-building system in static sub-language: efficient but less flexible
- Tape-based method: unrolls all control flow
- Overhead partly compensated by highly efficient kernels for machine learning applications

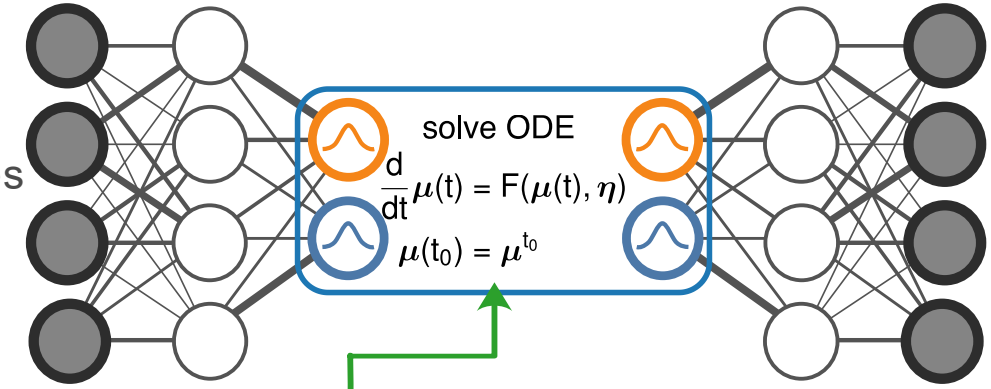
- Source-to-source transformation
- Supports full dynamism
  - E.g., straightforward differentiation through ODE solvers
  - Supports user-defined structs and recursion
- Keeps control flow intact

Describe individual SMA trajectories as ODEs in the latent space of a deep learning model



Describe individual SMA trajectories as ODEs in the latent space of a deep learning model

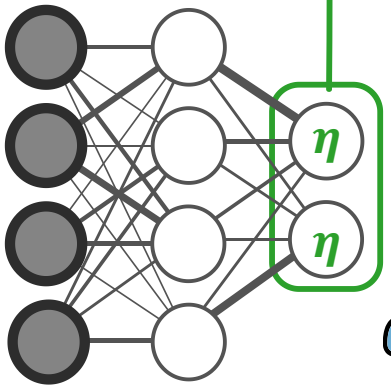
Time series of motor function test

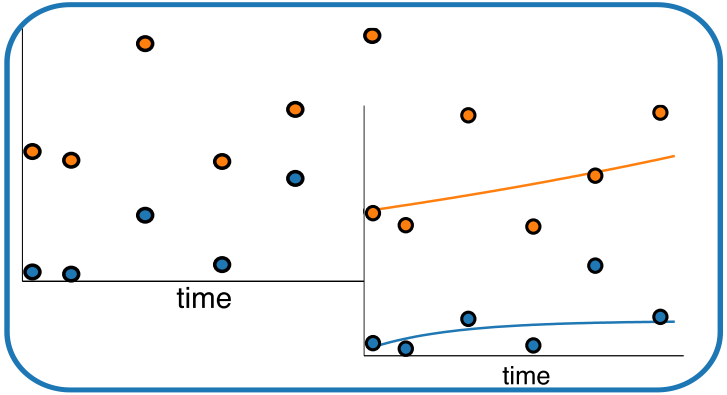


Recon-structed time series

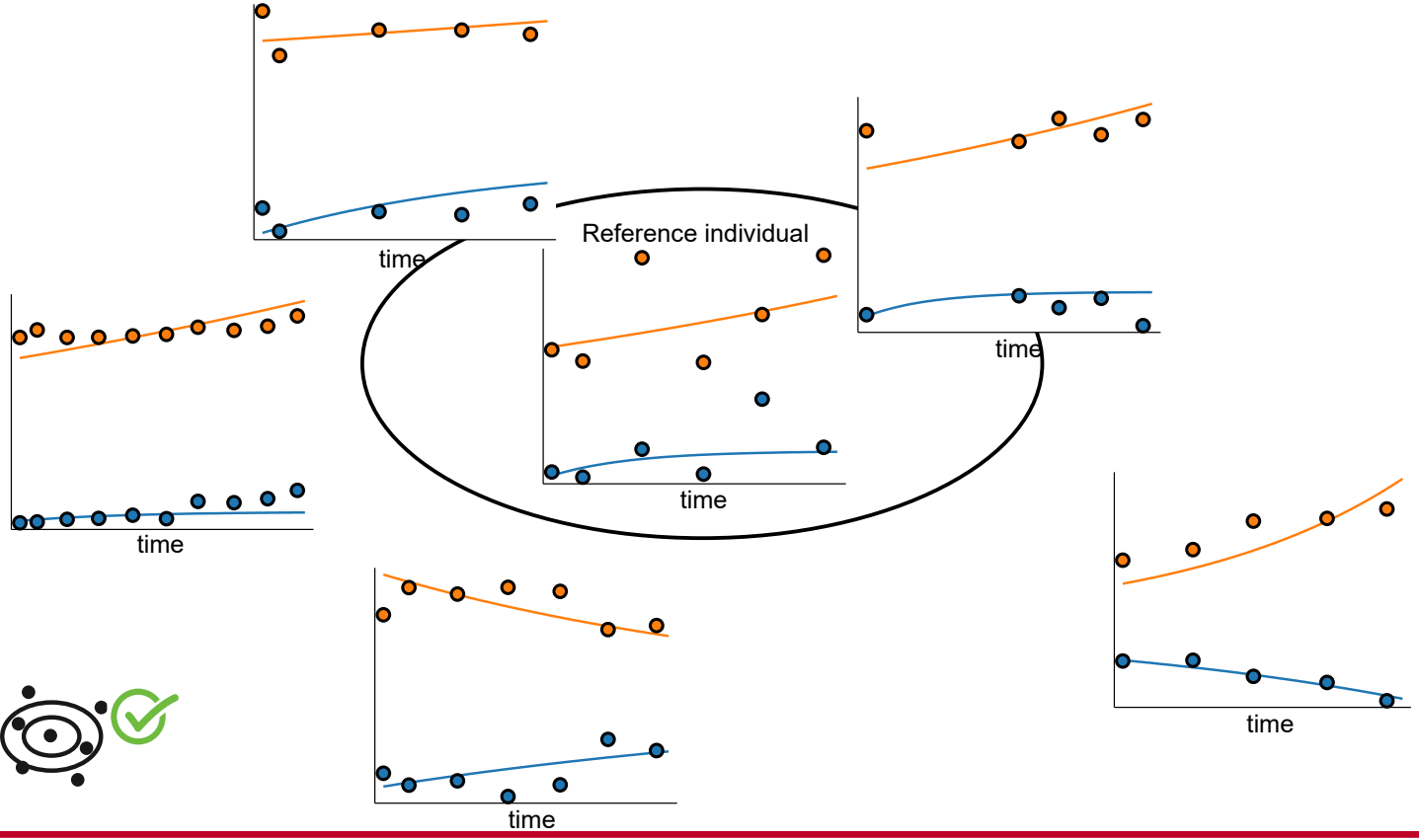
```
m = init_ODEVAE()
ps = getparams(m)
opt = ADAM(η)
trainingdata = zip(xs, xs_baseline, tvals)

for epoch in 1:epochs
    for (X, Y, t) in trainingdata
        grads = gradient(ps) do
            loss(X, Y, t, m, args=args)
        end
        update!(opt, ps, grads)
    end
end
```

Baseline variables

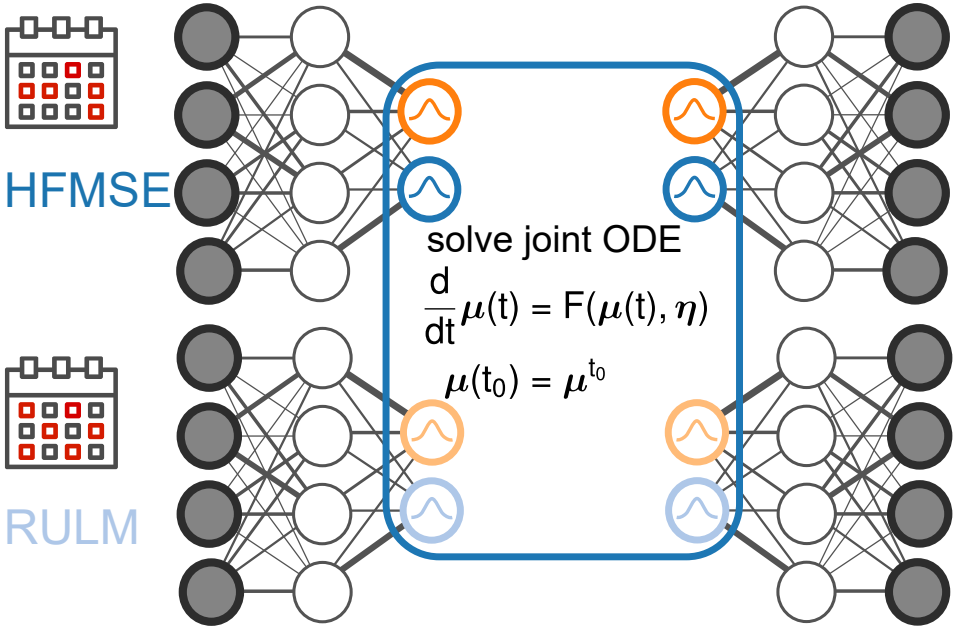


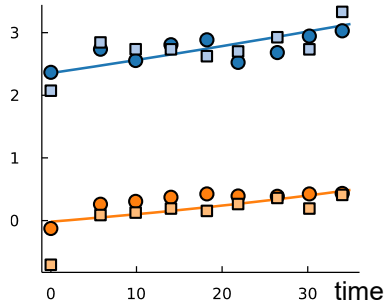


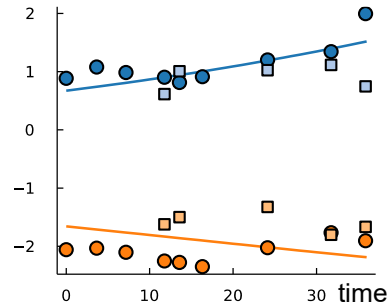
Jointly model groups of similar individuals

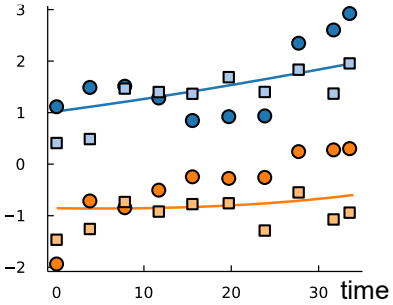


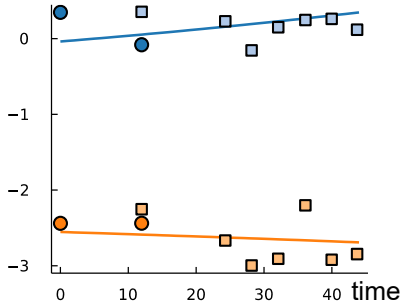
Integrating different motor function tests









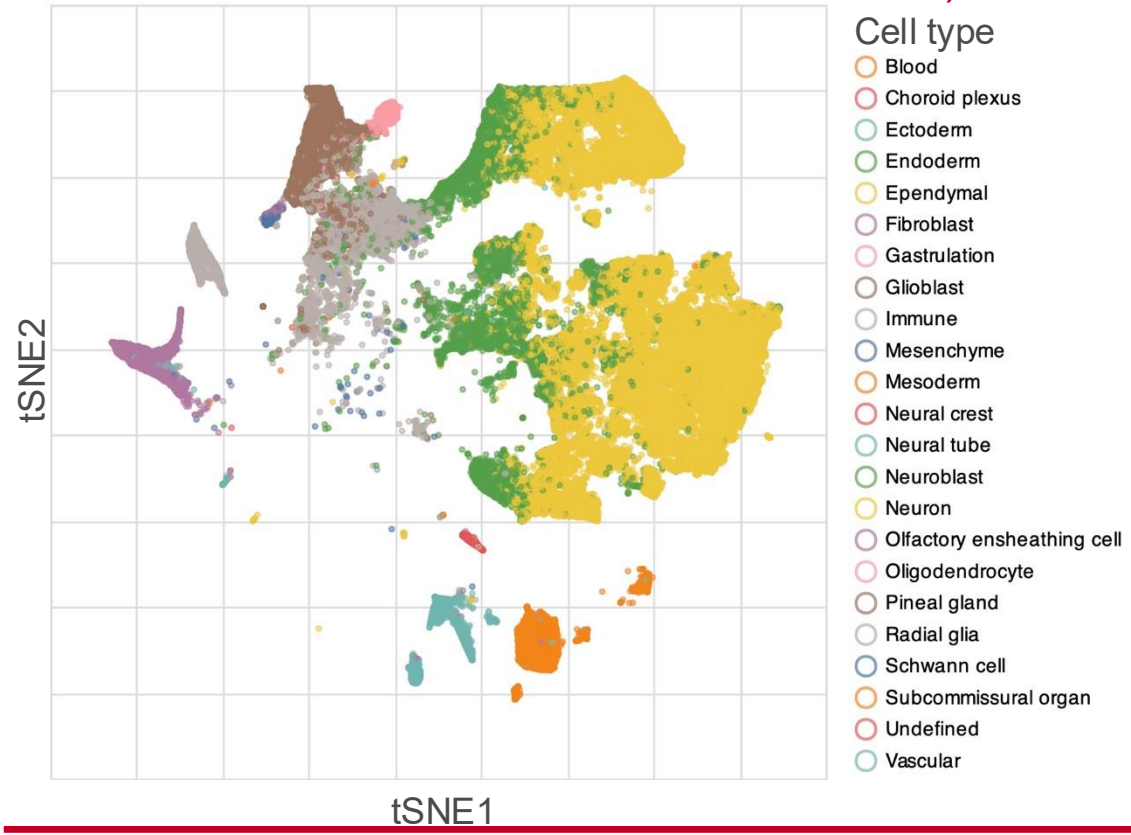


HFMSE measurements

RULM measurements

How do you typically look at single-cell RNA-seq data?

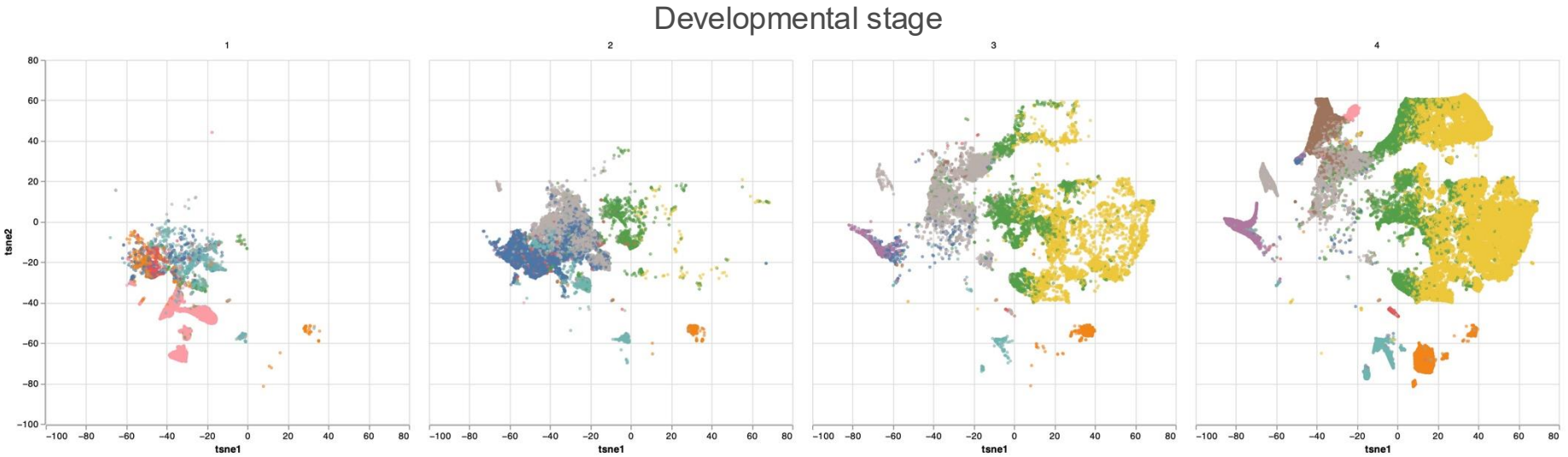
t-stochastic neighborhood embedding (tSNE) for dimension reduction



Dataset from La Manno *et al.* Molecular architecture of the developing mouse brain, *Nature* 2021.

How do you typically look at single-cell RNA-seq data?

t-stochastic neighborhood embedding (tSNE) for dimension reduction

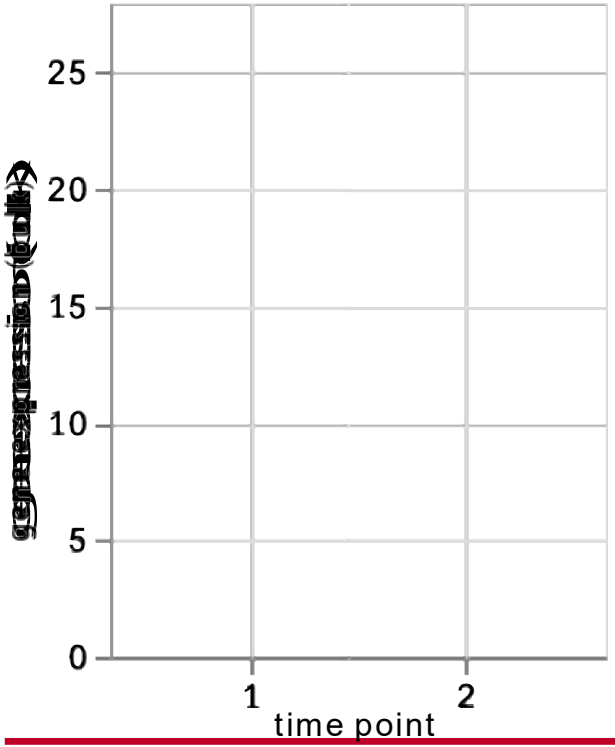


Dataset from La Manno *et al*. Molecular architecture of the developing mouse brain, *Nature* 2021.

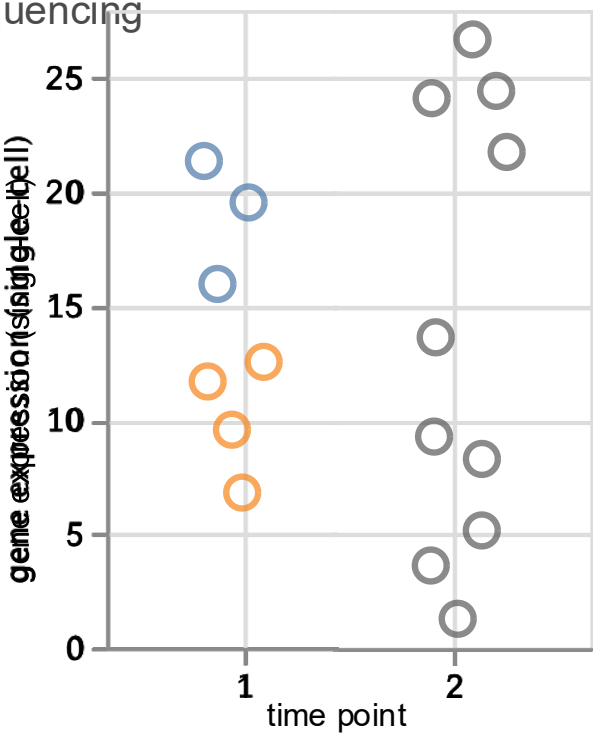
Time-series data: promising but challenging!

How can we infer developmental trajectories of distinct subgroups of cells?

Classical time-series scenario



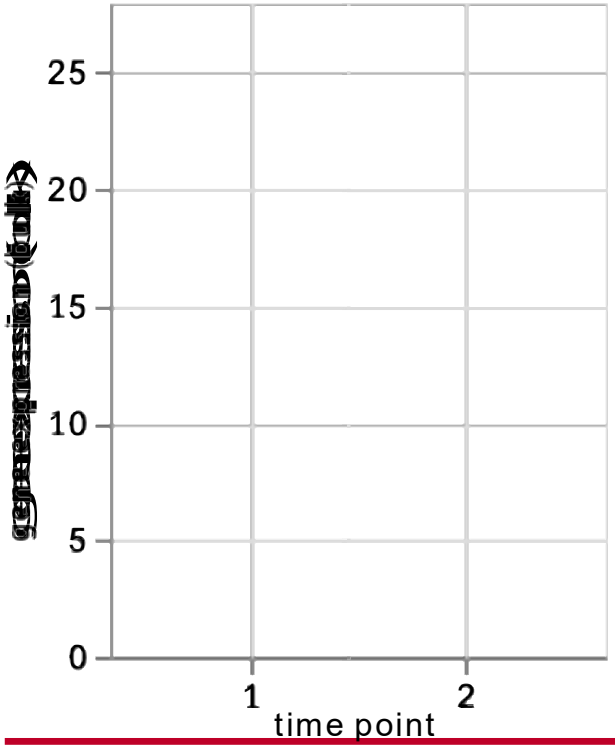
…and the scenario in single-cell sequencing



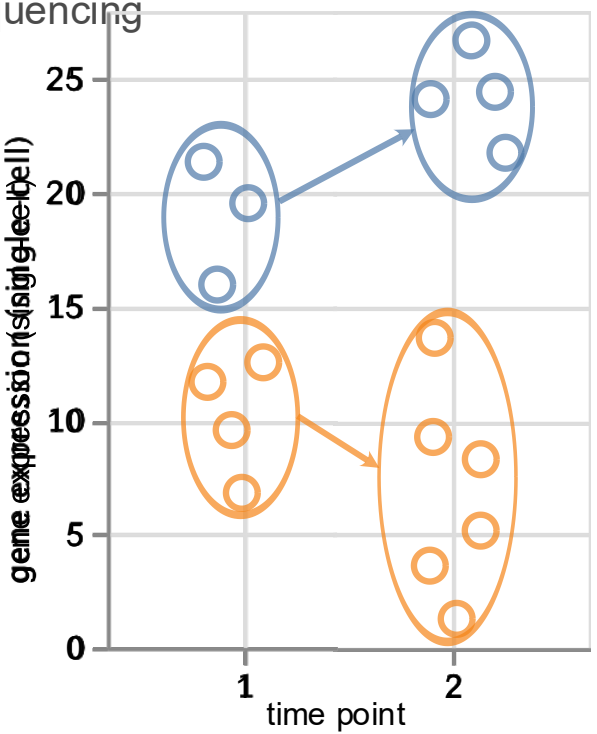
Time-series data: promising but challenging!

How can we infer developmental trajectories of distinct subgroups of cells?

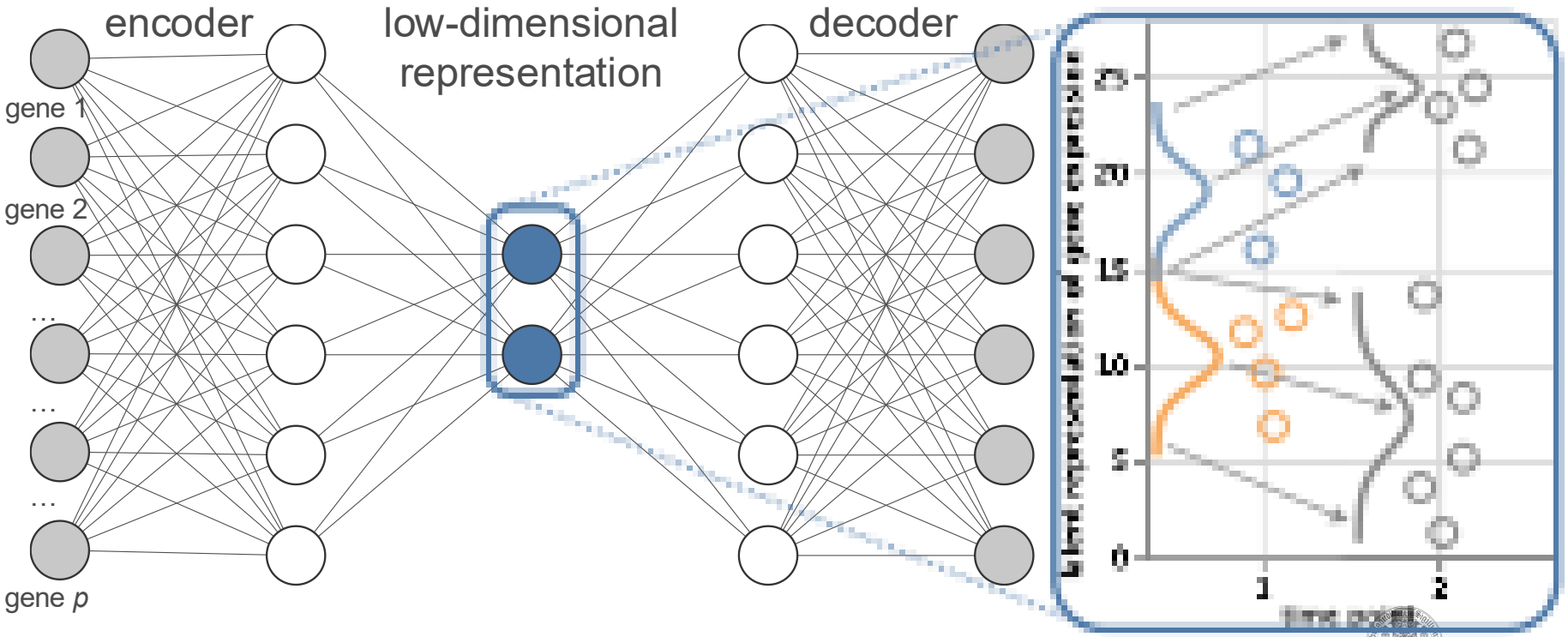
Classical time-series scenario



…and the scenario in single-cell sequencing



Combining VAEs with dynamic modeling for time-series single-cell RNA-sequencing



julia for generative models for single-cell analysis

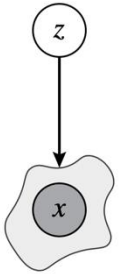
scvi-tools

Probabilistic models for single-cell omics data

scvi-tools accelerates data analysis and model development, powered by PyTorch and AnnData.

```
pip install scvi-tools
```

Get Started



End-to-end analysis

Dimensionality reduction, dataset integration, differential expression, automated annotation. scvi-tools contains models that perform a wide variety of tasks across many omics, all while accounting for the statistical properties of the data.

Easy-to-use implementations

Each model (e.g., scVI, scANVI, Stereoscope, totalVI) follows the same user interface that couples nicely with Scanpy, Seurat, or Bioconductor workflows. No more searching through source code.

julia for generative models for single-cell analysis

scVI

Search docs

Getting started

- Overview
- Installation
- Contents

Data processing

The scVAE model

The scLDVAE model

Model training

Model evaluation

Getting started

Edit on GitHub

scVI.jl

A Julia package for fitting VAEs to single-cell data using count distributions. Based on the Python implementation in the `scvi-tools` package.

Overview

https://github.com/maren-ha/scVI.jl

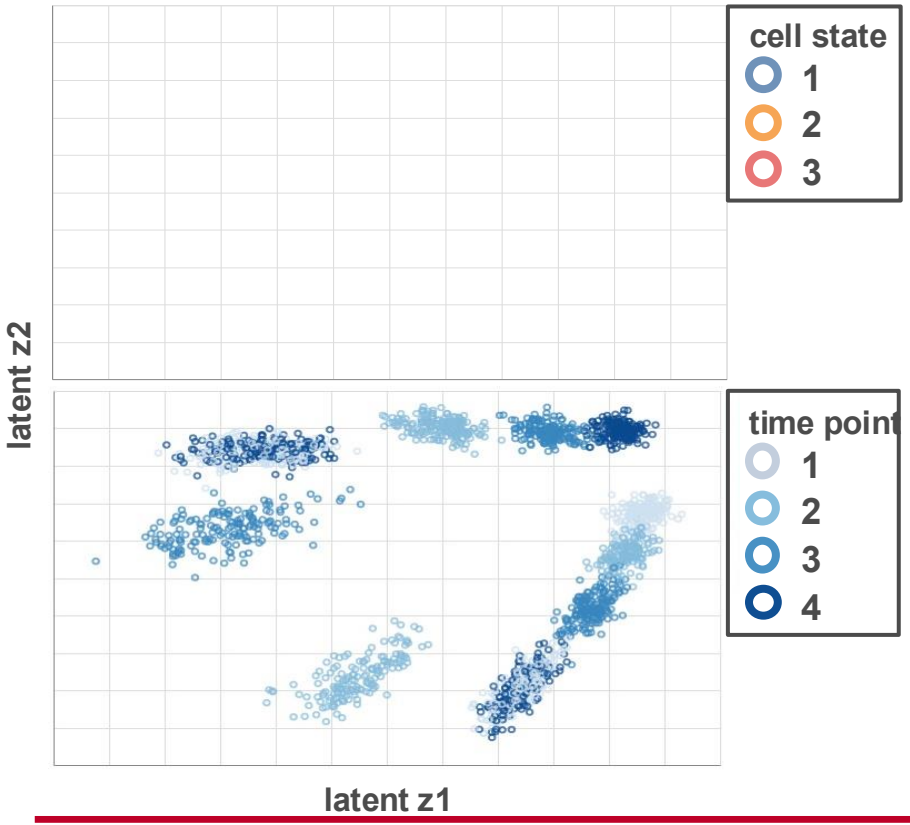
The scVI model was first proposed in Lopez R, Regier J, Cole MB *et al.* Deep generative modeling for single-cell transcriptomics. *Nat Methods* **15**, 1053-1058 (2018).

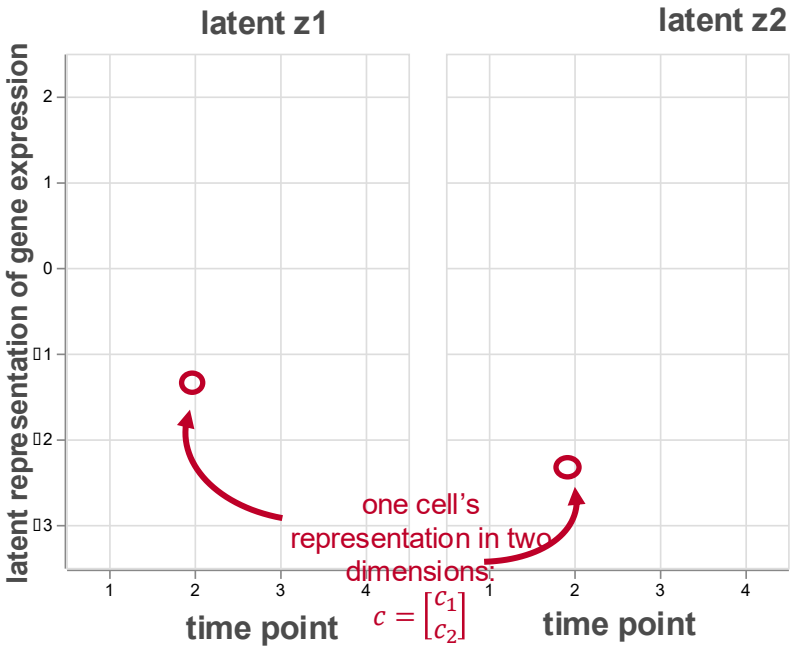
More on the much more extensive Python package ecosystem `scvi-tools` can be found on the website and in the corresponding paper Gayoso A, Lopez R, Xing G. *et al.* A Python library for probabilistic analysis of single-cell omics data. *Nat Biotechnol* **40**, 163–166 (2022).

UNIVERSITÄTS KLINIKUM FREIBURG

Simulation study results

Count data simulated with the splatter R package

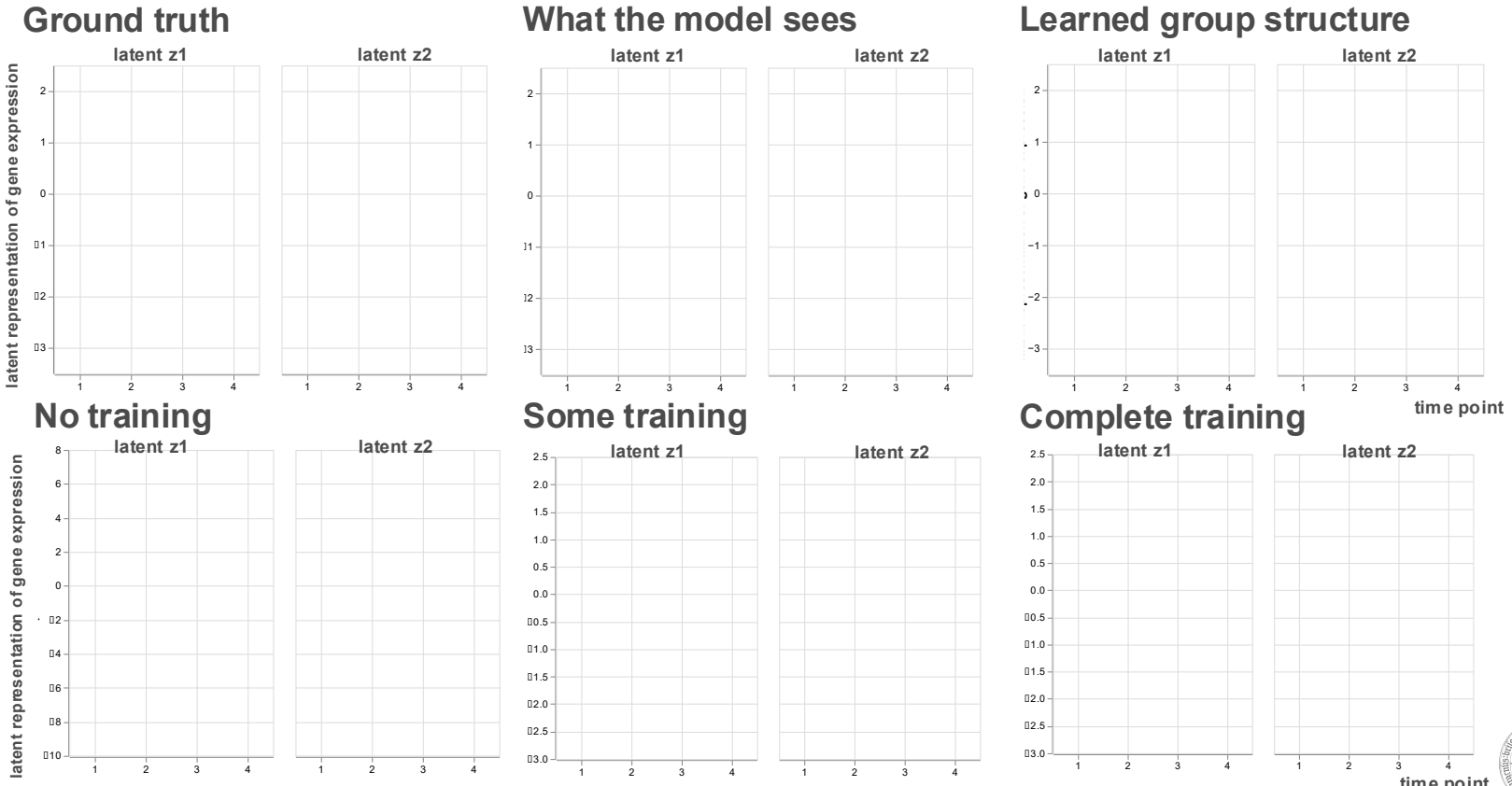




Each cell (circle) appears twice: dimension 1 of its low-dimensional representation in the left panel, dimension 2 in the right panel

Simulation study results

The dynamic model simultaneously infers underlying trajectories and cell states



VAE representation of original data

Model predictions

Julia – a pragmatic approach to scientific computing

Take home messages

- Fresh approach without legacy characteristics
- Fast yet easy to use
- Removes barriers that so far limit scientific ideas, e.g., by providing straightforward access to powerful automatic differentiation libraries
- Key tool for current trends such as differentiable programming:
  - Combining deep learning and dynamic modelling for individual disease trajectories in a rare disease registry…
  - … and for developmental trajectories in time-series single-cell omics

➢ Add Julia to your toolbox to expand the universe of modelling approaches available to your research!

Thanks to…

Special thanks to:

Harald Binder

and the AG Machine Learning

Colleagues at the IMBI:

Michelle Pfaffenlehner

Laia Canal

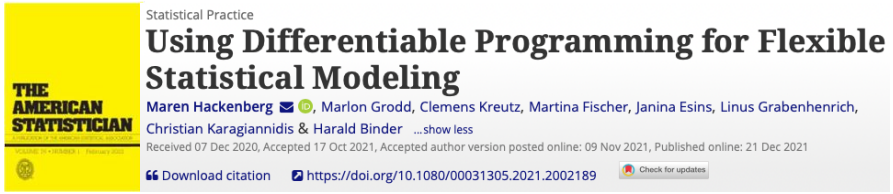
… **at the Institute of Mathematics:**
Philipp Harms, Thorsten Schmidt

… **and from the SMArtCARE registry:**
Astrid Pechmann, Janbernd Kirschner

And thanks to you for listening!

If you want to know more:

maren@imbi.uni-freiburg.de



RESEARCH ARTICLE | Open Access

Deep dynamic modeling with just two time points: Can we still allow for individual trajectories?

Maren Hackenberg, Philipp Harms, Michelle Pfaffenlehner, Astrid Pechmann, Janbernd Kirschner, Thorsten Schmidt, Harald Binder

First published: 06 April 2022 | https://doi.org/10.1002/bimj.202000366

https://doi.org/10.1080/00031305.2021.2002189

https://github.com/maren-ha/DifferentiableProgrammingForStatisticalModeling

https://doi.org/10.1002/bimj.202000366

https://github.com/maren-ha/DeepDynamicModelingWithJust2TimePoints