Designing and fitting neural ODEs

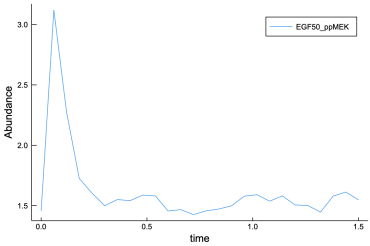
Background and preliminary results

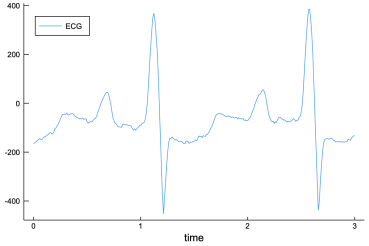
Elisabeth Rösch

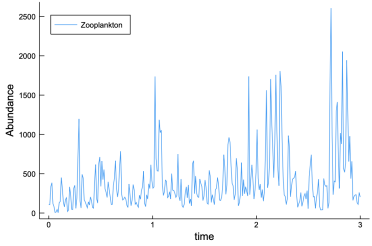
University of Melbourne

9 July 2019

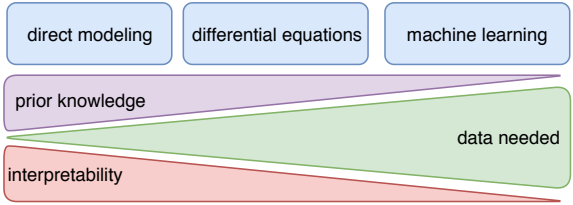
Example


EGF50_ppMEK
Abundance
time
3.0
2.5
2.0
1.5
0.0
0.5
1.0
1.5


ECG
time
400
200
0
-200
-400
0
1
2
3


Zooplankton
Abundance
time
2500
2000
1500
1000
500
0
0
1
2
3

Motivation


direct modeling
differential equations
machine learning
prior knowledge
data needed
interpretability

Ordinary Differential Equation (ODE)

$$\frac{\delta u}{\delta t} = f(u)$$

u: Species, t: Time, f: Function

Neural ODE [Chen et al., 2018]

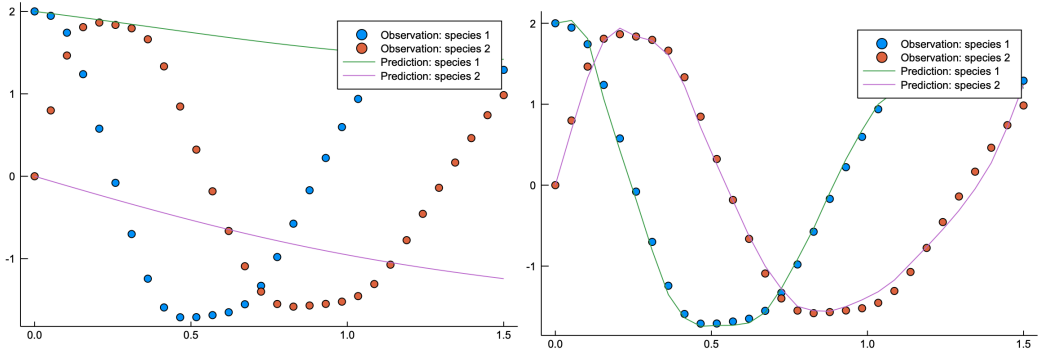
$$\frac{\delta u}{\delta t} = f(u)$$

u: Species, t: Time, f: **Neural net**

In Julia: DiffEqFlux.jl [Rackauckas et al., 2019]

```
# Derivative is modeled by a neural net.
dudt = Chain(x -> x.^3,
       Dense(2,50,tanh),
       Dense(50,2))
# Parameters of the model which are to be learnt.
ps = Flux.params(dudt)
```

Fitting neural ODEs: Optimize loss functions

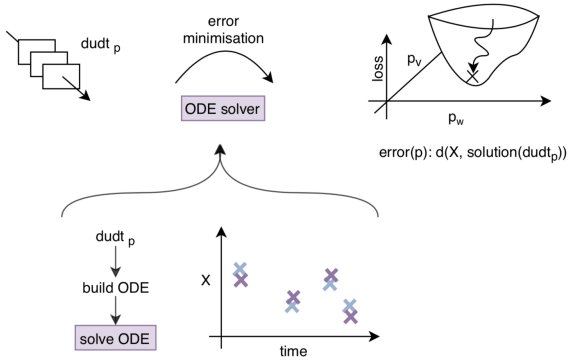


Before and after training: Observed and predicted species over time

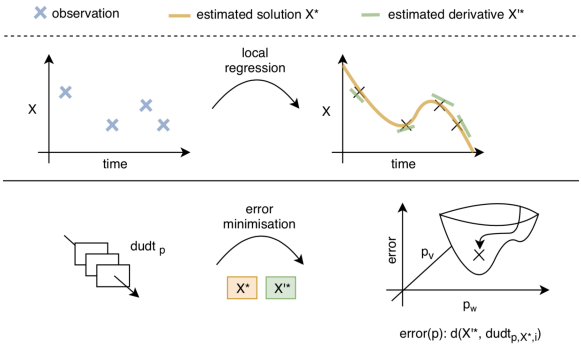
Loss functions

1. L2
2. Collocation based
3. Mixtures

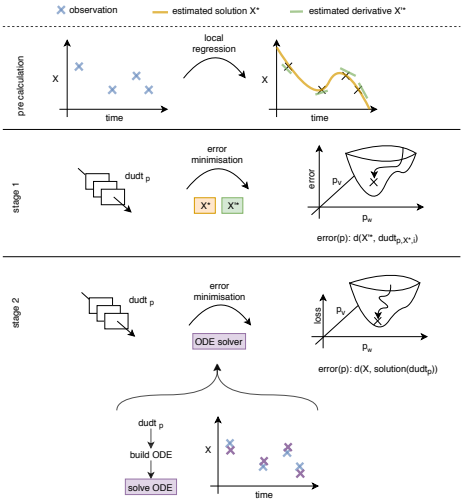
Loss function: L2



Loss function: Collocation based [Liang and Wu, 2008]

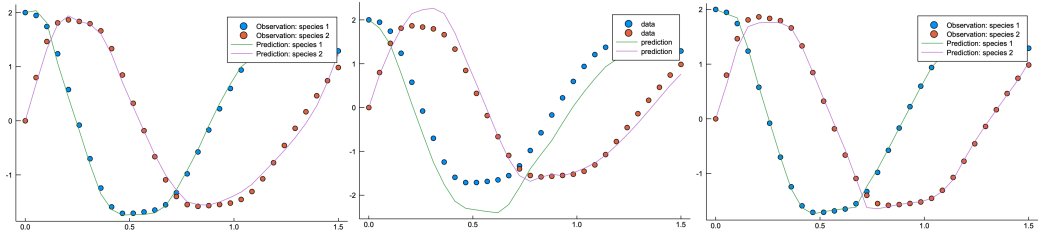


Loss function: Mixture



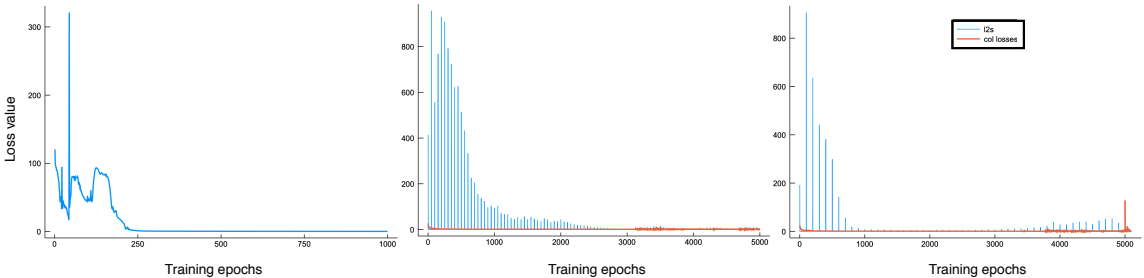
Performance: Accuracy

a. L2 norm as loss function b. Collocation as loss c. Mixture loss function

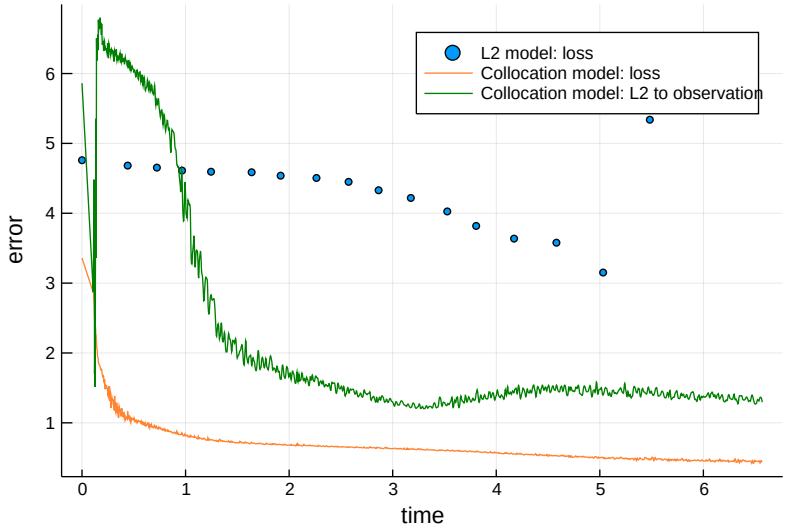


Performance: Convergence

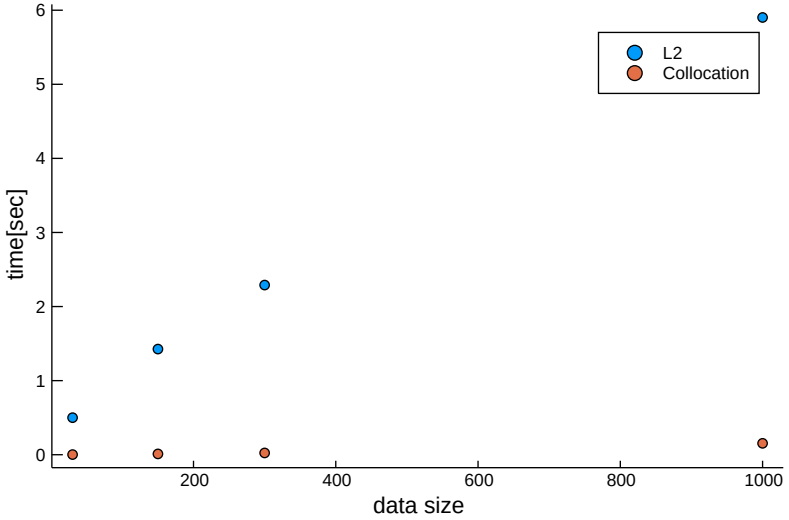
a. L2 norm as loss function b. Collocation as loss c. Mixture loss function



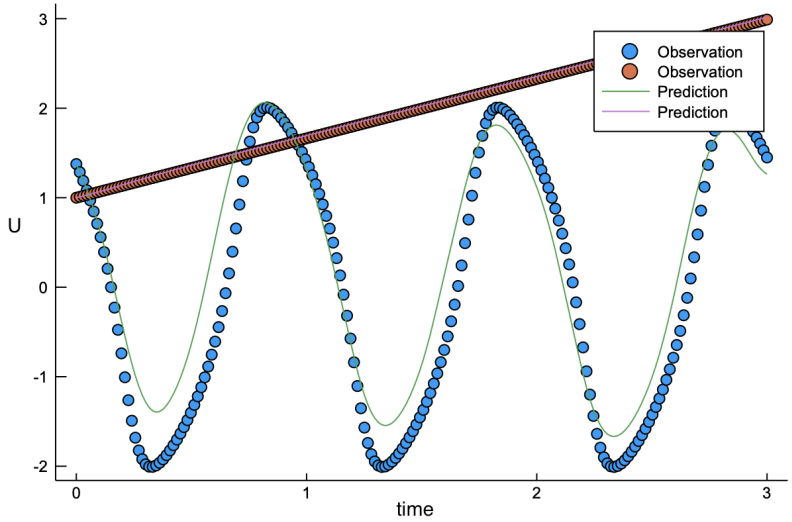
Performance: Time



Effect on performance: Data size



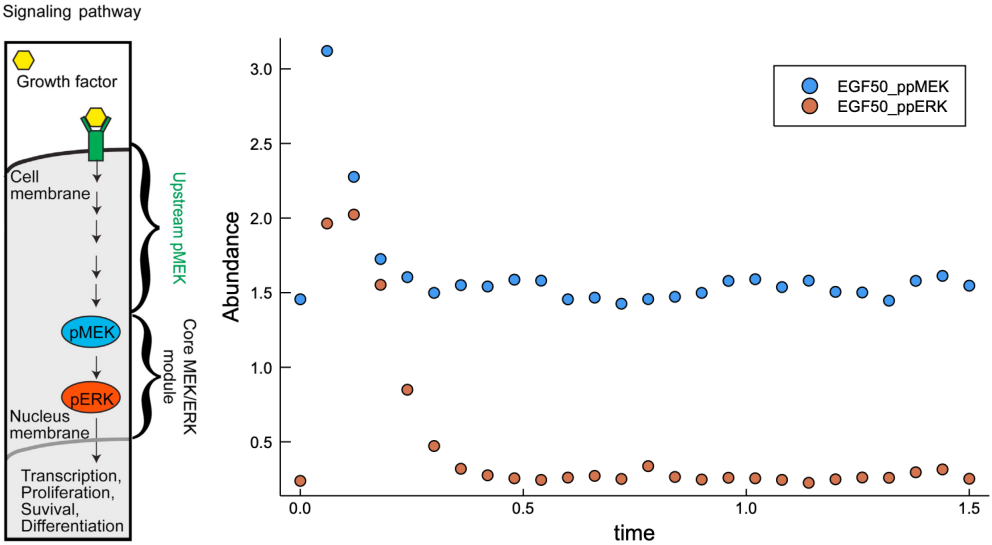
Biological application: Van der Pol Oscillator



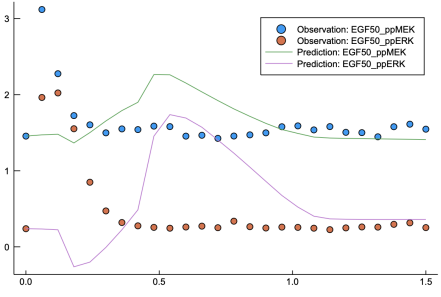
Biological application: FIND EASY

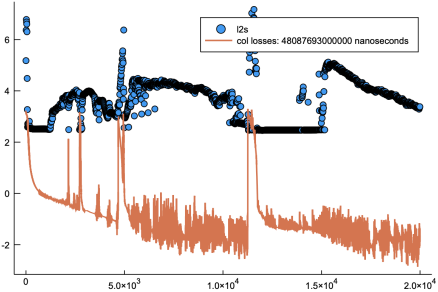
to add

Biological application: MEK-ERK dynamics [Filippi et al., 2016]



Biological application: Collocation based model

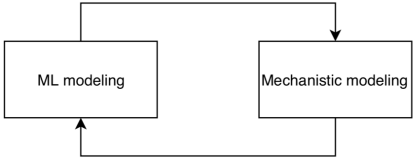




Biological application: Collocation based model with multiple shooting

to add

Outlook: Hybrid modeling



Acknowledgements

Supervision and mentoring:

- Michael Stumpf, Heejung Shim, Chris Rackauckas

Support and discussions:

- **Julia community**: Differential Equations/ Machine Learning – Chris Rackauckas, Lyndon White.
- **TheoSysBio group Melbourne and London**: Lucy Ham, Anissa Guillemin, Megan Anne Coomer, Leo Diaz, David Schnoerr, Rowan Brackston, Ivan Croydon Veleslavov, Leanne Massie, Tom Layshon.

Code:

- `https://github.com/LislPisl/neural_ODE_fitting`

Bibliography


Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018).
Neural ordinary differential equations.
CoRR.


Filippi, S. et al. (2016).
Robustness of MEK-ERK Dynamics and Origins of Cell-to-Cell Variability in MAPK Signaling.
Cell Reports, 15(11):2524–2535.


Liang, H. and Wu, H. (2008).
Parameter estimation for differential equation models using a framework of measurement error in regression models.
Journal of the American Statistical Association.


Rackauckas, C., Innes, M., Ma, Y., Bettencourt, J., White, L., and Dixit, V. (2019).
Diffeqflux.jl - a julia library for neural differential equations.
CoRR.