Startup meeting master project

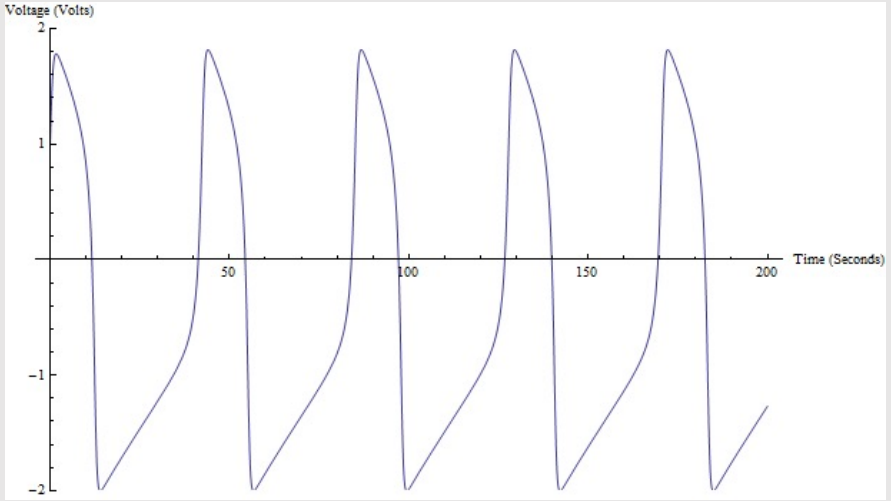
Characterization of cardiac cellular dynamics using Physics-Informed Neural Networks

03.09.21

simula

There exist numerous models for heart cells

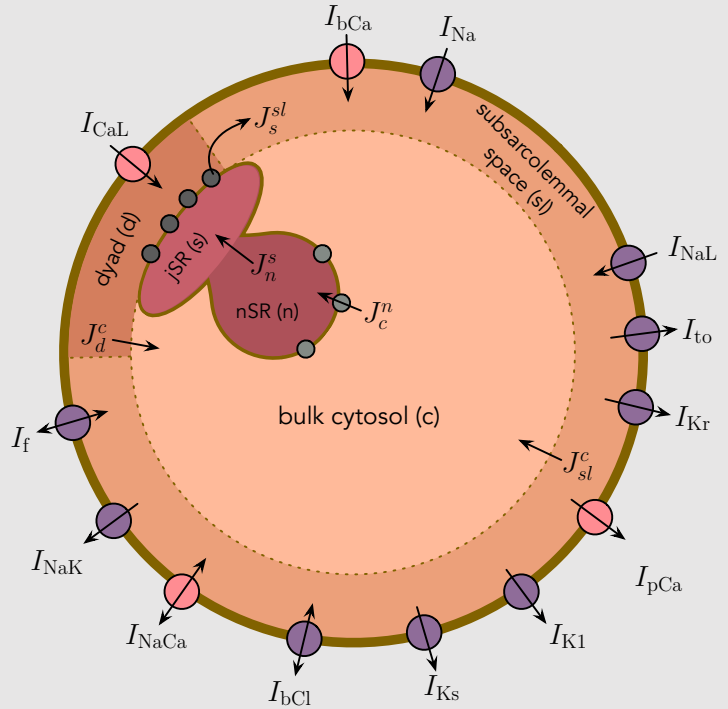
Fitzhugh-Nagumo (simple)



$$\begin{cases} \frac{dv}{dt} = v - v^3 - w + I_{\text{ext}} \\ \tau \frac{dw}{dt} = v - a - bw \end{cases}$$

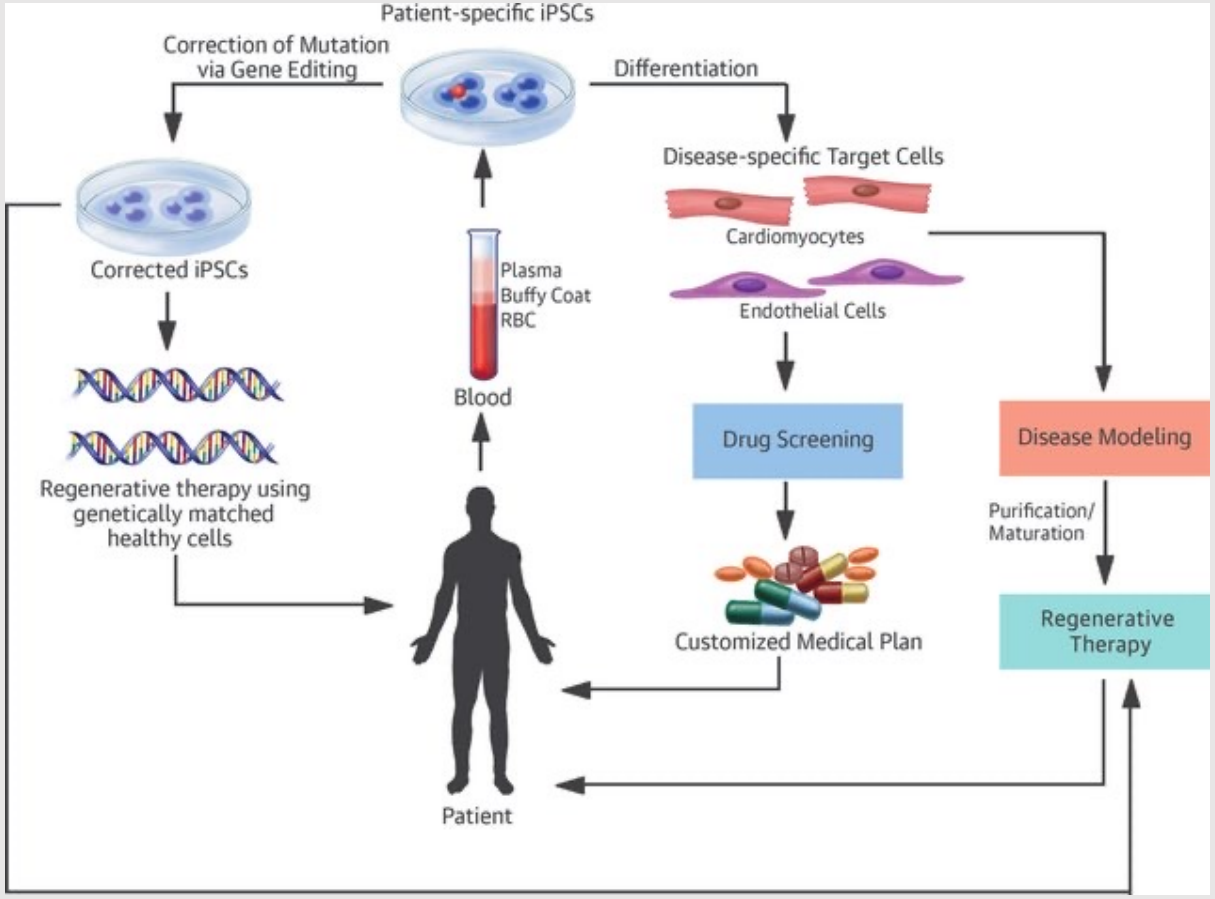
https://www.normalesup.org/~doulcier/teaching/modeling/excitable_systems.html

A model developed at Simula (complex)



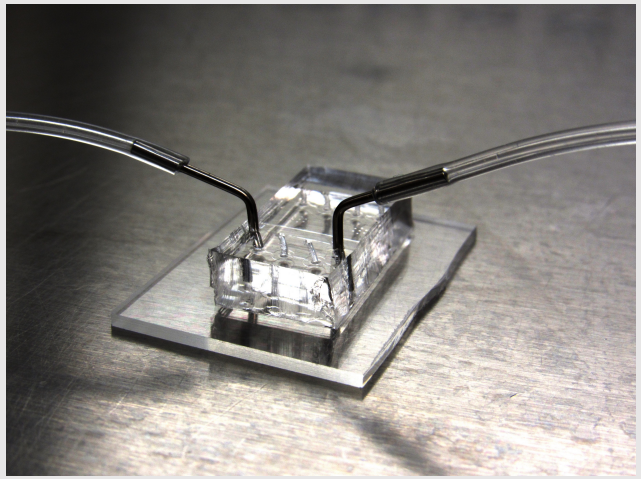
Jæger, K. H., Charwat, V., Charrez, B., Finsberg, H., Maleckar, M. M., Wall, S., ... & Tveito, A. (2020). Improved computational identification of drug response using optical measurements of human stem cell derived cardiomyocytes in microphysiological systems. *Frontiers in pharmacology*, *10*, 1648.

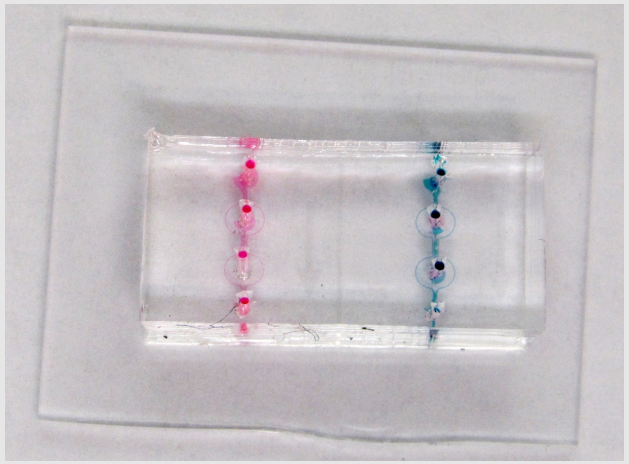
human induced Pluripotent Stem Cells (hiPSC) can be used in personalized drug screening

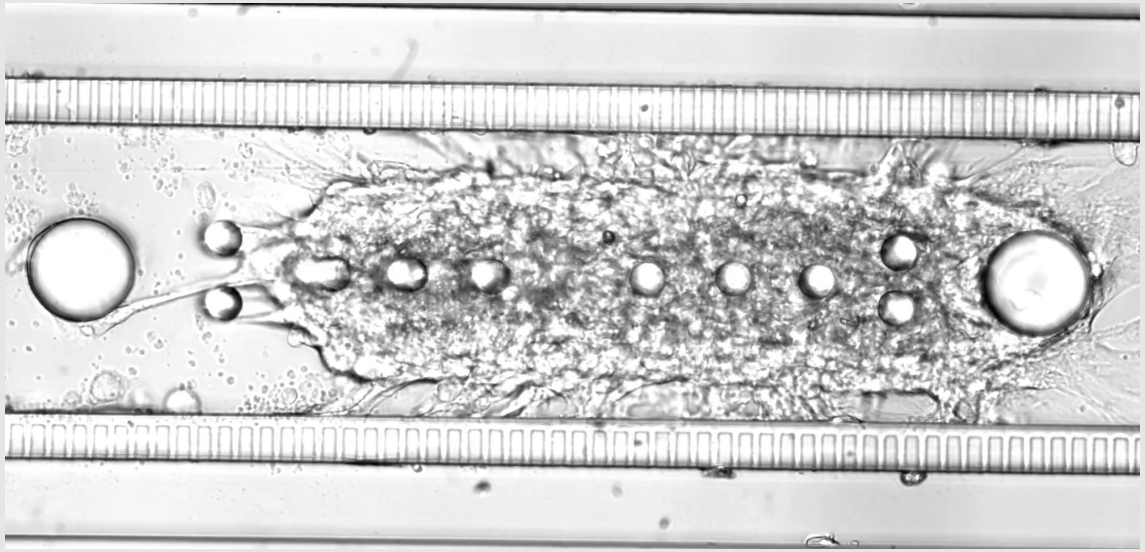


Translation of Human-Induced Pluripotent Stem Cells Nazish Sayed, Chun Liu, Joseph C. Wu Journal of the American College of Cardiology May 2016, 67 (18) 2161-2176

Microphysiological systems mimics the in vitro conditions





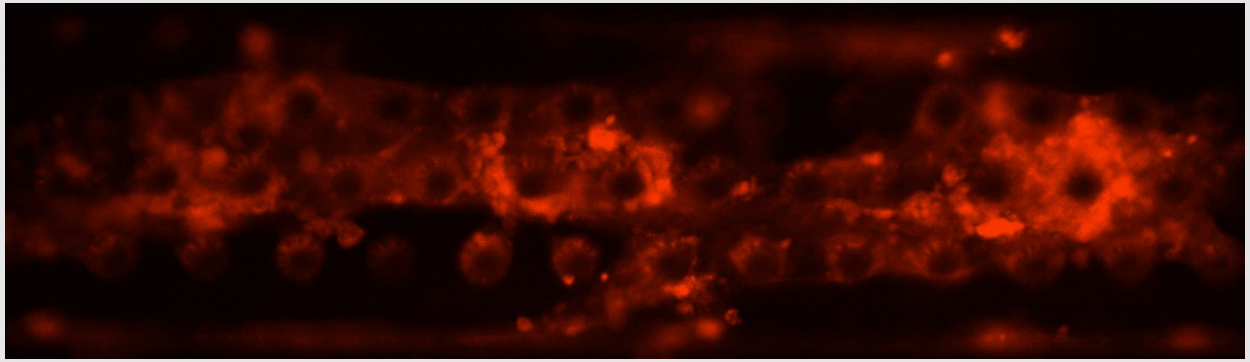


Organ-on-chip

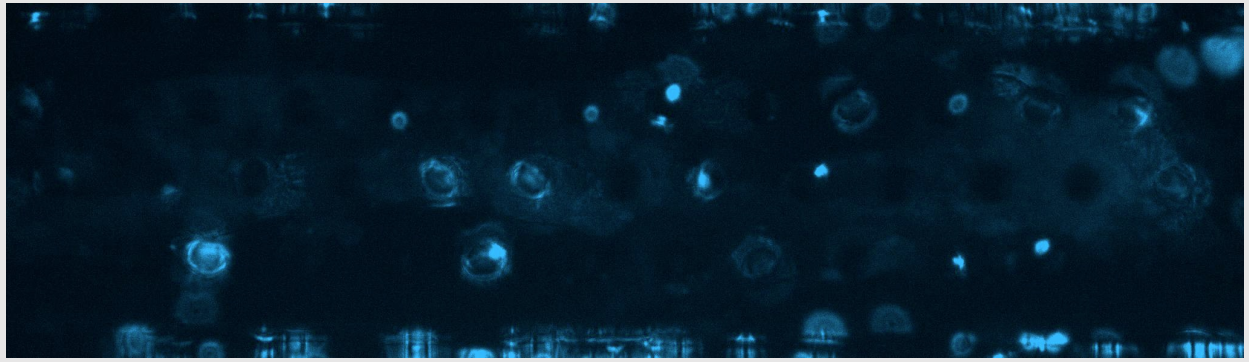
In vitro model for (personalized) human drug reponse

We want to be able to fit models to experimental data

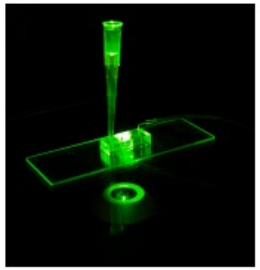
Voltage

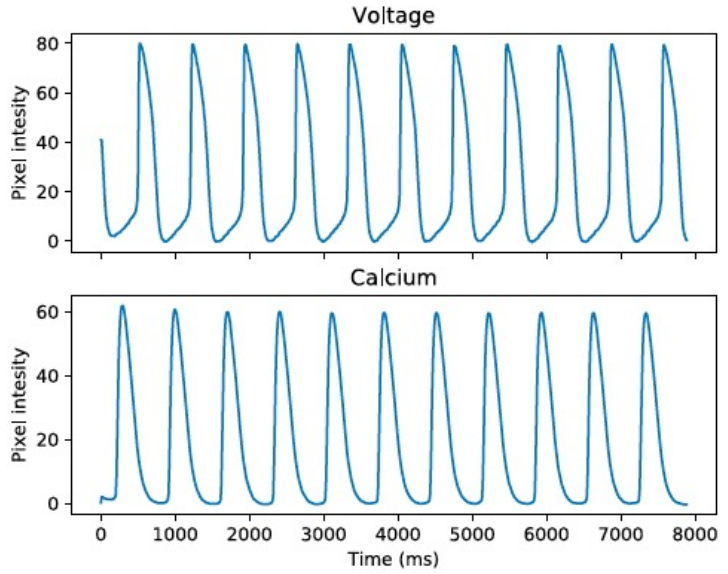


Calcium



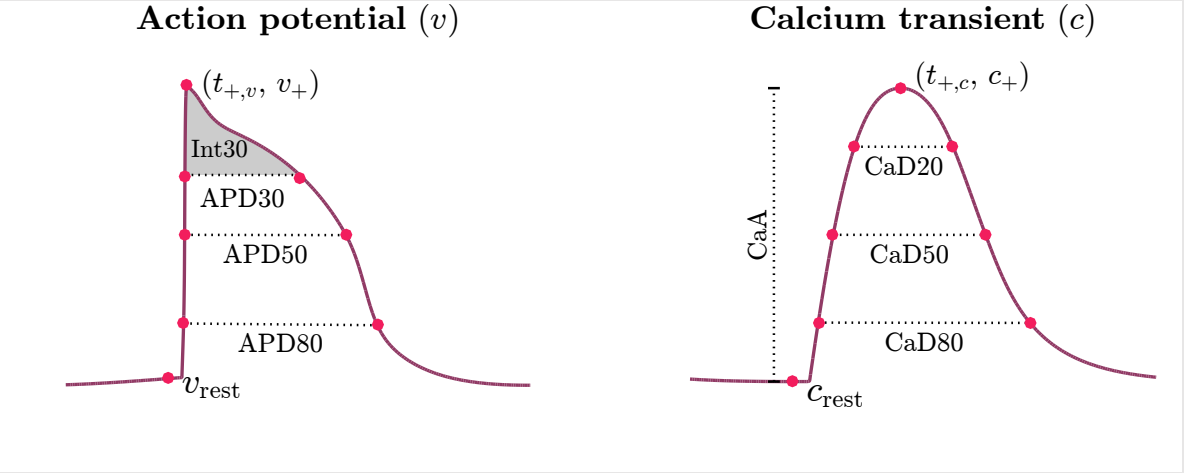
Optical Action Potential Measurements





The traditional approach is to form a cost function that represents the model-data mismatch

Compute features of the data and model



Choose control parameters (parameters that we want estimate to fit)

$$\lambda = (\lambda_{CaL}, \lambda_{NaL}, \lambda_{NaK}, \lambda_B, \lambda_{SERCA}, \lambda_{Kr}, \lambda_{NaCa}, \lambda_{diff}, \lambda_{K1}, \lambda_{RyR}),$$

Cost function:

$$J(\lambda) = \sum_i w_{i,v} \left(f_{i,v}^{\text{model}}(\lambda) - f_{i,v}^{\text{data}}(\lambda) \right)^2 + w_{i,c} \left(f_{i,c}^{\text{model}}(\lambda) - f_{i,c}^{\text{data}}(\lambda) \right)^2$$

Challenges with the traditional approach

- Cost function is highly non-convex – multiple local minima
- Gradient decent method does not work – gets stuck in local minima and is very dependent on initial starting point
- Currently we use a Brute Force method
  - Sample the parameter space
  - Run 1 billion simulations and store the resulting features in a database
  - Search through the database for best fit

Aim of thesis – compare deep learning approach with standard approaches

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Systems biology informed deep learning for inferring parameters and hidden dynamics

Alireza Yazdani[1☯], **Lu Lu**[2☯], **Maziar Raissi**[3], **George Em Karniadakis**[1*]

1 Division of Applied Mathematics, Brown University, Providence, Rhode Island, USA, **2** Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, **3** Department of Applied Mathematics, University of Colorado, Boulder, Colorado, USA

☯ These authors contributed equally to this work.
* george_karniadakis@brown.edu

Loss function consist of two supervised losses (data) and one unsupervised loss (ode)

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{p}) = \mathcal{L}^{data}(\boldsymbol{\theta}) + \mathcal{L}^{ode}(\boldsymbol{\theta}, \mathbf{p}) + \mathcal{L}^{aux}(\boldsymbol{\theta}), \tag{3}$$

where

$$\mathcal{L}^{data}(\boldsymbol{\theta}) = \sum_{m=1}^{M} w_m^{data} \mathcal{L}_m^{data} = \sum_{m=1}^{M} w_m^{data} \left[\frac{1}{N^{data}} \sum_{n=1}^{N^{data}} \left(y_m(t_n) - \hat{x}_{s_m}(t_n; \boldsymbol{\theta}) \right)^2 \right], \tag{4}$$

$$\mathcal{L}^{ode}(\boldsymbol{\theta}, \mathbf{p}) = \sum_{s=1}^{S} w_s^{ode} \mathcal{L}_s^{ode} = \sum_{s=1}^{S} w_s^{ode} \left[\frac{1}{N^{ode}} \sum_{n=1}^{N^{ode}} \left(\frac{d\hat{x}_s}{dt}\Big|_{\tau_n} - f_s(\hat{x}_s(\tau_n; \boldsymbol{\theta}), \tau_n; \mathbf{p}) \right)^2 \right], \tag{5}$$

$$\mathcal{L}^{aux}(\boldsymbol{\theta}) = \sum_{s=1}^{S} w_s^{aux} \mathcal{L}_s^{aux} = \sum_{s=1}^{S} w_s^{aux} \frac{\left(x_s(T_0) - \hat{x}_s(T_0; \boldsymbol{\theta}) \right)^2 + \left(x_s(T_1) - \hat{x}_s(T_1; \boldsymbol{\theta}) \right)^2}{2}. \tag{6}$$

Cardiac cell models have many similarities with the Yeast glycosis model

Table 1. Hyperparameters for the problems in this study. The fist and second number in the number of iterations correspond to the first and second training stage.

Model		NN depth	NN width	#Iterations
Yeast glycolysis	Noiseless	4	128	1000, 9×10^4
	Noisy	4	128	1000, 2×10^6
Cell apoptosis	Survival	5	256	0, 1.5×10^6
	Death	5	256	0, 1.5×10^6
Ultradian endocrine	Parameters only	4	128	2000, 6×10^5
	Hidden nutrition	4	128	2000, 1.5×10^6

Table 2. The feature layer used in the network for each problem.

Model	Features
Yeast glycolysis	$\tilde{t}, \sin(\tilde{t}), \sin(2\tilde{t}), \sin(3\tilde{t}), \sin(4\tilde{t}), \sin(5\tilde{t}), \sin(6\tilde{t})$
Cell apoptosis	$\tilde{t}, e^{-\tilde{t}}$
Ultradian endocrine	$\tilde{t}, \sin(\tilde{t}), \sin(2\tilde{t}), \sin(3\tilde{t}), \sin(4\tilde{t}), \sin(5\tilde{t})$

https://doi.org/10.1371/journal.pcbi.1007575.t002

Use the Yeast glycosis model as a starting point

- https://github.com/alirezayazdani1/SBINNs/blob/master/glycolysis.py
- Use a cardiac cell model instead. You can find a few implementations here: https://github.com/ComputationalPhysiology/ap_features/tree/master/demo
- Start with generating some synthetic data with known parameters and see if you can retrieve them using this method.
- Try without the ODE term – perhaps it is better without?
- Compare with other standard methods (see e.g https://docs.scipy.org/doc/scipy/reference/tutorial/optimize.html)
- Try adding some noise
- Look at parameter identifiability
- Experiment with settings for the neural network (Johannes)
- https://kent-and.github.io/OnWriting.pdf

simula