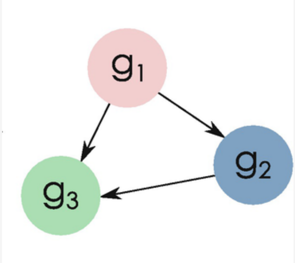
Using NeuralODEs to predict the dynamics of gene-regulatory networks

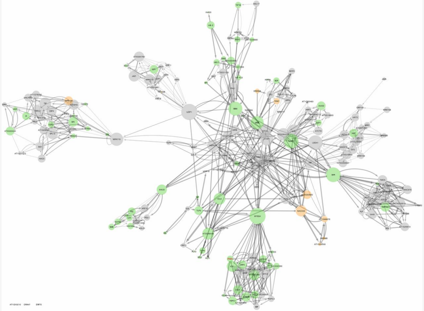
Intekhab Hossain
Advisors: Prof. John Quackenbush & Dr. Rebekka Burkholz

August 24, 2020

Gene-regulatory networks (GRNs)

- GRNs represent how multiple genes regulate (activate/repress) eachother to bring about observed levels of gene-expression.





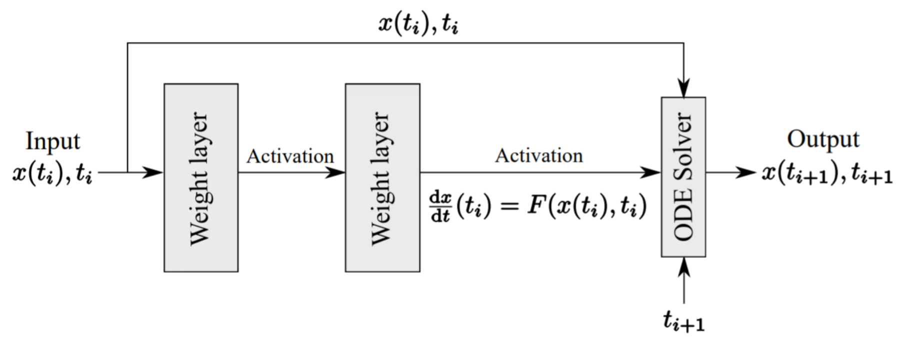
- Can study **long-term** behavior of networks (i.e. change in steady-state expression levels) upon perturbation (e.g. cancer).
- Inverse problem:
  - $g_1(t=\infty)$, $g_2(t=\infty)$, $g_3(t=\infty) \rightarrow$ predict $GRN(g_1, g_2, g_3)$

Extending GRNs to incorporate time - ODEs

- But often, **gene-expression dynamics** may also be of interest (gene-expression **trajectory** over time from $t = 0$ to $t = \infty$).
- Inverse problem:
  - Given:
    - $g_1(t = t_0), g_1(t = t_1), \ldots, g_1(t = \infty)$
    - $g_2(t = t_0), g_2(t = t_1), \ldots, g_2(t = \infty)$
    - $g_3(t = t_0), g_3(t = t_1), \ldots, g_3(t = \infty)$
  - Estimate **dynamics functions** f_1, f_2, f_3, where:
    - $dg_1/dt = f_1(g_1, g_2, g_3, t)$
    - $dg_2/dt = f_2(g_1, g_2, g_3, t)$
    - $dg_3/dt = f_3(g_1, g_2, g_3, t)$
- Any regression-based approach would mean restricting the functional forms of f_1, f_2, f_3, and simply **parametrizing** them.
- **NeuralODEs**: a deep neural network estimates f_1, f_2, f_3 without any restrictions on functional form.

The NeuralODE framework

- Chen *et al.* (NeurIPS 2018, best paper).



- Builds $\widehat{d\mathbf{x}/dt}$ by locally estimating $\frac{d\mathbf{x}}{dt}(t_i)$ at each time-point t_i.
- Has advantages over traditional ML tools for time-series (RNNs).
- PyTorch implementation with GPU-capacity.

Ground-truth simulator

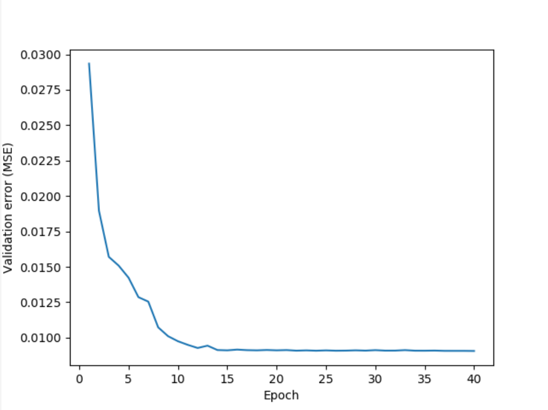
- Bhuva *et al.* (Genome Biology, BMC 2019) developed a simulator that uses the **Hill equation** to obtain steady-state gene levels.
- We modified their approach to generate time-series data.
- Toy example: $gene_A \xrightarrow{activates} gene_C \xleftarrow{represses} gene_B$
- $\frac{dA}{dt} = \frac{dB}{dt} = 0$
- $\frac{dC}{dt} = [f_{act}(A, E_{AC}, n_{AC}) - C] \times [1 - f_{act}(B, E_{BC}, n_{BC}) - C]$
- $f_{act}(X, E, n) = \frac{\beta X^n}{\beta - 1 + X^n}$, where $\beta = \frac{E^n - 1}{2E^n - 1}$
- Given a $GRN(g_1, g_2, ..., g_{150})$, our simulator can:
  - formulate $\frac{dg_1}{dt}, \frac{dg_2}{dt}, \ldots, \frac{dg_{150}}{dt}$.
  - generate $g_i(t = \tau), \forall i \in \{1, 2, \ldots, 150\}$ and $\forall \tau \geq 0$.

Experimental pipeline

- Sample 150 genes, relevant edges, and edge properties (activating vs repressive) from **yeast GRN**.
- Use ground-truth simulator (in R) to:
  - formulate the 150 ODEs ($\frac{dg_i}{dt}, \forall i \in \{1, 2, \ldots, 150\}$).
  - generate time-series data for all 150 genes.
    (multiple samples - varying initial conditions + Gaussian noise).
- Use NeuralODE framework (in PyTorch):
  - feed 90% of generated time-series samples for training.
  - learn $\{\widehat{\frac{dg_i}{dt}}\}_{i=1}^{150}$ using the deep NN of the framework.
  - evaluate $\{\widehat{\frac{dg_i}{dt}}\}_{i=1}^{150}$ - predictive performance on other 10% of samples.
- **Extract** analytical expression for $\{\widehat{\frac{dg_i}{dt}}\}_{i=1}^{150}$ from well-performing deep NN, and compare to ground-truth $\{\frac{dg_i}{dt}\}_{i=1}^{150}$.

Results so far

- Still in the phase of training a good NeuralODE.
- Poor predictive performance on validation set so far.



- Tuning NeuralODE hyperparameters to get better performance.

Next steps

- **Extract** analytical expressions for dynamics functions from well-performing NeuralODE, and compare to ground-truth.
- Deploy tool on new data to learn dynamics.
    - single-cell time-series data from **mouse GRN**.

Acknowledgements & References

- **Acknowledgements**:
  - Prof. John Quackenbush
  - Dr. Rebekka Burkholz
- **References**:
  - Chen, Ricky TQ, et al. "Neural ordinary differential equations." Advances in neural information processing systems. 2018.
  - Bhuva, Dharmesh D., et al. "Differential co-expression-based detection of conditional relationships in transcriptional data: comparative analysis and application to breast cancer." Genome biology 20.1 (2019): 1-21.
  - Karlsson, Daniel, and Olle Svanström. Modelling Dynamical Systems Using Neural Ordinary Differential Equations. MS thesis, Chalmers University. 2019.

Questions?

PROCESS

- sample 150 genes from **yeast GRN** ⟶
- formulate the 150 ODEs ⟶
- generate time-series data for all 150 genes ⟶
- training NeuralODE to learn dynamics (90% samples) ⟶
- test dynamics learned by NeuralODE (10% samples) ⟶
- extract analytical expressions for 150 dynamics functions from well-performing NeuralODE ⟶
- compare to 150 ground-truth ODEs