KSBi-BIML 2026

Bioinformatics & Machine Learning(BIML) Workshop for Life Scientists

생명정보학 & 머신러닝 워크샵 (온라인)

Dynamical description of complex biological systems using data, math, and deep learning

박계명 _ UNIST

KSBI KOREAN SOCIETY FOR BIOINFORMATICS | 한국생명정보학회

본 강의 자료는 한국생명정보학회가 주관하는 BIML 2026 워크샵을 목적으로
제작된 것으로 해당 목적 이외의 다른 용도로 사용할 수 없음을 분명하게 알립니다.

이를 다른 사람과 공유하거나 복제, 배포, 전송할 수 없으며 만약 이러한 사항을 위반할 경우
발생하는 **모든 법적 책임은 행위자 본인에게 있음**을 알립니다.

KSBi-BIML 2026

Bioinformatics & Machine Learning (BIML) Workshop for Life Scientists

한국생명정보학회가 주최하는 BIML-2026 동계 Bioinformatics &Machine Learning 교육 워크숍에 여러분을 초대합니다.

BIML 워크숍은 생명정보학 연구자들이 최신 AI바이오 분야의 인공지능 기반 분석 기술과 바이오 데이터 분석 기법을 이론과 실습을 통해 체계적으로 배울 수 있는 전문 교육 프로그램입니다. 2015년에 시작된 BIML 워크숍은 올해로 12년 차를 맞이하며, 국내 생명정보학 분야의 최초이자 최고 수준의 교육 프로그램으로 자리 잡았습니다. 이번 워크숍은 크게 인공지능바이오(AI바이오) 분야와 디지털바이오 분야, 두 분야로 구성됩니다.

AI바이오 분야에서는 생명정보 분석에 폭넓게 응용되고 있는 다양한 인공지능 기반 자료 모델링 기법을 다룰 예정입니다. 특히, 인공지능 심층학습을 활용한 단백질 구조 예측, 유전체 분석, 신약 개발에 대한 이론 및 실습 강의가 진행됩니다.

또한 디지털바이오 분야에서는 단일세포오믹스, 공간오믹스, 멀티오믹스, 메타오믹스에 대한 강의도 마련되어 있어, 연구자들의 분석 역량 강화에 실질적인 도움을 줄 것으로 기대됩니다.

또한 2024년부터 추가된 의료정보 자료 분석을 다루는 강의를 올해도 지속해서 운영하고자 합니다. 이는 최근 의료정보 자료 분석에 관한 연구 수요 증가를 반영한 것으로, 관련 연구를 수행하는 의과학자 및 의료정보 연구자들에게 유용한 지침을 제공할 것입니다.

또한, 올해도 생명정보학 기술의 다양화에 발맞춰 온라인 강좌를 대폭 확대했습니다. 올해는 무료 강좌 10개를 포함한 총 40개 이상의 강좌가 개설되며, 연구 주제에 맞는 강좌 추천과 강연료 할인 혜택도 제공합니다.

BIML-2026는 국내 주요 연구 중심 대학의 전임 교수 및 각 분야 최고 전문가들의 강의로 구성되어 있으며, 기초 이론부터 최신 연구 동향까지 아우르는 심도 있는 교육의 장이 될 것으로 확신합니다.

여러분의 많은 관심과 참여를 기대합니다!

2026년 2월

한국생명정보학회장 류 성 호

강의개요

Dynamical description of complex biological systems using data, math, and deep learning

생명현상은 다양한 세포, 분자들의 상호작용으로 인하여 결정되는 시공간적 동역학적 과정으로 일어난다. 그러나 오믹스로 대변되는 생명현상에 대한 조직, 세포, 및 분자 수준의 정량적 데이터를 다루는 생명정보학과 미분방정식 등을 활용하여 생명현상에 대한 동역학적 기술을 목표로 하는 수리생물학 간의 간극으로 인하여, 면역계와 같은 복잡한 생명현상을 있는 그대로 동역학적 모델링으로 다루고자 하는 시도는 많이 지체되어 있다.

본 강의에서는 단일세포 오믹스 데이터를 활용하여 주요 세포간/세포내 상호작용 네트워크를 추정하고 이 네트워크가 야기하는 동역학을 기술하는 미분방정식 기반의 수리 모델을 구축하며 이에 대한 비선형 동역학적 분석을 수행하는 방법을 다룬다. 또한 대상이 되는 시스템을 간략화 하는 전통적인 수리모델링 방법과 다르게, 복잡계적인 생명현상을 있는 그대로 동역학적으로 다루는 유망한 방법으로서 딥러닝 방법의 일종인 physics-informed neural network를 활용한 방법을 소개하고자 한다.

궁극적으로 본 강의를 통하여 정량적, 전산적 방법론으로서의 공통점을 지니나 상호간 간극이 큰 방향으로 발전한 생명정보학, 수리모델링, 딥러닝 방법론을 통합하여, 생명현상을 있는 그대로 동역학적으로 기술할 수 있는 방향의 연구가 가능함을 고찰하고자 한다.

강의는 다음의 내용을 포함한다:

- 단일 세포 전사체 데이터를 활용한 세포간/세포내 상호작용 네트워크 추정.
- 미분방정식을 이용한 동역학 모델링 및 비선형 동역학적 분석법 소개.
- Physics-informed neural network 소개.

* 참고강의교재: Nonlinear Dynamics and Chaos, 3rd edition by Steven Strogatz (CRC Press)

* 교육생준비물: 노트북 (메모리 8GB 이상, 디스크 여유공간 30GB 이상)

* 강의 난이도: 초급-중급

* 강의: 박계명교수 (울산과학기술원 의과학대학원 및 바이오메디컬공학과)

Curriculum Vitae

Speaker Name: Kyemyung Park, M.D., Ph.D.

▶ **Personal Info**

Name	Kyemyung Park
Title	Assistant Professor
Affiliation	Ulsan National Institute of Science and Technology (UNIST)

▶ **Contact Information**

Address	50, UNIST-gil, Ulju-gun, Ulsan, 44919
Email	kyemyung.park@unist.ac.kr

Research Interest

Systems biology/immunology/medicine, Bioinformatics, Mathematical biology, Biophysics

Educational Experience

2008	B.S. in Physics, Seoul National University
2014	M.D., Yonsei University College of Medicine
2020	Ph.D. in Biophysics, University of Maryland, College Park, USA

Professional Experience

2015-2020	Predoctoral Visiting Fellow, Laboratory of Immune System Biology, NIAID/NIA, USA
2020-2022	Fellow, Department of Pharmacology, Yonsei University College of Medicine
2022-2022	Senior Researcher, Korea Virus Research Institute, IBS

Selected Publications (3 maximum)

1. Tripathi, S., Tsang, J.S., and **Park, K.** (2023). Systems immunology of regulatory T cells: can one circuit explain it all? Trends Immunol. 44, 766–781. 10.1016/j.it.2023.08.007.
2. Wong, H. S., **Park, K.**, Gola, A., Baptista, A. P., Miller, C. H., Deep, D., Lou, M., Boyd, L. F., Rudensky, A. Y., Savage, P. A., Altan-Bonnet, G., Tsang, J. S., Germain, R. N. (2021). A local regulatory T cell feedback circuit maintains immunological homeostasis by pruning self-activated T cells. Cell 184, 3981-3997.e22.

KSBI | 한국생명정보학회

KSBi-BIML 2025: Dynamical description of complex biological system using data, math, and deep learning: Lecture

박계명 M.D., Ph.D.

Systems ImmunoDynamics Lab,
의과학대학원/바이오메디컬공학과,
울산과학기술원

Systems ImmunoDynamics Lab

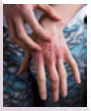
HST & BME

Ask and tackle immunological questions for predictive immunotherapy

Develop computational frameworks of scalable modeling

Infection

Cancer



Autoimmunity/
-inflammation

Across boundaries of

- Systems Biology/Immunology/Pharmacology
- Computational/Mathematical Biology
- Scalable/Multiscale Modeling
- Scientific Machine Learning
- Medical Data Science
- Statistical/Biological Physics

Lab website
https://sites.google.com/view/sysimm

Recruiting grad students and post-docs!
Email: kyemyung.park@unist.ac.kr

한국생명정보학회 연구분과

한국생명정보학회는 다양한 바이오정보 융합분야에 대한 학문증진과 학술교류를 위해 다음과 같은 연구분과를 운영하고 있다.

Artificial Intelligence and Information Technology 분과

딥러닝을 포함한 인공지능 및 최신 정보기술을 바이오의료 분야의 문제해결을 위해 적용하는 분야로서, 최근 학계는 물론 제약산업계에서 크게 각광받고 있다.

Genome Informatics 분과

유전체 전체 데이터를 해석하는 학문으로 유전체 데이터를 해석하고 다양한 유전 정보를 해독하며 이를 이용하여 유전체 수준에서 다양한 질병과의 연관성 연구를 수행하고 이를 위한 전산학적 알고리즘 개발까지 포함하는 학문이다.

Chemical Informatics 분과

화학분야의 다양한 문제들을 컴퓨터 및 정보 기술를 이용하여 연구하는 학문이며, 주로 신약개발 과정에서 선도물질 탐색 및 최적화를 위해 활용되고 있다.

Systems Biology 분과

생명현상을 개별 유전자나 세포 수준을 뛰어넘어 하나의 유기적인 시스템 수준에서 이해하고 분석하는 학문분야로서, 생물학은 물론 전산학, 수학과 같은 다양한 분야의 융합학문이라고 할 수 있다.

Translational Informatics 분과

분자세포생물학과 같은 기초과학 분야와 임상의료기술을 증개하기 위해 정보학적 이론과 방법을 활용하는 것이 특징이다. 많은 경우 임상실험 전 단계에서 임상적인 결과물 도출을 가속화하는데 사용된다.

Healthcare Informatics 분과

임상 데이터를 정보 기술을 이용하여 연구하는 학문을 말한다. 최근 임상 데이터 뿐 아니라 건강관리 데이타와 관련정보의 생성과 사용까지 포괄적으로 적용되고 있다.

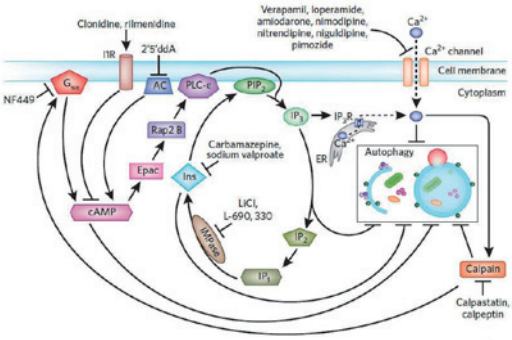
Intellectual Properties 분과

생명정보학은 생명과학과 정보기술의 융합분야로서, 특허를 포함한 지적재산권의 형태와 가치가 전통적인 분야와 매우 다른 양상으로 나타난다. 본 분과에서는 그러한 차별성을 식별하고 실무에 적용할 수 있는 연구를 수행한다.

Elements consisting of computational biology

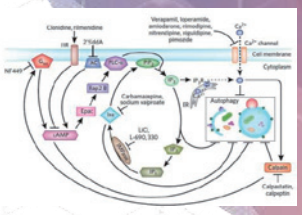
- Quantitative Data
  - Omics,
  - Clinical data from medical records, medical imaging, vital signals(e.g., ECG)
- Quantitative modeling
  - Mechanistic, rule-based, or dynamical modeling
    - Differential equations (ODE, PDE, SDE, CME)
    - Stochastic simulation (e.g., Gillespie's algorithm)
    - Boolean network
    - Agent-based modeling
  - Statistical or Data-driven modeling
    - Statistical models
    - AI/Deep learning models

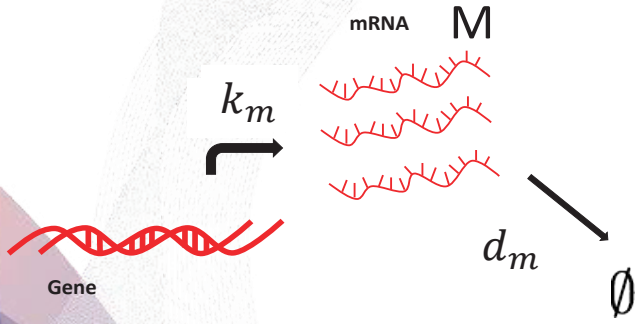
What is a model in biology?



Nature Chemical Biology ***7****, 9–17 (2011)*

Computational models

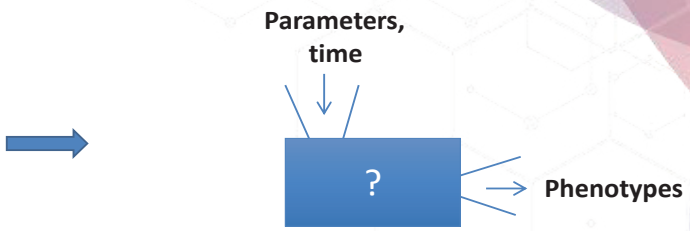




$$\frac{dM}{dt} = k_m - d_m \cdot M$$

Computational models

$$\frac{dM}{dt} = k_m - d_m \cdot M$$

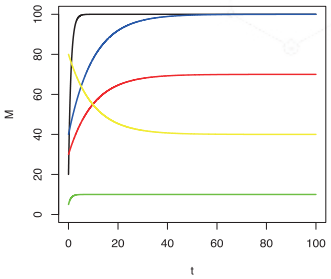


Mathematical solution

$$M(t) = \frac{k_m}{d_m} + \left(M_0 - \frac{k_m}{d_m}\right) \cdot e^{-d_m \cdot t}$$

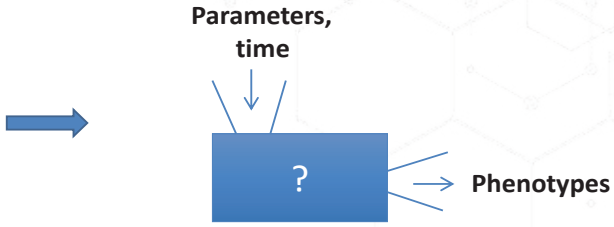
Simulation
(Numerical solution)

For different M_0, k_m, d_m



Computational models

$$\frac{dM}{dt} = k_m - d_m \cdot M$$



- **Predictive understanding** of biological systems
  - Time evolution
  - Responses upon perturbations of parameters → identification of personalized intervention modalities
  - Emergence phenomena

Mathematical modeling in biology

$$\frac{d\vec{X}}{dt} = \vec{f}(\vec{X}, \vec{k}, t)$$

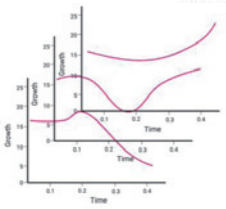
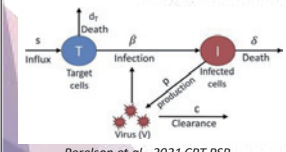


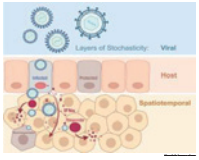
Table 1 Computational approaches and tools for systems biology

Modeling approach	Typical applications	Limitations	Tools
Individual particle-based stochastic	Small subcellular signaling processes, aspects of bacterial biochemistry	Applies only to small systems (in terms of space and chemical complexity)	MCell (32), Smoldyn (314), ChemCell (315), GetBonNie (nonspatial) (49)
Particle number stochastic	Signaling processes with important stochastic aspects (due to small system size or high sensitivity)	Applies only to small systems (in terms of space and chemical complexity), has less detail than individual particle simulation	MesoRD (35), SmartCell (33), GetBonNie (nonspatial)
Concentration-based spatial, nonstochastic	Cellular signaling processes with important spatial aspects	Provides either high spatial resolution or biochemical complexity, has no stochasticity	Virtual Cell (37), Simmune (36)
Concentration-based, nonspatial, nonstochastic	Cellular signaling processes without spatial aspects	Assumes global biochemical homogeneity in the simulated system	Copasi (46), E-cell (44), Cellware (45), Systems Biology Workbench (47), GetBonNie

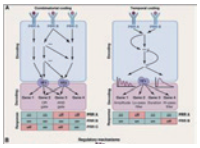
Germain et al. 2011 Annu. Rev. Immunol.



Perelson et al., 2021 CPT PSP

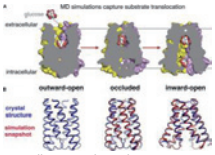


Van Eyndhoven et al., 2021 Trends in Immun.



Luecke et al. 2021 Immunity

Hong et al., 2021 Communications Biology

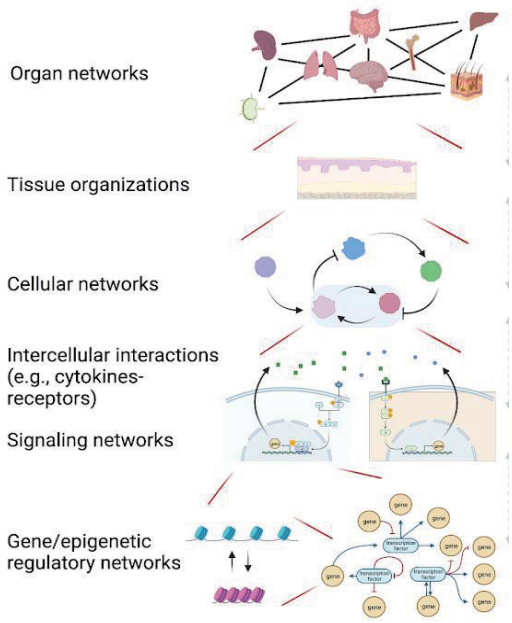


Hollingsworth et al., 2018 Neuron

Mechanistic, rule-based, or dynamical modeling

- Differential equations (ODE, PDE, SDE, CME)
- Stochastic simulation (e.g., Gillespie's algorithm)
- Boolean network
- Agent-based modeling

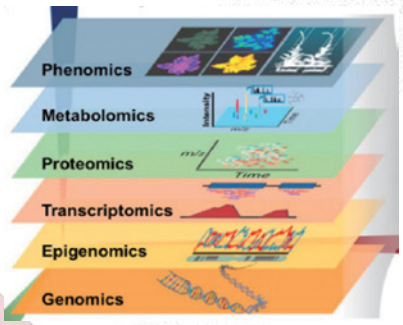
Complex and multiscale organization of the mammalian biological/immunological system



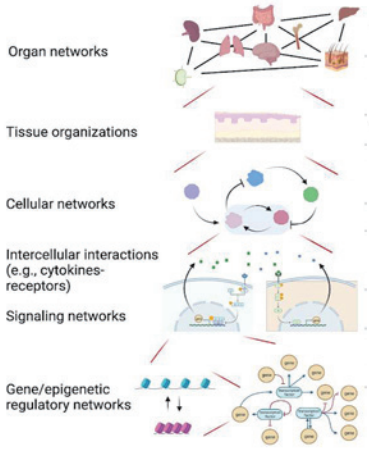
How can we establish scalable modeling of biological/immunological systems?

Data <-> Network <-> Dynamics

Data



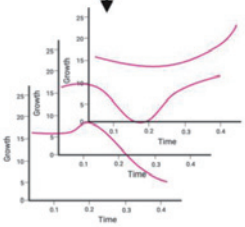
Network



Dynamics

$$\frac{d\vec{X}}{dt} = \vec{f}(\vec{X}, \vec{k}, t)$$

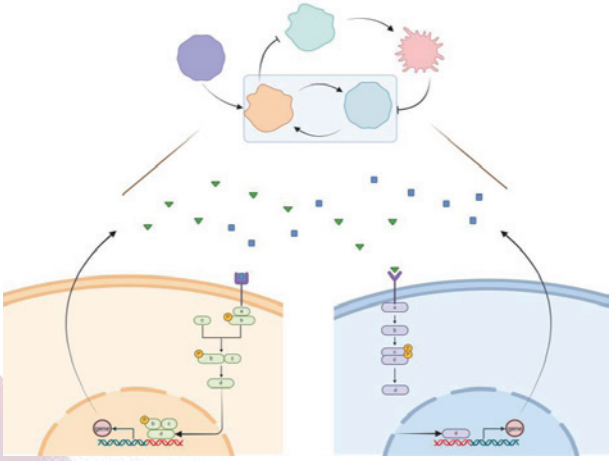
Prediction through simulations



Use deep learning to make these scalable (high-throughput)

Data → Network

Various tools are available for inferring networks across various layers using single-cell RNA sequencing data and established databases



Cell-Cell Communication (Ligand - Receptor)

Signaling Network (Receptor ~ Transcription Factor)

Gene Regulatory Network (Transcription Factor – Target gene)

CellphoneDB

CellChat

LIANA

NicheNet

Omni Path

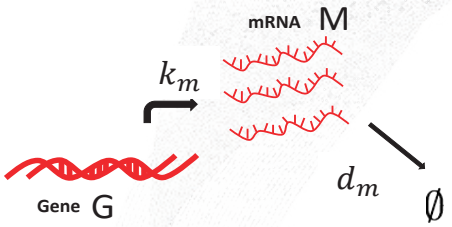
SCENIC+

CollecTRI

Network → Dynamics

Network → Reaction network → Differential equations

From the previous example



Reaction species:

- G (gene)
- M (mRNA)

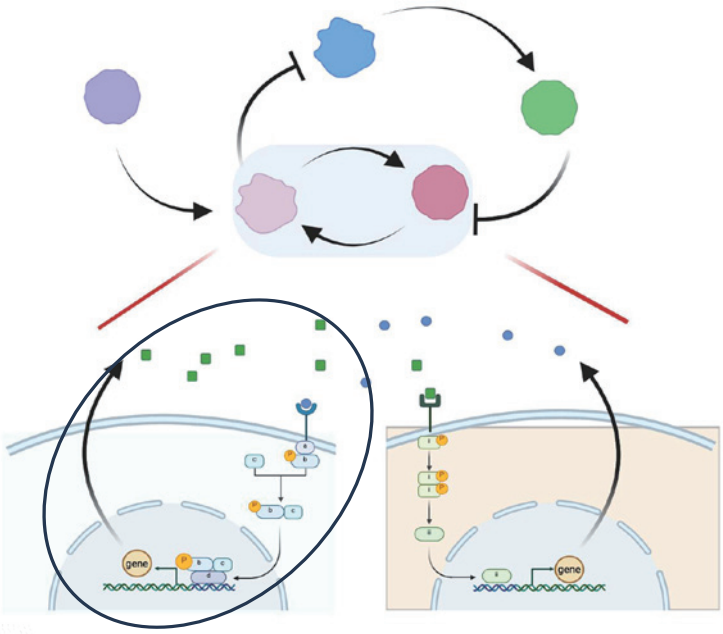
Reactions:

- $G \xrightarrow{k_m} Gene + M$ with a propensity, k_m
- $M \xrightarrow{d_m \cdot M} \emptyset$ with a propensity, $d_m \cdot M$

Equation(s)

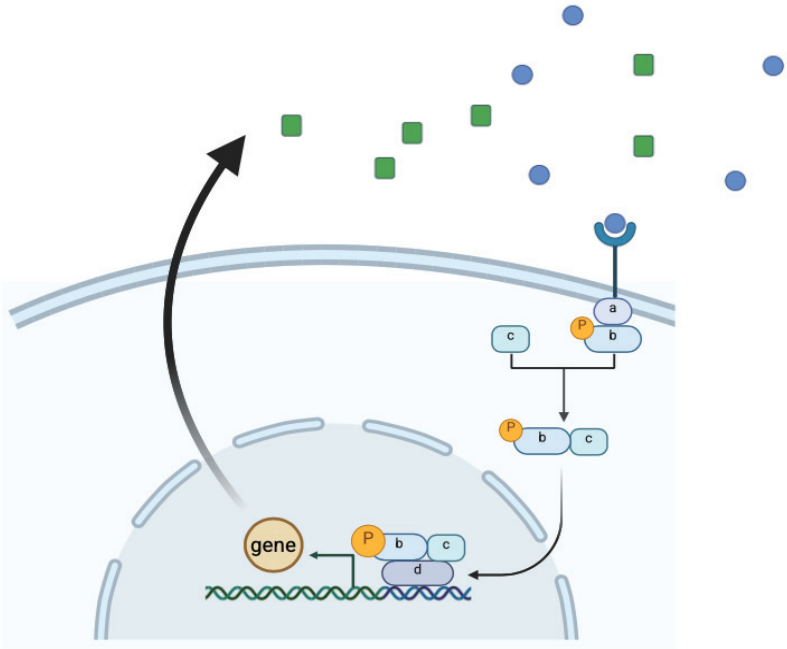
- $\frac{dM}{dt} = k_m - d_m \cdot M$

Network → Reaction network → Differential equations



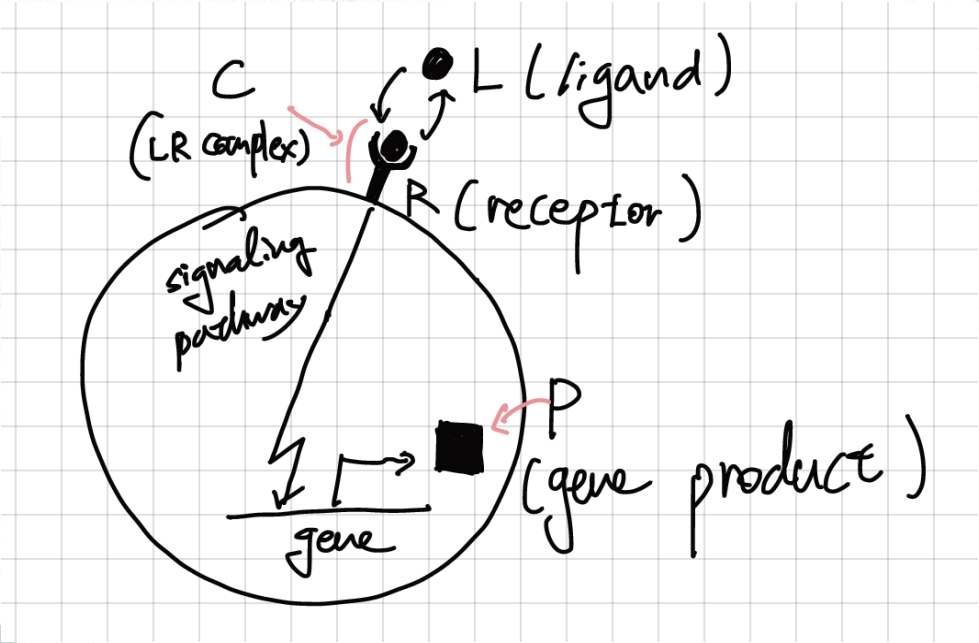
Network → Reaction network → Differential equations

A unit block in a multiscale network



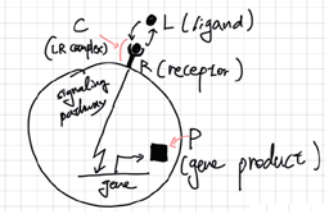
Network → Reaction network → Differential equations

A schematic view



Network → Reaction network → Differential equations

A schematic view



Reaction species

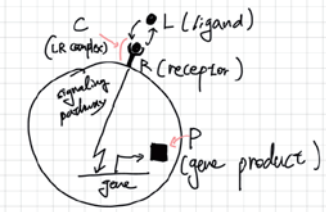
L: ligands
R: receptors
C: ligand-receptor complexes
G: gene
P: gene products

Reactions

- $L + R \rightleftarrows C$ w/ propensities
  - $\rightarrow$: $k_{on} L \cdot R$
  - $\leftarrow$: $k_{off} C$
- $G + C \longrightarrow G + C + P$
  w/ propensity $k_{syn} \cdot \underbrace{\frac{C^{n_{syn}}}{K_{syn}^{n_{syn}} + C^{n_{syn}}}}_{\text{Hill function}}$
- $P \longrightarrow \emptyset$ w/ propensity $k_{dec} \cdot P$

Network → Reaction network → Differential equations

A schematic view



Reaction species

L: ligands
R: receptors
C: ligand-receptor complexes
G: gene
P: gene products

- Equations

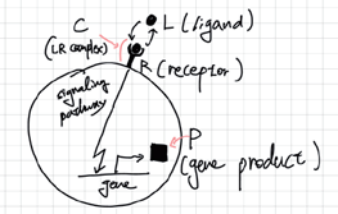
$$\frac{dL}{dt} = \frac{dR}{dt} = -k_{on} L \cdot R + k_{off} C$$

$$\frac{dC}{dt} = k_{off} C - k_{on} L \cdot R$$

$$\frac{dP}{dt} = k_{syn} \frac{C^{n_{syn}}}{K_{syn}^{n_{syn}} + C^{n_{syn}}} - k_{dec} \cdot P$$

Network → Reaction network → Differential equations

A schematic view



Reaction species

L: ligands
R: receptors
C: ligand-receptor complexes
G: gene
P: gene products

- Equations

$$\frac{dL}{dt} = \frac{dR}{dt} = -k_{on} L \cdot R + k_{off} C$$

$$\frac{dC}{dt} = k_{off} C - k_{on} L \cdot R$$

$$\frac{dP}{dt} = k_{syn} \frac{C^{n_{syn}}}{K_{syn}^{n_{syn}} + C^{n_{syn}}} - k_{dec} \cdot P$$

Typical time scales

→ in seconds

→ in hours

Network → Reaction network → Differential equations

Quasi-steady-state approximation for ligand-receptor binding

$$\frac{dR}{dt} = -k_{on} L \cdot R + k_{off} C \approx 0$$

Using $L_{tot} = L_{(free)} + C$

$R_{tot} = R_{(free)} + C$

and further assuming

$L_{tot} >> R_{tot}$, therefor $L_{tot} \approx L_{(free)}$

then we have

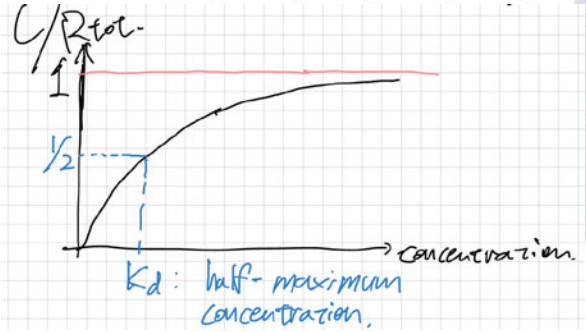
$$-k_{on} L (R_{tot} - C) + k_{off} C = 0$$

$$L (R_{tot} - C) - \frac{k_{off}}{k_{on}} C = 0$$

$\frac{k_{off}}{k_{on}} \equiv K_d$: dissociation coefficient.

$$C = R_{tot} \frac{L}{K_d + L}$$

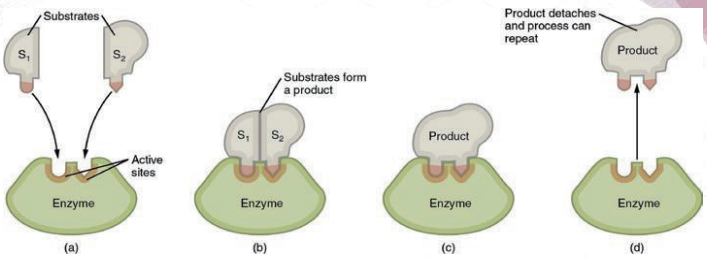
K_d in Molar concentration (M)

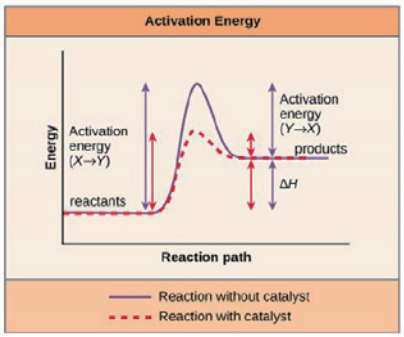


Network → Reaction network → Differential equations

A related topic: Michaelis-Menten equation

Enzymes are kinds of catalysts that speed up biochemical reactions of substrates by lowering activation energy.





The enzyme reaction can be decomposed as

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C \xrightarrow{k_2} E + P$$

where E: enzyme, S: substrate, C: ES complex, P: product

Network → Reaction network → Differential equations

A related topic: Michaelis-Menten equation

From

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C \xrightarrow{k_2} E + P$$

we can write down the kinetics of each species as (dropping [ ] for notational simplicity)

$$\frac{dS}{dt} = k_{-1}C - k_1 E \cdot S$$

$$\frac{dE}{dt} = k_{-1}C - k_1 E \cdot S + k_2 C$$

$$\frac{dC}{dt} = -k_2 C - k_{-1}C + k_1 E \cdot S$$

$$\frac{dP}{dt} = k_2 C$$

Network → Reaction network → Differential equations

A related topic: Michaelis-Menten equation

If the total amount of the enzyme E is limited as E_t and S >> E_t,

$$E_t = E + C = constant$$

If binding and unbinding of E and S are much faster than the rate of product synthesis (k_2), we can apply **the equilibrium approximation** as

$$\frac{dS}{dt} = k_{-2}C - k_1 E \cdot S = 0$$

$$C = \frac{k_1}{k_{-1}} E \cdot S$$

25

Network → Reaction network → Differential equations

A related topic: Michaelis-Menten equation

using $E_t = E + C$

$$\Rightarrow C = \frac{k_1}{k_{-1}}(E_t - C)\cdot S$$

solving this for C,

$$\left(\frac{k_{-1}}{k_1} + S\right)\cdot C = E_t$$

$$C = \frac{E_t S}{K_M + S} \quad \text{where } K_M = \frac{k_{-1}}{k_1}$$

26

Network → Reaction network → Differential equations

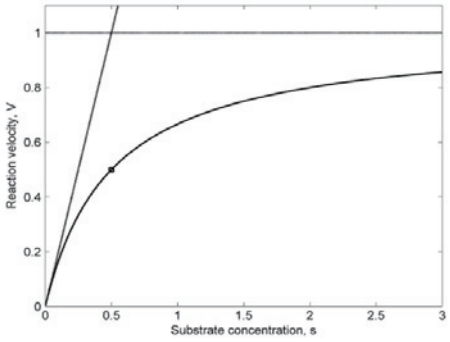
A related topic: Michaelis-Menten equation

Finally, the velocity of the reaction (the rate of formation of product) is

$$V = \frac{dP}{dt} = k_2 C = \frac{k_2 \cdot E_t \cdot S}{K_M + S}$$

$$= \frac{V_{max} \cdot S}{K_M + S} \quad \text{where } V_{max} \equiv k_2 E_t$$

$\Rightarrow$ Michealis – Menten equation



Network → Reaction network → Differential equations

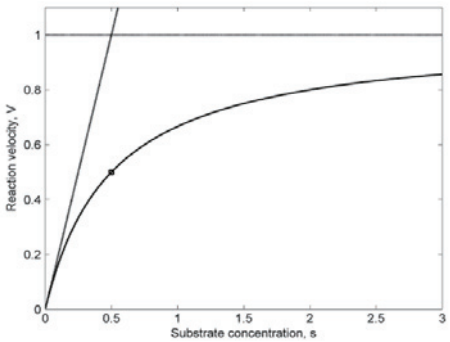
A related topic: Michaelis-Menten equation

Finally, the velocity of the reaction (the rate of formation of product) is

$$V = \frac{dP}{dt} = k_2 C = \frac{k_2 \cdot E_t \cdot S}{K_M + S}$$

$$= \frac{V_{max} \cdot S}{K_M + S} \quad \text{where } V_{max} \equiv k_2 E_t$$

$\Rightarrow$ Michealis – Menten equation



Network → Reaction network → Differential equations

Hill equation

- Cooperativity:
  - If the enzyme can bind more than one substrate molecule, and the binding of one substrate molecule affects the binding of subsequent molecules
  - Michaelis-Menten equation generalizes to Hill equation

$$V = \frac{V_{max} \cdot S}{K_M + S} \longrightarrow V = \frac{V_{max} \cdot S^n}{K_M^n + S^n}$$

Hill equation

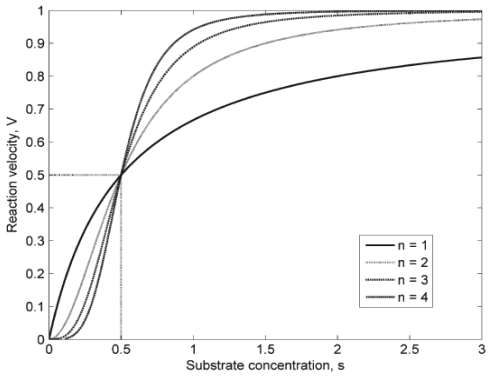
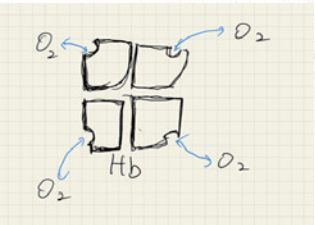


Fig. 1.8 Hill equation reaction behaviour, for different values of *n*

Network → Reaction network → Differential equations

Hill equation

Binding of hemoglobin and O_2 is a very good example.

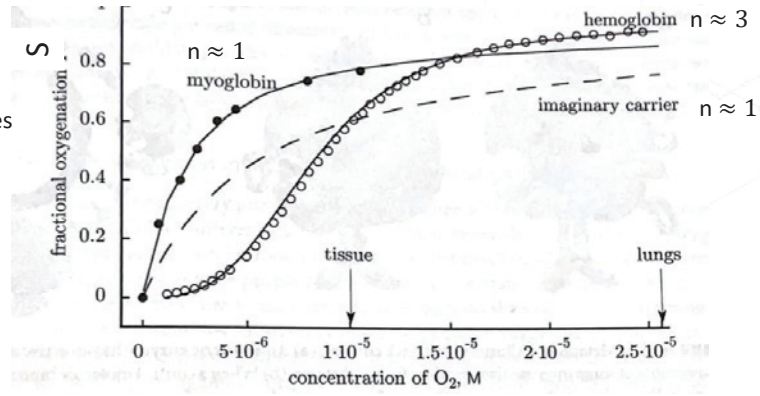


Tetramer: each subunit can bind to a O_2 molecule.

$$Hb + 4O_2 \rightleftarrows HbO_2 + 3O_2 \rightleftarrows Hb(O_2)_2 + 2O_2 \rightleftarrows Hb(O_2)_3 + O_2 \rightleftarrows Hb(O_2)_4$$

Fractional filling of available hemoglobin sites

$$S = \frac{[O_2]^n}{K^n + [O_2]^n}$$

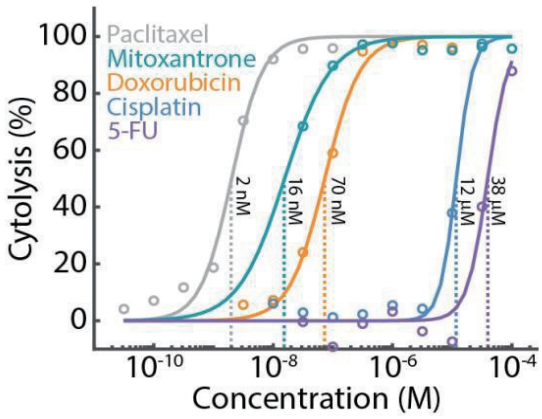


Network → Reaction network → Differential equations

Hill equation

$$V = \frac{V_{max} \cdot S^n}{K_M^n + S^n}$$

A good empirical way to quantitatively describe dose-response relationship in many cellular or subcellular behavior

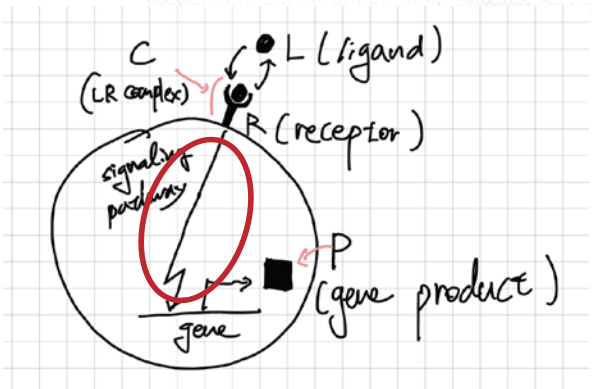


https://www.axionbiosystems.com/resources/application-note/dose-response-analysis-impedance-based-potency-assays

31

Network → Reaction network → Differential equations

Hill equation

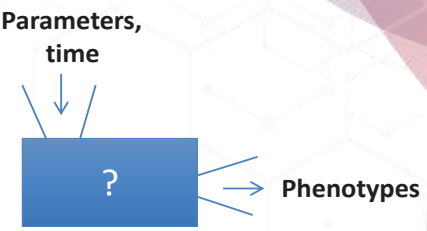


We assume that detailed biochemical reactions that comprise the signaling pathway collectively form a typical dose-response curve, which can be modeled by Hill functions (or Hill function of Hill function of …. as if activation functions do in deep neural network

$$\frac{dP}{dt} = k_{syn} \frac{C^{n_{syn}}}{K_{syn}^{n_{syn}} + C^{n_{syn}}} - k_{dec} \cdot P$$

How to solve differential equations using computers?

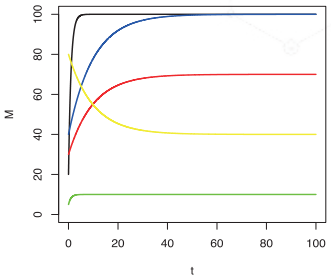
$$\frac{dM}{dt} = k_m - d_m \cdot M$$



Mathematical solution

Simulation
(Numerical solution)

For different M_0, k_m, d_m



$$M(t) = \frac{k_m}{d_m} + \left(M_0 - \frac{k_m}{d_m}\right) \cdot e^{-d_m \cdot t}$$

KSBI | 한국생명정보학회

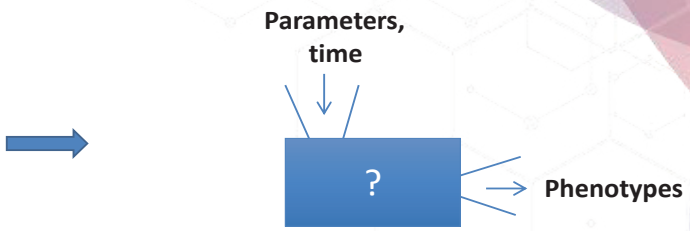
KSBi-BIML 2025: Dynamical description of complex biological system using data, math, and deep learning: Coding lab

박계명 M.D., Ph.D.

Systems ImmunoDynamics Lab,
의과학대학원/바이오메디컬공학과,
울산과학기술원

How to solve differential equations using computers?

$$\frac{dM}{dt} = k_m - d_m \cdot M$$

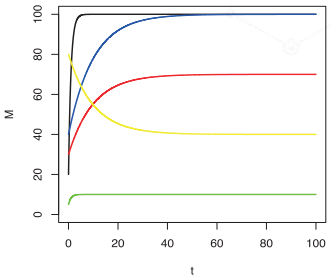


Mathematical solution

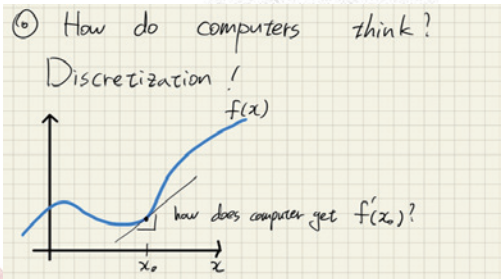
$$M(t) = \frac{k_m}{d_m} + \left(M_0 - \frac{k_m}{d_m}\right) \cdot e^{-d_m \cdot t}$$

Simulation
(Numerical solution)

For different M_0, k_m, d_m



How to solve differential equations using computers?



Based on Taylor expansion

$$f(x_0+h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + \cdots$$

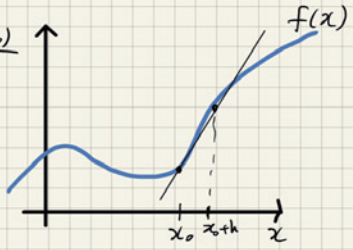
i) 1st order approximation

$$f(x_0+h) = f(x_0) + h f'(x_0) + O(h^2)$$

$$\Rightarrow f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} + O(h)$$

$$\Rightarrow f'(x_0) \approx \frac{f(x_0+h) - f(x_0)}{h}$$

within the accuracy of one order of h.



How to solve differential equations using computers?

ii) 2nd order approximation

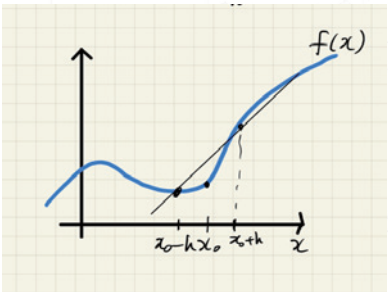
$$f(x_0+h) = f(x_0) + h f'(x_0) + \frac{h^2}{2!} f''(x_0) + \frac{h^3}{3!} f'''(x_0) + O(h^4)$$

$$f(x_0-h) = f(x_0) - h f'(x_0) + \frac{h^2}{2!} f''(x_0) - \frac{h^3}{3!} f'''(x_0) + O(h^4)$$

$$\Rightarrow f(x_0+h) - f(x_0-h) = 2h f'(x_0) + O(h^3)$$

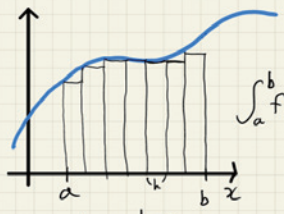
$$f'(x) = \frac{f(x_0+h) - f(x_0-h)}{2h} + O(h^2)$$

$$\simeq \frac{f(x_0+h) - f(x_0-h)}{2h}$$ within the accuracy of h^2.



How to solve differential equations using computers?

° Integration → sumation



$$\int_a^b f(x)dx = \sum_{n=0}^{N-1} f(a+nh)\cdot h + O(h^2)$$

Within the accuracy of h^2

$$h = \frac{b-a}{N}$$

$$\int_a^{a+h} f(x)dx = \int_0^h f(a+x)dx = \int_0^h (f(a) + xf'(a) + \cdots)dx$$

By keeping higher order terms in the tayler expansion, more accurate formula can be derived. such as trapezoidal rule or Simpson rule

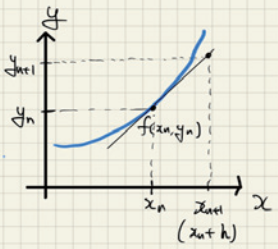
How to solve differential equations using computers?

◎ Solving differential equation

Again, integration → summation

$$\frac{dy}{dx} = f(y, x)$$

slope



$$\left.\frac{dy}{dx}\right|_{x=x_n} \approx \frac{y_{n+1} - y_n}{h} = f(x_n, y_n)$$

$$\Rightarrow y_{n+1} = y_n + h f(x_n, y_n)$$

⇒ Euler methods. ⇒ but not used widely due to large errors.

More accurate computation requires more terms for each increment.

39

How to solve differential equations using computers?

ex) Runge – Kutta method (4th order)

$$y_{n+1} = y_n + \frac{1}{6}h\left[f_1 + f_2 + f_3 + f_4\right]$$

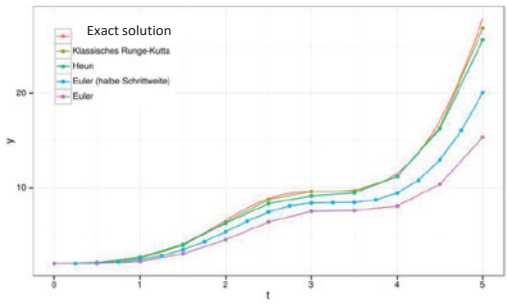
where

$$f_1 = f(x_n, y_n)$$

$$f_2 = f\left(x_n + \tfrac{1}{2}h,\ y_n + \tfrac{1}{2}hf_1\right)$$

$$f_3 = f\left(x_n + \tfrac{1}{2}h,\ y_n + \tfrac{1}{2}hf_2\right)$$

$$f_4 = f\left(x_n + h,\ y_n + hf_3\right)$$



https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods

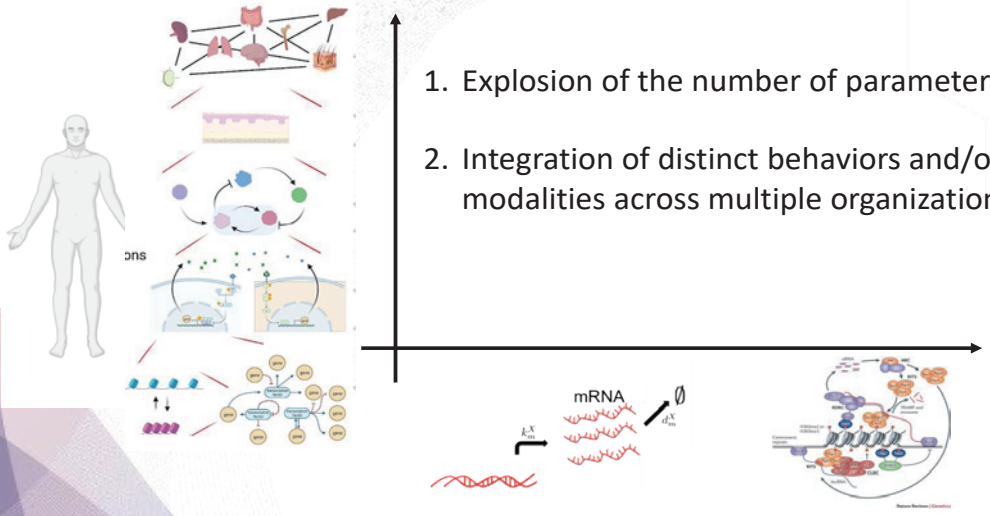
40

How to solve differential equations using computers?

- There are many good numerical solvers available in Fortran, C/C++, Matlab, R, Python, Julia ….
- All solvers in these languages share similar usages.
- Therefore, once you learn how to use solvers in one language, then you can easily adapt to other langua ges.
- We will practice using R.

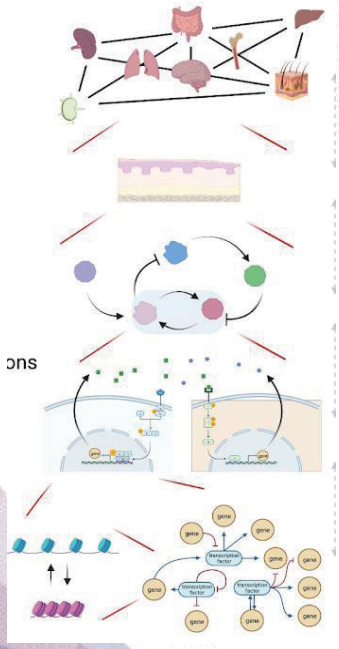
Scaling up [Data<->Netw ork<->Dynamics] using d eep learning

Challenges in scalable modeling



1. Explosion of the number of parameters
2. Integration of distinct behaviors and/or modeling modalities across multiple organizational levels

Challenges in Scalable modeling



Typical features in the immune system, which have not been addressed yet:

- Cells interact with each other via signaling molecules that change intracellular states that, in turn, affect intercellular interactions.
- Cells move in space and/or form spatial structures that, in turn, again affect intercellular interactions and/or intracellular states.
- There are many attributes (internal degrees of freedom) at the intracellular level.

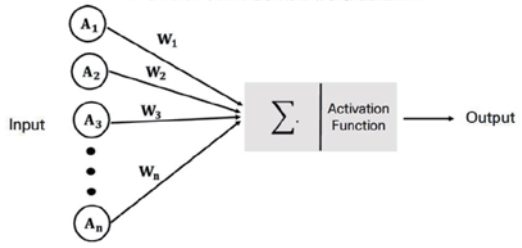
➔ How can we describe such complex, multiscale, nonlinear dynamical systems realistically (as opposed to a traditional way of modeling toward simplifying the system)?

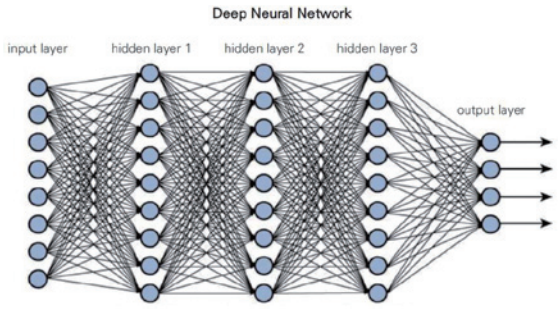
➔ Too many constituent players with too many parameters, with too many degrees of freedom (independent variables) →a curse of dimensionality!

➔ Can neural networks resolve this? (one of our research topics in our lab to enable scalable modeling using deep learning)

Physics-informed neural network

Neural networks





그림출처: https://repository.kisti.re.kr/bitstream/10580/19182/1/KISTI%20이슈브리프%20제74호.pdf

Universal approximation theorem:
Briefly, deep feedforward neural networks with proper activation functions can approximate any continuous functions within any arbitrary degree of accuracy given a sufficient number of neurons in the hidden layer.

https://en.wikipedia.org/wiki/Universal_approximation_theorem

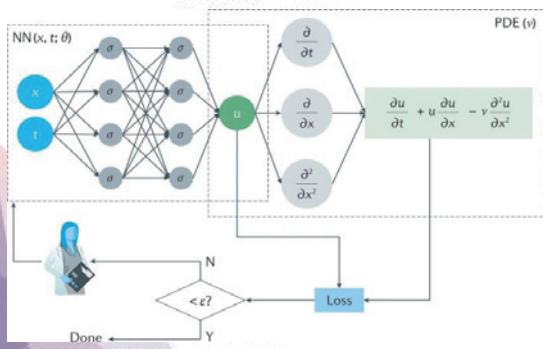
→ Can we let neural networks learn the solutions of differential equations that cannot be solved analytically and numerically?

Physics-informed neural network

Journal of Computational Physics
Volume 378, 1 February 2019, Pages 686-707

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi [a], P. Perdikaris [b], G.E. Karniadakis [a]



First appeared as “Physics-informed Neural networks” by the Karniadakis group in 2019

A given physics law by viscous Burger’s equation $\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}$

A surrogate neural netwok that learn the PDE solution $u(x,t)$ $NN(x,t,\theta)$

Construct a loss function $\mathcal{L} = w_{\text{data}}\mathcal{L}_{\text{data}} + w_{\text{PDE}}\mathcal{L}_{\text{PDE}},$

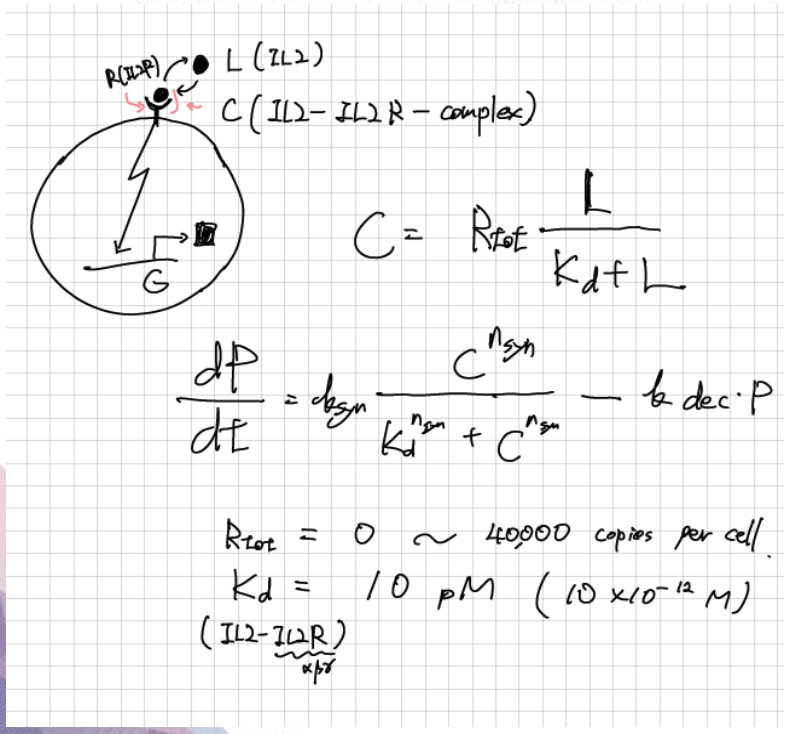
where

$$\mathcal{L}_{\text{data}} = \frac{1}{N_{\text{data}}}\sum_{i=1}^{N_{\text{data}}}(u(x_i,t_i) - u_i)^2 \quad \text{and}$$
$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}}\sum_{j=1}^{N_{\text{PDE}}}\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2}\right)^2|_{(x_j,t_j)}.$$

From Karniadakis et al., Nature Review Physics, 2019

Solving ODEs using R

IL2-IL2R-signaling pathway



Reaction species

L: ligands
R: receptors
C: ligand-receptor complexes
G: gene
P: gene products

Kinetic parameters

k_{syn}= 2000 (molecules/h)
k_{dec}= 0.05 /h

R coding

Please refer to a code file “ODE_lab.R”

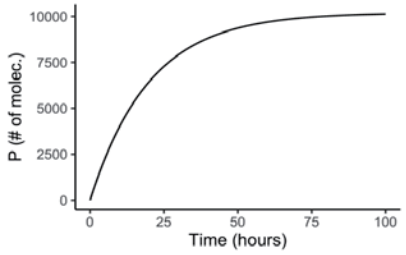
```
# Model parameters
parameters <- c(k_syn = 100*400*0.05,
                K_syn = 2000,
                n_syn = 2,
                k_deg = 0.05,
                k_on = 1.1 * 10^11,
                k_off = 0.83,
                L = 1*10^-12, # 1 pM
                R_tot = 10000)
```

$$K_d = \frac{k_{off}}{k_{on}} = 7.5\ pM$$

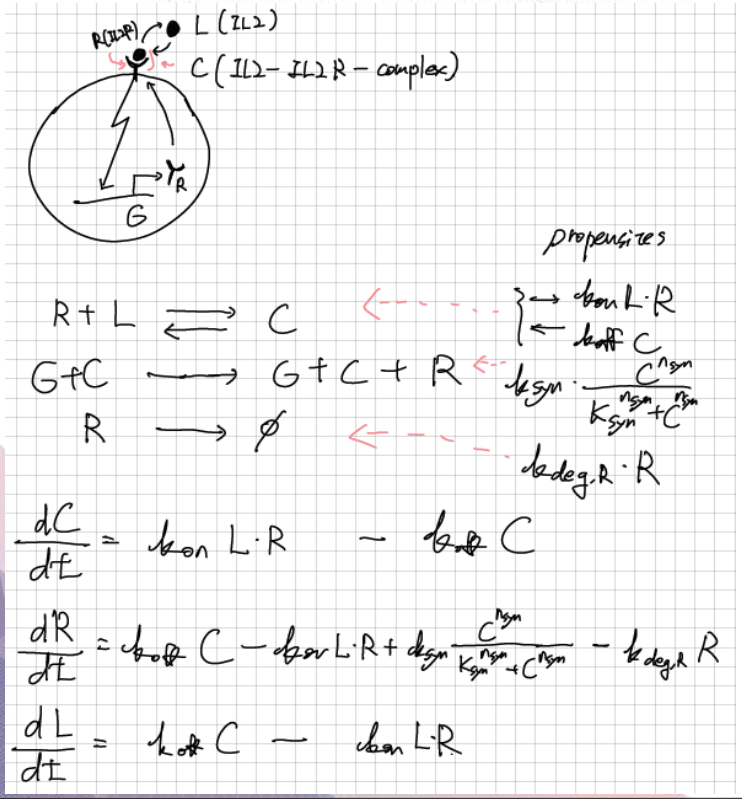
```
# Initial condition
state <- c(P = 0)

# Time specification
times <- seq(0,100, by =0.01)

# Model simulation
traj_P <- ode(y = state,
              times= times,
              func = mod_IL2_IL2R_signal,
              parms = parameters)
```

```
# Model equations
mod_IL2_IL2R_signal <- function(t, state, parameters){
  with(as.list(c(state, parameters)),{
    C <- R_tot*L/(k_off/k_on+L)

    dP_dt <- k_syn*C^n_syn/(K_syn^n_syn+C^n_syn) - k_deg*P
    list(c(dP_dt))
  })
}
```

```
# plot
ggplot(as.data.frame(traj_P), aes(x = time, y = P)) +
  geom_line() +theme_classic() +
  xlab("Time (hours)") + ylab("P (# of molec.)")
```



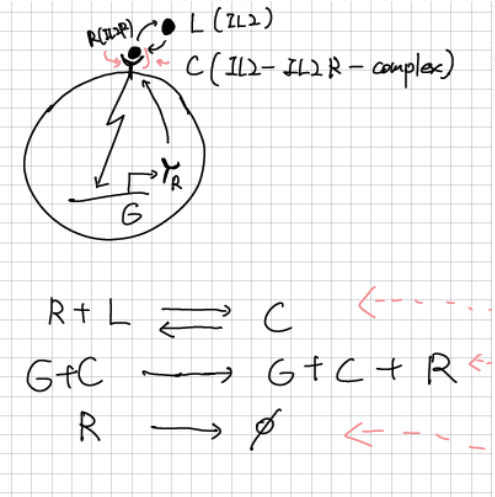
IL2-IL2R-signaling pathway with positive feedback



Reaction species

L: ligands
R: receptors
C: ligand-receptor complexes
G: gene
P: gene products

IL2-IL2R-signaling pathway with positive feedback



$$\frac{dC}{dt} = k_{on} L \cdot R - k_{off} C$$

$$\frac{dR}{dt} = k_{off} C - k_{on} L \cdot R + k_{syn} \frac{C^{n_{syn}}}{K_{syn}^{n_{syn}} + C^{n_{syn}}} - k_{deg,R} R$$

$$\frac{dL}{dt} = k_{off} C - k_{on} L \cdot R$$

fast slow

Quasi-steady-state approximation

$$C = \frac{R \, L}{K_d + L}$$

$$\frac{dR}{dt} = k_{syn} \frac{C^{n_{syn}}}{k_{syn}^{n_{syn}} + C^{n_{syn}}} - k_{deg,R} R$$

after dropping the subscript "tot" from R

R coding

Please refer to a code file "ODE_lab.R"

```r
# Model parameters
parameters <- c(k_syn = 100*400*0.05,
                K_syn = 2000,
                n_syn = 2,
                k_deg_R = 0.05,
                k_on = 1.1 * 10^11,
                k_off = 0.83,
                L = 1*10^-12)
```

```r
# Model equations
mod_IL2_IL2R_signal_R <- function(t, state, parameters){
  with(as.list(c(state, parameters)),{
    C <- R*L/(k_off/k_on+L)
    dR_dt <- k_syn*C^n_syn/(K_syn^n_syn+C^n_syn) - k_deg_R*R
    list(c(dR_dt))
  })
}
```

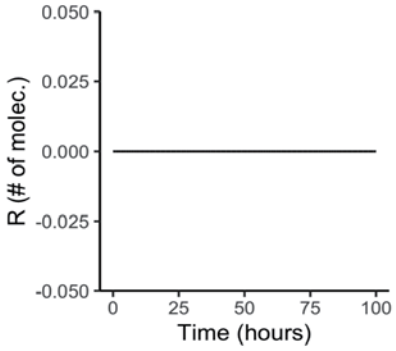
$$K_d = \frac{k_{off}}{k_{on}} = 7.5\ pM$$

```r
# Initial condition
state <- c(R = 0)

# Time specification
times <- seq(0,100, by =0.01)
```

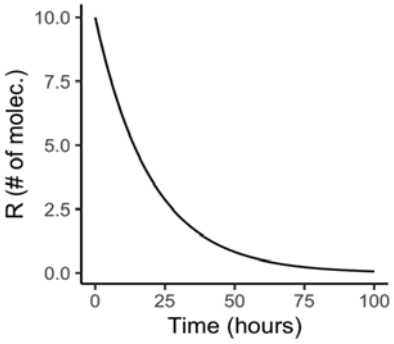
```r
# Model simulation
traj_R <- ode(y = state,
              times= times,
              func = mod_IL2_IL2R_signal_R,
              parms = parameters)
# Plot
ggplot(as.data.frame(traj_R), aes(x = time, y = R)) +
  geom_line() + theme_classic()
```

With different initial conditions, the model behaves very differently

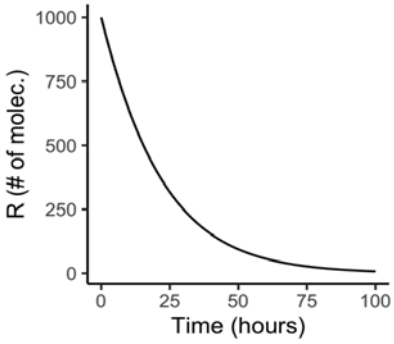
Initial condition R = 0



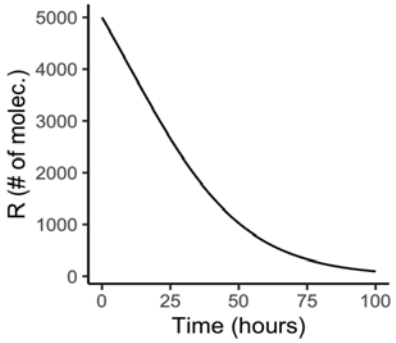
Initial condition R = 10



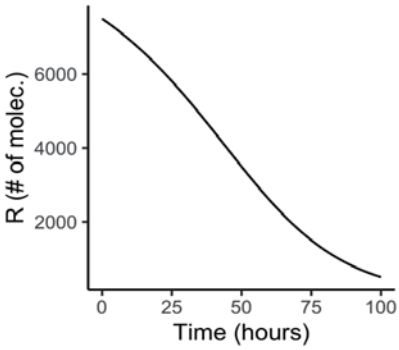
Initial condition R = 1000



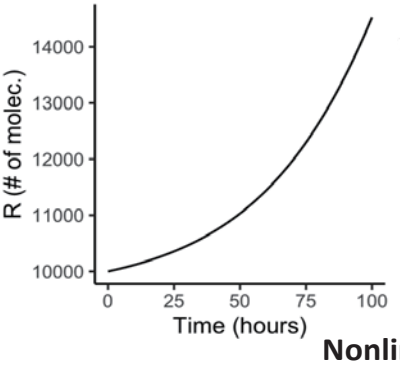
Initial condition R = 5000



Initial condition R = 7500



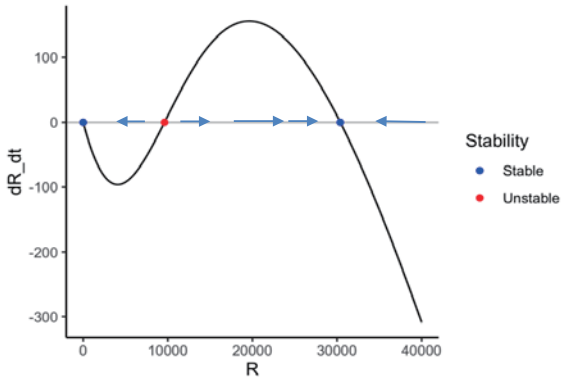
Initial condition R = 10000

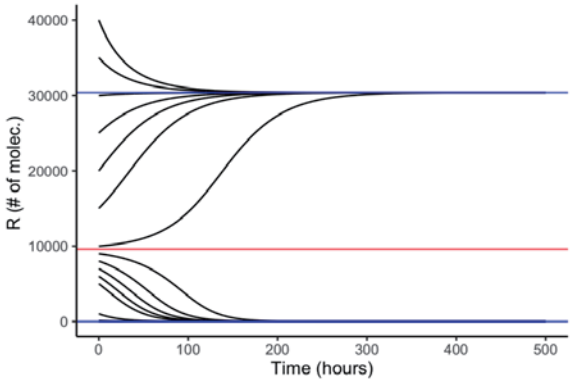


Nonlinear!

Phase portrait to analyze nonlinear systems (code available in ODE_lab.R)

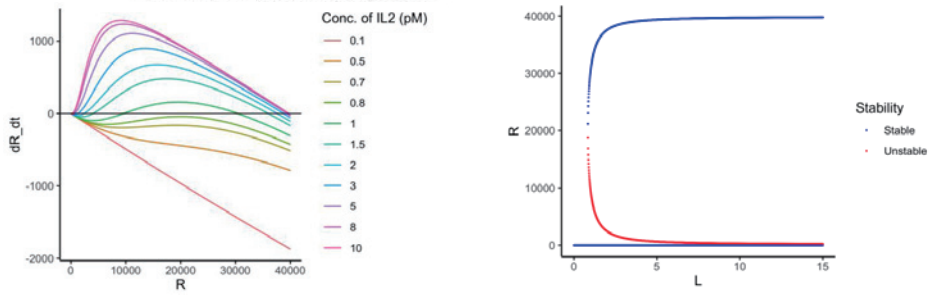
Plot $\frac{dR}{dt} = f(R, t, parameters)$



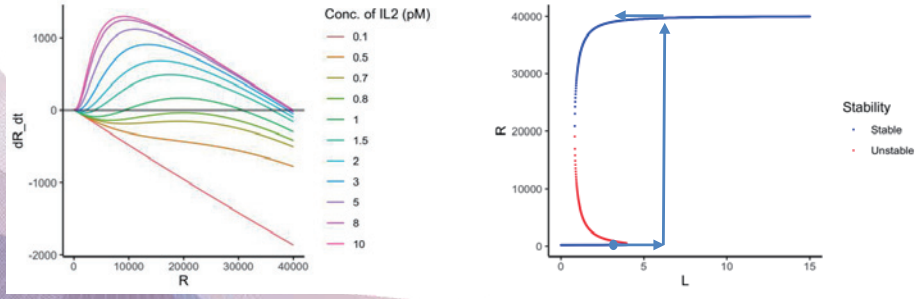


Bifurcation (code available in ODE_lab.R)

If parameters vary, the number and/or the types of fixed points can change
→ Bifurcation: one of the origin of many abrupt or nonlinear behaviors in the immune system shaped by nonlinear feedback loops



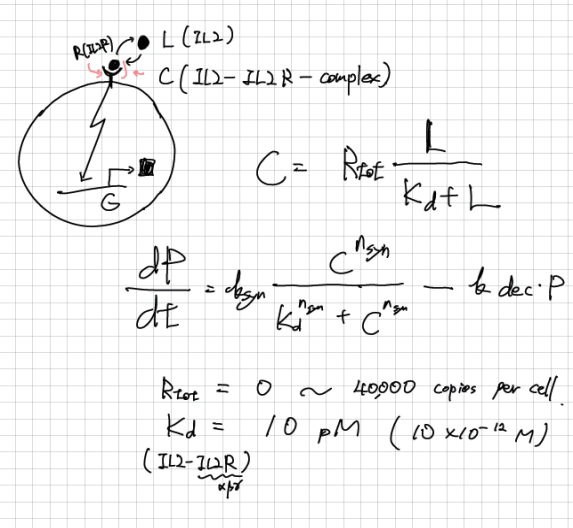
Add basal synthesis of IL2R (20 /h)



hysteresis

Solving differential equations using neural networks (physics-informed neural network)

Example from IL2-IL2R signaling (code avaialbe in PINN_lab.ipynb)



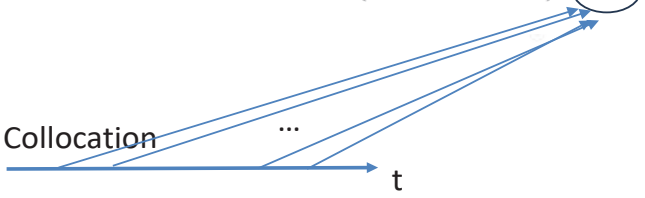
A surrogate neural netwok that learn the PDE solution $u(x,t)$ $NN(x,t,\theta)$

Construct a loss function $\mathcal{L} = w_{\text{data}}\mathcal{L}_{\text{data}} + w_{\text{PDE}}\mathcal{L}_{\text{PDE}}$

where

$$\mathcal{L}_{\text{data}} = \frac{1}{N_{\text{data}}}\sum_{i=1}^{N_{\text{data}}}(u(x_i,t_i) - u_i)^2 \quad \text{and}$$

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}}\sum_{j=1}^{N_{\text{PDE}}}\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2}\right)^2_{(x_j,t_j)}$$



Random sampling from the domain of independent variables (time, space)

References

- https://repository.kisti.re.kr/bitstream/10580/19182/1/KISTI%20이슈브리프%20제74호.pdf
- https://cran.r-project.org/web/packages/deSolve/vignettes/deSolve.pdf