Modeling Biological Systems - from Mechanistics to Machine Learning

Inaugural dissertation

for the attainment of the title of doctor
in the Faculty of Mathematics and Natural Sciences
at the Heinrich Heine University Düsseldorf

presented by Linlin Zhao

from Hunan

Düsseldorf, November, 2019

from the Institute of Mathematical Modeling of Biological Systems
at Heinrich-Heine-Universität Düsseldorf
Universitätstr. 1
40225 Düsseldorf

Published by permission of the
Faculty of Mathematics and Natural Sciences at
Heinrich-Heine University Düsseldorf

Supervisor: Prof. Dr. Markus Kollmann

Date of oral examination:

Declaration

Statement of authorship

I hereby declare that this dissertation is the result of my own work. No other person's work has been used without due acknowledgment. This dissertation has not been submitted in the same or similar form to other institutions. I have not previously failed a doctoral examination procedure.

Linlin Zhao

Düsseldorf, November, 2019

Acknowledgement

I have spent three wonderful and fruitful years in the Institute of Mathematical Modeling of Biological Systems. I was very lucky to have Markus as my supervisor. His great passion in research is always the source of inspiration and courage for me to overcome scientific challenges. His insightful directions always guide me towards the right spot of the projects. Especially, I remember the valuable days when I got criticized for not understanding the equations that I had solved. That helped me grow a lot. Thank you, Markus!

I am very grateful to Prof. Kai Stühler as well. Kai had been very supportive and walked me through the journey of building predictive models for secretory proteins. He showed me a different kind of flavour of doing research, which is also very inspiring. Thank you, Kai!

And Chris, you have been a great colleague and friend. Your Chris-style jokes made lunches in Mensa more enjoyable. Your corrections for the first draft of this thesis made it completely rewritten and improved to next level. And also, your home-made waffle was very delicious. Big thanks to you, Chris!

Then, sepass, dadash and abji! Nima the physicist, you are a great teacher with excellent ability to make complex concepts easy for anyone with any background. Nima the computer scientist and the philosopher, you are a truly badass and always helpful. I feel lucky to have worked with you. Nadia, Armin, Rahil and Sara, it is very precious to have you as colleagues and friends. Thank you all, my Iranian friends!

I would also like to thank Prof. Oliver Ebenhöh for reviewing my thesis. I cherish the valuable comments he wrote. And Prof. Laura Rose, Prof. Holger Schwender, and Prof. Egger, thank you very much for being on my defense committee.

Lastly, I would like to give my sincere thanks to all my colleagues in Molecular Proteomics Laboratory. Especially, thank Dr. Gereon Poschmann for his help and support.

In memory of my father, Fenping Zhao

To my mother, Qiulian Peng

To my wife, Tingting

To my boys, Kaixiang, Baoxiang and Ruixiang

A note to the readers

In order to quickly access relevant details, I would like to kindly invite readers to first have a look at this note before further reading.

The general introduction **Section 1.1** gives the basic concepts about modeling, complexity properties of modeling biological systems and the differences between mechanistic modeling and machine learning in biology.

The thesis includes six publications or submitted manuscripts:

- *An Operator-theoretic Approach to Synchronization of Dynamically Coupled Biological Rhythms*, published in Chinese Control Conference 2016, P. 45-52

  This paper relied on control theory and dynamical systems. Though it is hard to cover all technical details in the paper for general audience, the essentials of dynamical systems representations and stability analysis are introduced in **Section 1.2**.

- *Information integration and decision making in flowering time control*, submitted to Plos One, P. 55-86

  The manuscript involves probability theory (**Section 1.3.1, 1.3.3**), particle swarm optimization (**Section 1.5**), and artificial neural networks (**Section 1.4.3**).

- *Predicting gene expression level in E. coli from mRNA sequence information*, IEEE CIBCB 2019, P. 89-118

  The manuscript involves gradient boosting trees (**Section 1.4.4**) and model selection (**Section 1.4.5**). The biological background was introduced in **Section 1.6**.

- *Predicting eukaryotic protein secretion without signals.*, BBA-Proteins and Proteomics 2018, P. 121-129

  This is a review paper on different computational tools for predicting protein secretions without clear signals from protein sequences.

- *OutCyte: a novel tool for predicting unconventional protein secretion*, accepted by Scientific Reports, P. 130-160

The manuscript used gradient boosting trees (**Section 1.4.4**) and convolutional neural networks (**Section 1.4.3**). The biological background is in **Section 1.6**.

- *Automated computer-based detection of encounter behaviours in groups of honeybees*, 2017, Scientific Reports, P.163-172

  The paper relied on a previously published interactive machine learning framework, which relied on decision trees (**Section 1.4.4**).

From **Chapter 2** to **Chapter 6**, the manuscripts are presented with a short summary at the beginning of each chapter. The publication status and my contributions are stated before each manuscript.

Finally, **Chapter 7** summarizes the whole thesis.

Contents

1 Introduction

> *Wir müssen wissen, wir werden wissen.*
>
> — **David Hilbert**

1.1 General introduction

What is modeling?

Supposing we want to know how fast the enzymes in our stomach catalyze the digestion of the proteins in our food, we first need to understand in general how enzymatic reactions work. As early as 1903, Henri [Hen03] discovered that the enzymatic reactions were initiated by a binding interaction between enzymes and substrates. Later in 1913, Michaelis and Menten [MM13] extended Henri's discovery and mathematically described the kinetics of the enzymatic reactions. According to their findings, the enzymes (E) in the stomach bind to the proteins (S) to form the complexes (ES) which in turn produce peptides as the products (P) [Wike]. In reaction form, it can be represented as

$$E + S \underset{k_r}{\overset{k_f}{\rightleftharpoons}} ES \xrightarrow{k_{cat}} E + P, \tag{1.1}$$

where k_f, k_r and k_{cat} denote the forward rate, reverse rate and catalytic rate respectively.

Due to *the law of mass action*, which says that the reaction rate is proportional to the product of the concentrations of the reactants [VMO15], the reaction (1.1) can be described in mathematical equations as

$$\begin{aligned}
\frac{d[E]}{dt} &= -k_f[E][S] + k_r[ES] + k_{cat}[ES] \\
\frac{d[S]}{dt} &= -k_f[E][S] + k_r[ES] \\
\frac{d[ES]}{dt} &= -(k_r + k_{cat})[ES] + k_f[E][S] \\
\frac{d[P]}{dt} &= k_{cat}[ES],
\end{aligned} \tag{1.2}$$

where $[\cdot]$ denotes the concentration of the corresponding chemical substance, and the derivative $d[\cdot]/dt$ represents the change rate of the substance with respect to time. The positive terms on the right-hand side of the equations increase the change rates while the negative terms decrease them.

Up to now, the digestion process of proteins in our stomach has been *modeled* in reaction form (1.1) and mathematical equations (1.2). Essentially, *modeling* is to abstract the essentials from "real world" objects or phenomena to build their representations [Uni; MP12]. Models enable us to investigate ideas for generating scientific hypotheses [Mar17; BL16; MP12]. The models (1.1) and (1.2) have captured the key steps in enzymatic catalysis, without considering other non-essential facts such as how the enzymes have been produced, what proteins are present and so on. More specifically, the model (1.2) is a mathematical model which uses mathematics to describe the system of digesting proteins in the stomach, which involves proteins as the system input and peptides as the output.

In order to obtain the production rate $d[P]/dt$ from (1.2), further mathematical analyses require assumptions related to the system details. This is because the system of equations (1.2) is nonlinear due to the product terms $[E][S]$ and a direct solution is difficult to obtain. The key assumption is the *steady state approximation* which states that the concentration of the complex ES will rapidly reach

its steady state. Thus, $[ES]$ is regarded as a constant. And note that the total enzyme concentration is $[E]_T = [ES] + [E]$. Consequently, the production rate can be derived as

$$\frac{d[P]}{dt} = \frac{V_{max}[S]}{K_M + [S]}, \tag{1.3}$$

where $V_{max} = k_{cat}[E]_T$ is the maximum reaction rate and $K_M = (k_r + k_{cat})/k_f$ is the Michaelis constant. Equation (1.3) is the well-known *Michaelis-Menten Equation*, which states that the production rate $d[P]/dt$ depends only on the concentration of the input substrates $[S]$. $[P]$ and $[S]$ are system variables and K_M and V_{max} are system parameters. To determine the real value of production rate, not only the independent variable $[S]$ need to be measured, but also the values of the parameters V_{max} and K_M should be determined. To determine the parameters of a system of equations is often termed *parameterization* or *parameter fitting*, which is a key step in mathematical modeling methods.

The Michaelis-Menten equation was reported in 1913. Earlier than that, the use of mathematics to model biological systems can arguably [Hof15; Pea96; Mue79; Deu] date back to 1879 when Fritz Müller's used mathematics for his discovery of *Müllerian mimicry* [Mue79; Wikg] which describes phenomena such as the bees have evolved similar looking and stings as the wasps to avoid predators. Last decades have seen rapid growth of application of modeling in biological systems [Wikh; Hop95; May04; Hof15; Nob02; Kit02a; Mar17]. Especially with the massive data produced by high-throughput genomics and proteomics studies, systems can be investigated on much larger scales, for instance, systems with a large number of interconnected components. Systems biology and computational biology are fields that extensively use mathematical models to study complex biological systems. For example, a series of enzymatic reactions can form a metabolic pathway and then all the pathways will constitute the metabolic network which involves numerous reactants, enzymes and products. Modeling the metabolic network systematically can help understanding, for example, the causes of human diseases like obesity and diabetes [Lee+08; Ros+00], and the regulation of sugar utilization in yeast [Ide+01].

Why modeling biological systems is hard?

Biological systems are complex. A complex system is composed of a large number of interacting components that have a collective behavior as a whole [MP12; RHS07]. For example, the human brain is a complex system with billions of neurons connected with trillions of synapses (interconnections), giving the brain functionalities which individual neurons do not possess. Complex systems are examples of *nonlinear dynamical systems* whose states evolve over time according to certain rules [RHS07; Bod]. They are not linear because of not satisfying the *superposition principle* which characterizes linear systems. The superposition principle says that if A and B are solutions of a system, so is the sum $A + B$, which implies that a linear system can be solved by combining solutions of its subsystems. Clearly, the human brain and metabolic network violate this principle. The study of complex systems usually involves large number of variables for describing the system components, which lead to the high dimensionality of biological systems. The nonlinearity and high dimensionality are common characteristics of biological systems.

To tackle the difficulties in modeling biological systems, it is of key importance to recognize in general the complexity features of them. Besides the aforementioned nonlinearity and high dimensionality, biological systems involves different temporal and spatial scales [BC11; Kit02a]. For example, on the temporal scale, the turnover time of *Adenosine triphosphate* (ATP), the energy storage units of life systems, is around $1s$, while it is 4 months for red blood cells [Mil+09]. The

spatial scales can span from water molecules with size of $10^{-9}nm$ to *e. coli* with size of $10^{-6}\mu m$ to the large mammals. Though these biological objects or phenomena should obey certain physical laws, it is different from the well-defined approach in physics. Different areas in physics are based on the characteristic lengths of objects or characteristic time of phenomena [MP12]. Modeling on the *Intracellular* scale involves molecules or compartments within a cell. The translation from messenger RNA (mRNA) to proteins is a typical intracellular mechanism, whose dynamics involves a big protein complex called ribosome to bind and move along the mRNA. Systems on the cellular or intercellular level consist of large number of interacting cells and molecules. For example, the *quorum sensing* in bacterial cultures describes the phenomenon that individual bacteria produce and release molecules to culture medium to sense population density in order to regulate intracellular physiological activities [MB01]. The study of biological systems usually requires multiscale approach. Though creating a model of ecosystem dynamics may not need to be grounded on the molecular dynamics of the cell, the structure of macroscopic tissues depends on the intracellular dynamics like the gene expressions [BC11].

Another essential feature of biological systems is their *robustness* to maintain their states and functions under different environmental conditions. It usually involves systems control, redundancy, structural stability and modularity [Kit02b]. For example, the bacteria *E. coli* are able to reply on different carbon sources to survive in different environments. When switching from a glucose-rich environment to a fructose-rich one, *E. coli* need to sense the changing of environment and adapt their genetic regulatory network to such changes. Keeping some redundant genes can help *E. coli*' survival in different environments. And in terms of system control, the *chemotaxis* mechanism can guide them to move towards the place with high concentration of food and away from poisonous chemicals [WA04].

Biological *Heterogeneity* is also common and has been speculated as a fundamental property of biological systems [AW10; Rub90]. Generally, the heterogeneity can be stated as the differences among biological entities which belong to the same biological structure. There are many different levels of heterogeneities. For example, mutations in different genes in human may lead to the same disorder [MK10], which is referred to as genetic heterogeneity. On the contrary, the genetically identical cells often show significant differences in gene expression and phenotypic traits, which is referred to as phenotypic heterogeneity [Ack15]. Further, differences on the basis of molecules and cells can be termed molecular and cellular heterogeneities respectively. The heterogeneities are potentially beneficial for the biological systems to cope with fluctuating environments, to increase biological diversity and to increase survival or growth through natural selection [Ack15]. To model biological systems, it is important to understand the possible causes and effects of heterogeneities and to determine if the individual differences contain meaningful biological information. For instance, the plant *Arabidopsis thaliana* can have different flowering times for the same accession type in very close regions, which may be caused by climate or geographical differences. In this case, environmental data and flowering data need to be collected analyze the information processes for plants' flowering behaviors.

The *nonlinearity, high-dimensionality, multiscales, robustness and heterogeneities* are the most recognized complex features of biological systems [Orz12; BC11; MP12; Ree04]. Though they pose difficulties on mathematical modeling, understanding them can be helpful for reducing complexities. For example, the models can be on the proper scale for certain problems of interest to cope with the multiscale feature, on different functional subsystems to reduce dimensionality and nonlinearity, or based on sensible biological assumptions to reduce overall complexities.

Mechanistic modeling and machine learning modeling

Suppose the following question is investigated: *Does chocolate reduce blood pressure?* In terms of modeling, its answer can be approached in two ways.

The first is the mechanistic modeling. Like modeling the enzymatic reactions, all components involved in the system which facilitates the reduction by chocolates and their mechanistic interconnections need to be understood. Based on that, the causal relations between chocolates and the reduction of blood pressure can be mathematically formulated. Then with existing relevant data or dedicated experimental data, the model can be used to make deductions and draw conclusions. However, such a mechanistic model is hard to obtain due to many unknown details. For example, the high blood pressure can be resulted from disorders of different organs, of which the mechanisms are probably different. Then detailed steps of chocolates to reduce blood pressures become hard to determine.

The second is the statistical modeling which counts uncertainties in system variables and aims at capturing the relationships between the variables without involving details of system components. By *meta analysis*, a statistical analysis by combining data from multiple studies [Hai10], researchers found that different dosages of dark chocolates were indeed correlated with the reduction of blood pressures [Rie+10]. Essentially, statistical models are models of data, of which the objective is to mathematically approximate the truth of the data generating process so as to make predictions. For example, if the regression model in [Rie+10] approximated the truth well, for a given dosage of dark chocolate, the model should be able to predict to what extent the blood pressure can be reduced. However, due to the quality and quantity of data and the complexity of the problem, such models are usually far from accurate to make reliable predictions.

Machine learning is a discipline closely related to statistical modeling, with focus on building computer systems to make predictions. Tom Mitchell [Mit17] provided a widely accepted definition by describing machine learning as a computer program which learns from *experience* with respect to some *tasks* and *performance* measures. For example, to build learning models to recognize cats (task) from images, a huge number of images (experience) are needed for the models to capture the general patterns of cats such that the learned models are able to recognize cats in unseen images (performance). Machine learning has been widely used for decades [Mit17; Mur12], and is becoming more and more popular due to the success of its subfield, deep learning, in a wide range of applications [LBH15a]. Deep learning are the machine learning algorithms for learning multiple levels of representations from the data. For example, for recognizing cats in images, deep learning models do not require human-engineered characteristics of cats, but learn first the basic representations like the curves of ears and eyes and then assemble them to next levels of higher representations [Le+11; Qin+18].

Statistical modeling and machine learning are largely overlapped but have slight distinction in emphasis and terminology [Sha; Ros; BAK18; Mur12]. They are both concerned with the same question: how do we learn from data [Was]? For example, given measured expression levels of a set of genes, the primary concern is what regulatory factors underlying the gene sequences are related to the expression levels. Statistics emphasizes formal inferences about which the regulatory factors are associated with a specific probability model, while machine learning emphasizes predicting the expression levels of unseen gene sequences using general purpose algorithms to find patterns from the measured sequences. Though the emphases are different, they are both concerned with trying to find out how the gene expressions work and what will happen next. One notable distinction is that

specific statistical models usually rely on problem-specific assumptions whereas machine learning makes minimal assumptions about data generating systems [BAK18]. Due to this distinction, the use of "machine learning" in this thesis is more suitable as the modelings without mechanistic relations in later chapters used general purpose algorithms to learn patterns from data. Another reason for the favor of "machine learning" is the rapidly growing mass of data from high-throughput techniques, which requires efficient and powerful general-purpose learning algorithms to learn patterns out of the massive data.

The rest of this chapter will introduce the basic principles and knowledge related to methods and results but not explicitly explained in the manuscripts of later chapters.

1.2 Dynamical systems

> *System dynamics provides a common foundation that can be applied wherever we want to understand and influence how things change through time.*
>
> *Jay W. Forrester*

Dynamics is the subject dealing with changes in systems over time [Str18]. Dynamical systems are ubiquitous, for instance, chemical kinetics, mechanical systems, growing cell cultures and cardiovascular cycles. Formally, a dynamical system is a system whose state evolves with time over a *state space* according to *fixed rules* [Nyk]. The state space is the collection of all possible configurations of the system. For instance, if an oak tree is regarded as a dynamical system for the interest of modeling its height, then depending on different environmental conditions, all possible growth curves over time are the state space of the system.

In this section, the representations and stability of dynamical systems are introduced for *Chapter 2* and the stochastic modeling in *Chapter 3*.

1.2.1 Dynamical system representations

Differential equations

The simple harmonic oscillator, a mass attached to a spring without any other driving or damping forces (Fig. 1.1), is used as an example to introduce the notions about dynamical systems.

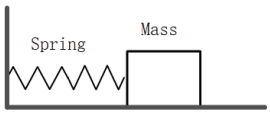


Fig. 1.1: A simple harmonic oscillator consists of a mass on a smooth surface and a spring with one end fixed on the wall.

The rules for governing the oscillator are Newton's second law and Hooke's law. Newton's second law states that an object's acceleration depends on its mass and the forces acted on it, and Hooke's law states that the deforming force from the relative small deformation of an object is proportional to the displacement of the deformation. According to the two laws, the dynamics of a simple harmonic oscillator can be modeled as

$$F = ma = m\frac{d^2x}{dt^2} = -kx, \tag{1.4}$$

where F denotes spring force, x the displacement, m the mass, a the acceleration and t the time.

By denoting the second derivative d^2x/dt^2 as $\ddot{x}$, Eq. (1.4) can be written as

$$m\ddot{x} + kx = 0, \tag{1.5}$$

which is a *second-order ordinary differential equation*, because the derivatives is only with respect to time t. A harmonic oscillator is a linear system because of the constant coefficients m and k, and (1.5) can be solved analytically as

$$x(t) = A\cos(\sqrt{\frac{k}{m}}t + \phi) \tag{1.6}$$

which shows the oscillation frequency is $\sqrt{k/m}$. To get the displacement $x(t_1)$for a given time t_1, we ought to determine the oscillation amplitude A and phase constant ϕ. The amplitude A is determined by the total energy in the system, for instance, the work has been done to pull the spring at the beginning. The phase constant is determined by the system states at the initial time $t = 0$.

Real systems, especially life systems, are usually high-dimensional and nonlinear, leading to more complex differential equation representations. The analytical way like dealing with simple harmonic oscillator is usually difficult. Another two examples of more complex systems, the toggle switch [GCC00] and repressilator [EL00] can be found in the paper of Chapter 2.

State Space representation

The state space representation of a dynamical system is reformalizing a single higher-oder differential equation to a group of first-order ones by introducing extra state variables. Suppose the simple harmonic oscillator is perturbed by an external force $u(t)$, the new dynamics can be modified from (1.5) as

$$m\ddot{x} + kx = u. \tag{1.7}$$

If the displacement x is denoted as x_1, and the velocity $\dot{x}$ is denoted as x_2, then the acceleration is $\dot{x}_2$ and (1.7) can be rewritten as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{k}{m}x_1 + \frac{1}{m}u \end{aligned} \tag{1.8}$$

The state space representation is made up of the dynamics of displacement and velocity, which are both first-order differential equations. It becomes easy to see that the oscillator is a linear system as the terms at right hand side contain only first-order power. And further Eq. (1.8) can be written in matrix format as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u, \tag{1.9}$$

If we measure the displacement x_1 as system output y, then

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{1.10}$$

The equations (1.9) and (1.10) together define a linear time-invariant system (LTI) with single input u and single output y (SISO). A system is termed time-invariant when the relation between input and output are time-independent. [DB11; K. 09]

Though the perturbed SISO oscillator system is used to illustrate state space representations, its advantage is actually that it can be used for analyzing multiple input and multiple output (MIMO)

systems in the time domain [DB11; K. 09]. The paper in *Chapter 2* factorized the toggle switch and the repressilator models to state space form for analyzing their behaviors in a network.

Transfer function representation

For linear systems, transfer function is derived from the ratio of the Laplace transforms of the output to the input with all initial conditions assumed to be zero, which is instrumental for studying the relation between input and output [DB11; K. 09].

For real time function $f(t)$, i.e. $t > 0$, the one-sided Laplace transform is defined as

$$F(s) = \int_0^\infty f(t)e^{-st}dt. \tag{1.11}$$

Following this definition and assuming zero initial conditions, Eq. (1.7) can be transformed as

$$s^2mX(s) + kX(s) = U(s) \implies G(s) = \frac{X(s)}{U(s)} = \frac{1}{ms^2 + k}, \tag{1.12}$$

with $G(s)$ the transfer function denoting the relation between output displacement $X(s)$ and input external force $U(s)$ in frequency domain. In control theory the stability and response analyses based on transfer function were well-developed for SISO systems [K. 09]. For MIMO systems, state space representations are more popular.

1.2.2 Stability of dynamical systems

The stability of dynamical systems is critical in engineering as perturbations or noises are usually presented, which impose challenges on system design to sustain the stability under different circumstances. The term stability in dynamical system can be referred to as either the stability of motion or the stability of the equilibrium. The stability of motion is concerning the changes of system trajectories if the initial state or input is perturbed. If the system is settled at equilibrium (stationary state), it is a particular motion and the stability analysis is to investigate system behaviors after the equilibrium is perturbed.

Consider a generic dynamical system [Mel]

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= g(\mathbf{x}, \mathbf{u}) \end{aligned} \tag{1.13}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^k$ denote the state vector, input vector and output vector respectively. The relations among them are characterized by function $f(\cdot)$ and $g(\cdot)$. Note that all the vector variables are time dependent.

Motion Stability

To define the motion stability, the reference motion is introduced as

$$\tilde{\mathbf{x}}(t) = \psi\Big(t, t_0, \tilde{\mathbf{x}}(t_0), \tilde{\mathbf{u}}(\cdot)\Big), \tag{1.14}$$

for initial time t_0. Then the motion difference resulted from the initial state perturbations can be written as

$$\Delta\mathbf{x}_1(t) = \psi\Big(t, t_0, \tilde{\mathbf{x}}(t_0) + \Delta\mathbf{x}_1(t_0), \tilde{\mathbf{u}}(\cdot)\Big) - \tilde{\mathbf{x}}(t), \tag{1.15}$$

and the difference resulted from input perturbations is

$$\Delta\mathbf{x}_2(t) = \psi\Big(t, t_0, \tilde{\mathbf{x}}(t_0), \tilde{\mathbf{u}}(\cdot) + \Delta\tilde{\mathbf{u}}(\cdot)\Big) - \tilde{\mathbf{x}}(t). \tag{1.16}$$

The system (1.13) *motion is stable with respect to the initial state perturbations if*

$$\forall t_0, \forall\epsilon, \exists\eta > 0 \implies \|\Delta\mathbf{x}_1(t)\| < \epsilon, \forall t \le t_0 \text{ if } \|\Delta\mathbf{x}_1(t_0)\| < \eta \tag{1.17}$$

and is stable with respect to input perturbations if

$$\forall t_0, \forall\epsilon, \exists\eta > 0 \implies \|\Delta\mathbf{x}_2(t)\| < \epsilon, \forall t \le t_0 \text{ if } \|\Delta\mathbf{u}(t)\| < \eta. \tag{1.18}$$

Note that the equilibrium or stationary state of the system is a special case of (1.14).

Equilibria Stability

An equilibrium $\mathbf{x}^*$ is *attracting* if the perturbed equilibrium $\tilde{\mathbf{x}}^*$ is guaranteed to eventually converge back to x^*. Mathematically, if $\exists\delta$ such that $\lim_{t\to\infty}\mathbf{x}(t) = \mathbf{x}^*$ if $\|\mathbf{x}(0) - \mathbf{x}^*\| < \delta$ [Str18].

$\mathbf{x}^*$ is *Lyapunov stable* if the perturbed equilibrium $\tilde{\mathbf{x}}^*$ is close to and stays close to $\mathbf{x}^*$ for all future time. For example, a simple harmonic oscillator is Lyapunov stable.

$\mathbf{x}^*$ is *neutrally* stable if it is Lyapunov stable but not attracting. And furthermore $\mathbf{x}^*$ is *asymtotically* stable if it is both Lyapunov stable and attracting. For example, a damped harmonic oscillator is asymtotically stable after a perturbation, it will eventually converge to steady state due to friction.

Stability analysis of linear systems

For the simplicity of illustration and without loss of generality, consider a two-dimensional linear dynamical system

$$\mathbf{x}' = A\mathbf{x}, \tag{1.19}$$

where the constant matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ and state vector } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

According to the superposition principle, if the *eigenvalues* and *eigenvectors* of A are λ_1, λ_2 and $\mathbf{v}_1$, $\mathbf{v}_2$ respectively, then the general solution can be written as [Str18]

$$\mathbf{x}(t) = c_1 e^{\lambda_1 t}\mathbf{v}_1 + c_2 e^{\lambda_2 t}\mathbf{v}_2 \tag{1.20}$$

where c_1 and c_2 are constants depending on the initial conditions of the system. For real valued λ_1 and λ_2, if either of them is positive, then $\mathbf{x}(t)$ is exponentially growing as $t \to \infty$. Hence the equilibria of such systems are unstable.

In the case of complex eigenvalues, $\lambda_{1,2} = \alpha \pm i\omega$ and by Euler's formula, $e^{i\omega t} = \cos\omega t + i\sin\omega t$, $\mathbf{x}(t)$ is combination of terms involving $e^{\alpha t}\cos\omega t$ and $e^{\alpha t}\sin\omega t$. It can be seen that if $\alpha > 0$ then $\mathbf{x}(t)$ is growing as $t \to \infty$.

To sum up and generalize the stability conditions of n-dimensional linear system, if $\Re(\lambda_i) \leq 0, i = 1, \cdots, n$, then the system is stable; if $\Re(\lambda_i) < 0, i = 1, \cdots, n$, then the system is asymtotically stable, where $\Re$ stands for the real part of a complex number.

Lyapunov function method

A Lyapunov function a scalar continuous differentiable function defined on the state space of a dynamical system, which can be used to prove the stability of an equilibrium [Kha; Str18; Mat].

Consider a dynamical system of the form

$$\dot{\mathbf{x}} = f(\mathbf{x}), \quad \mathbf{x} = (x_1, \cdots, x_n) \tag{1.21}$$

with an equilibrium at $\mathbf{x}^*$.

Given a domain $D \subset \mathbb{R}^n$ including the equilibrium, if $V(\mathbf{x}), D \to \mathbb{R}$, a continuously differentiable function, has the following properties:

- *$V(\mathbf{x}) > 0$ for all points in D except $\mathbf{x}^*$, and $V(\mathbf{x}^*) = 0$; (V is positive definite.)*
- *$\dot{V} \leq 0$ for all points in D; ($\dot{V}$ is seminegative definite.)*

then the equilibrium $\mathbf{x}^*$ is stable. Moreover, if $\dot{V} < 0$ for all points in D except $\mathbf{x}^*$ ($\dot{V}$ is negative definite.), then $\mathbf{x}^*$ is asymtotically stable.

The given stability criteria using Lyapunov function is the so-called *Lyapunov function method*, which is not intuitive. But for mechanical or electrical systems, a Lyapunov function can be interpreted as an energy storage function. If the stored energy is neither decreasing or increasing, then the system stays near the equilibrium (Lyapunov stable). If the stored energy is dissipated, the system eventually converges to its equilibrium (asymtotically stable). For instance, a simple harmonic oscillator is Lyapunov stable, and a dampened harmonic oscillator is asymtotically stable. The simple harmonic oscillator in Eq. (1.7) can be written as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{k}{m} x_1, \end{aligned}$$

with the equilibrium at $(0, 0)$. The total mechanical energy consists of the potential energy of the spring and the kinetic energy of the mass, then the Lyapunov function can be taken as

$$V(x_1, x_2) = \frac{1}{2}\frac{k}{m}x_1^2 + \frac{1}{2}x_2^2. \tag{1.22}$$

It is obvious that $V(0, 0) = 0$ and $V(x_1, x_2) > 0$ for $(x_1, x_2) \neq (0, 0)$. And further

$$\begin{aligned} \dot{V}(x_1, x_2) &= \frac{\partial V}{\partial x_1}\dot{x}_1 + \frac{\partial V}{\partial x_2}\dot{x}_2 \\ &= \frac{k}{m}x_1 x_2 + x_2(-\frac{k}{m}x_1) = 0 \end{aligned}$$

Therefore, according to the Lyapunov function method, the simple harmonic oscillator is stable. And for the dampened harmonic oscillator, a damping term renders the differentiation $\dot{V}$ negative, which asserts that the dampened oscillator is asymtotically stable.

The Lyapunov method has the advantage of not requiring detailed solution of the systems to determine the stability, but the disadvantage is that there is no systematic way to construct a Lyapunov function for a specific system. Despite the disadvantage, Lyapunov function played a crucial role in stability analysis of dynamical systems, especially nonlinear systems where solutions are usually hard to obtain.

The input-output stability of dynamical systems

Given an input to a dynamical system, if its output falls in a range confined by the input, then the system can be defined as a stable one without requiring the knowledge of the internal structure of the system. The stability from the perspective of input-output relations is formally introduced in this section. Its relation to Lyapunov function method will also be shown.

Consider a dynamical system of the form

$$\mathbf{y} = \mathcal{H}\mathbf{u}, \tag{1.23}$$

where $\mathcal{H}$ denotes an operator specifying output $\mathbf{y}$ in terms of input $\mathbf{u}$. The signals $\mathbf{u}$ and $\mathbf{y}$ are functions that map the time interval $[0, +\infty)$ to Euclidean space. For example, the input signal $\mathbf{u}(t)$ can be defined as $[0, +\infty) \to \mathbb{R}^m$ which maps the time interval to m dimensional vector spaces.

Typical spaces for signal $\mathbf{u}$ include the $\mathcal{L}_\infty^m$ space, which is the space of continuous, bounded functions with their norms defined as

$$\|\mathbf{u}\|_{\mathcal{L}_\infty} = \sup_{t \geq 0} \|\mathbf{u}(t)\| < +\infty \tag{1.24}$$

and the $\mathcal{L}_2^m$ space, which is the space of piecewise continuous, square-integrable functions with their norms defined by

$$\|\mathbf{u}\|_{\mathcal{L}_2} = \sqrt{\int_0^\infty \mathbf{u}^T(t)\mathbf{u}(t)dt} < +\infty \tag{1.25}$$

and more generally the $\mathcal{L}_p^m$ space, which is the space of all piecewise continuous functions with their norms defined by

$$\|\mathbf{u}\|_{\mathcal{L}_p} = \Big(\int_0^\infty \|\mathbf{u}(t)\|^p dt \Big)^{\frac{1}{p}} < +\infty. \tag{1.26}$$

For example, the signal $u(t) = t$ does not belong to the space $\mathcal{L}_\infty$ as it is not upper bounded in the time interval $[0, +\infty)$. However, its truncation

$$u_\tau(t) = \left\{ \begin{array}{ll} t & 0 \leq t < \tau \\ 0 & t > \tau \end{array} \right\}$$

belongs to $\mathcal{L}_\infty$ for every finite τ. To cope with signals like $u(t) = t$, the extended space $\mathcal{L}_e^m$ of $\mathbf{u}$ is introduced as

$$\mathcal{L}_e^m = \{\mathbf{u} : \mathbf{u}_\tau \in \mathcal{L}^m, \forall \tau \in [0, +\infty)\} \tag{1.27}$$

where

$$\mathbf{u}_\tau = \left\{ \begin{array}{ll} \mathbf{u}(t) & 0 \leq t < \tau \\ 0 & t > \tau \end{array} \right\}$$

and $\mathcal{L}^m$ is the abbreviation of $\mathcal{L}_p^m$ for $1 \leq p < \infty$. It can be seen that $\mathcal{L}_e^m$ is equivalent to $\mathcal{L}^m$ when τ goes to $+\infty$.

With above definitions the input-output stability of systems (1.23) can be defined as following [Kha]:

A mapping $\mathcal{H} : \mathcal{L}_e^m \to \mathcal{L}_e^q$ is $\mathcal{L}$ stable if there exists a class κ function α and a constant $\beta \geq 0$ such that

$$\|(\mathcal{H}\mathbf{u})_\tau\|_\mathcal{L} \leq \alpha(\|\mathbf{u}_\tau\|_\mathcal{L}) + \beta \tag{1.28}$$

for all $\mathbf{u} \in \mathcal{L}_e^m$ and $\tau \in [0, +\infty)$. It is finite $\mathcal{L}$-gain stable if there exists $\gamma, \beta \geq 0$ such that

$$\|(\mathcal{H}\mathbf{u})_\tau\|_\mathcal{L} \leq \gamma\|\mathbf{u}_\tau\|_\mathcal{L} + \beta. \tag{1.29}$$

The class κ function α is strictly increasing and $\alpha(0) = 0$.

For simplicity, the relation of input-output stability to Lyapunov function is shown for linear systems and $\mathcal{L}_2$ signals [Isi]. The relation also holds for nonlinear systems [DZK12; HM80]. Consider a linear system of the form

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} &= C\mathbf{x} + D\mathbf{u}, \end{aligned} \tag{1.30}$$

with state $\mathbf{x} \in \mathbb{R}^n$, input $\mathbf{u} \in \mathbb{R}^m$ and output $\mathbf{y} \in \mathbb{R}^q$.

Suppose the linear system is asymtotically stable, then there exists a positive definite matrix P such that the quadratic Lyapunov function $V(\mathbf{x}) = \mathbf{x}^T P\mathbf{x}$ satisfied the dissipation inequality

$$\frac{\partial V}{\partial \mathbf{x}}\dot{\mathbf{x}} \leq -\epsilon\|\mathbf{x}\|^2 + \gamma^2\|\mathbf{u}\|^2 - \|\mathbf{y}\|^2. \tag{1.31}$$

For $u = 0$ this inequality reduced to

$$\frac{\partial V}{\partial \mathbf{x}}A\mathbf{x} \leq -\epsilon\|\mathbf{x}\|^2 < 0 \tag{1.32}$$

which is consistent with the asymtotical stability assumption. The dissipation inequality is originated from the fact that for a physical system supplied with external energy, its energy storage rate should be bounded by the supply rate.

Integration of (1.31) on the interval $[0, T]$ leads to

$$V(\mathbf{x}(T)) \leq V(\mathbf{x}(0)) + \gamma^2 \int_0^T \|\mathbf{u}\|^2 dt - \int_0^T \|\mathbf{y}\|^2 dt \tag{1.33}$$

for any initial state $\mathbf{x}(0)$. Moreover, since $V(\mathbf{x}) \geq 0$, assuming $\mathbf{x}(0) = 0$ yields

$$\int_0^T \|\mathbf{y}\|^2 dt \leq \gamma^2 \int_0^T \|\mathbf{u}\|^2 dt. \tag{1.34}$$

According to the definition of $\mathcal{L}_2$, (1.34) can be written as

$$\|\mathbf{y}\|_{\mathcal{L}_2} \leq \gamma\|\mathbf{u}\|_{\mathcal{L}_2}. \tag{1.35}$$

Then according to the definition of *input-output stability*, the linear system (1.30) is *finite $\mathcal{L}_2$ gain stable*. Thus, the connection between input-output stability and Lyapunov function method has been established.

1.3 Probability Theory

> *Probability theory is nothing but common sense reduced to calculation.*
>
> *Pierre-Simon Laplace*

The concept of uncertainty was said to be as old as civilization [DS12] as humans have always had to deal with uncertainties from weather forecast and food supply to potential dangers. Life systems ought to make correct decisions in reacting to uncertainties in their environments in order to survive and reproduce. For instance, plants in temperate regions need to fight against the uncertain weather conditions in spring from year to year in order to make correct flowering decisions. Probability theory provides a consistent and solid mathematical foundation for quantifying and analyzing uncertainties. Paradigms as machine learning and stochastic modeling heavily rely on probability theory.

1.3.1 Probability and distributions

Concepts of probability

Interestingly, despite of the fundamental importance of probability in a wide range of disciplines, there is no consensus concept for it [DS12]. The *relative frequency* view of probability, i.e., probability as the ratio of number of occurrence of an event to the total number of trials, requires that the process should be repeated a large number times under similar conditions. This is usually referred to as frequentist view of probability. However a Bayesian would ask frequentists what is the probability of sun explosion tomorrow which cannot be repeated. Instead he interprets probability as degree of belief or plausibility. Besides frequentist and Bayesian, the third interpretation is based on the *equally likely outcomes* [DS12]. For example, each coin has only tail or head sides so the probability of tail is 0.5.

Different interpretations have their application cases but also drawbacks in other cases. The relative frequency of view is well rooted in some repeated control biological experiments but is not suited for unrepeatable events such as the death of a person or the explosion of the sun. The Bayesian interpretation is subjective and the degree of belief can be updated by observations. Meanwhile the subjectiveness renders it not able to maintain its consistency in experiments with high dimensional sample space.

Though different people can hold different interpretations of probability, the calculus of probability theory applies universally.

Mathematical formulation of probability

Closely related to set theory, the *sample space* Ω is defined as the set of all possible events from an experiment. For instance, the sample space for tossing a coin is $\{head,\ tail\}$. With the foundation on set theory, the set operations can be applied to events and the mathematical definition of probability can be stated as

The probability measure on a sample space Ω is the realization of numbers $p(A)$ for all events A such that the following axioms are satisfied:

Axiom I: $p(A) \geq 0$ for all A

Axiom II: $p(\Omega) = 1$

Axiom III: if events $\{A_1, A_2, A_3, \cdots\}$ are disjoint, then

$$p\Big(\bigcup_{i=1}^{\infty} A_i\Big) = \sum_{i=1}^{\infty} p(A_i). \tag{1.36}$$

Random variables and distributions

A random variable is real-valued function that quantifies the sample space by mapping the experiment outcomes to real numbers. Random variables can be discrete or continuous. Given a discrete random variable X, the set of all possible values $\{x_1, x_2, \cdots\}$, the *probability mass function* is defined as

$$f_X(x_i) = p(X = x_i), i \in \{1, 2, \cdots\}. \tag{1.37}$$

For continuous random variable X, the probability of X falling a certain range of values (x_1, x_2) is derived by integrating the *probability density function* $f_X(x)$ over the range, that is

$$p(x_1 < X < x_2) = \int_{x_1}^{x_2} f_X(x)dx. \tag{1.38}$$

When a dice is rolled, the sample space is six different sides so that the random variable X can take the values of $\{1, 2, 3, 4, 5, 6\}$. Then the probability mass function can be written as $f_X(i) = 1/6, i = \{1, 2, 3, 4, 5, 6\}$. It is worthy of noting the difference between *events* and *random variables*. The probability is defined on events while probability distributions are defined in terms of random variables. X assigns numbers to each elementary outcome (each of six sides), while an event can be a set of outcomes, which is a subset of sample space. For example, the event can be $\emptyset$ which is an impossible event, or "the sums of rolling dices adding up to 3".

Conditional probability and independence

When flipping a coin, the event A "the first trial showing tail" and the event B "the second trial showing head" are independent. Formally, two events A and B are independent if $p(A, B) = p(A)p(B)$. Obviously the relation holds for A and B, thus they are independent. Similarly, two random variables X and Y are independent if

$$p\Big(X = x, Y = y\Big) = p(X = x)p(Y = y), \forall x, y. \tag{1.39}$$

Though the probability definition does not involve conditional probability, in a more rigorous sense probabilities are conditioned on what is known. For example, the probability of showing each face of a dice is $1/6$, because it is known that a dice has six faces and it is assumed to be a fair dice. Therefore it is of utmost importance to have conditional probability in probability theory. It can be defined as

If $p(B) > 0$ then the conditional probability of A given B is

$$p(A \mid B) = \frac{p(A, B)}{p(B)}. \tag{1.40}$$

The independence can be defined in terms of conditional probability as

A and B are independent if and only if $p(A \mid B) = p(A)$.

1.3.2 Maximum likelihood

Maximum Likelihood Estimate (MLE), introduced by Fisher in 1920s [Ald+97; Sti07], is a method for estimating parameters of statistical models to best fit the observed data under certain model hypothesis. In other words, in order to model observed data, MLE presumes the fixed certain models, then seeks for the optimal parameters which make the models best fit the present data. MLE lays the probabilistic foundation for a wide range of machine learning algorithms (e.g.*Section 1.4.2, 1.4.2*).

Assume the given data $x_1, x_2, \cdots, x_n$ are values of a random sample drawn from some distribution (continuous or discrete) with probability density (or mass) function $f_X(x \mid \theta)$ with unknown parameter θ. Then the probability of observing the random sample is

$$f_X(x_1, x_2, \cdots, x_n \mid \theta) = L(\theta \mid x_1, x_2, \cdots, x_n), \tag{1.41}$$

where $L(\theta \mid x_1, x_2, \cdots, x_n)$ is the likelihood function. The likelihood function, as a function of the parameter θ, can be interpreted as the probability of observing the given data from the distribution parameterized by θ.

A common assumption for statistical modeling is that $x_1, x_2 \cdots, x_n$ are *identically independently distributed (i.i.d.)*, which leads to

$$L(\theta \mid x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} f_X(x_i \mid \theta) \tag{1.42}$$

In order to maximize the likelihood function, the common trick to simplify the calculation is to take the logarithm of it because of the monotonic increasing of logarithm functions

$$\log L(\theta \mid x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n} \log f_X(x_i \mid \theta). \tag{1.43}$$

As a simple example, suppose three *Heads* are observed from tossing a coin three times, one needs to estimate the probability p for the Bernoulli experiment. It is reasonable to believe that the three tosses are independent, then the likelihood function is

$$L(p) = p^3, \tag{1.44}$$

which is monotonically increasing with $p \in [0, 1]$. It is obvious that if one wants to find the value of p which gives highest likelihood to observe the three *Heads* in three trials, then $L(p)$ should be maximized. And $p = 1$ maximizes $L(p)$, which means that the MLE would result in that we will observe *Heads* for ever.

In the context of machine learning algorithms, the negative logarithmic likelihood function is taken as loss function for measuring the error between the model fits and given data. Then maximizing likelihood is equivalent to minimizing the loss function, which connects the MLE to building machine learning models.

However, as seen from the above simple example, MLE can easily lead to overparameterization and poor generalization, which is known as over-fitting to the given data. To apply the "Occam's Razor" principle which promotes simple but working models, adding regularization in machine learning algorithms is used to prevent the overfitting (details in **Section 1.4.1 and 1.4.5**), especially the extreme estimate of p in the coin tossing example.

1.3.3 Probability generating functions

Probability generating function (PGF) provides a power series based representation of probability mass of discrete random variables, which is closely related to concepts like z-transform [Wei] and momentum generating functions [Wikf]. PGF can serve as a useful tool for deriving expectations and variances of probability distributions, it also played an important role in obtaining the analytic solution of master equation modeled in *Chapter 3*. In this section, we briefly introduce the essence of PGF.

The term "generating function" is usually referred to as *ordinary generating function*, which is defined as

$$G(s) = \sum_{i}^{\infty} a_i s^i \tag{1.45}$$

for a given sequence $\{a_1, a_2, \cdots, a_i \cdots\}$. Intuitively, the generating function can be seen as a clothesline for hanging the series of numbers $\{a_i\}$[Wil05]. When the sequence $\{a_i\}$ is specified by a probability distribution of a discrete random variable X with probability mass function $p_i = p(X = i), i = 1, 2, \cdots$, we have probability generating function

$$G(s) = \sum_{i}^{\infty} p_i s^i = \mathbf{E}[s^X]. \tag{1.46}$$

Note that $|s| \leq 1, G(1) = 1$. By differentiating the generating function, it yields

$$G'(s) = \sum_{i}^{\infty} i \cdot p_i \cdot s^{i-1} \implies G'(1) = E[X]. \tag{1.47}$$

And taking the second derivative of $G(s)$ gives rise to

$$G''(s) = \sum_{i}^{\infty} i \cdot (i-1) \cdot p_i \cdot s^{i-2} \implies G''(1) = E[X(X-1)] = E[X^2] - E[X]. \tag{1.48}$$

The derivatives of $G(s)$ at $s = 1$ can promptly lead to the derivation of the variance of X as

$$\begin{aligned} Var[X] &= E[X^2] - (E[X])^2 \\ &= G''(1) + G'(1) - (G'(1))^2. \end{aligned} \tag{1.49}$$

The relations (1.47) and (1.49) provide a neat way of deriving expectations and variances of distributions. For instance, the generating function of Poisson distribution is derived as

$$G(s) = \sum_{i}^{\infty} \frac{\lambda^k}{k!} e^{-\lambda} s^k = e^{-\lambda} \sum_{i}^{\infty} \frac{(\lambda s)^k}{k!} = e^{\lambda(s-1)}. \tag{1.50}$$

Then as a comparison to the derivation in [Pro], the expectation and variance of Poisson distribution can be computed as following

$$E[X] = G'(1) = \lambda e^{\lambda(s-1)} \mid_{s=1} = \lambda \tag{1.51}$$

and

$$Var[X] = G''(1) + G'(1) - (G'(1))^2 = \lambda^2 e^{\lambda(s-1)} \mid_{s=1} + \lambda - \lambda^2 = \lambda. \tag{1.52}$$

1.4 Machine Learning

> *A theory with mathematical beauty is more likely to be correct than an ugly one that fits some experimental data.*
>
> *Paul Dirac*

The term machine learning can be interpreted as "a computer program which learns from experience with respect to some tasks and performance measures.". Statistically speaking, it can be translated as "fitting a parametric or nonparametric model to data in order to make predictions". Based on the data which directly related to the problems to be solved, different machine learning subfields are commonly categorized to supervised learning, unsupervised leaning and reinforcement leaning[JM15; Mur12]. In this section, the focus is on supervised learning which served as one of the major methods throughout the thesis. The core elements of supervised learning, namely labeled data, hypothetical function (or approximation function), loss function and optimization, are first introduced along with a real life analogy.

1.4.1 Supervised Learning

Machine learning tasks which are supervised by given input-output pairs and learn the mapping function from the input to the output are intuitively named as supervised learning, which are further divided into tasks of classifications and regressions. As a real life analogy, supposing that a pupil learner sitting in a classroom tries to learn arithmetic operations, he is supervised by the teacher who gives him many examples such as $1+1=2, 1+2=3$ and so on and needs to do exercises to master the operations. At the very beginning, the learner knows nothing and makes a lot of mistakes so that the teacher tells him where goes wrong and learner needs to rewire his brain neurons in order to make less mistakes. When coming to examine how well he learns, the learner has to solve the exam questions not seen before without supervision. The exam grade indicates the goodness of learning. The analogy is not perfect but sensible since learning arithmetics is actually empirical as the teacher cannot explicitly explain why $1+1=2$ and $1+2=3$ to pupils but show that putting one items together with another two items makes in total three items. Machine learning is aimed at automating extractions of patterns from data, without writing explicit algorithms to describe all details about the patterns. This is like the pupil learns the arithmetic addition by counting and exercising without knowing all the algebraic reasoning for proving why $1+1=2$. At the beginning of learning, machine learning models are not better than random guesses therefore criteria indicating goodness of learning are needed for improving them, which are termed loss functions that are defined as the total distance between all model outputs and the real outputs. The learning procedure is repeatedly showing the data pairs to *train* the models such that the loss functions are minimized. A learned model should be *tested* by new data pairs to show its performance.

To put all these formally, given N data pairs $(\mathbf{x}_i, \mathbf{y}_i),\ i=1,2,\cdots,N$ with $\mathbf{x}_i \in \mathbb{R}^d$ the inputs with d dimensional features, $\mathbf{y}_i \in \mathbb{R}^m$ the m dimensional outputs which is usually also called the targets. $f(\cdot): \mathbb{R}^d \to \mathbb{R}^m$ denotes the hypothesis function learned from the data to approximate the underlying truth. Then the prediction from the model is given by

$$\hat{\mathbf{y}}_i = f(\mathbf{x}_i, \theta), \tag{1.53}$$

with $\hat{\mathbf{y}}_i$ the model estimation for $\mathbf{y}_i$, θ as the model parameters. Intuitively, the learning goal is to make $\hat{\mathbf{y}}_i$ as close as possible to $\mathbf{y}_i$. However this is usually not practically favored due to *overfitting*, a problem that an overly complex model is deployed such that it fits well on the training data but predicts badly in application on unseen data. The performance gap is due to the fact that the training data is usually either corrupted with certain level of noises or not fully representative for the true relation between input features and output targets. On the contrary, if an overly simple model is fitted to the training data, it lacks the capacity to capture the input-output relation, which is usually termed *underfitting*. The overfitting and underfitting problem is often examined as variance-bias tradeoff, which will be elaborated in more details by error decomposition in *Section 1.4.5*. When training a model, one needs to find the sweet spot where variance and bias are well balanced such that the performance gap between training and predicting is as small as possible. A natural question arises as that how one can find the good tradeoff between variance and bias. For answering that, it should be noted that it is easy to increase the complexity of models to fit well the training data, for instance, as reported in [Zha+16], deep neural networks with enough capacity (complexity) can easily fit data with random labels to have zero training error, but not easy to control the model complexity in order to discover real pattern underlying the training data. Regularization techniques play a prominent role in preventing overfitting, which add regularizers to the training objective in order to punish overly complex models.

With all that said, the training objective in a supervised learning task can be defined in general as

$$Loss(\theta) = \sum_{i=1}^{N} \mathcal{E}(\mathbf{y}_i, f(\mathbf{x}_i, \theta)) + \Omega(\theta), \tag{1.54}$$

which is referred to as the loss or cost function, with $\mathcal{E}(\cdot)$ as the error function defining the distance between model estimations and the targets for the entire training data, $\Omega(\cdot)$ the regularizer parameterized by θ. Essentially all supervised learning tasks converge to optimizing the loss function.

Up to now, the core principles of supervised learning have been introduced. It shall be seen later that different machine learning algorithms only differ in specifying $f(\cdot)$, $\mathcal{E}(\cdot)$, $\Omega(\cdot)$ and optimization techniques to minimize the loss functions. For example, backpropagation and stochastic gradient decent algorithms are widely employed in neural networks to optimize weights [LBH15b], and additive training is used in gradient boosting trees to grow optimal trees [CG16].

In practice, popular libraries like Scikit-Learn[Ped+11] in Python implemented various classification, regression and clustering algorithms, which makes a wide range of algorithms out-of-box for application. However, insights of algorithm details are helpful to choose suitable models, loss function, regularizations and optimization routines for solving particular problems.

1.4.2 Logistic regression and cross-entropy as loss

Given the task of assigning input data to classes $\{0, 1\}$, where class 0 is usually referred to as the negative class and 1 as the positive class, logistic regression provides a simple and straightforward solution for the binary classification. Logistic regression algorithm uses logistic function as the hypothetical function to approximate the true mapping between the labeled classes and data. For the sake simplicity and illustration, the logistic function with independent variable x with its weight w can be written as

$$y = f(x, w) = \frac{1}{1 + e^{-wx}}, \tag{1.55}$$

which is sketched in Fig1.2 for different values of w. The sigmoidal curves all intersect at $(0, 0.5)$ and the value of $f(x, w)$ has the property of

$$\begin{cases} 0.5 < y < 1 & \text{if } wx > 0 \\ 0 < y \leq 0.5 & \text{if } wx \leq 0, \end{cases} \tag{1.56}$$

which can be interpreted as the conditional probability $P(class = 1 \mid x, w)$ of being in class of 1 for the given x and w. Based on this probability, decision boundary can be set to make classifications. If the boundary is taken to be 0.5, then any x leading to $y < 0.5$ is classified as a negative instance, and vice versa. The boundaries for negative examples and positive examples can be set differently in order to increase the classification confidence. For example, to diagnose if patients have diabetes, the blood sugar level of two hours after the meals can be used to make decisions. Due to variances in sugar levels among different people, a single boundary for diagnosing is not practical. The 2-hour glucose in the blood of a healthy person is less than $140mg/dl$, but only people with 2-hour glucose level higher than $200mg/dl$ are diagnosed as diabetes [Org+06].

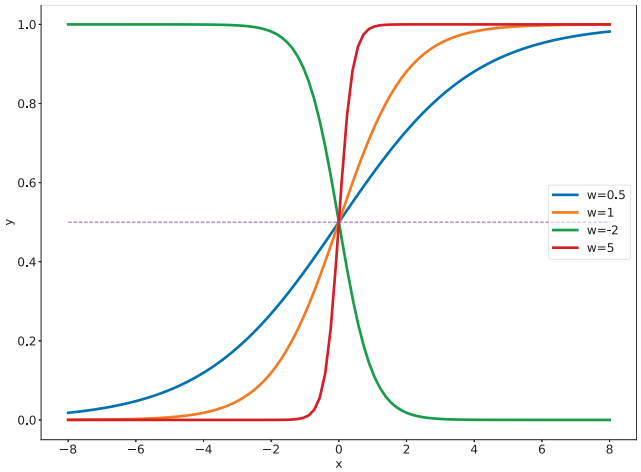


Fig. 1.2: The sigmoidal curves of logistic function for different weights.

In real applications, x is mostly multidimensinal, which will be denoted as $\mathbf{x}$, and a bias b is introduced for shifting the sigmoidal curve positions. Consequently the logistic function becomes

$$f(\mathbf{x}, \mathbf{w}, b) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}, \tag{1.57}$$

where $\mathbf{w}$ is the vector form of w.

With the general form of logistic regression model (1.57), the derivation of the error function is shown in the following from a probabilistic perspective. Given the dataset $\mathcal{D}$: $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}, i \in \{0, 1, \cdots, N\}$, the probability for observing a positive or negative instance can be written as

$$p(y_i \mid \mathbf{x}_i, \mathbf{w}, b) = [f(\mathbf{x}_i, \mathbf{w}, b)]^{y_i} [1 - f(\mathbf{x}_i, \mathbf{w}, b)]^{1 - y_i}. \tag{1.58}$$

With the assumption that the data is i.i.d., the likelihood of having the data is

$$L(\mathbf{w}, b) = \prod_{i=1}^{N} [f(\mathbf{x}_i, \mathbf{w}, b)]^{y_i} [1 - f(\mathbf{x}_i, \mathbf{w}, b)]^{1 - y_i}. \tag{1.59}$$

Taking the negative logarithm of the likelihood leads to the error function

$$\mathcal{E}(\mathbf{w}, b) = -\log L(\mathbf{w}, b) = -\sum_{i=1}^{N} y_i \log f(\mathbf{x}_i, \mathbf{w}, b) + (1 - y_i) \log(1 - f(\mathbf{x}_i, \mathbf{w}, b)), \tag{1.60}$$

which is the cross entropy[SJ80] of target distribution in data $\mathcal{D}$ and the model estimated target distribution.

Intuitively, Kullbach-Leibler(KL) divergence which quantifies the distance between two distributions can be taken as the error function, but why the cross entropy arises for classification problems? First recall that entropy is used to measure the uncertainty of a system, which is defined as

$$S(v) = -\sum_{i} p(v_i) \log p(v_i), \tag{1.61}$$

for $p(v_i)$ as the probabilities of different states v_i of the system. From an information theory point of view, $S(v)$ is the amount of information is needed for removing the uncertainty. For instance, the event A "I will die eventually" is almost certain (the aging problem might be solved for including the word "almost"), therefore it has low entropy which requires only the information of "might solve the aging problem" to make it certain. However, the event B "The president will die in 50 years" is much more uncertain than A, thus it needs more information to remove the uncertainties.

Now look at the definition of KL divergence between events A and B

$$D_{KL}(A \parallel B) = \sum_{i} p_A(v_i) \log p_A(v_i) - p_A(v_i) \log p_B(v_i), \tag{1.62}$$

where the first term of the right hand side is the entropy of event A, the second term can be interpreted as the expectation of event B in terms of event A. And the D_{KL} describes how different B is from A from the perspective of A.

To relate cross entropy to entropy and KL divergence, the cross entropy in (1.60) needs to be reformalized in terms of events A and B as

$$H(A, B) = -\sum_{i} p_A(v_i) \log p_B(v_i). \tag{1.63}$$

From (1.61), (1.62) and (1.63), it can be seen that

$$H(A, B) = D_{KL}(A \parallel B) + S_A. \tag{1.64}$$

From (1.64), the fact that if S_A is a constant, then minimizing $H(A, B)$ is equivalent to minimizing $D_{KL}(A \parallel B)$ can answer why the cross entropy error function arises from the likelihood function of the model. A further question follows naturally as how the entropy can be a constant. A machine learning task is started with a dataset (denoted as $P(\mathcal{D})$) which represent the problem to be solved, and the learning purpose is to make the model estimated distribution (denoted as $P(model)$) as close as possible to true distribution of the problem (denoted as $P(truth)$). $P(truth)$ is unknown and represented by $P(\mathcal{D})$. Therefore in an ideal world, one expects

$$P(model) \approx P(\mathcal{D}) \approx P(truth) \tag{1.65}$$

and minimize $D_{KL}(P(\mathcal{D}) \parallel P(model))$. And luckily, in practice $\mathcal{D}$ is given, which means its entropy $S(D)$ is fixed as a constant. Now it is clear that the equivalence of minimizing cross entropy and KL divergence in a classification problem for given dataset, which shows the cross entropy can be the proper error function.

Without loss of generality, the regularizer is specified as L_2 norm of weight vector $\mathbf{w}$. Therefore the loss function can be summed up as

$$Loss(\mathbf{w} \mid \mathbf{x}, y) = \mathcal{E}(\mathbf{w}, b) + \Omega(\mathbf{w}) \tag{1.66}$$

$$= -\sum_{i=1}^{N} y_i \log f(\mathbf{x}_i, \mathbf{w}, b) + (1 - y_i) \log(1 - f(\mathbf{x}_i, \mathbf{w}, b)) + \parallel \mathbf{w} \parallel^2 \tag{1.67}$$

The training of a logistic regression model is cast into searching the optimal $\mathbf{w}$ in the parameter space as

$$f^* = f(\mathbf{w}^*) = \arg\min_{\mathbf{w}} Loss(|\mathbf{x}, y) \tag{1.68}$$

The optimization procedure is introduced in *Section 1.5.1.*

1.4.3 Neural networks and error backpropagation

Neural network algorithms have been growing from the simple perceptrons[Ros62] in 1950s and 1960s to nowadays deep learning as one of the most influential fields in machine learning [Sch15; Nie15]. As a tip of an iceberg, a dense fully connected neural network with two feedforward layers is used to introduce the basic elements of a neural network. Then a powerful type of network named convolutional neural networks is illustrated. Further the soul of training neural networks, error backpropagation, is introduced via a simple but representative network.

Feedforward dense neural networks

A feedforward neural network with two layers is illustrated in Fig. 1.3, which can be mathematically described as

$$F_k(\mathbf{x}, W) = f\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{d} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right), \tag{1.69}$$

where $F_k(\mathbf{x}, W)$ denotes the kth output, d, M denote the dimension of input $\mathbf{x}$ (number of nodes in the input layer) and number of nodes in the hidden layer (second layer) respectively. $W = \left\{W^{(1)}, W_k^{(2)}\right\}$ where $W^{(1)}$ is the weight matrix with $w_{ji}^{(1)}$ connecting ith input to jth node in the hidden layer, $W_k^{(2)}$ is the weight matrix with $w_{kj}^{(2)}$ connecting jth hidden node to kth output node. In the context of regression, the output is usually one-dimensional, the index k can be dropped.

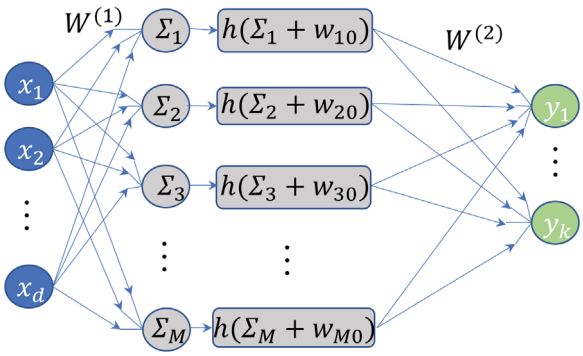


Fig. 1.3: A fully connected feedforward neural network.

For a given *i.i.d.* dataset $\mathcal{D}$: $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $i \in 1, 2, \cdots, N$, assuming that the targets y have a Gaussian distribution centered on $F(\mathbf{x}, W)$ gives rise to

$$p(y \mid \mathbf{x}, W) = \mathcal{N}(y \mid F(\mathbf{x}, W), \beta), \tag{1.70}$$

with β the variance of Gaussian noise. With defining the mean squared error between the targets and their model estimates as

$$\mathcal{E}(W) = \frac{1}{N} \sum_{i=1}^{N} \|F(\mathbf{x}_i, W) - y_i\|^2, \tag{1.71}$$

the likelihood function can be written as

$$\mathcal{L}(W) = p(y \mid \mathbf{x}, W) = \prod_{i=1}^{N} p(y_i \mid \mathbf{x}_i, W, \beta) = \frac{\exp(-\beta \mathcal{E}(W))}{Z(\beta)} \tag{1.72}$$

where $Z(\beta) = (2\pi/\beta)^N$. Taking negative logarithm of (1.72) gives

$$-\ln \mathcal{L}(W) = \mathcal{E}(W) + c, \tag{1.73}$$

where c is a constant related to β which can be neglected for the purpose of minimizing negative logarithm likelihood. The equation (1.73) shows that minimizing error function is equivalent to maximizing the likelihood function.

Consequently adding L_2 norm regularization leads to the final loss function as

$$Loss(W \mid \mathbf{x}, y) = \mathcal{E}(W) + \Omega(W) \tag{1.74}$$

$$= \frac{1}{N} \sum_{i} \|F(\mathbf{x}_i, W) - y_i\|^2 + \| \mathbf{w} \|^2 . \tag{1.75}$$

Convolutional neural networks

Convolutional neural networks are a special type of multilayer neural networks, each layer of which typically consists of convolution, pooling and nonlinear activation.

Mathematically, given two functions g and f, a convolution is defined as the integral of the product of the two functions with one reversed and shifted:

$$f * g := \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)d\tau. \tag{1.76}$$

To have a better visualization of the convolution, the first layer in Fig. 1.4a shows a two-dimensional vector (a filter or a kernel) of weights $\mathbf{w} := (w_1, w_2)$ convolves with a six dimensional input vector $\mathbf{x} := (x_1, \cdots, x_6)$, which yields five dimensional vector. When comparing to the structure of dense fully connected layers, the number of weights in each convolution has been greatly reduced. This is usually referred to as the weight sharing property of convolutional neural networks.

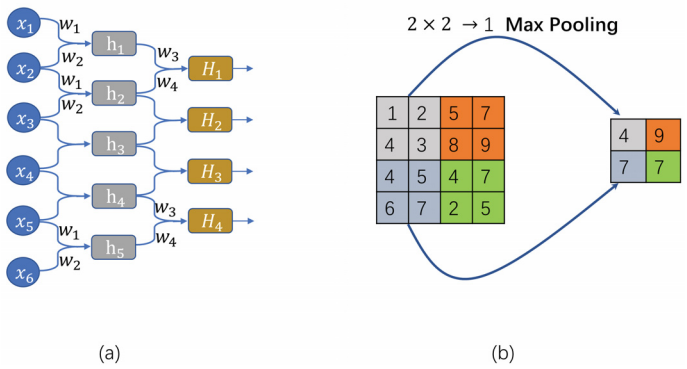


Fig. 1.4: Convolutional neural networks. (a). A simple illustration of weight sharing in convolutional neural networks, adapted from Colah's blog [Ola]; (b). The $2 \times 2 \to 1$ mapping shows a maximum pooling, i.e., taking the maximum value of the 2×2 cell in the left to the corresponding cell in the right.

As the primary application of convolutional neural networks is image recognition with images numerically stored as matrices, the pooling operation can not only reduce the image dimension but also extract image features and reduce invariance. As shown in Fig. 1.4b, the pooling reduced the matrix from 4×4 to 2×2, which can largely reduce computational complexity for large input images. For images, as neighbouring pixels are related, the max pooling can extract extreme local features, for example, the edges of objects. Similar to fully connected layers, the pooling output is then taken as the input of activation functions to be transformed nonlinearly or linearly.

The basic components of convolutional neural networks have been introduced. In real practice, the filter sizes, the number of filters, types of pooling and pooling sizes are all problem-dependent and regarded as hyperparameters during the network training.

Error backpropagation

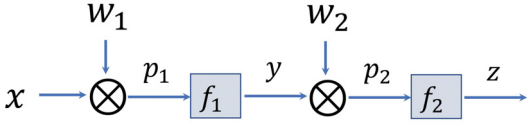


Fig. 1.5: Illustration of error backpropagation by the simplest neural net with scalar input and output

A simple but informative model is shown in Fig. 1.5, with scalar input x and output z, with scalar weights w_1 and w_2 to elaborate the well-known error backpropagation algorithms for training neural networks. Intermediate computing steps are explicitly shown, where $\otimes$ denotes the multiplication with p_1, p_2 as the multiplication products, and f_1, f_2 are activation functions. The squared error function is taken as the loss function $\mathcal{E}(\mathbf{w}|x, d) = \frac{1}{2}(z-d)^2$ with d the target values. Start with calculating the first order derivatives of $\mathcal{E}(\mathbf{w})$ with respect to w_1 and w_2

$$\begin{aligned}\frac{\partial \mathcal{E}}{\partial w_2} &= \frac{\partial \mathcal{E}}{\partial z}\frac{\partial z}{\partial p_2}\frac{\partial p_2}{\partial w_2} \\ &= (d-z)\frac{\partial z}{\partial p_2}y\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{E}}{\partial w_1} &= \frac{\partial \mathcal{E}}{\partial z}\frac{\partial z}{\partial p_2}\frac{\partial p_2}{\partial y}\frac{\partial y}{\partial p_1}\frac{\partial p_1}{\partial w_1} \\ &= (d-z)\frac{\partial z}{\partial p_2}w\frac{\partial y}{\partial p_1}x\end{aligned}$$

Then the derivatives are written in vector form as

$$g = (\frac{\partial \mathcal{E}}{\partial w_1}\ \frac{\partial \mathcal{E}}{\partial w_2})^T. \tag{1.77}$$

According to steepest gradient descent (*Section 1.5.1*), the weight vector can be updated at step k by

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha g. \tag{1.78}$$

This simple model is informative since calculation of gradients g can be directly extended to higher dimensions and the chain-rule based propagation of the errors is intuitive in the scalar input-output case.

1.4.4 Gradient boosting methods and gradient boosting trees

Ensemble algorithms combine a collection of hypothetical functions as the approximation function in order to form a better hypothesis for approximating the underlying true mapping between inputs and targets. Boosting is one of such algorithms to sequentially combine multiple weak learners which perform poorly on learning to form a single powerful hypothesis. The weak learners in principle can be broadly different models, for example, linear models or decision trees. The gradient boosting methods follow the idea of iteratively using weak hypothesis which points the negative gradient directions to optimize an differentiable loss function [Bre97; Fri01; Mas+99; Fri02]. Gradient boosting trees, as the most popular gradient boosting algorithm, has been used in later chapters. Explicitly, gradient boosting trees are ensembles of Classification and Regression Trees (CART) which are growing sequentially to fit pseudo-residues (explained in next paragraph). The advantage of CART over a normal decision tree is that it assigns scores for tree leaves in order to provide richer interpretations beyond classification while the latter gives only decision values to each leaf. Gradient boosting methods are conceptually and mathematically introduced with a focus on gradient boosting trees.

To grab the intuition of the core idea underlying gradient boosting methods, we start with the simple scenario of assuming that y is the target to be fitted with from independent variable x.

The first hypothesis is F_1 which gives rises to the first residue $e_1 = y - F_1(x)$. Usually e_1 is not satisfying for being too large, therefore a further hypothesis F_2 is deployed to fit e_1 as the target, which leads to a new residue $e_2 = e_1 - F_2(x)$. The errors $e_1,\ e_2$ are referred to as pseudo-residues. The procedure is continued until the final pseudo-error falls in an acceptable range. For a well defined loss function, at step i, the hypothesis F_i points into the direction of the negative gradient of the loss function, which connects the boosting trees to gradient descent of a loss function.

To formally elaborate gradient boosting trees (GBT), assuming dataset $\mathcal{D}$: $\mathbf{x}_i \in \mathbb{R}^d,\ y_i \in \{0, 1\}, i \in \{0, 1, \cdots, N\}$, the ensemble trees as hypothesis function F approximating the relation between $\mathbf{x}$ and $\mathbf{y}$ is given by

$$\hat{y}_i = F(\mathbf{x}_i) = \sum_{k=1}^{\mathcal{K}} f_k(\mathbf{x}_i), f_k \in \mathcal{F}. \tag{1.79}$$

This notation is actually a general definition for a broad range of algorithms which consist of multiple basis hypothses (functions), with $\mathcal{F}$ denoting the hypothesis space. In terms of GBT, $\mathcal{F} = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}$ is the space of CARTs, with $q : \mathbb{R}^d \to T$ denoting the structure of CART which has T leaves. The leaf values $w \in \mathbb{R}$ are usually named as leaf weights or scores. The loss function can be accordingly defined as

$$Loss(y, F(\mathbf{x})) = \sum_{i=1}^{N} \mathcal{E}(\hat{y}_i, y_i) + \sum_{k=1}^{\mathcal{K}} \Omega(f_k), \tag{1.80}$$

with the first term at right characterizes the distance between $\hat{y}_i$ and y_i and second term regularizes the complexity of trees in the ensemble. The training of GBT to obtain the optimal mapping F is then summarized as

$$\begin{aligned} F(\mathbf{x}) &= \underset{f_j, j=1, \cdots, \mathcal{K}}{\arg\min} Loss(y, F(\mathbf{x})) \\ &= \underset{f_j, j=1, \cdots, \mathcal{K}}{\arg\min} \sum_{i=1}^{N} \mathcal{E}\left(\sum_{k=1}^{\mathcal{K}} f_k(\mathbf{x}_i), y_i\right) + \sum_{k=1}^{\mathcal{K}} \Omega(f_k). \end{aligned} \tag{1.81}$$

This equation might look tedious but it is mainly to show that the training is actually optimizing the loss in function space for finding the optimal hypothesis sequence $f_1, f_2, \cdots, f_{\mathcal{K}}$. The training procedure follows an additive strategy which optimizes one new tree at a time given the preceding optimized trees and add newly optimized tree to existing trees to form an updated ensemble.

Next the gradient of the loss function is connected to the additive training by specifying the error function as squared errors

$$Loss(y, F(\mathbf{x})) = \frac{1}{2} \sum_{i=1}^{N} \left(F(\mathbf{x}_i) - y_i\right)^2 + \sum_{k=1}^{\mathcal{K}} \Omega(f_k). \tag{1.82}$$

Assuming at step t, one ought to optimize tree f_t to optimize the loss (1.80) given that all preceding steps $1, \cdots, t-1$ have obtained optimized CARTs. Taking the derivatives of (1.80) at step $t-1$ yields

$$\frac{\partial Loss_{t-1}}{\partial F_{t-1}(\mathbf{x}_j)} = F_{t-1}(\mathbf{x}_j) - y_j + \frac{\partial \Omega(F_{t-1}(\mathbf{x}_j))}{\partial F_{t-1}(\mathbf{x}_j)}, \tag{1.83}$$

which can be rearranged as

$$y_j - F_{t-1}(\mathbf{x}_j) = -\left(\frac{\partial Loss_{t-1}}{\partial F_{t-1}(\mathbf{x}_j)} + \frac{\partial \Omega(F_{t-1})}{\partial F_{t-1}}\right). \tag{1.84}$$

It shows that the residue $F_{t-1}(\mathbf{x}_j) - y_j$ after accomplishing step $t-1$ is the regularized negative gradient of the loss with respect to F_{t-1}. Then at step t, for finding the optimal f_t, one needs only to fit it to this negative gradient. In the case of squared errors in loss function, the close form of residue is perfectly the negative gradient. However, the flexibility of gradient boosting methods allows arbitrary differentiable loss functions, for which the negative gradient is more proper as the targets of tree f_t than the pseudo-residue. Now the update rule for the ensemble hypothesis update rule can be written as

$$F_t \leftarrow F_{t-1} + \alpha f_t, \tag{1.85}$$

where α denotes the shrinkage rate which is similar to the learning rate in steepest gradient descent algorithm, providing an extra way for regularizing the ensemble hypothesis. The update rule continues until certain criteria are satisfied for convergence, for example, the number of steps and gradient value threshold. In one word, the main principle ideas underlying gradient boosting methods is to update the ensemble hypothesis by adding new basis hypothesis associated with the negative gradient of the hypothesis.

Next the searching of optimal tree f_t is shown for the algorithm XGBoost [CG16] which is a popular variant of GBT with improved scalability and efficiency comparing the standard GBT. At step t, the optimal tree f_t should minimize the loss

$$Loss_t(y, F_t(\mathbf{x})) = \sum_{i=1}^{N} \mathcal{E}(y_i, F_{t-1}(\mathbf{x}_i) + \alpha f_t(\mathbf{x}_i)) + \Omega(f_t(\mathbf{x}_i)). \tag{1.86}$$

To efficiently optimize (1.86), the second-order approximation of $\mathcal{E}$ can be used by expanding it around $F_{t-1} - f_t$ as

$$\mathcal{E}(y_i, F_{t-1}(\mathbf{x}_i) + \alpha f_t(\mathbf{x}_i)) = \mathcal{E}(y_i, F_{t-1}(\mathbf{x}_i)) + g_i \alpha f_t(\mathbf{x}_i) + \frac{1}{2} h_i \alpha^2 f_t^2(\mathbf{x}_i) \tag{1.87}$$

with $g_i = \partial_{F_{t-1}} \mathcal{E}(y_i, F_{t-1})$ and $h_i = \partial^2_{F_{t-1}} \mathcal{E}(y_i, F_{t-1})$. As for now, only f_t is concerned therefore $F-1$ is a constant and can be removed for optimizing $Loss_t$. The regularizing function Ω can be specified as $\lambda T + \gamma \sum_j^T w_j^2$ to restrict the number of leaves and the leaf scores simultaneously. Consequently

$$Loss_t(y, F_t(\mathbf{x})) = \sum_{i=1}^{N} \left(g_i \alpha f_t(\mathbf{x}_i) + \frac{1}{2} h_i \alpha^2 f_t^2(\mathbf{x}_i)\right) + \lambda T + \gamma \sum_j^T w_j^2 \tag{1.88}$$

Reformalizing the equation (1.88) in terms of scores $\mathbf{w}^{(t)}$ in f_t as

$$Loss_t(y, \mathbf{w}^{(t)}) = \sum_{j=1}^{T} \left(\alpha w_j \sum_{i \in I_j} g_i + \frac{1}{2} \alpha^2 w_j^2 (\sum_{i \in I_j} h_i + \gamma)\right) + \lambda T \tag{1.89}$$

where $I_j = \{i \mid q(\mathbf{x}_i) = j\}$ as the instance set of leaf j over the whole dataset, the problem of hypothesis optimization (optimize f_t) is transformed to parameter optimization (optimize $\mathbf{w}^{(t)}$). For a fixed tree structure $q(\mathbf{x})$, the optimal w_j^* of leaf j can be obtained from (1.89) as

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\alpha(\sum_{i\in I_j} h_i + \gamma)}. \tag{1.90}$$

And the optimal $Loss_t$ becomes

$$Loss_t^* = -\frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i\in I_j} g_i)^2}{\alpha(\sum_{i\in I_j} h_i + \gamma)} + \lambda T \tag{1.91}$$

which can be used as the criterion for estimating the structure quality of f_t. To further search the space of all possible tree structures for f_t, XGBoost [CG16] used a greedy algorithm of optimizing one level of the tree at a time. By assuming I_L and I_R as left and right nodes respectively after splitting, $Loss_t$ is transformed to splitting score as

$$score = \frac{1}{2}\left(\frac{(\sum_{i\in I_L} g_i)^2}{\alpha(\sum_{i\in I_L} h_i + \gamma)} + \frac{(\sum_{i\in I_R} g_i)^2}{\alpha(\sum_{i\in I_R} h_i + \gamma)} - \frac{(\sum_{i\in I} g_i)^2}{\alpha(\sum_{i\in I} h_i + \gamma)}\right) - \lambda \tag{1.92}$$

By each splitting, the $score$ is maximized to have optimal split candidates.

Thus far the principle idea of updating the ensemble hypothesis sequentially by the associated gradient has been introduced for gradient boosting methods. Specifically, the searching of optimal trees at each step for XGBoost completes the introduction of full gradient boosting algorithm.

1.4.5 Error decomposition and model selection

Bias and variance trade-off

To measure the efficacy of machine learning models is of fundamental importance as fallacious conclusions can be easily drawn due to underfitting or overfitting. It is intuitive from the name to accept that underfitting is fitting a too simple model to the data while overfitting is fitting a overly complex model, both of which lead to high prediction errors.

To unravel the full details of different sources of misfitting, the trade-off between bias and variance is introduced, which will be addressed both conceptually and mathematically. Conceptually, the predictive errors caused by bias is due to difference between the expected prediction from a set of models and the true targets, and errors caused by variance is due to the variance of a set of model predictions for a given data point. A high bias model would miss the true relation between input features and their target in the data, and a high variance model would mistake the random noises in the data as true signals. As illustrated in Fig. 1.6, generating a few data points from a sin function $g(x) = \sin(2\pi x)$ and corrupting them by Gaussian noises, a straight line fitting leads to high bias as it is underfitting the data, whereas a 10th order polynomial tries to fit exactly each data points, which leads to an overfitting model with high variance. An intuitive bulls-eye definition of bias and variance predictive models can be found in [FR]

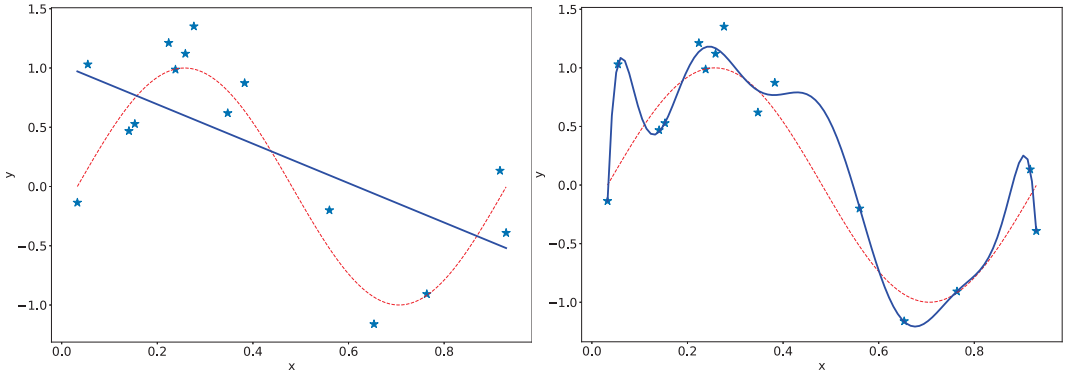


Fig. 1.6: Example illustration of underfitting and overfitting, reproduced from [Bis06].

The prediction error can be decomposed into variance and bias in terms of squared error between predictions and targets. Assume the dataset $\mathcal{D} = \{(x_i, y_i) \mid i = 1, \cdots, N\}$ is generated from the relation

$$y_i = f(\mathbf{x}_i) + \epsilon_i \tag{1.93}$$

where f is the true relation between $\mathbf{x}_i$ and y_i, ϵ_i is the irreducible error (e.g. measurement errors) with $E[\epsilon] = 0$. And y_i is estimated by $\hat{y}_i$ from an arbitrary model. For the sake of notation simplicity, the index i is neglected for calculation and $f \leftarrow f(\mathbf{x}_i)$. Then the expected mean sqaured errors at $\mathbf{x}_i$ can be written as

$$\begin{aligned} E[(y - \hat{y})^2] &= E[(y - f + f - \hat{y})^2] \\ &= E[(y - f)^2] + E[(f - \hat{y})^2] + 2E[(y - f)(f - \hat{y})] \end{aligned}$$

Note that $\epsilon = y - f$, f is deterministic, $E[y] = f$, and ϵ is independent from the model estimation $\hat{y}$, then

$$E[(y - f)^2] = E[\epsilon^2];$$

$$\begin{aligned} E\Big[(y - f)(f - \hat{y})\Big] &= E[yf] - E[f^2] + E[f\hat{y}] - E[y\hat{y}] \\ &= f^2 - f^2 + E[f\hat{y}] - E[(f + \epsilon)\hat{y}] \\ &= E[\epsilon]E[\hat{y}] \\ &= 0; \end{aligned}$$

$$\begin{aligned} E[(f - \hat{y})^2] &= E\Big[(f - E[\hat{y}] + E[\hat{y}] - \hat{y})^2\Big] \\ &= E[(f - E[\hat{y}])^2] + E[(E[\hat{y}] - \hat{y})^2] + 2E\Big[(f - E[\hat{y}])(E[\hat{y}] - \hat{y})\Big] \\ &= Bias^2(\hat{y}) + Var(\hat{y}). \end{aligned}$$

Now

$$E[(y - \hat{y})^2] = E[\epsilon^2] + Bias^2(\hat{y}) + Var(\hat{y}), \tag{1.94}$$

which shows that the total error consists of the variance of irreducible errors $E[\epsilon^2]$, the bias which describes how far off the predictions are from the truth, and the variance which describes the

spread of the predictions from truth. From the decomposition, one can easily see that to minimize the total error, the errors due to bias and variance should be minimized. This is not that trivial due to the trade-off between bias and variance. In the example of fitting sin function with polynomials, a 10th order polynomial model with high capacity, which can fit the data well, would result in low bias but high variance in a long run. However, a simple straight line leads to high bias and low variance. Therefore, one needs to find a model with relatively both low bias and low variance, i.e. with suitable model complexity. Unfortunately, in practice, there is no analytical ways to find the optimal model complexity. Instead one needs accurate error metrics for measuring model performances with different complexities to find the optimal complexity of the model of interest. The metrics are usually used together with data sampling technique *cross validation* for evaluating different models.

Cross Validation

To obtain the optimal model from the given data, accurately estimating the prediction errors is of central interest. The common treatment is to split the whole dataset into training, validating, and testing data sets. Training and validating data sets are used during training procedure for preventing overfitting by techniques such as early stopping, while testing data is absolutely blind for the model during training to mimic the real predicting scenario. However, the given data generally does not perfectly represent the problem to be solved due to measurement noises and its limited size. In order to avoid biases caused by single splittings, cross validation is usually a necessary procedure for estimating testing errors, especially for small datasets.

As an example, the *leave-one-out* cross validation is illustrated. Given that the model $f(\mathbf{x})$ is learned from $D = \{\mathbf{x}_i, y_i, i = 1, \cdots, N\}$, the error for data point i is measured by $\mathcal{E}(f(\mathbf{x}_i), y_i)$, assume data point j is selected for testing the model $f_j(\mathbf{x})$ which is trained on the data set excluding $(\mathbf{x}_j, y_j)$. Then the unbiased error can be taken from the average of the errors from applying *leave-one-out* cross validation on the whole dataset as

$$\hat{E} = \frac{1}{N}\sum_{j=1}^{N} \mathcal{E}\Big(f_j(\mathbf{x}_j), y_j\Big). \tag{1.95}$$

This procedure makes the model predict on all data points without seeing them during training. The averaged error estimate can avoid claiming the low error on a certain single split as the final model performance, which is probably biased. Thus the models selected from this procedure would have better generalization in future predictions.

k-fold cross validation is also widely used in practice, which divides the whole data set evenly into k groups and each group is iteratively used as the testing data, the rest $k-1$ is used as the training data. The averaged error from k iterations is used as the metric for training one models. It is worthy of noting that in practice if the testing data is involved in any way of optimizing either model parameters or hyperparameters, nested cross validation would be needed to have an unbiased metric for the selected model. Thus the overestimate of performance can be avoided.

1.5 Optimization

Optimization is a key step in machine learning and has been involved in later chapters. In this section, the details of steepest gradient descent and stochastic gradient are introduced for illustrating the concepts of optimization and understanding the training of neural networks. Particle swarm as a global optimization routine which played a key in *Chapter 3* is also introduced.

1.5.1 Gradient descent

Concepts of steepest gradient descent

For minimizing a differentiable multivariable function $f(\mathbf{x})$ with initial value of $\mathbf{x}_1$, the fastest decreasing would be the direction of negative gradient of $f(x)$ since its gradient points to fastest ascending direction. The updating rule

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t), t = 1, 2 \cdots \tag{1.96}$$

with sufficiently small η leads to $f(\mathbf{x}_{t+1}) < f(\mathbf{x}_t)$. The scalar constant η is the step size determining how far the updating moves in the negative gradient direction, which is usually called *learning rate* in machine learning model training. This iterative procedure attempts to make $f(\mathbf{x})$ converge to local or global minimum. The convergence is not guaranteed due to possible reasons like improper step size, complex function surface or initial conditions.

A graphical illustration of gradient descent searching with initial values of x_1 and y_1 is shown in Fig 1.7 for the non-convex surface of a nonlinear function. The searching converges to the respective local minimum. However, for an initial point at the saddle ridge between two valleys, the gradient at each direction is small which can potentially trap the searching such that the minima cannot be reached.

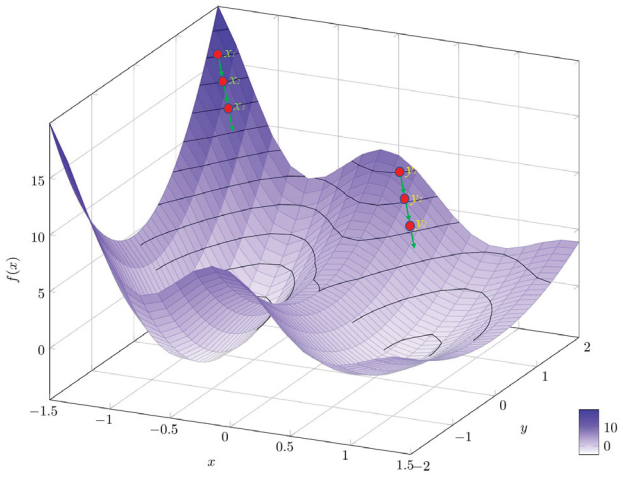


Fig. 1.7: Steepest gradient descent on non-convex surface.

Gradient descent as a first-order iterative optimization method

To gain the mathematical interpretation of why negative gradient is chosen for the updating, assume the general context of

$$\min_{\mathbf{x}} f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^m \tag{1.97}$$

with initial condition $\mathbf{x} = \mathbf{x}_0$.

The goal of first updating is to find $\mathbf{x}_1 = \mathbf{x}_0 + \Delta\mathbf{x}$ such that

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_0), \tag{1.98}$$

where $\Delta\mathbf{x}$ is the updating vector.

First $f(\mathbf{x}_1)$ is approximated with its first-order Taylor expansion as

$$f(\mathbf{x}_1) = f(\mathbf{x}_0 + \Delta\mathbf{x}) = f(\mathbf{x}_0) + \Delta\mathbf{x}^T \nabla f(\mathbf{x}_0), \tag{1.99}$$

which leads to

$$f(\mathbf{x}_1) - f(\mathbf{x}_0) = \Delta\mathbf{x}^T \nabla f(\mathbf{x}_0). \tag{1.100}$$

Now to ensure the semi-negative definite of $f(\mathbf{x}_1) - f(\mathbf{x}_0)$, one can simply choose

$$\Delta\mathbf{x} = -\eta \nabla f(\mathbf{x}_0) \tag{1.101}$$

and then we have

$$f(\mathbf{x}_1) - f(\mathbf{x}_0) = -\eta \|\nabla f(\mathbf{x}_0)\|^2 \leq 0, \tag{1.102}$$

for some scalar constant η, and $\|\nabla f(\mathbf{x}_0)\|^2 = 0$ if and only if $f(\mathbf{x}_0)$ is already the minimum.

To sum up and generalize to arbitrary step t, the updating rule

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t) \tag{1.103}$$

drives the searching towards the minimum given that the proper value of η can ensure that x_{t+1} is close enough to x_t such that the first-order approximate is a valid approximate with tolerable error.

Stochastic gradient descent

Now in the context of minimizing loss function for a machine learning algorithm, the objective is

$$\min_{\mathbf{w}} \mathcal{J}(\mathbf{w}, D), \tag{1.104}$$

with $\mathbf{w} \in \mathbb{R}^m$ as the model parameters, and $D = \{\mathbf{x}_i, y_i | i = 1, \cdots, N\}$ the dataset.

The term "batch gradient descent" is used when the entire dataset is used to compute $\mathcal{J}(\mathbf{w}, D)$. The parameters are repeatedly updated using the complete information from the dataset with the updating rule

$$\mathbf{w} := \mathbf{w} - \eta \nabla \mathcal{J}(\mathbf{w}, D). \tag{1.105}$$

The disadvantages of batch gradient descent include the expensive computing of the gradient of the loss function over entire dataset for each update, and the slow convergence caused by improper choices of learning rate of η.

Different from batch gradient descent, stochastic gradient descent (SGD) uses single examples from the dataset for computing the loss function of each update as

$$\mathbf{w} := \mathbf{w} - \eta \nabla \mathcal{J}(\mathbf{w}, D_i), \tag{1.106}$$

for $D_i = \{\mathbf{x}_i, y_i | i \in \{1, \cdots, N\}\}$. The single example based updating leads to light computation of the loss function's gradient but also high variance of parameter search. The high variance is due to the stochastic nature, which comes from the single-example approximation of the true loss function.

SGD is an unbiased approximation of the batch gradient descent. By the definitions, the relation between $\mathcal{J}(\mathbf{w}, D)$ and $\mathcal{J}(\mathbf{w}, D_i)$ can be derived as

$$\mathcal{J}(\mathbf{w}, D) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{J}(\mathbf{w}, D_i). \tag{1.107}$$

For the simplicity of notation, it is rewritten as

$$\mathcal{J} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{J}_i. \tag{1.108}$$

In stochastic gradient descent, the gradient of $\mathcal{J}$ is approximated by the gradient of $\mathcal{J}_i$ of a single example D_i

$$\tilde{\nabla} \mathcal{J} := \nabla \mathcal{J}_i. \tag{1.109}$$

Then expectation of the approximation

$$E\left(\tilde{\nabla} J\right) = \frac{1}{N} \sum_{i=1}^{N} \nabla J_i = \nabla \frac{1}{N} \sum_{i=1}^{N} J_i = \nabla J \tag{1.110}$$

is truly the gradient of $\mathcal{J}$, which shows that the approximation is not biased.

The advantage of SGD is the stochasticity introduced by the single-example loss function, which makes SGD has the potential to jump out of one minimum to another and possibly result in a better local optimum. However, it also usually slows down the convergence. It requires good strategy of adapting the learning rate to converge the optimizing, and many variants of SGD are available [Rud16].

Mini-batch gradient descent is improved from SGD, but is also usually referred to as SGD, of which a subset of dataset is used for each updating as

$$\mathbf{w} := \mathbf{w} - \eta \nabla \mathcal{J}(\mathbf{w}, D_{i,n+i}), \tag{1.111}$$

for $D_{i,n+i} = \{\mathbf{x}_j, y_j | j \in \{i, \cdots, i+n\} \subset \{1, \cdots, N\}\}$.

This way of updating leads to not only efficiently computing loss function but also reducing the searching variance. However, the subset size is empirical and problem dependent. To this date, there is no systemic way of finding it. It is usually suggested to be between 50 and 256, which is shown to be effective in practice.

1.5.2 Gradient free optimization methods

The premise of gradient descent methods is the differentiability of objective functions, which makes them not applicable to non-differentiable objective functions which can be discrete, discontinuous or noisy. Many gradient free methods are available, for example, exhaustive search, simulated annealing, genetic algorithms and particle swarm. In this section, particle swarm optimization (PSO) used in *Chapter 3* is introduced.

Concepts of particle swarm optimization

PSO is a stochastic optimization method with a population of particles searching for the optima through the solution space, where each particle is a candidate solution. The concept of PSO originated from studying collective behavior of social animals like ants, bees and birds, of which the individuals interact with the environment and each other. The simulations of simplified social systems from bird flock or fish schools were found to be useful for optimization [ESI01; PV02; MW15]. The principle idea of PSO is that, for example, a bird in a flock searching for the food source will constantly adjust its direction and velocity according to its best position so far and the best position within the flock, and eventually all birds end up at the position of the food source.

The basic particle swarm algorithm and improvements

In the context of

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) \tag{1.112}$$

where objective function $f(\mathbf{x})$ is usually referred to as fitness function, PSO is initialized as a population of n random particles (solutions) $\{\mathbf{x}_i | i \in \{1, 2, \cdots n\}, \mathbf{x} \in \mathbb{R}^m\}$. The updating rule of ith particle at step t is

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}, \tag{1.113}$$

where the velocity $\mathbf{v}_i^{t+1}$ is determined by its previous velocity, the personal (cognitive) best position $\mathbf{p}_i$ so far and the global (social) best or neighbourhood best position $\mathbf{g}_i$ as

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + c_1(\mathbf{p}_i - \mathbf{x}_i^t)\mathbf{R}_1 + c_2(\mathbf{g}_i - \mathbf{x}_i^t)\mathbf{R}_2, \tag{1.114}$$

where $c_1,\ c_2$ are accelerating constants named as cognitive coefficient and social coefficient respectively, $\mathbf{R}_1$ and $\mathbf{R}_2$ are diagonal matrices with uniformly distributed entries in $[0, 1]$ for adding stochasticity to the influences from personal and social behaviors.

One important constraint on the updating rule is that the velocities should not exceed a threshold of v_{max} such that the velocities would not explode and the searching is confined in the fitness resolution i.e. the region between present position and best position. Large values of v_{max} encourage global exploration and small values lead to exploitation. However, a too small v_{max} leads to insufficient exploration in the solution space while a too large v_{max} may result in skipping of optima. An empirical suggestion for the value of v_{max} is

$$v_{max} = \alpha \frac{\mathbf{x}_{max} - \mathbf{x}_{min}}{2} \tag{1.115}$$

where $\mathbf{x}_{max}, \mathbf{x}_{min}$ are the searching upper and lower boundaries of all $\mathbf{x}_i$.

To better control the exploration and exploitation, an inertia weight w was introduced for the velocity

$$\mathbf{v}_i^{t+1} = w_t \mathbf{v}_i^t + c_1(\mathbf{p}_i - \mathbf{x}_i^t)\mathbf{R}_1 + c_2(\mathbf{g}_i - \mathbf{x}_i^t)\mathbf{R}_2. \tag{1.116}$$

As suggested in [ESI01], w is usually linearly decreasing from 0.9 to 0.4 to have a good tradeoff between exploration and exploitation.

To better ensure the convergence, constriction factor K as a function of the cognitive coefficient c_1 and social coefficient c_2 was introduced for the updating rule

$$\mathbf{v}_i^{t+1} = K\Big(w_t \mathbf{v}_i^t + c_1(\mathbf{p}_i - \mathbf{x}_i^t)\mathbf{R}_1 + c_2(\mathbf{g}_i - \mathbf{x}_i^t)\mathbf{R}_2\Big). \tag{1.117}$$

The further details and proofs of constriction method can be found in [Cle99]

1.6 Molecular Biology 101

1.6.1 The general picture of information flow

The central dogma of molecular biology is as simple as

$$\text{DNA} \xrightarrow{Transcription} \text{RNA} \xrightarrow{Translation} \text{Protein.}$$

DNAs are double strands of 4-letter ("ATCG") sequences which are the instructions for building all known living organisms and many viruses [Hiy+11; Wika]. Genes are slices of DNA, which will be *transcripted* to messenger RNAs. Messenger RNAs are sequences consisting of "AUCG"s, which in turn will be *translated* to proteins. Proteins are the building blocks of life systems, which carry out different functionalities for fulfilling different biochemical activities.

The complexity and mystery of either the translation or transcription processes level up drastically when the system details are investigated. For example, the whole life-cycle dynamics of an mRNA involves questions like how the secondary structure is formed, how its structure incorporates the elongation of the translation machines, ribosomes, how it knows the cell locations it should go. When life systems are studied on the whole genome scale, more complicated and challenging problems can emerge. For instance, *E. coli*' genome includes in total around 4000 genes which encode proteins [Bla+97]. The ∼4000 genes make up a large genetic network for maintaining the life activities of *E. coli*, which enables them live under different environment conditions. When the *E. coli* are moving from a glucose rich environment to a lactose rich one, they first need to sense the changing. On the molecular level, the information should trigger the *lac* genes which are capable of catalyzing and transporting lactose, meanwhile the production of proteins which deal with glucose is reduced. In the case of eukaryotes, for example, plants and humans, the genome sizes are usually larger than bacteria and many more complicated mechanisms are involved.

The information flow in life systems makes them interact with and adapt to their environments. Essentially, a genome, the complete set of DNA in an organism, is the blueprint of the organism. The information underlying this blueprint is transformed on different molecular layers, involving RNA and proteins [LH16]. The information transformations interact with their environments which turn the blueprint to specific phenotypes on cellular, tissue and organism levels. The phenotypes in turn influence the organism's survival and reproduction in specific environments. The robustness of life systems is determined by their ability to adapt to the current environment and flexibility to adjust to environmental changes. For example, plants in temperate regions usually make use of the memory of cold winter for determining the upcoming spring and then decide when is the best time to flower. The primary information from the environment includes temperature and day length. The temperate plants developed a genetic regulatory mechanism for processing the signals from temperature and day length and accordingly producing the signaling molecules to trigger the transition of plants from growing to reproducing.

1.6.2 Transcription, translation and gene regulatory networks

In biology, the transcription is a process of transferring information from a gene on DNA to a RNA which is also usually called a transcript; the translation is a process of encoding the information

on an mRNA to a chain of amino acids, which is a polypeptide or a protein. As mentioned in the last section, many explicit details of the processes are still under intensive investigations. For simplicity, an introductory overview of transcription and translation are presented for bacteria. The hierarchical relationships among associated regulators and targeted genes in the processes of transcription and translation constitute the *gene regulatory networks (GRN)*, which govern the levels of RNA and proteins within a cell [SD18; Wikb]. GRN is illustrated by the example of flowering time regulation in plant model *Arabidopsis thaliana*.

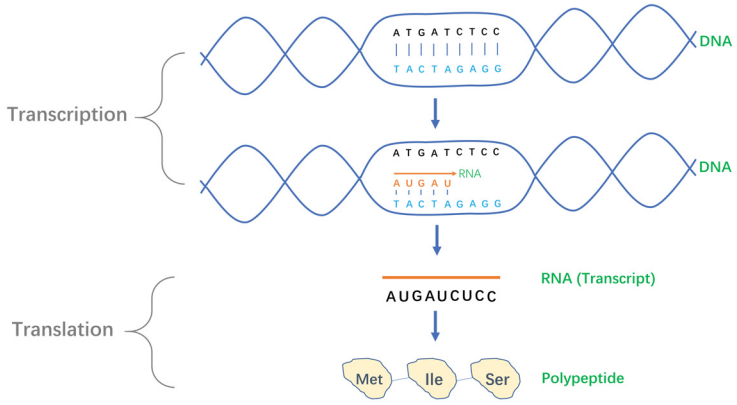


Fig. 1.8: A simplified illustration of the general picture of the transcription and translation. The plot is adapted from [Aca].

Transcription

The transcription, which copies the information on a gene to a RNA, has three phases: initiation, elongation and termination. The enzyme RNA polymerase plays a central role in transcription. For initiation, it binds to a segment of DNA sequence at the upstream of the gene, which is termed *promoter*, then separates the DNA double strands to make a single-stranded template for transcription. The binding of RNA polymerase can be activated or repressed by transcription factors which are proteins for controlling the rate of transcription. At elongation, the RNA polymerase moves along the single-stranded gene sequence to synthesize RNA. As transcription moves on to the terminating part, the sequence signal called terminator causes the stopping of transcription and then a complete RNA molecule is produced. The RNA has the same information as another strand except that thymine (T) is replaced with uracil (U).

Translation

The translation process of mRNA also consists of initiation, elongation and termination. Two types of molecules with central roles in the processes are *transfer RNA (tRNA)* and ribosomes. Different *tRNA*s with different anti-codons carry the corresponding amino acids to the ribosomes. Ribosomes are complex molecular machines which give slots for different *tRNA*s to match the right codons on *mRNA* and move along *mRNA* to form amino acid chains. The initiation takes place at the untranslated region (*UTR*) of an mRNA, where the ribosomes dock on. More technically speaking, a subregion called ribosomal binding site at *UTR* is where the ribosomes actually land on *mRNA*. Once the binding is ready, the first *tRNA* carries methionine to match the starting codon "AUG". After the drop of methionine from the *tRNA*, the ribosome moves on to next codons. Therefore, the elongation produces a chain of amino acid, which can be a polypeptide or a complete protein. The

termination of translation happens when the ribosomes encounter one stop codon ("UAG", "UAA", or "UGA").

The transcription and translation were superficially introduced to give a general picture such that when facing a problem, one knows which part of the central dogma it belongs to. For instance, in the case of translational efficiency regulation, the entangled regulatory factors on mRNA sequences, including codon bias, special motifs, secondary structures and other potential regulators, need to incorporate with cellular resources and requirements to produce the desired amount of proteins.

Gene regulatory networks

The flowering regulatory networks is extended with details to illustrated the concepts of gene regulatory networks in general.

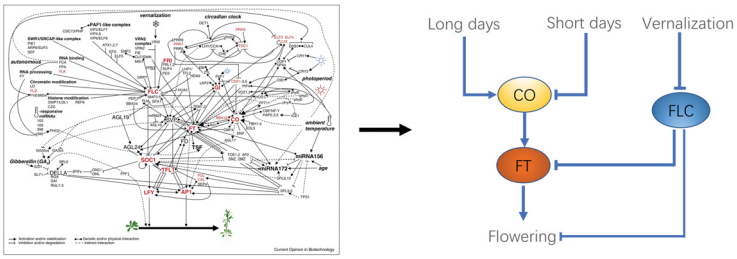


Fig. 1.9: The example for gene regulatory networks. (a). Taken from [BDJ15], it showed the known genetic interactions for regulating flowering time in *Arabidopsis thaliana*. The details are made invisible on purpose for not only showing the scale and complexity of the network but also for the credits of the original authors of the plot. (b). The core of the flowering regulatory network, adapted from [AC12].

The goal of gene regulatory networks is to regulate the synthesis of gene products at proper rate such that certain biological functions can be sustained. The produced protein from one gene can be a regulator of another gene to either activate or repress its expression. In *Arabidopsis thaliana*, the gene *FLOWERING LOCUS T (FT)* acts as the converging point of signals from both the day length and temperature, and its product eventually induces the flowering [BT15]. The expression of FT genes is promoted by the proteins produced from the gene *CONSTANS (CO)*, which are instable in the dark and only being produced about 12 hours after the dawn [BT15; AC12]. Thus the condition that day length is long enough to produce stable CO proteins is necessary for initiating the flowering of these long-day plants [TFC08; AC12; BT15]. Another necessary condition is the turn-off of the inhibitor, *FlOWERING LOCUS C (FLC)*, of *FT*, which is fulfilled by a process called vernalization. It involves the exposure of plants in a prolonged period of cold temperature, which induces a histone modification reaction on the molecular level [Cso+14; Ang+11; Ang+15]. Once the number of modified histone sites exceed a certain threshold, the expression of *FLC* will be repressed (turned off) and then in turn, *FT* can be expressed. Thus, to activate the expression of *FT* in the right time, plants of this type need to incorporate the signals of both day length and temperature for determining flowering seasons.

As can be seen from the example network, biological entities need sophisticated biochemical interactions among network components and their environments to achieve certain biological functions.

1.6.3 Genomics, transcriptomics and proteomics

Genomics is the study of the genome which is the complete DNA set of a cell or an organism. For example, the genome of *e. coli* K-12 consists of 4,639,221 DNA nucleotides, of which 4288 protein-coding genes are annotated [Bla+97]. To make sense of the genome, genomics aims at systematic characterization and understanding of the structure, functions, mapping and editing of it [HK11; Wikc]. Similarly, the terminology can be applied to RNA and proteins as shown in Fig. 1.10.

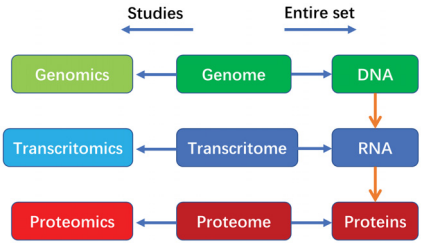


Fig. 1.10: The relationships of *-ome* and *-omics* on the levels of DNA, RNA and proteins.

The advances of technologies in omics have brought the studies to a level that they are able to produce system-wide quantitative and qualitative data of DNA, RNA and proteins [LH16]. The large amount of data shed light on the system-wide understanding of biology, for instance, the mapping from genotypes to phenotypes. Sophisticated algorithms for handling the massive data became fundamental [Mar17; HK11; Kit02a]. For example, efficient computational alignment played a central in assembling the DNA fragments from sequencing [Con+01]. And *Next Generation Sequencing* completely rely on the advances in computational biology to analyze enormous amount of short DNA reads [FB09; Mar17].

As an example of proteomics, the study of secretome (the entire set of secreted proteins) in muscle cells [Gru+18] is introduced to show the typical workflow of mass-spectrometry based proteomics. The data generated from this study has also been used in *Chapter 6*. As shown in Fig. 1.11, the cell culture has been divided into the medium and cell, from which the proteins were collected and processed to have the secretome and cellular proteome. The cellular proteome here was specifically referred to as all the proteins staying inside the cell. The obtained protein samples were then digested by enzymes to small fragments before fed into the mass spectrometer. The spectra of detected peptides were further analyzed by computational tools for identifying and quantifying the detected proteins. A few challenges exist in such proteomics studies. For example, the sample preparation had to make cell lysate in the medium as less as possible to have a good separation of secretome and cellular proteome; the computational tools had to make *false discovery rate (FDR)* as small as possible. FDR is the expected proportion of incorrectly reject the null hypothesis, or the number of false positives in all of the rejected hypotheses [Col14], which is often used as a measure of protein identification and quantification in mass spectrometry [Bur17].

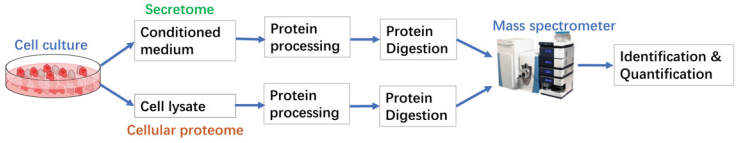


Fig. 1.11: The workflow of studying secretome using mass spectrometry techniques, adapted from [Gru+18].

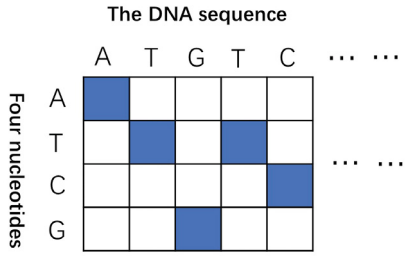


Fig. 1.12: The illustration of one-hot-key scheme for DNA sequences. The blue squares represent '1's, and white ones represent '0's.

1.6.4 Biological data for machine learning

The huge amount of available "omics" data including *genomics, transcriptomics, and proteomics* pave the way for machine learning to succeed in solving biological problems. Making sense of the increasing data in order to understand more about life systems becomes the central interest of fields of computational biology and bioinformatics. In terms of data preprocessing for machine learning, omics data can be used in the following ways.

The primary sequences as learning features

As the instructions of life systems, DNA sequences have the complete information for guiding biological processes. For example, to regulate the expression of a certain gene, the instructions should include how many mRNAs it produces in unit time and in turn how many proteins will be encoded in unit time. To conserve the information in DNA sequences, a natural way of representing sequences is to convert them to numerical vectors such that computers are able to comprehend. One-hot-coding is the most straightforward and common way to convert a character sequence to numerical vectors. The technique is also commonly used in natural language processing, for example, the vocabulary of a given document includes 5000 different words which are sorted by alphabetic order, then each word can be represented by a 5000 dimensional vector where the position the word sits is 1 and the rest are 0's. Similarly, an illustration for encoding DNA sequences is shown in Fig. 1.12, each letter is represented by a four dimensional vector. Then a DNA sequence with $n$bp can be represented by a $4 \times n$ matrix, which can be used as the feature matrix for properly designed model.

Features generated by biological prior knowledge

Imagining a 100bp long sequences, the number of possible sequences is 4^{100} which is an enormous space for a machine learning model to search for the right pattern. To reduce the search complexity, it is of great benefit to incorporate biological prior knowledge into either model building, e.g. neural networks structures, or feature generation.

For example, in a simplified procedure, the first step in translating an mRNA is that the ribosomes find the right position on the sequence to dock on. The positions are termed ribosome binding sites (RBS). It was found that special motifs, e.g. Shine-Dalgano sequence " AGGAGGU", exist in RBS to accelerate the binding process of ribosomes. Therefore, features can be constructed to represent the Shine-Dalgano motif. One simple way of doing that is to obtain matching scores of the consensus motif " AGGAGGU" to the actual mRNA sequences by sliding the motif over each position of the

mRNA. Without knowing this knowledge, when building models for investigating the regulatory factors of translation efficiency, one needs more efforts to make the model to capture it.

Generating features by prior knowledge to represent the problems of interest actually falls into the traditional way of applying machine learning methods, which requires less data than deep learning models which tend to learn the feature representation instead of encoding prior knowledge manually.

2 Synchronization Analysis of Complex Networks

2.1 Summary

Inspired by quorum sensing, a machinery of individuals to sense the state of the population in bacteria [MB01], the paper presented in *Section 2.2* studied the synchronization conditions of a network system which was made up of a population of biological compartments coupled by a dynamical common medium. Each compartment was assumed to be a biological oscillator consisting of a number of system components. The network system was then an adaptation of quorum sensing networks in such a way that all components of individual compartments are able to diffuse among compartments and the common medium. As an illustration, four compartments in Fig. 2.1 are interconnected through the common medium, each of which consists of four components $M_1,\ M_2,\ M3$ and M_4 in a cyclic structure.

The synchronization of the proposed networks was influenced by the dynamical properties of the components of individual compartments and the couplings on two levels: the interconnections among compartment components and the interconnections between the compartments and the common medium. It was shown that individual compartments should satisfy the following stability criteria: the bounded input and output relation and the diagonal stability for multi-dimensional systems to assure either stable oscillations or equilibria. And condition imposed on the dynamical coupling is determined by both the coupling strength and input-output properties of individual compartments. In the case of compartments being oscillators, the input-output dynamical properties and the couplings conditions ensured the synchronization of networked oscillators. Further, the synchronization characteristics were specified for biological systems formulated by the state-space representation, where the individual compartments were factorized into relations among system states, and the relation between output and systems states. The main results were presented as a theorem, stating that if the conditions were satisfied, the asynchronous rates between individual compartments were bounded by the external perturbations.

The derived theorem was applied to a group of coupled repressilators as a case study. The repressilator is a synthetic biological oscillator [EL00], consisting of three genes which form a repressing cyclic structure as shown in Fig. 2.2a. Fig. 2.2b shows the oscillation of a single repressilator. When the repressilators form a network structure as shown in Fig. 2.1, the simulation result (Fig. 2.2c) verified that satisfying the conditions of the theorem guaranteed the synchronization of the repressilators.

The results in *Section 2.2* extended the input-output approach reported by Scardovi et.al [SAS10] for the quorum-sensing inspired networks. However, this particular extension limited the generality of the derived results to oscillators that do not possess a cyclic structure and networks which do not form a structure as shown in Fig. 2.1.

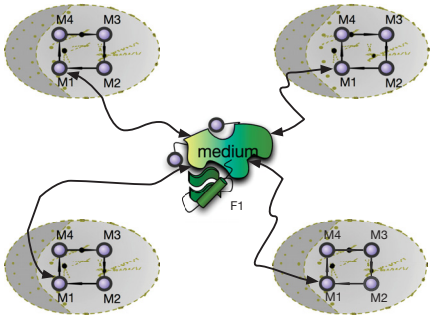
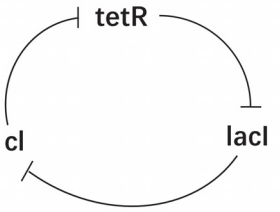
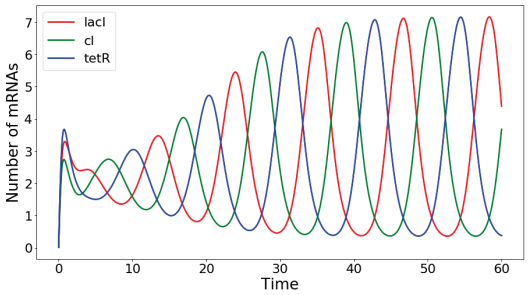


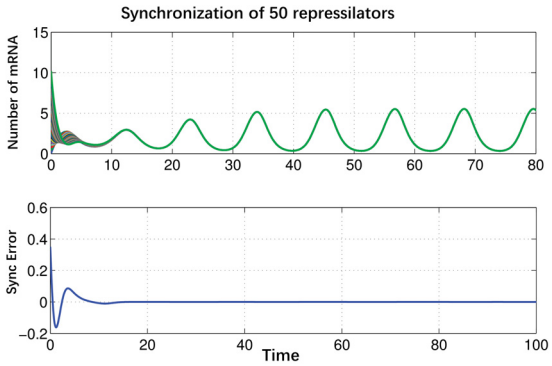
Fig. 2.1: Four compartments interconnected by a common medium and each consisting of four components forming a cyclic structure. In this illustration, the component M_1 is diffusing through the network. The way of constructing the network system also fixed the network structure.



(a) The repressilator, reproduced from [EL00].



(b) The oscillation of a single repressilator.



(c) Synchronization of a group of repressilators.

Fig. 2.2: Application of the main theorem in *Section 2.2* to repressilator networks. (a) The synthetic oscillator, repressilator, consists of three genes (*tetR, lacI, cI*) in a cyclic repressing structure. (b) It shows the oscillation of three mRNA levels of a single repressilator. (c) 50 repressilators which formed a network structure as shown in 2.1 achieved synchronization when the conditions in the main theorem were satisfied.

2.2 An Operator-theoretic Approach to Synchronization of Dynamically Coupled Biological Rhythms

Publication status

Linlin Zhao, and Dong Xue. "An operator-theoretic approach to synchronization of dynamically coupled biological rhythms" In 2016 35th Chinese Control Conference (CCC), pp. 9342-9348. IEEE, 2016.

Linlin Zhao's contributions

1. Extended the diffusive network system in [SAS10] to mathematically describe the quorum sensing inspired network systems.
2. Extended the theorems proposed by Scardovi et. al [SAS10] to characterize system properties such that synchronization can be achieved.
3. Applied the extended theorems to the state space formalism of repressilators and Goodwin oscillators.
4. Carried out the proofs and simulations, with help and discussions from the coauthor Dong Xue and Dr. Luca Scardovi.

An Operator-theoretic Approach to Synchronization of Dynamically Coupled Biological Rhythms

ZHAO Linlin[1], XUE Dong[2],

1. Institute for Mathematical Modeling of Biological Systems, Heinrich-Heine-University Dusseldorf, Dusseldorf 40225, Germany
E-mail: zhao@hhu.de

2. Institute for Information-Oriented Control, Technical University of Munich, Munich 80333, Germany
E-mail: dong.xue@tum.de

Abstract: This paper studies synchronization mechanisms for networks of biological rhythms. The network is made up of compartments (e.g., *E. coli* in a cell culture) which consist of heterogeneous subsystems (e.g., reaction pathways) interconnected by internal signaling. The compartments are, in turn, interrelated through common medium. Based on this structural foundation, synchronization conditions are provided from operator-theoretic view of point, which involve the input-output properties of individual compartment together with topological structure of underlying networks. Furthermore, as an additional goal, the paper also provides synchronization criterion for the networks modeled in the formalism of state-space. Finally, the proposed theory finds bio-chemical applications in the networks of toggle switches and repressilators, respectively.

Key Words: Synchronization, input-output operator, passivity, biological rhythms

1 Introduction

Biological rhythms are central to life and social community as well as generalized large complex systems [1]. The natural phenomenon covers a wide range of diverse physiological processes, for example cardiac system [2], circadian rhythms [3] etc.

The rhythms in nature usually interact with each other by various mechanisms, and also subject to external fluctuating due to exposing in noisy environment. Additionally, synthetic genetic circuit usually involves nonlinear components such as toggle switches [4] and oscillators [5]. Hence, to investigate the network-induced behavior, fully understanding of isolated individual dynamics is inadequate. Due to the complex nature of biological rhythms, researchers in many disciplines impose much effort on theoretical and experimental studies of gene regulatory networks, obtaining numerous profound results on modeling and analyzing such rhythms[4–10]. Among the studies in this area, synchronization is of fundamental significance in the coordination of rhythmic behavior among individual subsystems in a large networks.

The analysis of synchronization phenomena in networks has become an important topic in systems and control theory, motivated by diverse applications in physics, biology, and engineering[11–13]. However, in studying the literature on the synchronization analysis of biochemical network, one is stuck by the level of sophistication and technical complexity. To avoid involving the internal modeling complex, the input-output method has been applied to biological systems, which turns out to rely hardly on the knowledge of the intrinsic physical laws regulating the systems, and hence is fairly adequate to deal with systems with parameter and structural uncertainties [14]. Motivated by this pioneering work, we develop the analysis of biologically synchronous pattern from the input-output perspective associated with operator theory. In particular, we generalize differential passivity [15] to the operator field and then connect the synchronization property to energy-based concepts.

our motivation The seminal work of Bassler et. al unravels the underlying machinery of quorum sensing and its importance for communication in bacteria[16]. Inspired by the machinery where each cell secrets auto-inducers to a common medium then in turn individual cells sense the density of auto-inducer in the medium to monitor the state of the whole population, we extend it in such a way that all components of individual compartments are able to diffuse between compartments and the common medium. In case of compartments being oscillators, the synchronization conditions are heavily influenced by the interaction. Therefore a sound theoretical analysis for synchronization in such a scenario is needed. Theoretically armed with the input-output approach, in addition by exploring internal coupling within compartments (intracellular) and external interconnected in a common medium (intercellular), we are able to attain the conditions for synchronization.

main contribution In this work, we study the synchronization problem of biological rhythms with dynamical couplings in both intracellular and intercellular levels. Aiming at characterizing synchronization conditions, a network model is first proposed under the consideration that interaction of species attributes to internal signaling and intercellular signaling is accomplished by a common medium rather than directed cell-to-cell communication. Then, an input-to-output approach is employed to study synchronization properties of compartments. It is shown that the synchronization behaviors depend not only on input-output characteristics of the compartments, but also the diffusive coupling between the common medium and compartments. In addition, we also specify the synchronization characteristics of biological systems with models in state-space, as such is terminus a quo for utilization of theoretic and experimental tools derived on the basis of the state-space representation in control engineering.

The organization of the paper is as follows. In Section II,

This work is supported by CSC scholarship

some necessary notations and the model formulation are presented; In the Section III, we provide the main results based on the proposed model. And the main results are shown to be applicable to networks modeled in the formalism of state space. In Section IV, It is shown that biological model with very general interconnections can be applied with diagonal stability test and synchronization conditions for networks of these biological rhythms which are dynamically coupled are obtained. Finally the simulation results verify the conditions numerically.

2 Preliminaries and Notations

In this section, some preliminary knowledge and necessary notations are presented for later illustration. For a given signal a: $[0,\infty) \to \mathbb{R}$ in the extended space $\mathscr{L}_{2e}$, the restriction $a_T = a|_{[0,T]}$ belongs to $\mathscr{L}_2(0,T)$, for any $T > 0$. Given any $a \in \mathscr{L}_{2e}$ and any $T > 0$, let $\|a\|_T$ denote the $\mathscr{L}_2$ norm of the restriction a_T, and let the inner product of a_T and b_T be denoted by $\langle a,b\rangle_T$ for given $a,b \in \mathscr{L}_{2e}$ and any $T > 0$.

Bacterial cells in a cell culture exchange information with each other through the growing environment by sensing certain chemicals or nutrients. Different components within a cell also exchange information or energy through cytoplasm. We consider a network consisting n compartments, each composed of N subsystems which are called species[14], communicate with each other through a common medium. Assuming the kth species of all compartments are diffusing through the network, the input-output behavior of the jth compartment can be written as

$$y_{k,j} = M_k u_{k,j} \tag{1}$$

$$u_{k,j} = z_{k,j} + \sum_{i=1}^{N} a_{k,i} y_{i,j} + b_k(s_k - y_{k,j}), \tag{2}$$

where $k = 1,2,\cdots,N$, $j = 1,2,\cdots,n$ and $u_{k,j}$ is input of mapping M_k. Exogenous signal is denoted by $z_{k,j}$, and $\sum_{i=1}^{N} a_{k,i} y_{i,j}$ describes the interaction among different species in the jth compartment, which is referred to species interaction, and b_k describes the diffusive interaction strength between each species and the common medium. The coefficients $a_{k,i} \in \mathbb{R}$, $k,i = 1,2,\cdots,N$, represent the interconnection among different species, and are identical in each compartment. The interconnection matrix for species within each compartment can be written as $A := [a_{k,i}]$, $k,i = 1,2,\cdots,N$. s_k represents the concentration of the corresponding species of kth species in the common medium, whose dynamics can be formulated as

$$s_k = F_k w_k \tag{3}$$

$$w_k = \sum_{i=1}^{n} c_k(y_{k,i} - s_k), \tag{4}$$

where w_k is the input of mapping F_k. It is assumed that the common medium is equally connected to all compartments, i.e. interaction strength between the medium and all compartments is the same. For instance, for species k, the diffusive strength from the medium to each compartment is b_k, from each compartment to the medium c_k. The mechanism can be illustrated by the graph as Figure 1.

From a biochemical point of view, the network model (1)-(4) can describe as a population of cells living in a common

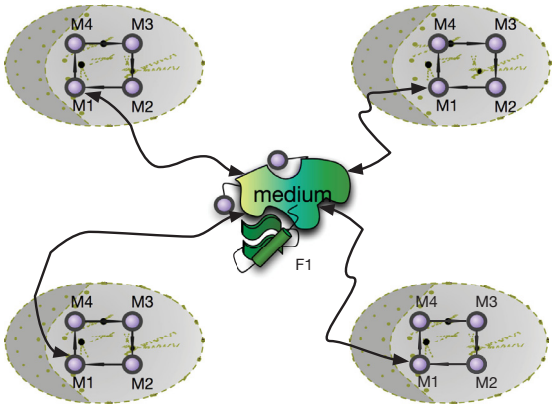


Fig. 1: Four compartments interconnect by a common medium and each consists of four species forming a cyclic structure

medium and such medium can be considered as a large cell which is adjacent to all normal cells. Assuming the volumes of the cell and the medium are X_c, X_m respectively and defining constant $\varepsilon = X_c/X_t$, the relation between coefficients b_k and c_k is $c_k = \varepsilon b_k$, of which the related work can be found in [17–20].

Grouping the outputs of species k in n compartments into vector form gives rise to $Y_k = (y_{k,1}, y_{k,2}, \cdots, y_{k,n})^T$. Similarly, inputs and external signals of them can be denoted as $U_k = (u_{k,1}, u_{k,2}, \cdots, u_{k,n})^T$, $Z_k = (z_{k,1}, z_{k,2}, \cdots, z_{k,n})^T$ respectively. The output of the medium s_k diffuses to kth species of all n compartments as their inputs, thus we write $S_k = (s_k, s_k, \cdots, s_k)^T$. Using those notations, the interconnections (2) can be rewritten as

$$U_k = Z_k + \sum_{i=1}^{N} a_{k,i} Y_i + b_k(S_k - Y_k), \quad k = 1,2,\cdots,n. \tag{5}$$

The coefficients $a_{k,i} \in \mathbb{R}, k,i = 1,2,\cdots,N$ represent the interconnection between different species. The interconnection matrix of each compartment can be written as $A := [a_{k,i}], k,i = 1,2,\cdots,N$.

The differences among the outputs of the same species in different compartments, which can be defined by $Y_k^\Delta := [y_{k,1} - \bar{Y}_k, \cdots, y_{k,n} - \bar{Y}_k]^T$ where the average of all outputs of species k is $\bar{Y}_k := \frac{1}{n} 1_n^T Y_k$, characterize the synchronization of the network system (1)-(2). The same notation is used to define Z_k^Δ and X_k^Δ. Further we introduce the operator ψ as

$$\psi = Q^T Q = I_n - \frac{1}{n} 1_n 1_n^T, n \in \mathbb{Z}_+,$$

where

$$Q = \begin{bmatrix} -1+(n-1)\nu & 1-\nu & -\nu & \cdots & -\nu \\ -1+(n-1)\nu & -\nu & 1-\nu & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & -\nu \\ -1+(n-1)\nu & -\nu & \cdots & -\nu & 1-\nu \end{bmatrix}_{n\times(n-1)}$$

with $\nu = \frac{n-\sqrt{n}}{n(n-1)}$. From a straightforward calculation, one can see that $\psi Y_k = Q^T Q Y_k = Y_k^\Delta$, which means the operator

ψ measures the disagreement between each element and the average of the elements in a vector. Moreover we define $\hat{Y}_k := QY_k$, and the same for $\hat{U}_k, \hat{Z}_k, \hat{S}_k$.

Before moving on proceeding, we need the notion of cocoercivity and a critical lemma from [14] which provided as follows.

Definition 2.1. *Let $M : \mathscr{L}_{2e}^m \to \mathscr{L}_{2e}^m$. Then M is relaxed cocoercive if there exists some $\theta_c \in \mathbb{R}$ such that in every pair of inputs $u_1, u_2 \in \mathscr{L}_{2e}^m$*

$$\theta_c \|Mu_1 - Mu_2\|_T^2 \leq \langle Mu_1 - Mu_2, u_1 - u_2 \rangle_T, \quad \forall T \geq 0. \tag{6}$$

If (6) holds with $\theta_c \geq 0$, then M is called monotone. If (6) holds with $\theta_c > 0$, then M is cocoercive.

Lemma 2.1. *Consider the open-loop system $y_{k,j} = M_k u_{k,j}$. If the mappings $M_k, k = 1,2,\cdots,N$ are relaxed cocoercive then*

$$\theta_k \|\hat{Y}_k\|_T^2 \leq \langle \hat{Y}_k, \hat{U}_k \rangle_T, \quad k = 1,2,\cdots,N, \tag{7}$$

for each $T > 0$ and every $X_k \in L_{2e}^n$.

Without interactions between the medium and compartments, i.e. $b_k = 0, k = 1,2,\cdots,N$, the n compartments are isolated and their stability depends on the interactions among the internal species. The notion of *dissipativity matrix* is introduced to characterize stability [21], [22], [23], which is defined as

$$E_b := A - \Theta_b,$$

where $\Theta_b := diag(b_1, b_2, \cdots, b_N)$, $b := col(b_1, \cdots, b_N)$ and A is the interconnection matrix $[a_{k,i}]$. The dissipative matrix E_b is dianogally stable if there exists diagonal matrix $D > 0$ such that $E_b^T D + DE_b < 0$. The stability of E_b implies stability of the isolated systems.

When isolated compartments are connected by a common medium to form a quorum-sensing-like network, dynamic of each compartment is influenced by the network in such a way that the diagonal entries of its dissipativity matrix are augmented with network properties. In bacterial colony where bacteria sense the information from the environment and detect the density of the population by secreting autoinducer, whereas in our network model, each species in individual compartments is allowed to diffuse through the common medium which is a more general scenario.

3 Synchronization results for networks with dynamical coupling

In this section it is shown that input-output properties of subsystems of each compartment and the diffusing strength are key ingredients of synchronization conditions.

3.1 Main results

We now consider that the common medium displays the same dynamics to all compartments such that the input from the medium to each compartment is identical. The common medium is influenced by all the compartments and the output of the medium dynamic diffuses back to compartments.

Theorem 3.1. *Consider the close loop system (1)-(4) and suppose the following assumptions:*

1) *Each mapping M_k are θ_k-relaxed cocoercive, $k = 1,2,\cdots,N$;*
2) *For $k = 1,2,\cdots,N$, $\hat{\theta}_k = \theta_k + b_k > 0$, where b_k is the compartment coupling strength of species k;*
3) *The dissipativity matrix $E_{\hat{\theta}_k}$, defined as*

$$E_{\hat{\theta}_k} = A - \Theta_{\hat{\theta}_k}.$$

where $A = [a_{k,i}]$, $k,i = 1,2,\cdots,N$ and $\Theta = diag(\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_N)$ is diagonally stable.

Then for all $z_{k,j}$, $y_{k,j}$, $k = 1,2,\cdots,n$ that satisfy (1)-(4) we have

$$\|Y^\Delta\|_T \leq \eta \|Z^\Delta\|_T.$$

for some $\eta > 0$, and all $Z \in \mathscr{L}_{2e}^{Nn}$.

Proof. Consider inputs $U_k = V_k + b_k(S_k - Y_k)$, then

$$\hat{U}_k = \hat{V}_k + b_k QS_k - b_k \hat{Y}_k.$$

Since $QS_k = 0$, we obtain

$$\hat{U}_k = \hat{V}_k - b_k \hat{Y}_k. \tag{8}$$

Mappings $M_k, k = 1,2,\cdots,n$ are θ_k-relaxed cocoercive, therefore using Lemma 2.1 and (8) yields

$$\begin{aligned} \theta_k \|\hat{Y}_k\|_T^2 &\leq \langle \hat{Y}_k, \hat{U}_k \rangle_T \\ &= \langle \hat{Y}_k, \hat{V}_k - b_k \hat{Y}_k \rangle_T \\ &= \langle \hat{Y}_k, \hat{V}_k \rangle_T - b_k \|\hat{Y}_k\|_T^2. \end{aligned}$$

It follows that

$$\hat{\theta}_k \|\hat{Y}_k\|_T^2 \leq \langle \hat{Y}_k, \hat{V}_k \rangle_T, \tag{9}$$

with $\hat{\theta}_k = \theta_k + b_k$.

We define

$$W_k = Z_k + \sum_{i=1}^{N} a_{k,i} Y_i. \tag{10}$$

By stacking vectors and using Kronecker product, (10) can be rewritten as

$$W = Z + (A \otimes I_n) Y. \tag{11}$$

Assumption 3 states the dissipativity matrix $E_{\hat{\theta}_k} = A - \Theta_{\hat{\theta}_k}$ is diagonally stable with $\Theta = diag(\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_N)$. Thus there exists $D = diag(d_1, d_2, \cdots, d_N)$ satisfying

$$DE_{\hat{\theta}_k} + E_{\hat{\theta}_k}^T D < 0, \tag{12}$$

such that $DE_{\hat{\theta}_k} + E_{\hat{\theta}_k}^T D < -2\beta$ holds for some small $\beta > 0$. Noticing that

$$\begin{aligned} \langle Dv, E_{\hat{\theta}} v \rangle_T &= \frac{1}{2} \int_0^T v^T(t) (DE_{\hat{\theta}_k} + E_{\hat{\theta}_k}^T D) v(t) dt \\ &\leq -\beta \|v\|_T^2 \end{aligned}$$

and combining with (9) we have $\langle d_k \hat{Y}_k, \hat{W}_k - \hat{\theta} \hat{Y}_k \rangle_T \geq 0$ for $k = 1,2,\cdots,k$. Stacking the vectors gives

$$\left\langle (D \otimes I_{n-1}) \hat{Y}, \hat{W} - (\Theta_{\hat{\theta}} \otimes I_{n-1}) \hat{Y} \right\rangle_T \geq 0. \tag{13}$$

Substituting (11) into (13) leads to

$$\left\langle (D \otimes I_{n-1}) \hat{Y}, \hat{Z} + (E_{\hat{\theta}} \otimes I_{n-1}) \hat{Y} \right\rangle_T \geq 0 \tag{14}$$

Consequently we have

$$\begin{aligned}\alpha\|\hat{Z}\|_T\|\hat{Y}\|_T &\geq \left\langle (D\otimes I_n)\hat{Y}, Z\right\rangle_T \\ &\geq \left\langle (D\otimes I_n)\hat{Y}, (E_{\hat{\theta}}\otimes I_n)\right\rangle_T \\ &\geq \beta\|\hat{Y}\|_T^2\end{aligned}$$

for some $\alpha > 0$. Directly it comes to

$$\|\hat{Y}\|_T \leq \eta\|\hat{Z}\|_T, \quad \forall T > 0, \tag{15}$$

where $\eta = \alpha/\beta$.

□

It is worthy of noting that no constraints are imposed on the dynamic of the medium since the species are diffusive and the dynamic of the medium is heavily influenced by the dynamics of individual compartments. And also the assumptions have no requirements about the structural properties of network.

3.2 Applications to state space formalism

Theorem 1 is applicable to synchronization of networked systems formulated in state space.

$$\begin{aligned}\dot{x}_{k,j} &= f_k(x_k, v_{k,j}) \\ y_{k,j} &= h_k(x_{k,j}) \\ \dot{w}_k &= g_k(w_k, \xi_k) \\ s_k &= p_k(w_k),\end{aligned} \tag{16}$$

where

$$v_{k,j} = z_{k,j} + \sum_{i=1}^{N} a_{k,i}y_{i,j} + b_k(s_k - y_{k,j}) \tag{17}$$

$$\xi_k = \sum_{i=1}^{n} c_k(y_{k,i} - s_k) \tag{18}$$

for all $k = 1,2,\cdots,N$, and all $j = 1,2,\cdots,n$. The state of the medium with respect to species k is denoted by w_k. And f_k, h_k, g_k, p_k are continuous functions. The following corollary concludes the synchronization conditions of the networked system.

Corollary 3.1. *Consider system* (16) *and assume the mappings M_j, $j = 1,\cdots,n$ and F_k, $k = 1,2,\cdots,N$ with zero initial conditions are well defined. Consider the closed loop system defined by* (16) *with inputs as in* (2) *and* (4). *If 1) θ-relaxed cocoercive mapping M_k from $v_{k,j}$ to $y_{k,j}$ can be factorized into two functions f_k and h_k, and the mapping F_k from ξ_k to s_k can be factorized into g_k and p_k, and 2) the dissipativity matrix $E_{\hat{\theta}}$ as defined in assumption 3 of Theorem 1 is diagonally stable. Then, there exists $\lambda > 0$, such that*

$$\|Y^{\Delta}\|_T \leq \lambda\|Z^{\Delta}\|_T,$$

If $Z = 0$, the outputs asymptotically synchronize.

Due to the assumption that θ-relaxed cocoercive mapping M_k can be factorized to function f_k and h_k, the proof follows the same reasoning of the proof of Theorem 1.

Remark: For the biological rhythms of interest, in order to obtain the synchronization conditions using our input-output method, a primary assumption on the dynamics of the system is the subsystems are required to be relaxed cocoercive. However, biological systems in general possess high complexity which leads to high nonlinearity in mathematical formulations. In turn, many realistic systems can hardly satisfy the assumption directly. To tackle this sophistication, we can usually decompose the systems in order to obtain the cocoercivity. For instance, $\dot{x}_1 = -x_1 + 1/(1+x_2^n)$, i.e. a subsystem containing a Hill function of the state of another subsystem will render it not relaxed cocoercive. Without any alteration to the dynamic of the subsystem, once we separate the nonlinearity out, the equation can be factorized as $\dot{x}_1 = -x_1 + u$, which is relaxed cocoercive with gain 1, and $u = 1/(1+x_2^n)$, which is also relaxed cocoercive with gain $\frac{4n}{n^2-1}\sqrt[n]{\frac{n-1}{n+1}}$. This technique will be utilized in case study.

3.3 Dissipativity Matrix and Diagonal stability

Besides relaxed cocoercivity of mappings and algebraic connectivity, another key ingredient for concluding synchronization of interconnected systems is the diagonal stability of the corresponding dissipativity matrix which incorporates information of cocoercivity of subsystems, interconnection of compartments and signs of interconnection terms. The work in [23] proposed a result for proving diagonal stability of interconnected systems with cactus structures that can be interpreted as combinations of arbitrary numbers of cyclic sub-structures. Therefore systems with more general structures become easy to tackle based on framework of diagonal stability and secant criterion. To analyze interconnected systems with general interconnection such as networked repressilator(see the following application), we introduce the following lemma from the results of [23].

Lemma 3.1. *Consider a matrix with the form of*

$$\hat{E} = \begin{bmatrix} -1 & 0 & \cdots & e_{1n} \\ e_{21} & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & e_{n,n-1} & -1 \end{bmatrix}, \tag{19}$$

which consists of a single loop. A matrix E that can be brought to the form of $\hat{E}$ in (19) *upon a permutation is diagonally stable if and only if:*

$$|\gamma|\Phi(sgn(\gamma), n) < 1,$$

where

$$\Phi(sgn(\gamma), n) = \begin{cases} cos^n(\pi/n) & \text{if } \gamma < 0 \\ 1 & \text{if } \gamma > 0. \end{cases}$$

In the classic cyclic structures, $e_{21}\cdots e_{n,n-1}$ are required to be positive while e_{1n} to be negative. However for many practical biological interconnections where repressions usually exist in intermediate reactions, the cyclic structure is not suitable to characterize them. Lemma 3.1 does not require specific signs for terms of $e_{21}\cdots e_{n,n-1}, e_{1n}$ but only concerns the sign of their product. Therefore it is applicable to a wide range of biological models.

4 Further Discussions and Applications

In this section, we apply the proposed theory to analyze synchronization in networks of genetic regulatory systems:

toggle switches and repressilators. The networked switches or oscillators interact with each other through a common dynamical medium where each species is allowed to diffuse into.

4.1 Case 1: toggle switches

A genetic toggle switch is constructed by means of mutual inhibition using two repressors and two promoters [4]. To generate bistability of the toggle switch, each promoter is inhibited by the repressor which is transcribed the opposing promoter. To illustrate the cyclic mechanism, take a natural switch from the bacteriophage λ for example [6], two repressors *cl* and *cro*, two promoters P_R and P_{RM}, the promoter P_{RM} controls expression of the gene *cl*, and protein *CI* represses P_R, whereas P_R controls expression of the gene *cro*, and protein *Cro* represses P_{RM}.

A dimensionless model was proposed in [4] to understand and approximately describe the biological dynamics of the toggle switch:

$$\dot{w} = \frac{a_1}{1+v^b} - w \tag{20}$$

$$\dot{v} = \frac{a_2}{1+w^c} - v, \tag{21}$$

where w is the concentration of the first repressor, v is the concentration of the second repressor, a_1, a_2 are the effective rate of synthesis of the first repressor, b is the cooperativity of repression of second promoter and c is the cooperation of repression of first promoter. The first term in each equation models the cooperative repression of constitutively transcribed promoters and the second term represents the degradation.

Now we use Lemma 3.1 to obtain sufficient condition for the diagonal stability of toggle switches. Separating out two nonlinearities, input-output form of (20), (21) can be written as

$$M_1 : \dot{x}_1 = -x_1 + \alpha_1 u_1, \quad u_1 = -y_4 \tag{22}$$

$$M_2 : y_2 = -\frac{1}{1+u_2^b}, \quad u_2 = x_1 = y_1 \tag{23}$$

$$M_3 : \dot{x}_3 = -x_3 + \alpha_2 u_3, \quad u_3 = -y_4 \tag{24}$$

$$M_4 : y_4 = -\frac{1}{1+u_4^c}, \quad u_4 = x_3 = y_3 \tag{25}$$

where x_1 is the concentration of the first repressor corresponding to w in (20), x_3 is the concentration of the second repressor corresponding to v in (21), y_2, y_4 are separated nonlinearities from (20), (21) and are also considered as two species, u_1-u_4 are the associated inputs of four species. Therefore the original toggle switch model is recomposed to a cyclic-like loop as

$$M_1 \xrightarrow{+} M_2 \xrightarrow{-} M_3 \xrightarrow{+} M_4 \xrightarrow{-} M_1$$

From the relation between inputs and outputs of four species, we have the interconnection matrix of the modified toggle switch model as

$$A = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

By a straightforward calculation(section V of [14]), we have cocoercive gains of M_1, M_2, M_3, M_4 as $\theta_1 = 1/a_1, \theta_2 = \frac{4c}{c^2-1}\sqrt[c]{\frac{c-1}{c+1}}, \theta_3 = 1/a_2, \theta_4 = \frac{4b}{b^2-1}\sqrt[b]{\frac{b-1}{b+1}}$ respectively. The dissipativity matrix associated with (22) is

$$E = A - diag\{\theta_1, \theta_2, \theta_3, \theta_4\} = \begin{bmatrix} -\theta_1 & 0 & 0 & -1 \\ 1 & -\theta_2 & 0 & 0 \\ 0 & -1 & -\theta_3 & 0 \\ 0 & 0 & 1 & -\theta_4 \end{bmatrix}.$$

By Lemma 3.1, dissipativity matrix E is diagonally stable if

$$\theta_1\theta_2\theta_3\theta_4 > 1, \tag{26}$$

since E can be brought to the form of $\hat{E}$ in (19) by permutation PE where $P = diag\{\frac{1}{\theta_1}, \frac{1}{\theta_2}, \frac{1}{\theta_3}, \frac{1}{\theta_4}\}$ and obviously the product of the off-diagonal elements of PE is positive.

Now we consider a network of n toggle switches which are dynamically coupled. In each toggle switch, M_2, M_4 are separated nonlinearities and only M_1, M_3 can diffuse through the network. Then, the dynamics of the networked system can be modeled as provised in Table 1.

Table 1: Toggle switches

	dynamics	input
M_1	$\dot{x}_{1,j} = -x_{1,j} + a_1 u_{1,j} + b_1(s_{1,j} - x_{1,j})$	$u_1 = -y_4$
M_2	$y_{2,j} = -\frac{1}{1+u_{2,j}^c}$	$u_2 = x_1 = y_1$
M_3	$\dot{x}_{3,j} = -x_{3,j} + a_2 u_{3,j} + b_3(s_{3,j} - x_{3,j})$	$u_3 = -y_4$
M_4	$y_{4,j} = -\frac{1}{1+u_{4,j}^b}$	$u_{4,j} = x_{4,j} = y_{4,j}$
F_1	$\dot{s}_{1,j} = -s_{1,j} + c_1 \sum_{i=1}^{n} (x_{1,i} - s_{1,j})$	
F_3	$\dot{s}_{3,j} = -s_{3,j} + c_3 \sum_{i=1}^{n} (x_{3,i} - s_{3,j})$	

The dissipativity matrix $\hat{E}$ should be constructed following the assumption 3 in Theorem 1. The sufficient condition for diagonal stability is

$$(\theta_1 + b_1)(\theta_3 + b_3) > \frac{1}{\theta_2\theta_4}. \tag{27}$$

To determine values of cocoercive gains, parameters in (20)-(21) has to be chosen to assure the existence of bistability of toggle switches. The dynamical analysis of (20), (21) in [4] showed that the bistability of toggle switch is favoured by $b, c > 1$ that are cooperative repressions of transcription and balanced rates of synthesis, that is, difference between a_1 and a_2 cannot be too large. Thus we choose the parameter set as $b = 4, c = 3, a_1 = 10, a_2 = 20$.

From the discussion in section V of [14], straightforward calculation of cocoercive gains gives rise to $\theta_1 = 0.1, \theta_2 = 1.2, \theta_3 = 0.05, \theta_4 = 0.9$ and substituting them into (27) yields

$$(0.1 + b_1)(0.05 + b_3) > 1 \tag{28}$$

Therefore, the validity of (28) and discussion above result in synchronization of the networked switches by Theorem 1.

4.2 Case 2: repressilators

In this part, we analyze the synchronization of repressilators which share the common dynamical medium and assume the medium displays the same dynamic to all repressilators.

The seminal work of Elowitz et. al [5] gave rise to the model of repressilator as

$$\dot{w}_i = -w_i + \frac{a}{1+z_j^p} + a_0 \tag{29}$$

$$\dot{z}_i = -b(z_i - w_i), \tag{30}$$

where w represent mRNA concenstrations, z represent the corresponding protein concenstrations ($i = lacI, tetR, cl$, $j = cl, lacI, tetR$); a_0 is the leakiness of the given type of promoter in the presence of saturating repressors and $a + a_0$ in the absence; b is the lifetime ratio of mRNA and protein; p is the Hill coefficient.

Table 2: Repressilator Network

	dynamics	input
M_1	$\dot{x}_{1,j} = -x_{1,j} + av_{1,j}$	$u_{1,j} = -y_{9,j}$
M_2	$\dot{x}_{2,j} = -bx_{2,j} + bv_{2,j}$	$u_{2,j} = y_{1,j}$
M_3	$y_{3,j} = -\frac{1}{1+v_{3,j}^p}$	$u_{3,j} = y_{2,j}$
M_4	$\dot{x}_{4,j} = -x_{4,j} + av_{4,j}$	$u_{4,j} = -y_{3,j}$
M_5	$\dot{x}_{5,j} = -bx_{5,j} + bv_{5,j}$	$u_{5,j} = y_{4,j}$
M_6	$y_{6,j} = -\frac{1}{1+v_{6,j}^p}$	$u_{6,j} = y_{5,j}$
M_7	$\dot{x}_{7,j} = -x_{7,j} + av_{7,j}$	$u_{7,j} = -y_{6,j}$
M_8	$\dot{x}_{8,j} = -bx_{8,j} + bv_{8,j}$	$u_{8,j} = y_{7,j}$
M_9	$y_{9,j} = -\frac{1}{1+v_{9,j}^p}$	$u_{9,j} = y_{8,j}$
F_k	$\dot{s}_k = -cs_k + \sum_{i=1}^{n} c_k(x_{k,i} - s_k)$	

The repressilator model determines the kinetics of a synthetic oscillatory network consisting of three repressor genes $cI, tetR$, and $lacI$. The protein of the first repressor cI, CI, inhibits the transcription of the second gene $lacI$, whose product protein in turn inhibits cI's expression. In this cycle, one gene is the repressor for another, and without repressor, it will continuously produce protein. The ideal case is that in the presence of saturating repressors, there will be no product from the repressed gene. The corresponding deterministic model can be described as (29)-(30) with $\alpha_0 = 0$ which is the case considered in this section. Due to different parameter choices and initial conditions, the repressilator model has at least two dynamical behaivors: limit-cycle oscillations or stable steady state. Large value of a, low leakiness with small a_0, and proper b are beneficial for oscillatory behavior of repressilators.

We modify the model (29)-(30) into input-output form that is given in Table 2 in which $j = 1, 2, \cdots, n$, $y_{k,j}$, $k = 3, 6, 9$, $x_{k,j}$, $k = 1, 2, 4, 5, 7, 8$ and $v_{k,j}$, $k = 1, 2, \cdots, 9$ determine the input-output relations between subsystems. F_k, $k = 1, 2 \cdots, 9$ define the same dynamic of the medium to all compartments, with c_k, $k = 1, 2, \cdots, 9$

By the similar tracking as the case of toggle switches, we have cocoercive mappings M_k, $k = 1, \cdots, 9$ with cocoercive gains $(a, 1, \frac{1}{\theta_h}, a, 1, \frac{1}{\theta_h}, a, 1, \frac{1}{\theta_h})$ respectively, where $\theta_h = \frac{4p}{p^2-1}\sqrt[p]{\frac{p-1}{p+1}}$. Similarly the sufficient condition for diagonal stability of dissipativity matrix $\hat{E}$ is

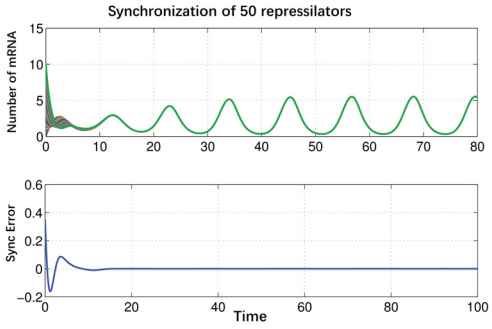


Fig. 2: Synchronization of 50 repressilators which are dynamically coupled by the common medium.

$$\prod_k (\theta_k + b_k) > \frac{1}{\theta_3\theta_6\theta_9}\cos^9(\frac{\pi}{9}), k = 1, 2, 4, 5, 7, 8. \tag{31}$$

Substituting the gains $(a, 1, \frac{1}{\theta_h}, a, 1, \frac{1}{\theta_h}, a, 1, \frac{1}{\theta_h})$ into (31), it follows that

$$(\frac{1}{a} + b_1)(1 + b_2)(\frac{1}{a} + b_4)(1 + b_5)(\frac{1}{a} + b_7)(1 + b_8) > \frac{1}{\theta_h^3}\cos^9(\frac{\pi}{9}) \tag{32}$$

with $\theta_h = \frac{4p}{p^2-1}\sqrt[p]{\frac{p-1}{p+1}}$ which can be verified as an decreasing function with Hill coefficient p. Now one has to chooose proper values for a and p in such a way that the oscillation of repressilators is certain.

Following the discussion in [5], we choose values for parameters as $a = 10, b = 1, p = 2$ to ensure sustained oscillation of repressilators. Consequently we have

$$(0.1 + b_1)(1 + b_2)(0.1 + b_4)(1 + b_5)(0.1 + b_7)(1 + b_8) > \frac{1}{\theta_h^3}\cos^9(\frac{\pi}{9}) := c \tag{33}$$

Then if we only allow the first species in each compartment can diffuse into the medium and assume the coupling strength is ε, i.e. $b_k = \varepsilon, for\ k = 1$ and $b_k = 0, for\ k = 2, 4, 5, 7, 8$. It follows the condition

$$0.1 + \varepsilon > 100c$$

The simulation results are shown in Figure 2, where a network of 50 repressilators are interconnected based on the model (1)-(4). Under the condition of $\varepsilon > 16$, synchronization of repressilators is achieved.

5 Conclusion and Future work

Based on input-output approach, synchronization conditions for networks with dynamical coupling are presented in the paper. The conditions rely heavily on input-output properties of subsystems, diffusive strength between the common

medium and each compartment, and diagonal stability of the dissipativity matrix corresponding to the compartments. The case that biological rhythms which possess complex interconnections are coupled dynamically by a common medium is applied with the proposed theoretical results to obtain synchronization conditions. The simulation results shows the effectiveness the results.

6 Acknowledgement

The authors appreciate Dr. Luca Scardovi for insightful advices and discussions and CSC for funding the work.

References

[1] L. Glass, "Synchronization and rhythmic processes in physiology," *Nature*, vol. 410, no. March, pp. 277–284, 2001.

[2] J. G. Cleland, J.-C. Daubert, E. Erdmann, N. Freemantle, D. Gras, L. Kappenberger, and L. Tavazzi, "The effect of cardiac resynchronization on morbidity and mortality in heart failure," *New England Journal of Medicine*, vol. 352, no. 15, pp. 1539–1549, 2005.

[3] N. Komin, A. C. Murza, E. Hernandez-Garcia, and R. Toral, "Synchronization and entrainment of coupled circadian oscillators," *Interface focus*, vol. 1, pp. 167–176, 2011.

[4] T. Gardner, C. R. Cantor, and J. Collins, "Construction of a genetic toggle switch in escherichia coli," *Nature*, vol. 403, pp. 339–342, 2000.

[5] E. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, pp. 335–338, 2000.

[6] J. Hasty, D. Mcmillen, and J. Collins, "Engineered gene circuits," *Nature*, vol. 420, pp. 224–230, 2002.

[7] H. Kobayashi, M. Karn, M. Araki, K. Chung, T. S. Gardner, and C. R. Cantor, "Programmable cells: Interfacing natural and engineered gene networks," *Proceedings of the National Academy of Science of the USA*, vol. 101, pp. 8414–8419, 2004.

[8] T. Danino, O. Mondragon-Palomino, L. Tsimring, and J. Hasty, "A synchronized quorum of genetic clocks," *Nature*, vol. 463, pp. 326–330, 2010.

[9] J. Hasty, D. McMillen, F. Isaacs, and J. Collins, "Computational studies of gene regulatory networks: in numero molecular biology," *Nature Review. Genetics*, vol. 2, pp. 268–279, 2001.

[10] P. Purnick and R. Weiss, "The second wave of system biology: from modules to systems," *Nature Review. Molecular cell biology*, vol. 10, pp. 410–422, 2009.

[11] S. Strogatz, "From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators," *Physica D: Nonlinear Phenomena*, vol. 143, Sep. 2000.

[12] T. Danino, O. Mondragón-Palomino, L. Tsimring, and J. Hasty, "A synchronized quorum of genetic clocks." *Nature*, vol. 463, no. 7279, pp. 326–30, Jan. 2010.

[13] H. Kori, C. G. Rusin, I. Z. Kiss, and J. L. Hudson, "Synchronization engineering: Theoretical framework and application to dynamical clustering," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 18, no. 2, p. 026111, 2008.

[14] L. Scardovi, M. Arcak, and E. Sontag, "Synchronization of interconnected systems with application to biochemical networks: an input-output approach," *IEEE Transactions on Automatic Control*, vol. 55, 2010.

[15] F. Forni, R. Sepulchre, and A. van der Schaft, "On differential passivity of physical systems," *CoRR*, vol. abs/1309.2558, 2013.

[16] C. Waters and B. Bassler, "Quorum sensing: cell-to-cell communication in bacteria," *Annu. Rev. Cell Dev. Biol.*, vol. 21, pp. 319–346, 2005.

[17] J. Garcia-Ojalvo, M. Elowitz, and S. Strogatz, "Modeling a synthetic multicellular clock: repressilators coupled by quorum sensing," *Proceedings of the National Academy of Science of the USA*, vol. 101, pp. 10 955–10 960, 2004.

[18] G. Katriel, "Synchronization of oscillators coupled through an environment," *Physica D: Nonlinear Phenomena*, vol. 237, no. 22, pp. 2933–2944, 2008.

[19] S. De Monte, F. d'Ovidio, S. Danø, and P. Sørensen, "Dynamical quorum sensing: Population density encoded in cellular dynamics," *Proceedings of the National Academy of Sciences*, vol. 104, no. 47, p. 18377, 2007.

[20] A. Kuznetsov, M. Kærn, and N. Kopell, "Synchrony in a population of hysteresis-based genetic oscillators," *SIAM Journal on Applied Mathematics*, pp. 392–425, 2004.

[21] M. Arcak and E. Sontag, "A passivity-based stability criterion for a class of biochemical reaction networks," *Mathematical Bioscience and Engineering*, vol. 5, 2008.

[22] ——, "Diagonal stability of a class of cyclic systems and its connection with the secant condition," *Automatica*, vol. 42, pp. 1531–1537, 2006.

[23] M. Arcak, "Diagonal stability on cactus graphs and application to network stability analysis," *IEEE Transactions on Automatic Control*, vol. 56, pp. 2777–2777, 2011.

3 Analytical and Data-driven Analysis of Flowering Time Determination

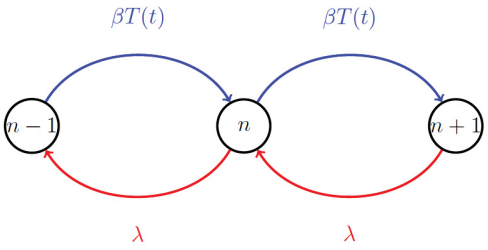
3.1 Summary

In order to successfully reproduce, plants must sense changes in their environment and flower at the correct time. Many plants in temperate regions utilize day length and *vernalization*, a process of exposing plants in prolonged cold, to determine when to flower. On the molecular level of the model plant *Arabidopsis*, vernalization accelerates flowering by down-regulating the protein FLC which prevents flowering. The down-regulation involves histone modifications on the *FLC* locus. When the expression of *FLC* is silenced, the flowering signal FT is produced to trigger flowering. Based on this qualitative understanding of flowering regulation in *Arabidopsis*, the manuscript in *Section 3.2* investigated the regulatory mechanism from an information point of view using temperature and day length data.

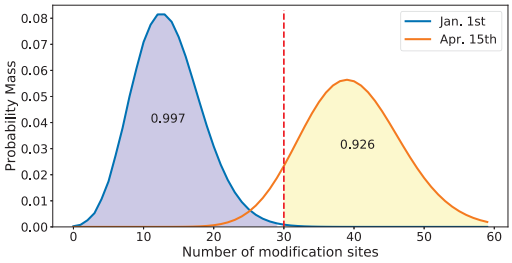
First, it was assumed that the prolonged cold winter contains sufficient information for constructing robust switch-behaviors of FLC. And for different climates, due to the differences in temperature dynamics, vernalization requires different memory spans of cold winter. The temperature dynamics was first dissected by decomposing them into seasonal changes and temperature fluctuations due to weather changes. It was found that temperature fluctuations of temperate climates such as Cologne and Auckland have exponentially decaying autocorrelations. The autocorrelation means that the fluctuation in one day is correlated with neighbouring days. Therefore, the plants should detect the seasonal changes and avoid the influences from the autocorrelations in temperature fluctuations. Based on the molecular basis of vernalization, a simple stochastic model was established for the histone modification reaction to incorporate the temperature dynamics. The model is mathematically described as a master equation and shown in reaction form in Fig. 3.1a. To model the temperature sensitive behavior of the reaction, the production rate $\beta T(t)$ was temperature-dependent and the degradation rate λ of modifications was temperature independent. The probability of having n histone modifications was only dependent on its neighbouring states. Due to the exponential decay of correlations in temperature fluctuations, the master equation was analytically solved for climates such Cologne. Solving the master equation yielded the probability distribution of having different modification states with the parameters β and λ. With such a setting, a flowering decision objective was optimized to obtain the optimal β and λ which specified the probability distribution which is shown in Fig 3.1b. Further, it was shown in Fig 3.1c that the switch behavior of FLC can be reconstructed. The simple stochastic model was only applicable to temperate climates as they have exponentially decaying correlations in temperature fluctuations.

Secondly, to relax the restriction on climate properties and system details, artificial neural networks were deployed to learn the idealized expression pattern of FLC from temperatures and day lengths of different climates. It was shown in Fig 3.1d that, using data from Cologne, the

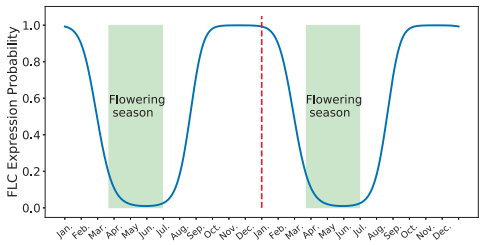
neural network models trained solely on temperature memory roughly reconstructed the idealized expression patterns of *FLC* but with a local optimum in September due to seasonal similarity between Spring and Autumn. This local minimum might lead to wrong flowering decisions. Interestingly, the addition of short-term day length signal eliminated the local optimum, which indicated that the combined signal led to more precise decisions (Fig 3.1e). However, for climates with less seasonal changes such as Kahului and San Francisco, the expression patterns could not be reconstructed from the temperature memory. These findings suggested that the plants in temperate region may use long-term cold temperatures to determined the season and short-term day lengths to trigger the flowering.



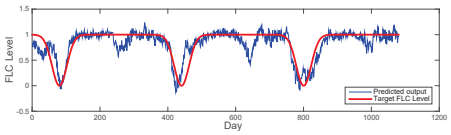
(a) Modeling the temperature dependent histone modifications.



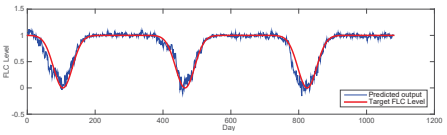
(b) The probability distributions of two time points in a year.



(c) Switch behavior of the core regulatory gene *FLC*.



(d) Predictions with only temperature



(e) Predictions with temperature and day length.

Fig. 3.1: Decision making in flowering time. (a) The histone modification is simplified as birth-death process. n stands for the number of modified histones, the production rate $\beta T(t)$ depends on the temperature $T(t)$ and λ is the degradation rate. Solving the model led to the distribution $p(n,t)$ which is parameterized by β and λ; (b) With optimal β and λ, the probability distribution $p(n,t)$ over number of modifications n (in total 60 sites) for two time points: 1st January and 15th April; (c) The *FLC* expression switching behavior was constructed from $p(n,t)$ of having less than 30 modification sites over two years, with green areas for potential flowering seasons. (d) *Cologne*, prediction result of the idealized *FLC* (in red) expressions using 42 days of temperature as the input features gave a local minimum due to similarity between spring and autumn; (e) *Cologne*, the local minima were eliminated by adding 2 days of day lengths as extra features.

3.2 Information integration and decision making in flowering time control

Publication status

Linlin Zhao, Sarah Richards, Franziska Turck, and Markus Kollmann, "Information integration and decision making in flowering time control", submitted to PLOS One.

Linlin Zhao's contributions

1. Built a stochastic model characterizing the simplified histone modification reaction.
2. Analyzed the temperature data to obtain the correlations in temperature fluctuations.
3. Solved the stochastic differential equation analytically under assumptions based on properties of temperature data.
4. Applied machine learning (neural networks) to reconstruct the idealized expression patterns of core regulatory genes from a wide range of climates.
5. Rewrote the code for evolutionary simulations in Matlab written by the coauthor Sarah Richards to Python, and carried out the simulations.
6. Wrote the manuscript, with help and discussions from other coauthors.

Information integration and decision making in flowering time control

Linlin Zhao[1*], Sarah Richards[2], Franziska Turck[3], Markus Kollmann[1*],

1 Institute of Mathematical Modeling for Biological Systems, Heinrich-Heine-University Dusseldorf, Dusseldorf, Germany
2 Institute of Population Genetics, Heinrich-Heine-University Dusseldorf, Dusseldorf, Germany
3 Max-Planck Institute for Plant Breeding Research, Cologne, Germany

* markus.kollman@hhu.de, linlin.zhao@hhu.de

Abstract

In order to successfully reproduce, plants must sense changes in their environment and flower at the correct time. Many plants utilize day length and vernalization, a mechanism for verifying that winter has occurred, to determine when to flower. Our study used available temperature and day length data from different climates to provide a general understanding how this information processing of environmental signals could have evolved in plants. For climates where temperature fluctuation correlations decayed exponentially, a simple stochastic model characterizing vernalization was able to reconstruct the switch-like behavior of the core flowering regulatory genes. For these and other climates, artificial neural networks were used to predict flowering gene expression patterns. For temperate plants, long-term cold temperature and short-term day length measurements were sufficient to produce robust flowering time decisions from the neural networks. Additionally, evolutionary simulations on neural networks confirmed that the combined signal of temperature and day length achieved the highest fitness relative to neural networks with access to only one of those inputs. We suggest that winter temperature memory is a well-adapted strategy for plants' detection of seasonal changes, and absolute day length is useful for the subsequent triggering of flowering.

Introduction

Plants must make correct flowering time decisions in a noisy environment in order to successfully reproduce. As key environmental signals, day length and temperature are processed by plants' genetic networks for detecting seasonal changes. The core genes and their interplays have been well understood in the model plant, *Arabidopsis thaliana* [1,2], as shown in Fig 1. The gene *FLOWERING LOCUS T (FT)* merges signals from both day length and temperature, and its encoded protein eventually induces the flowering [3]. The expression of FT genes is promoted by the expression of the gene *CONSTANS (CO)*, whose gene products are produced about 12 hours after dawn and quickly degrade in the dark [1,3]. Thus, the condition that day length is long enough to produce stable CO proteins is necessary for initiating the flowering of the so-called long-day plants [1,3,4]. In particular, for winter annuals of *Arabidopsis thaliana* [5,6], vernalization is required to cease expression of *FlOWERING LOCUS C (FLC)*, the inhibitor of *FT*. Vernalization involves the exposure of plants to a prolonged

period of cold temperature, which induces histone modifications on the epigenetic level for silencing *FLC* [7–9]. The repressed *FLC* allows *FT* to be expressed under long days. Similarly, in perennial *Arabis alpina*, the orthologs of *FLC*, *PERPETUAL FLOWERING 1 (PEP1)*, downregulates the orthologs of *FT*, *AaFT1* and *AaFT3*, and needs to be silenced by vernalization in order for the perennials to flower in the right time [6]. Moreover, for perennial *Arabidopsis halleri*, Nagano et.al [10] demonstrated the role of cooperation between the oscillations of temperature and day length in adaptation to seasonal changes. Thus, the integration of signals from temperature and day length is crucial for both annual and perennial plants to make flowering decisions. Despite the qualitative understanding of the genetic regulation of flowering time, it is unclear from an information point of view why plants have evolved vernalization from fluctuating winter temperatures and how it relates to day length in flowering decisions.

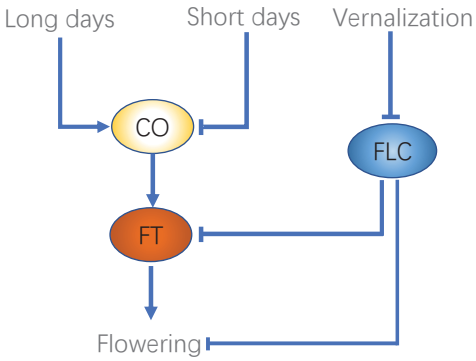


Fig 1. Flowering time regulation in *Arabidopsis thaliana*. In *Arabidopsis Thaliana*, long days promote the expression of FLOWERING LOCUS T (FT). The vernalization process also promotes its expression by turning-off its repressor FLOWERING LOCUS C (FLC).

Many theoretical studies have contributed to the understanding of the vernalization mechanism. It was shown to be an inheritable and stable epigenetic switch for the expression of *FLC* [11–15]. Dodd et.al [15] developed a stochastic model for *Schizosaccharomyces pombe* showing that gene expressions bistability can be established by accumulating histone modifications, which acted as an epigenetic memory. The work of Angel et al. [9] extended their approach by incorporating histone modifications of *FLC* in *Arabidopsis thaliana* and investigating how the different epigenetic states could be controlled. Several studies have reported that FLC repression was cell-autonomous and that cold temperature memory was encoded by the fraction of cells with repressed FLC [8, 9, 16, 17]. Due to the positive feedbacks that lead to adding more of the same type of histone modifications, a particular cell would mostly have one type of modifications at *FLC*. To understand how plants utilize fluctuating temperature in vernalization, Antoniou-Kourounioti et al. developed a model by incorporating thermosensing on multiple timescales and suggested that the sensing was broadly distributed in plants [18].

Investigations of flowering time regulation in *Arabidopsis* exploited mathematical modeling and experiments. Wilczeck et al. developed a model to incorporate the impacts of temperature, day length, and vernalization on flowering initialization in different accessions of *Arabidopsis thaliana* [5, 19]. Their analyses yielded a photothermal measure of plant states which was able to accurately predict flowering time of *Arabidopsis Thaliana* field plants. Another dynamic model described interplays

between *FT* and *FLC* in *Arabidopsis halleri* to study impacts of temperature on flowering decisions [20]. It was able to reproduce the observed seasonal expression changes and estimate the climate change-induced reduction of flowering season. Hepworth et al. reported that the spikes of temperatures above 15°C may have deleterious consequences for vernalization [21]. These studies have significantly refined our understanding of the effects of temperature and day length on *Arabidopsis* flowering, but the mechanisms employed by a variety of plants in various environments cannot be described by the same processes [23]. A method leading to a more general understanding of climate information processing would promote understanding of flowering-time decision making for a greater variety of plants.

Our study focuses on the extraction of available information from climate data (temperature and day length) and its usefulness in making precise flowering decisions. We first established a simple stochastic model for vernalization in perennials, which showed that the idealized expression patterns of *FT* or *FLC* can be reconstructed for temperate climates due to exponentially decaying correlation in temperature fluctuations. The stochastic model does not apply to other climates where temperature fluctuations correlate differently. To relax this restriction on climate properties, we employed artificial neural networks to learn idealized gene expression patterns from several climate datasets. We showed that, in temperate city Cologne, the neural network models trained solely on temperature memory roughly reconstructed the idealized expression patterns of *FLC*. However day length data was required to resolve the danger of incorrect flowering time decisions based on a local optimum in September rather than April. Further, to simulate the evolutionary adaptation to environmental conditions, individual neural networks were used in a simulation of evolution for plants with access to temperature, day length, or both. Simulations with different mutation rates and population sizes showed a persistent selective advantage for the neural networks with access to the combined temperature and day length data.

Materials and methods

Datasets

The temperature and day length data of several climate regions (Table 1) were retrieved from NOAA [24] and PTAFF [25]. Temperatures were recorded as the daily maximum and minimum, and data from different stations are considered distinct. The mean of daily maximum and minimum is regarded as the daily average temperature. To account for the effect of noise in daily light quantity [26–28] on the day length, Gaussian noise was used to corrupt the day length data and simulate real variations due to weather conditions.

Table 1. Selected regions and cities for collecting climate data.

Cold Regions	Obvious Seasonal Changes	Less Seasonal Changes
Oslo	Cologne	Kahului
	Auckland	San Francisco

Master Equation and Hermite polynomial

Chemical master equations are used to model the probabilistic states of chemical reactions over time [29–31]. Following the previous work of modeling birth-death process [32], for the reaction $\phi \underset{\lambda}{\overset{\beta}{\rightleftharpoons}} \mathrm{B}$, the probability p of having n molecules of B can

be described as Eq. (1) for time t.

$$\partial_t p(n,t) = \beta\big(p(n-1,t) - p(n,t)\big) - \lambda n p(n,t) + \lambda(n+1)p(n+1,t) \qquad (1)$$

In equilibrium, $p(n)$ is Poisson distributed (see Supplement Section 3.1). In our study, a modified form of this model was used with B representing the cellular state of having active modifications at the *FLC* locus. Different from the basic model, the production rate is adapted to be temperature dependent. Since the temperature recordings are time series, the temperature dependence of the model make it able to integrate temperature properties. This modification necessitates the use of Hermite polynomials [33, 34] to solve the master equation (see Supplement Section 3.1).

The original daily average temperature from different regions have been used for the analytical deductions based on the master equation.

Neural network models and data features

Artificial neural networks was used as a complementary method to extract information from temperature and day length. A feedforward neural network is comprised of a number of neurons to transmit information in only one direction, from the input data through hidden neurons to output neurons. Each neuron can be regarded as either a computing node or a decision maker which outputs a decision by weighing and transforming the information it receives from upstream neurons. A detailed formal description of neural networks is in Supplementary section 4.

Fully connected feed-forward neural networks with one hidden layer were used to classify different time windows in each year and regression of the idealized expression patterns of *FT* and *FLC* in Arabidopsis perennials.

For the classification, each year was shrank to 360 days for simplicity. For example, it will be divided into 12 windows corresponding to 12 months if the window size is set to 30 days. The neural networks then need to determine which month the input window belongs to, given 30 days of temperatures within the window. The daily temperatures were summarized by daily maximum and minimum.

For the regression, without loss of generality, *FT*'s idealized pattern is characterized by a normal distribution $p(t)$, peaked in April every year as shown in Fig 2, where t denotes certain day of a year. Whereas the *FLC* is featured as an upside-down normal distribution centered at March 15th [35], which is 15 days earlier than the peak of *FT* [20]. To learn the expression patterns from the climate data, the input features consist of daily temperature maxima and minima and day lengths of the past days. That is, the expression level on a specific day is determined by the plant's memory of temperature and day length of the past days.

The Neural Networks Toolbox in Matlab [36] was used to build the classification and regression models.

Evolution of individual neural networks

Plant selection pressure and evolution were simulated using neural networks to represent individual plants, with weights of individual networks analogous to plant genotypes (see details in Supplement Section 5). A group (population) of networks trained on distinct subsets of the same climate data were used to evaluate the flowering decision-making strategies of plants based on climate information. There were three populations in each simulation, each with access to only temperature, only day length, or both. Each generation of individuals was trained on randomly selected subsets of climate data from Cologne. At the end of each reproductive cycle, fitness was measured by the Kullbach-Leibler distance between target and neural network-predicted gene expression

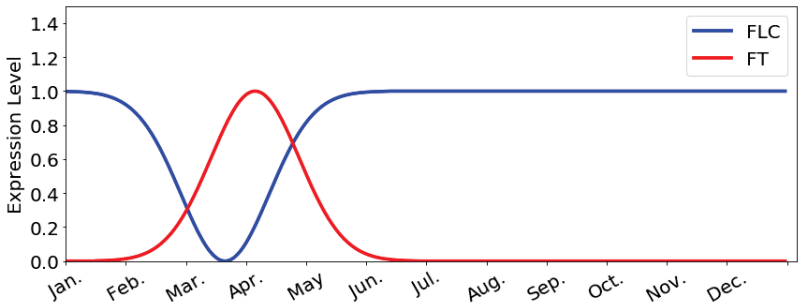


Fig 2. Idealized gene expressions. The idealized expression level of FT is characterized by a normal distribution peaked at April 1st, and that of FLC is a flipped normal distribution centered at March 15th.

level, and mutations (in the form of neural network weight perturbations) were introduced at specific rates which varied across simulations for the next generation. The simulation procedure is summarized in Fig. S8. It was implemented in Python 2.7, and the Theano package [37] was used to optimize individual networks. The results of each simulation were reported as the proportion of individuals from each of the three populations throughout the entire simulation, which would be greater for neural networks with higher fitness leading to better reproductive odds.

Results

Expression patterns reproduced from modeling key reactions

Plants need to avoid the disastrous effect of flowering at the wrong time as a reaction to sudden and sustained temperature fluctuations. We hypothesize that long-cold based vernalization is not only for capturing the winter cold temperature but also for canceling this effect and sensing the winter robustly. The vernalization machinery can be interpreted that plants have evolved biochemical processes to capture and accumulate the information in cold temperature that is a reliable signal in temperate regions. Inspired by the machinery, we modeled the which is driven by real temperature to reconstruct the idealized expression patterns of FLC/FT of *Arabidopsis* perennials.

In the following, we denoted daily average temperature as $T(t)$ for day $t \in [1, \cdots, 365]$. To investigate the seasonal changes and fluctuations in real temperature, it is decomposed into three parts as $T(t) = \bar{T} + \langle T(t) \rangle + \delta T(t)$ with $\bar{T}$ denoting the average yearly temperature, $\langle T(t) \rangle$ the seasonal temperature changes, and $\delta T(t)$ the remaining temperature fluctuations. The temperature dynamics comprising of $\langle T(t) \rangle$ and $\delta T(t)$ are shown in Fig 3 for Cologne. The Fourier fitting of $\langle T(t)$ was detailed Supplementary section 2. The temperature fluctuation on a specific day typically correlates with its neighboring days. And the longer periods of unseasonally cold or warm temperatures may confuse the plants more than the shorter periods. Therefore, we analyzed the autocorrelation times, which quantify the correlation length in a time series, in the fluctuations of temperature from five different regions (Table 1). The autocorrelation times in Cologne, Auckland and Oslo decay exponentially, and decay faster in Auckland than in Cologne and Oslo (Supplementary Section 2, Fig S3a, S3b). This is consistent with experimental observations that some plants in Auckland required only two weeks of vernalization [38], while vernalization requires typically around 6 weeks in Cologne [5] and even three months in North Sweden [39]. Having

verified the exponential decay of temperature fluctuations, without loss of generality, we used the climate data of Cologne for the successive modeling, where the autocorrelation in averaged yearly-cycle temperature fluctuations decays exponentially with an approximate half life of 4 days (Fig 4b) To further evaluated the exponential fitting of the fluctuation decay, the bootstraping of temperature data with block length of 50 showed that the fitted coefficients of the exponential function indeed located in the bootstrapped confidence interval(further details in supplementary Section 2).

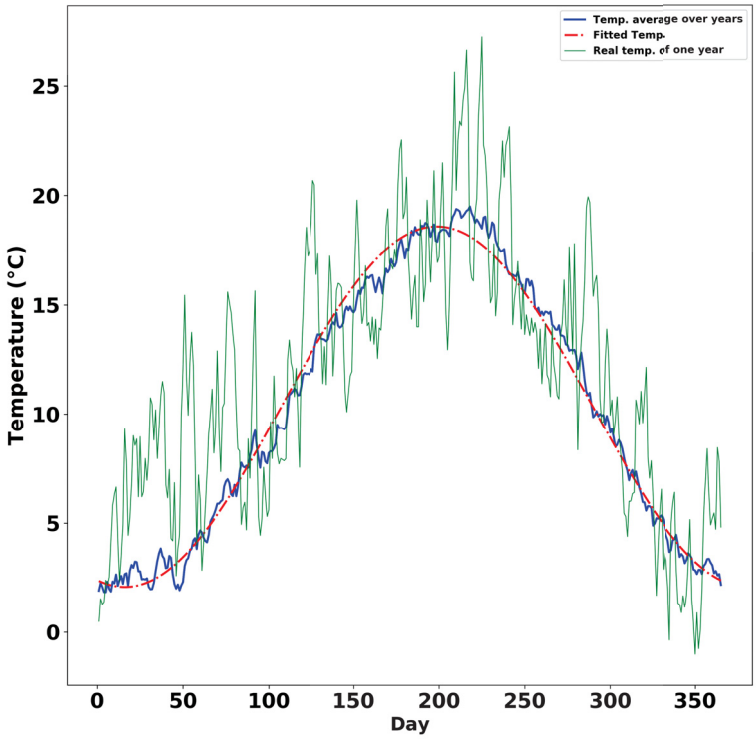


Fig 3. Temperature dynamics in Cologne. The temperature dynamics consist of the seasonal changes and the daily temperature fluctuations, which were fitted by a second order Fourier series. The dynamical data were obtained by averaging 93 years of temperatures in temperate city Cologne. The first day in the plot was January 1st.

To investigate the effect of temperature dynamics on vernalization, we modeled the number of cells which have their histones all with active modifications as a death-birth process. The assumption of having all active modifications in a cell relied on the fact that the fraction of histones in one state affects modifications in their own vicinity, which makes histone modification a relative fast process. As shown in Fig. 4B, the probability of having n active cells at time t is denoted as $p(n,t)$:

$$\partial_t p(n,t) = \beta T(t)\big(p(n-1,t) - p(n,t)\big) - \lambda n p(n,t) + \lambda(n+1)p(n+1,t), \qquad (2)$$

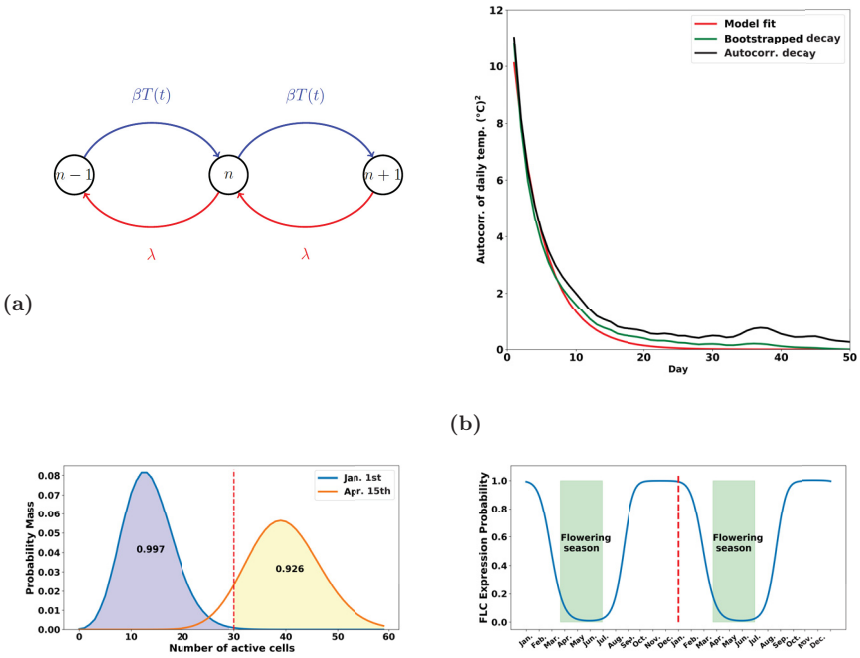


Fig 4. Reconstruct the switch behavior of *FLC* (a): The vernalization was simplified as birth-death process for actively modified cells. n stands for the number of modified histones, the production rate $\beta T(t)$ depends on the temperature $T(t)$ and λ is the degradation rate. Solving the model led to the distribution $p(n,t)$ which is parameterized by β and λ; (b): For the data from Cologne, the autocorrelation in daily temperature decays exponentially, the bootstraping was performed by using block length of 50 days; (c): The probability $p(n,t)$ distribution over number of active modifications (in total 60 sites) on 1st January and 15th April; (d): The switching *FLC* expression behavior was constructed from the probabilities of having less than 30 modification sites over two years, with green areas for potential flowering seasons.

with $\beta T(t)$ the temperature dependent production rate and λ the temperature-independent degradation rate. Due to the fact that the autocorrelation time of temperature fluctuations decays exponentially, the stochastic differential equation has a concise form of solution given by

$$p(n,t) = e^{a-b}\frac{\alpha^n}{n!}H_n\Big(\frac{b+2\alpha^2}{2\alpha}\Big). \tag{3}$$

with $a := \frac{\beta^2\sigma^2}{2\lambda(\lambda+\tau^{-1})}$ and $b = \frac{\bar{T}\beta}{\lambda} + \beta D(\lambda,t)$. Here, we denoted by $D(\lambda,t) = \int_{-\infty}^{t}\langle T(t)\rangle e^{-\lambda(t-t')}dt'$ the expected memorized temperature. The parameters σ^2 and τ denote the averaged variance and autocorrelation time of temperature fluctuations respectively. $H_m(\cdot)$ denotes the mth Hermite polynomial. The parameterization for a, b, and α is detailed in the supplementary.

To reconstruct the expression patterns, we need to define an objective for getting the optimal reaction rates based on the probability $p(n,t)$. We assumed that the unit of the

number of cells n was scalable for the simplicity of computation. Due to the limited cellular resources, it was further assumed that $N_{\max}$ cells were available and N_c cells was sufficient to turn off the expression of *FLC* on the plant level. The objective function of flowering probability in time window between March and June and non-flowering probability between July and next February can be defined as

$$F(\beta,\lambda) = \int_{Mar}^{Jun} \sum_{n=N_c}^{N_{max}} p(n,t)dt + \int_{Jul}^{Feb} \sum_{n=0}^{N_c-1} p(n,t)dt \tag{4}$$

Maximizing this objective is equivalent to maximizing the probability of flowering (i.e. having at least N_c active cells) in flowering season and non-flowering (i.e. having at most N_c active cells) during non-flowering season. The optimal reaction rates which maximized the objective led to time dependent optimal probability sequences which were capable of reproducing the idealized expression pattern of *FLC* that is switched off during flowering season. In Fig. 4C, two different time points are chosen to show the typical probability distributions in flowering and non-flowering seasons. On 1st January, the most density (99.2%) of the probability located below the critical value of 30 unit of cells, whereas on 15th April most density (92.2%) was distributed above the critical value. As shown in Fig 4D that the probability of having at least N_c unit of active cells at different time of a yearly cycle preserved the idealized expression pattern of *FLC*, which was active during the non-flowering season and then gradually switched off in flowering season.

Under the condition that the autocorrelation length of temperature fluctuations in time decays exponentially, it was shown that the idealized expression patterns of *FLC* could be rebuilt from the stochastic model. By relaxing the autocorrelation condition, we would also like to investigate the effect of climate information on flowering time decision using machine learning, which is typically not requiring great details of the system thus can be more broadly applied to different climates.

Long-term cold temperature and short-term day lengths together as a robust signal

To make flowering time decisions upon environmental cues such as temperatures and day lengths, plants are essentially information processing units for extracting critical environmental signals in order to survive by making correct reproduction transit. We employed artificial neural networks to approximate the information processing in plants by predicting the flowering season. The networks were trained to learn the idealized expression pattern of *FLC* from temperatures and day length from different climates, and the results relied on climate data in Cologne. The approach was broken into two tasks: to determine the effective memory length of determining season and to reconstruct the idealized expression pattern of *FLC* of *Arabidopsis* perennials.

The first step was to determine the number of days in the past that the plants need to remember in order to recognize the current season. It was cast into a classification problem as for given consecutive L days of temperature, the neural networks learned to classify which time window the temperature belonged to. The results showed that for Cologne, the prediction achieved MCC score of 0.964 for temperature memory of over 40 days. And increasing the memory length did not increase the score accordingly. The result was consistent with experimental result [5]. This reflected an expected tradeoff between sufficiently long memory to reduce the fluctuations of temperature signal (variance) and the loss of season specific averaged temperature if the memory stretches beyond the length of the season (bias). The detailed classification result can be found in Supplementary.

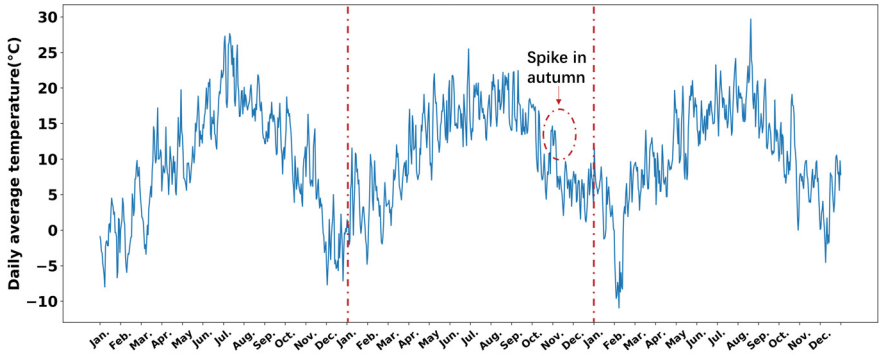


Fig 5. Three years of test temperatures Three years of temperatures were used to test the regression models. In the second year, one can observe a long temperature spike starts from late October to early November. This is in correspondence with a predicted local minimum in Fig 6a.

Having determined the effective memory length, we were able to construct the input features of the neural networks for fitting the idealized expression pattern of *FLC* of *Arabidopsis* perennials. With only the temperature memory fed into the neural networks, the trained neural network was tested on three years of temperature as shown in Fig 5. Local minima arose in autumn as shown in Fig 6A, which resulted from the temperature similarity between spring and autumn. Especially for the second test year, a clear local minimum was observed in October, which has a correspondence to the high temperature spike in October as indicated in Fig 5. The local minima could potentially give a high chance of making wrong flowering decision in autumn. In order to remove the local minima and have a robust detection of spring, two days of day-length signal were added to the input features. It can be seen from Fig 6B that the local minima are eliminated, leading to more precise regression of the *FLC* expression pattern. The precise and robust reconstruction of the *FLC* signal was critical for precise flowering decisions in the right season. As a comparison, the climate data from Kahului, Hawaii was used as well to learn the *FLC* pattern. Fig 6C showed that the temperature memory only features led to a high-error fitting, which probably due to both the flat seasonal changes and the long autocorrelation in temperature fluctuations as shown in Supplementary Fig S4e. The integration of two days of day lengths increased the regression but still being noisy. The result may provide an explanation that vernalization in natural tropical climates was not established.

The neural networks based method was broadly applicable to different climates (the results for other climates were shown in supplementary), unlike the stochastic model which required more system details and climate properties. Its fitting results from the climates data reflected that for temperate regions, long-term temperature and short-term day length together were deployed as a robust signal for determining the flowering transition, while for other regions, merely temperature and day length signals were not sufficient to have a mechanism such as the interplay between *FLC* and *FT*.

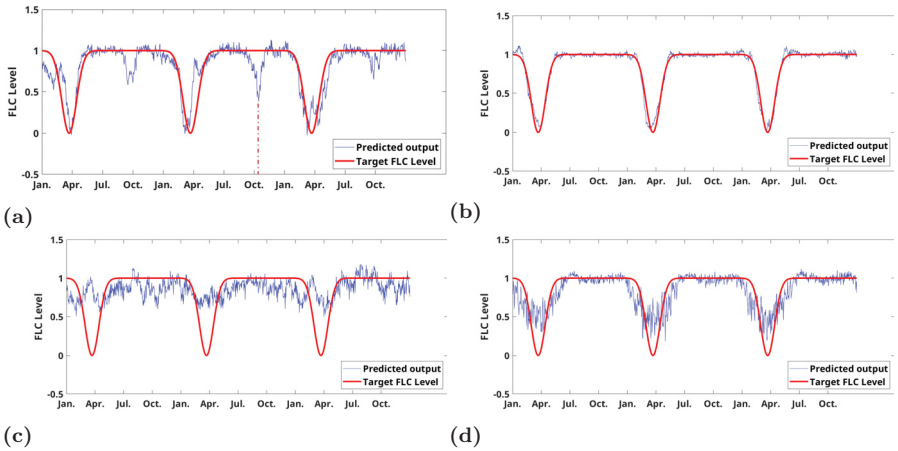


Fig 6. Predicted expression patterns by neural networks (a): *Cologne* , fitting result of the idealized *FLC* expressions using 42 days of temperature as the input features gave a local minimum in September due to similarity between spring and autumn; (b): *Cologne*, fitting result from 42 days of temperature and 2 days of day lengths with eliminated local minima; (c): *Kahului*, fitting result from 42 days of temperatures; (d): *Kahului*, fitting result from 42 days of temperature and 2 days of day lengths.

Evolutionary simulation favors integration of temperature and day length

To investigate the possible effects of evolution on the role of temperature and day length in achieving the idealized expression pattern of FT, we conducted simulations using neural networks as plants' agents. In each simulation, three groups of virtual plants were given access to either temperature, day length, or both. The Kullbach-Leibler Divergence (KLD) between the fitted and idealized expression patterns was used as the fitness measure. Fitnesses were calculated for individuals from each group and normalized to the population of each generation to have a fitness probability for each individual. Individuals then reproduced according to their fitness probability, that is, individuals with higher fitnesses had more offspring accordingly. Multiple offspring were possible for each individual, and offspring had access to the same input type as their parents (temperature, day length, or both). For each reproduced generation, a fixed number of mutations, represented by randomly selected neural network weights, were applied to each individual. Reproduction continued until 500 cycles were simulated or until all surviving individuals were the offspring of only one of the three groups. Each simulation was repeated 50 times, and the proportion of offspring per group throughout the entire simulation were tallied to estimate fixation probability.

To simulate the effects of strongly or weakly deleterious mutations, simulations were run with various numbers of mutations per generation. To investigate the effects of population size and genetic drift, the number of individuals per group was varied. The results of the simulations are shown in Fig.7, as violin plots of group offspring proportions under different mutation rates and population sizes.

The group with access to both temperature and day length generally had the most offspring, and the group with access to day length alone had the least. The temperature-only group performed somewhere between the two others. Consistent with the basic principles of genetic drift, larger population sizes resulted in less variance in

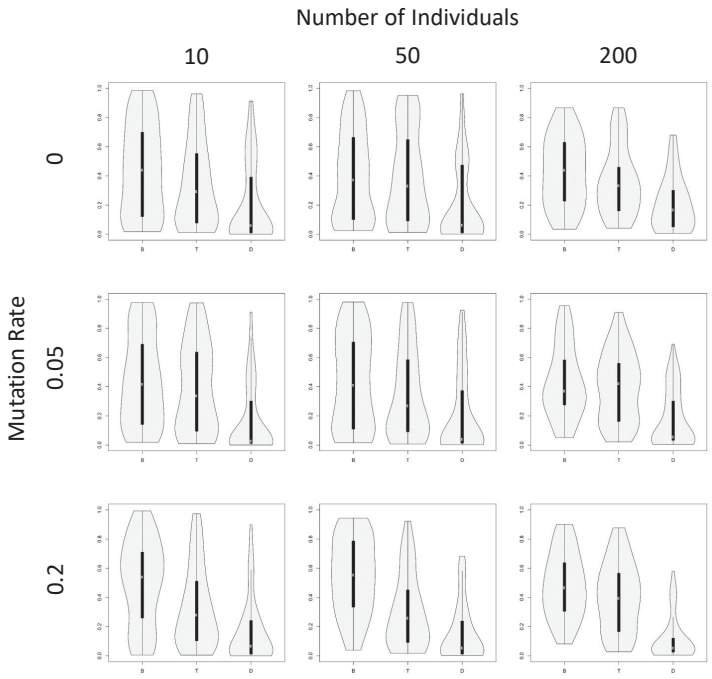


Fig 7. Evolution simulation. Distributions of group offspring proportions for mutation rates of 0, 0.05, and 0.2, for group sizes of 10, 50, and 200 individuals. "B" stands for group with access to both temperature and day length, "T" for temperature and "D" for day length.

the fixation probability. Possibly due to the small number of input variables in the neural network, the group with access to day length alone performed notably worse as the mutation rate increased.

Conclusion

Our study verified that the temperature and day length in temperate regions like Cologne have the information premise for plants such as *Arabidopsis* to establish the epigenetic switch vernalization. For different climates, the vernalization requires different memory spans due to different temperature seasonal properties. The memory spans depended on the autocorrelations in temperature fluctuations, which decayed differently from climate to climate. For regions with flat temperature dynamics like San Fransisco and Hawaii, the autocorrelations decay slower than regions like Cologne, and have autocorrelation length above 100 days, which might be the reason to fail the establishment of vernalization. Our stochastic model, describing the dynamics of the number of cell with repressed *FLC*, was able to integrate the temperature dynamics as well as the temperature fluctuations. For regions with exponential decays in temperature fluctuation autocorrelations, the switch behavior of *FLC* in *Arabidopsis* perennials can be reconstructed. Further, without requirements on climate properties

and system details, our machine learning approach showed that the idealized expression patterns of *FLC* can be robustly reconstructed by the combination of prolonged cold and short term of day length. The strategy of combining long-term cold and short-term day length is proven to be also favored by an evolution simulation where neural networks were regarded as the agents of plants for processing climate information.

Although in natural environments, temperate plants need to cope with other signals such as ambient temperature using additional genes like *FLM* [40], it might indicate the backbone of flowering mechanism is that the plants utilized long-term temperature to detect seasonal changes and used absolute day lengths to decide the eventual flowering days. The reason is that, for temperate climates, a cold winter is guaranteed and it is easier to track the absolute day length than the variations in day lengths which is 4min in maximum from day to day. It could also be a good strategy for cold regions like Norway since the autocorrelation decays similarly to Cologne, but due to lower temperature and shorter summer, the fast life cycles such as summer annuals of *Arabidopsis* was adapted [5]. In this case, ambient temperature and light intensity might play a more important role in plants' vegetative or reproductive timing. In tropical regions where locate the most diverse and abundant plant species on earth, more factors have to be taken into consideration and merely temperature and day length are not sufficient for flowering decision making. For instance, flowering is mostly rain-season dependent, which might play a more critical role than the temperature and day length as they contain less seasonal information than that of temperate regions.

By investigating flowering decision making from an information point of view, our study suggested that, for temperate regions, cold winter memory and short term of day length can serve as a robust strategy for plants to determine flowering season.

Supporting information

Supplementary.pdf Details of data preprocessing, proofs of analytical solution to stochastic model, regression of the idealized *FLC* expressions based on different climates.

code.zip It includes the code *autocorrelation.py* for analyzing temperature data, *optimization.py* for optimizing the flowering objective, *evolution.py* for the evolutionary simulations and *neuralnets.m* for cleansing temperature data and training neural networks.

data.zip It includes csv files for the temperature and day length data of Cologne, Norway, Auckland, Kahului and San Francisco.

Acknowledgments

The authors would like to thank DFG (Deutsche Forschungsgemeinschaft) for financial support (KO3442-9).

References

1. Andrés F, Coupland G. The genetic basis of flowering responses to seasonal cues. Nature reviews Genetics. 2012;13(9):627–39. doi:10.1038/nrg3291.

2. Song YH, Ito S, Imaizumi T. Flowering time regulation: photoperiod- and temperature-sensing in leaves. Trends in plant science. 2013; p. 1–9. doi:10.1016/j.tplants.2013.05.003.

3. Bratzel F, Turck F. Molecular memories in the regulation of seasonal flowering: From competence to cessation; 2015.

4. Turck F, Fornara F, Coupland G. Regulation and Identity of Florigen: FLOWERING LOCUS T Moves Center Stage. Annual Review of Plant Biology. 2008;59(1):573–594. doi:10.1146/annurev.arplant.59.032607.092755.

5. Wilczek AM, Roe JL, Knapp MC, Cooper MD, Lopez-Gallego C, Martin LJ, et al. Effects of genetic perturbation on seasonal life history plasticity. Science (New York, NY). 2009;323(5916):930–4. doi:10.1126/science.1165826.

6. Hyun Y, Vincent C, Tilmes V, Bergonzi S, Kiefer C, Richter R, et al. A regulatory circuit conferring varied flowering response to cold in annual and perennial plants. Science. 2019;363(6425):409–412. doi:10.1126/science.aau8197.

7. Csorba T, Questa JI, Sun Q, Dean C. Antisense COOLAIR mediates the coordinated switching of chromatin states at FLC during vernalization. Proceedings of the National Academy of Sciences. 2014;111(45):16160–16165. doi:10.1073/pnas.1419030111.

8. Angel A, Song J, Yang H, Questa JI, Dean C, Howard M. Vernalizing cold is registered digitally at FLC. Proceedings of the National Academy of Sciences. 2015;112(13):4146–4151. doi:10.1073/pnas.1503100112.

9. Angel A, Song J, Dean C, Howard M. A Polycomb-based switch underlying quantitative epigenetic memory. Nature. 2011;476(7358):105–8. doi:10.1038/nature10241.

10. Nagano AJ, Kawagoe T, Sugisaka J, Honjo MN, Iwayama K, Kudoh H. Annual transcriptome dynamics in natural environments reveals plant seasonal adaptation. Nature Plants. 2019;5(1):74–83. doi:10.1038/s41477-018-0338-z.

11. Mukhopadhyay S, Nagaraj VH, Sengupta AM. Locus dependence in epigenetic chromatin silencing. BioSystems. 2010;102(1):49–54. doi:10.1016/j.biosystems.2010.07.012.

12. Kelemen JZ, Ratna P, Scherrer S, Becskei A. Spatial epigenetic control of mono- and bistable gene expression. PLoS Biology. 2010;8(3). doi:10.1371/journal.pbio.1000332.

13. David-Rus D, Mukhopadhyay S, Lebowitz JL, Sengupta AM. Inheritance of epigenetic chromatin silencing. Journal of Theoretical Biology. 2009;258(1):112–120. doi:10.1016/j.jtbi.2008.12.021.

14. Sedighi M, Sengupta AM. Epigenetic chromatin silencing: Bistability and front propagation. Physical Biology. 2007;4(4):246–255. doi:10.1088/1478-3975/4/4/002.

15. Dodd IB, Micheelsen Ma, Sneppen K, Thon G. Theoretical analysis of epigenetic cell memory by nucleosome modification. Cell. 2007;129(4):813–22. doi:10.1016/j.cell.2007.02.053.

16. Song J, Angel A, Howard M, Dean C. Vernalization - a cold-induced epigenetic switch. Journal of cell science. 2012;125(Pt 16):3723–31. doi:10.1242/jcs.084764.

17. Satake A, Iwasa Y. A stochastic model of chromatin modification: cell population coding of winter memory in plants. Journal of theoretical biology. 2012;302:6–17. doi:10.1016/j.jtbi.2012.02.009.

18. Antoniou-Kourounioti RL, Hepworth J, Heckmann A, Duncan S, Qüesta J, Rosa S, et al. Temperature Sensing Is Distributed throughout the Regulatory Network that Controls FLC Epigenetic Silencing in Vernalization. Cell Systems. 2018;7(6):643–655.e9. doi:10.1016/j.cels.2018.10.011.

19. Chew YH, Wilczek AM, Williams M, Welch SM, Schmitt J, Halliday KJ. An augmented Arabidopsis phenology model reveals seasonal temperature control of flowering time. The New phytologist. 2012;194(3):654–65. doi:10.1111/j.1469-8137.2012.04069.x.

20. Satake A, Kawagoe T, Saburi Y, Chiba Y, Sakurai G, Kudoh H. Forecasting flowering phenology under climate warming by modelling the regulatory dynamics of flowering-time genes. Nature communications. 2013;4:2303. doi:10.1038/ncomms3303.

21. Hepworth J, Antoniou-Kourounioti RL, Bloomer RH, Selga C, Berggren K, Cox D, et al. Absence of warmth permits epigenetic memory of winter in Arabidopsis. Nature communications. 2018;9(1):639.

22. Topham AT, Taylor RE, Yan D, Nambara E, Johnston IG, Bassel GW. Temperature variability is integrated by a spatially embedded decision-making center to break dormancy in Arabidopsis seeds. Proceedings of the National Academy of Sciences. 2017;114(25):6629–6634. doi:10.1073/pnas.1704745114.

23. Weigel D. Natural Variation in Arabidopsis: From Molecular Genetics to Ecological Genomics. Plant Physiology. 2012;158(1):2–22. doi:10.1104/pp.111.189845.

24. NOAA. Climate data online;. Available from: `https://www.ncdc.noaa.gov/cdo-web/`.

25. ptaff. Sunrise, sunset, daylight in a graph;. Available from: `https://ptaff.ca/soleil/?lang=en_CA`.

26. Whiting D, Roll M, Vickerman L. Plant Growth Factors: Light; 2016.

27. WILSON D, COOPER JP. Effect of light intensity during growth on leaf anatomy and subsequent light-saturated photosynthesis among contrasting Lolium genotypes. New Phytologist. 1969;68(4):1115–1123. doi:10.1111/j.1469-8137.1969.tb06511.x.

28. Powles SB. Photoinhibition of Photosynthesis Induced by Visible Light. Annual Review of Plant Physiology. 1984;35(1):15–44. doi:10.1146/annurev.pp.35.060184.000311.

29. Gillespie DT. A rigorous derivation of the chemical master equation. Physica A: Statistical Mechanics and its Applications. 1992;188(1):404–425. doi:https://doi.org/10.1016/0378-4371(92)90283-V.

30. Risken H. Fokker-Planck Equation. Berlin, Heidelberg: Springer Berlin Heidelberg; 1996. Available from: `https://doi.org/10.1007/978-3-642-61544-3{_}4`.

31. Gardiner CW. Handbook of stochastic methods: for physics, chemistry and the natural sciences. Springer; 2004.

32. Walczak AM, Mugler A, Wiggins CH. Analytic methods for modeling stochastic regulatory networks. In: Computational Modeling of Signaling Networks. Springer; 2012. p. 273–322.

33. Weisstein EW. Hermite Polynomial. From MathWorld—A Wolfram Web Resource; Available from: `http://mathworld.wolfram.com/HermitePolynomial.html`.

34. doublle (https://mathoverflow net/users/57344/doublle). An interesting calculation of derivative;. MathOverflow. Available from: `https://mathoverflow.net/q/179031`.

35. Aikawa S, Kobayashi MJ, Satake A, Shimizu KK, Kudoh H. Robust control of the seasonal expression of the Arabidopsis FLC gene in a fluctuating environment. Proceedings of the National Academy of Sciences of the United States of America. 2010;107(25):11632–7. doi:10.1073/pnas.0914293107.

36. MATLAB. R2014a. Natick, Massachusetts: The MathWorks Inc.; 2014.

37. Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, et al. Theano: a CPU and GPU Math Expression Compiler; 2010.

38. Adhikari KN, Buirchell BJ, Sweetingham MW. Length of vernalization period affects flowering time in three lupin species. Plant breeding. 2012;131(5):631–636.

39. Duncan S, Questa J, Irwin J, Dean C, Holm S, Grant A. Seasonal shift in timing of vernalization as an adaptation to extreme winter. eLife. 2015;4(JULY2015):1–11. doi:10.7554/eLife.06620.

40. Posé D, Verhage L, Ott F, Yant L, Mathieu J, Angenent GC, et al. Temperature-dependent regulation of flowering by antagonistic FLM variants. Nature. 2013;doi:10.1038/nature12633.

Information integration and decision making in flowering time control-Figures

Linlin Zhao[1*], Sarah Richards[2], Franziska Turck[3], Markus Kollmann[1*],

1 Institute of Mathematical Modeling for Biological Systems, Heinrich-Heine-University Dusseldorf, Dusseldorf, Germany
2 Institute of Population Genetics, Heinrich-Heine-University Dusseldorf, Dusseldorf, Germany
3 Max-Planck Institute for Plant Breeding Research, Cologne, Germany

* markus.kollman@hhu.de, linlin.zhao@hhu.de

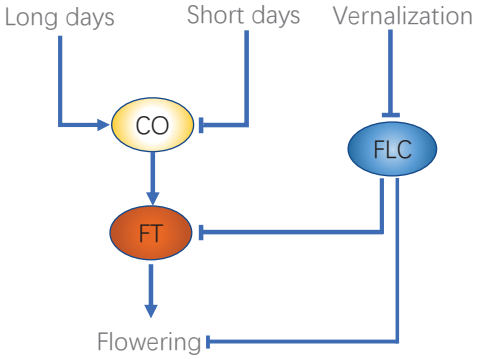


Fig 1. Flowering time regulation in *Arabidopsis thaliana*.

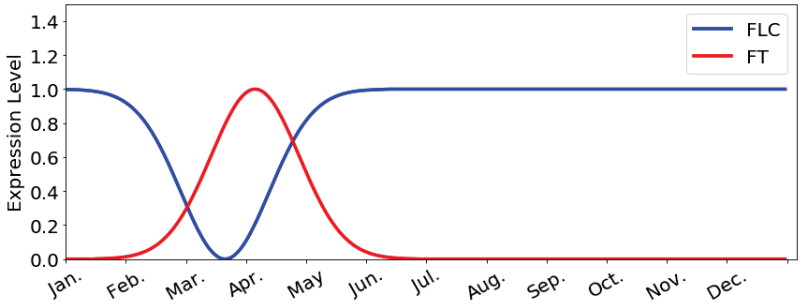


Fig 2. Idealized gene expressions.

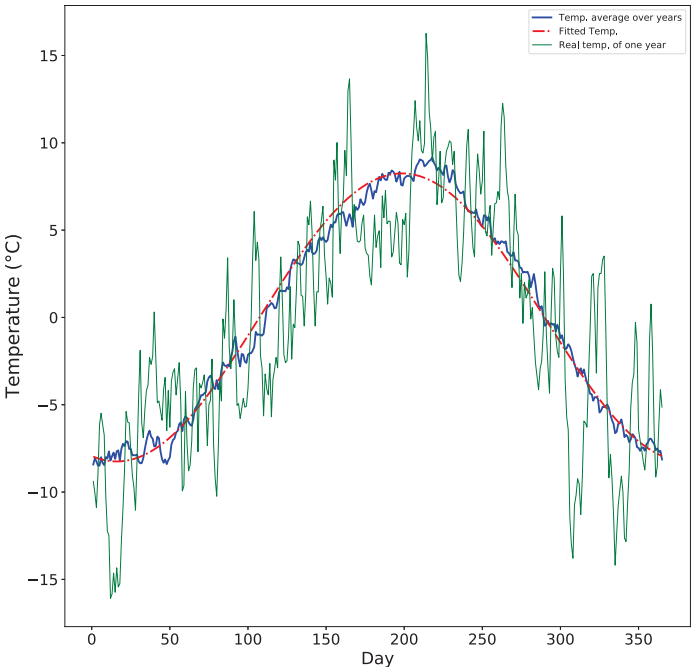


Fig 3. Temperature dynamics in Cologne.

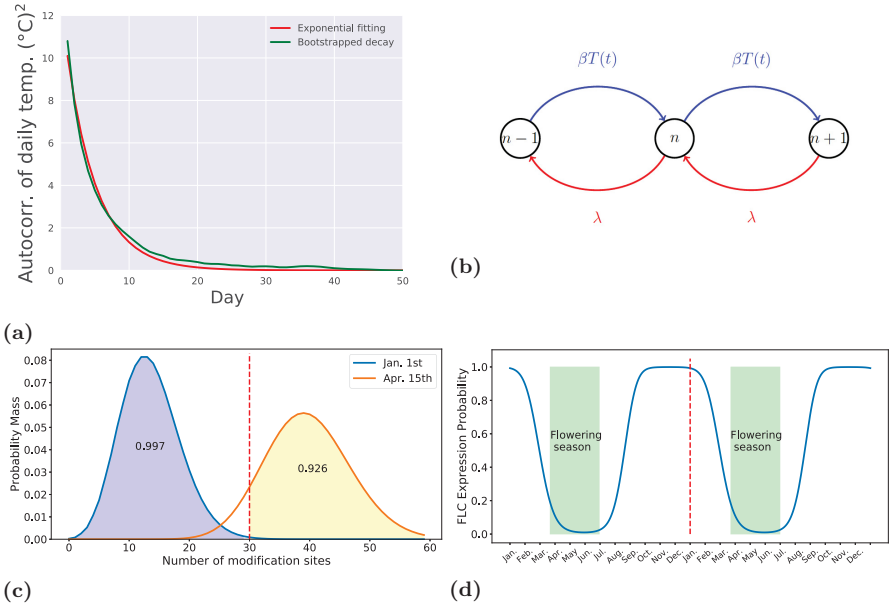


Fig 4. Reconstruct the switch behavior of *FLC*

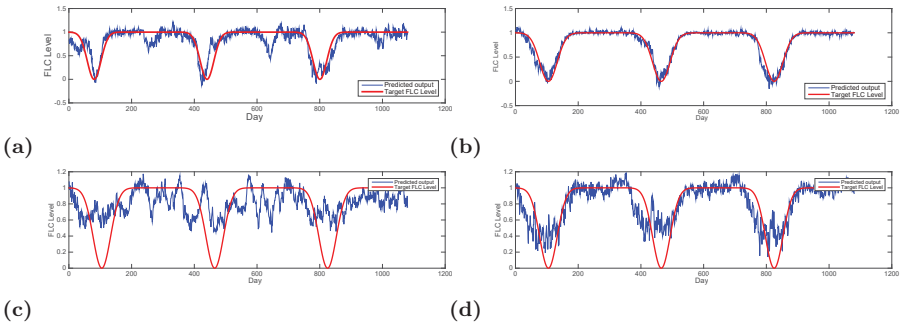


Fig 5. Predicted expression patterns by neural networks

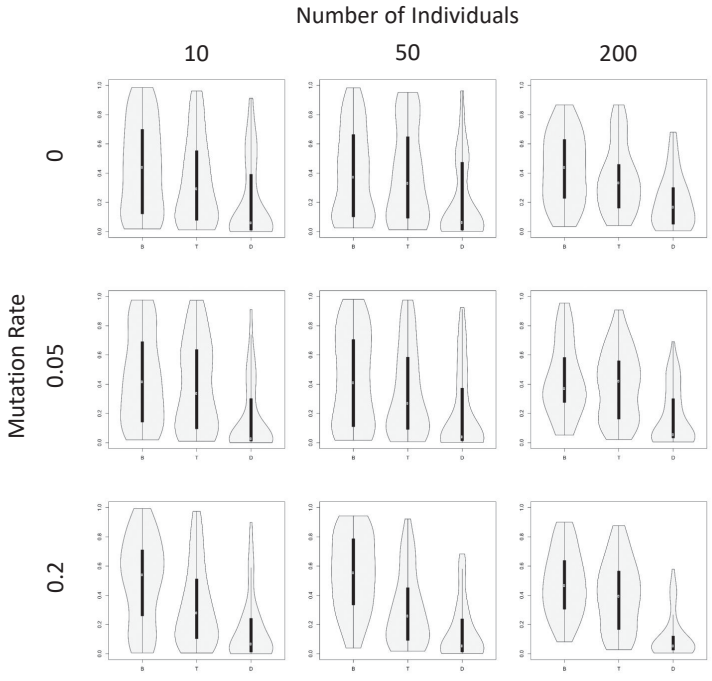


Fig 6. Evolution simulation.

Information integration and decision making in flowering time control-Supplementary

Linlin Zhao[1,*], Sarah Richards[2], Franziska Turck[3], and Markus Kollmann[1,*]

[1]Institute of Mathematical Modeling for Biological Systems, Heinrich-Heine-University Dusseldorf, Dusseldorf, Germany
[2]Institute of Population Genetics, Heinrich-Heine-University Dusseldorf, Dusseldorf, Germany
[3]Max-Planck Institute for Plant Breeding Research, Cologne, Germany
[*]markus.kollmann@hhu.de, zhao@hhu.de

1 Data preprocessing

1.1 Source data cleaning

The data preparation is illustrated by using data from Cologne. The retrieved temperature data from NOAA includes recordings of daily maximal and minimal temperatures of 2 stations in Cologne. A snapshot of the customized CSV data file is shown in Fig. S1. Each station independently recorded daily temperatures. Thus, for Cologne, we have 98 years of daily temperature summaries (Table S1).

Noises of different sources are presented in the temperature data, for instance, missing dates or missing records in a day. We refer to the missing values as "holes" of the data. The dataset was cleansed as follows:

- For the simplicity of computation, each year was trimmed to 360 days and each month was modified to 30 days for the neural networks based models. For the stochastic model, yearly cycle of 365 days was used;
- The holes were filled by averaging neighbouring two days or just copying the neighbouring date if the hole was less than 10 days. For holes larger than 10 days, they were filled with the same dates from a nearby station;

	1	2	3	4
1	Station	Date	Daily max (x10)	Daily min (x10)
2	GHCND:GME00121042	19580101	88	64
3	GHCND:GME00121042	19580102	76	-36
4	GHCND:GME00121042	19580103	5	-76
5	GHCND:GME00121042	19580104	-2	-65
6	GHCND:GME00121042	19580105	84	-14
7	GHCND:GME00121042	19580106	120	-3
8	GHCND:GME00121042	19580107	115	16
9	GHCND:GME00121042	19580108	50	12
10	GHCND:GME00121042	19580109	83	42
11	GHCND:GME00121042	19580110	68	38

Figure S1: A snapshot of the customized CSV file retrieved from NOAA with 4 columns.

Data from the stations which have more than 20% of holes were discarded.

The day length data of Cologne were downloaded from `ptaff.ca`. The orbit of the earth around the sun is stable such that the day length variation remains the same from year to year. Thus we just replicated the yearly day length for Cologne to 98 years. Meanwhile, to count influence of weather conditions on day lengths, Gaussian noises with adjustable strength were added to different years.

The same cleaning procedure was applied to other climates. The datasets of all retrieved climates are summarized in Table S1.

Table S1: Number of Years temperature for different regions

Regions	Cologne	Oslo	Kahului	Auckland	San Francisco
Number of Years	98	57	237	421	154

2 Time series of temperature data

2.1 Decomposing and analyzing temperature data

The temperature data from Cologne were analyzed. The consecutive daily recordings of temperatures are time series data where neighbouring days of temperatures are correlated. For instance, a storm brings rains as well as temperature drops. The dropped temperatures are usually correlated for a few days. The changes in temperatures due to weather conditions are termed temperature fluctuations and the seasonal temperature changes are termed deterministic temperature dynamics. The temperature fluctuation on day t can be mathematically described as

$$\delta T(t) = T(t) - (\bar{T} + \langle T(t) \rangle)$$

where $T(t)$ denotes the real temperatures, $\langle T(t) \rangle$ is the deterministic temperature dynamics with period of 365 days and $\bar{T}$ is the arithmetic average of temperature. The daily average temperature is obtained by averaging the maximal and minimal temperatures. The 98 years of temperature recordings from Cologne as

$$\text{Temperature fluctuations of each year} = \text{Each of 98 years} - (\langle T(t) \rangle + \bar{T}).$$

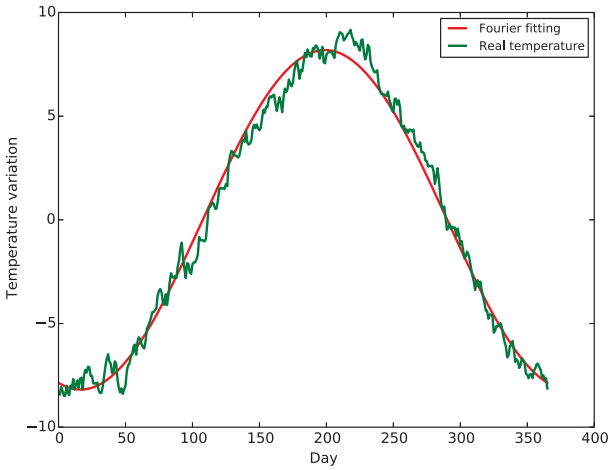


Figure S2: The real temperature is obtained from averaging 98 years of temperatures in Cologne. The fitted temperature curve is based on a second order Fourier series.

Fourier series based regression for $\langle T(t)\rangle$

To mimic the long term adaptation of plants to local climates, the average of 98 years of temperature over one-year period gives an estimate of temperature dynamics and averaged temperature fluctuations. The temperature dynamics was fitted by a second-order Fourier series as

$$\langle T(t)\rangle = a_0 + a_1\cos(wt) + b_1\sin(wt) + a_2\cos(2wt) + b_2\sin(2wt) \tag{1}$$

with determined parameters

$$(a_0, a_1, b_1, a_2, b_2, w)_T = (0, -7.775, -2.669, -0.1552, 0.4052, 0.0172). \tag{2}$$

The fitted result is shown in Fig. S2.

The exponential decay in temperature time series

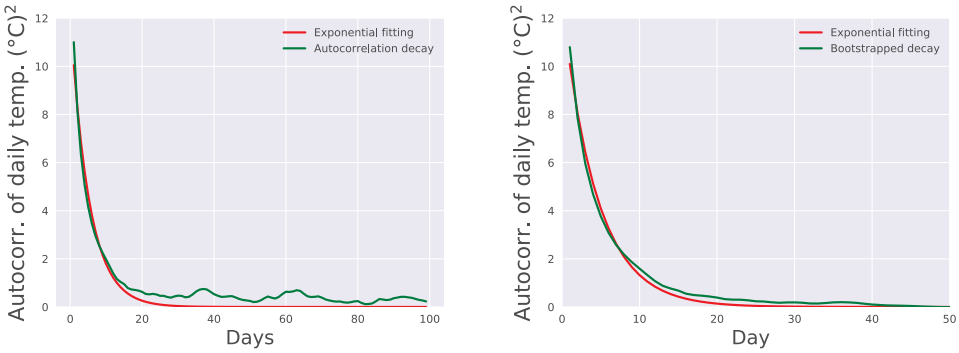
The autocorrelation in temperature fluctuations is defined as

$$R(t) = \frac{1}{n-k}\sum_{t=1}^{n-k}\delta T_t \delta T_{t+k}, \tag{3}$$

where n is the total number of days, and k is the presumed maximal correlated number of days, which was set as 80 in our calculation. It is shown in Fig. S3a that the autocorrelations in temperature fluctuations follow an exponential decay which can be fitted as

$$autocorrelation(t) = \sigma^2 e^{-t/\tau}, \tag{4}$$

where $\sigma^2 = 12.67$ and $\tau^{-1} = 0.22$ with a fitting R^2 score of 0.934. In addition, we performed a bootstrap analysis to estimate the confidence interval of the fitted coefficients. Since the fluctuations are correlated, in order to preserve the correlation, the sampling used a block length of 50 days. From the bootstrap with 10000 samples, the confidence intervals for σ^2 and τ were estimated as $(11.90, 13.42)$ and $(0.199, 0.253)$ respectively. The fitted coefficients indeed located in the confidence interval. The bootstrapped fitting results are shown in Fig S3b.



(a) (b)

Figure S3: The real decay of autocorrelations exhibits random fluctuations after about 15 days. The exponential decay fitting smoothed these fluctuations to zero.

2.2 Autocorrelation analysis for other climates

Similar to the calculation of autocorrelation in temperature fluctuations for Cologne, the procedure was applied to other selected climates as shown in Fig S4. It can be observed that the autocorrelations for Cologne, Oslo and Auckland showed exponential decays but with different lower bounds, which may be due to different noise levels in temperatures. And for regions with less seasonal changes like Hawaii and San Francisco, exponential decays were not observed.

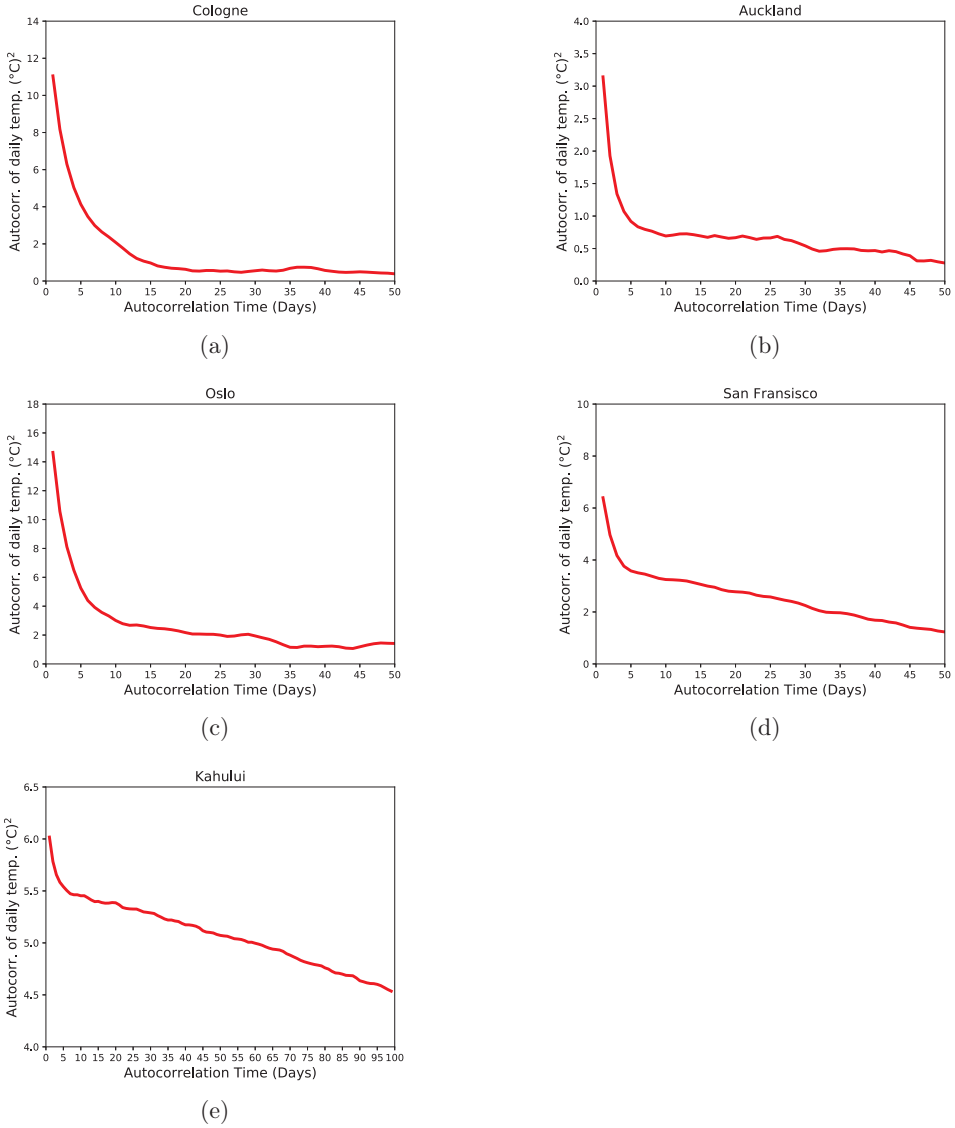


Figure S4: The autocorrelation time decays for different climates.

3 Modeling the simplified histone modification reaction

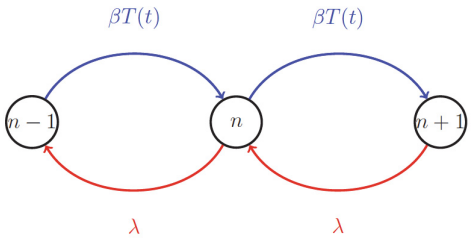


Figure S5: Temperature-driven histone modification. n stands for the number of modified histones, the production rate $\beta T(t)$ depends on the temperature $T(t)$ and λ is the degradation rate of histones.

3.1 The master equation and its analytical solution

Following the previous work of modeling simple birth-death process for one species [2, 3, 1], the histone modification reaction is simplified as $\phi \underset{\lambda}{\overset{\beta T(t)}{\rightleftharpoons}} \mathrm{M}$ which is further illustrated in Fig. S5. The production rate of active modification of histones $\beta T(t)$ depends on the temperature $T(t)$ and λ is the degradation rate of active modifications. And the maximal number of required modification sites is N. The master equation for describing the reaction can be written as

$$\partial_t p(n,t) = \beta T(t)\big(p(n-1,t) - p(n,t)\big) - \lambda n p(n,t) + \lambda(n+1)p(n+1,t). \tag{5}$$

To solve the partial differential equation, we first need to introduce the time-dependent probability generating function $G(s,t) = \sum_{n=0}^{\infty} s^n p(n,t)$. Then multiplying s^n to the equation (5) and summing over n can transform the equation to

$$\partial_t G(s,t) = \beta T(t)(s-1)G(s,t) - \lambda(s-1)\partial_s G(s,t). \tag{6}$$

First neglecting the time dependence of the temperature, the solution of the equation (6) is

$$G(s,t) = (1 + (s-1)e^{-\lambda t})^N e^{\beta T(s-1)\int_0^t e^{-\lambda t'} dt'} \tag{7}$$

(the derivation of the solution can be found in chapter 7 of [3]). Now considering the time dependence of T, we can assume the solution becomes

$$G(s,t) = \big(1 + (s-1)e^{-\lambda t}\big)^N e^{\beta(s-1)F(t)}. \tag{8}$$

To find $F(t)$, substituting (8) into (6) yields

$$F'(t) + \lambda F(t) = T(t). \tag{9}$$

By solving the differential equation (9), we have

$$F(t) = ce^{-\lambda t} + e^{-\lambda t}\int_0^t T(t')e^{\lambda t'}dt'. \tag{10}$$

Under the assumptions of $c = 0$ and $T(t) = d + \langle T(t)\rangle + \delta T(t)$ where d is the constant for converting temperature in Celsius to Kelvin, $\langle T(t)\rangle$ is the temperature average over years and $\delta T(t)$ is the temperature fluctuation, the equation (8) can be written as

$$G(s,t) = \left(1 + (s-1)e^{-\lambda t}\right)^N e^{\beta(s-1)e^{-\lambda t}\int_0^t \left(d + \langle T(t')\rangle + \delta T(t')\right)e^{\lambda t'}dt'}. \tag{11}$$

Considering the stationary state of $G(s,t)$, which follows a yearly cycle and is still time dependent, we have

$$G^*(s,t) = e^{\frac{d\beta}{\lambda}(s-1)}\left\langle e^{\beta(s-1)e^{-\lambda t}\int_{-\infty}^t \delta T(t')e^{\lambda t'}dt'}\right\rangle e^{\beta(s-1)e^{-\lambda t}\int_{-\infty}^t \langle T(t')\rangle e^{\lambda t'}dt'} \tag{12}$$

Noise-free temperature

Now, starting with the simple scenario, we assume temperature dynamics are noise-free, which means $\delta T(t) = 0$. Consequently, the steady state solution becomes

$$G^*(s,t) = e^{\frac{d\beta}{\lambda}(s-1)}e^{\beta(s-1)e^{-\lambda t}\int_{-\infty}^t \langle T(t')\rangle e^{\lambda t'}dt'} \tag{13}$$

Zooming into the equation (13), we need to deal with the part related to temperature dynamics: $e^{-\lambda t}\int_{-\infty}^t \langle T(t')\rangle e^{\lambda t'}dt'$. For the simplicity of notation, we define

$$D(\lambda, t) := e^{-\lambda t}\int_{-\infty}^t \langle T(t')\rangle e^{\lambda t'}dt'. \tag{14}$$

In the case of temperatures from Cologne, the temperature dynamics $\langle T(t)\rangle$ can be numerically approximated by a second-order Fourier series (Fig. S2). Using the parameters fitted in the approximation (1), we have

$$\begin{aligned} D(\lambda,t) = \frac{a_0}{\lambda} + \frac{a_1}{w^2+\lambda^2}\left(\lambda\cos(wt) + w\sin(wt)\right) + \frac{b_1}{w^2+\lambda^2}\left(\lambda\sin(wt) - w\cos(wt)\right) + \\ \frac{a_2}{(2w)^2+\lambda^2}\left(\lambda\cos(2wt) + 2w\sin(2wt)\right) + \frac{b_2}{(2w)^2+\lambda^2}\left(\lambda\sin(2wt) - 2w\cos(2wt)\right), \end{aligned}$$

where $t \in [0, 2L]$ with $2L$ the period of temperature cycle. By further defining $b := \frac{d\beta}{\lambda} + \beta D(\lambda, t)$, the stationary generating function becomes

$$G^*(s,t) = e^{b(s-1)}, \tag{15}$$

which is a generating function for a Poisson distribution. Therefore, in the case of noise-free temperatures, the stationary solution to the master equation (5) is

$$p(n,t) = \frac{b^n}{n!}e^{-b}. \tag{16}$$

Noisy real temperature

Now considering the real noisy temperature, we need to deal with the part caused by temperature fluctuations: $\left\langle e^{\beta(s-1)e^{-\lambda t}\int_{-\infty}^{t}\delta T(t')e^{\lambda t'}dt'}\right\rangle$. Since the fluctuations has been shown to have an exponential decay in autocorrelations for temperatures in Cologne, by Doob's theorem [4], they follow a Gaussian Markovian Process. Therefore, we have

$$\left\langle e^{\beta(s-1)e^{-\lambda t}\int_{-\infty}^{t}\delta T(t')e^{\lambda t'}dt'}\right\rangle = e^{\frac{1}{2}\beta^2(s-1)^2\int_{-\infty}^{t}\int_{-\infty}^{t}\left\langle\delta T(t_1)e^{\lambda(t_1-t)}\delta T(t_2)e^{\lambda(t_2-t)}\right\rangle dt_1dt_2} \tag{17}$$

$$= e^{\frac{1}{2}\beta^2(s-1)^2\int_{-\infty}^{t}\int_{-\infty}^{t}\left\langle\delta T(t_1)\delta T(t_2)\right\rangle e^{\lambda(t_1+t_2-2t)}dt_1dt_2} \tag{18}$$

$$= e^{\frac{1}{2}\beta^2(s-1)^2\int_{-\infty}^{t}\int_{-\infty}^{t}\sigma^2 e^{-\frac{|t_2-t_1|}{\tau}}e^{\lambda(t_1+t_2-2t)}dt_1dt_2} \tag{19}$$

$$= e^{\frac{\beta^2\sigma}{2\lambda(\lambda+A)}(s-1)^2}. \tag{20}$$

With $a := \frac{\beta^2\sigma^2}{2\lambda(\lambda+\tau^{-1})}$ and $b = \frac{d\beta}{\lambda} + \beta D(\lambda, t)$, the stationary $G(s,t)$ (12) can be written in a closed form as

$$G^*(s,t) = e^{a(s-1)^2+b(s-1)}. \tag{21}$$

Due to the quadratic term in the exponential of (21), finding the associated probability density function requires the introduction of an auxiliary generating function which can be expanded in terms of Hermite Polynomial

$$G_a(x,t) = e^{-t^2+2xt} = \sum_{n=0}^{\infty} H_n(x)\frac{t^n}{n!}. \tag{22}$$

By changing variables $a = -\alpha^2$, $t = \alpha s$ and $x = \frac{b+2\alpha^2}{2\alpha}$, we have the following deduction

$$G^*(s,t) = e^{b(s-1)+a(s-1)^2} = e^{a-b}e^{-\alpha^2s^2+(b+2\alpha^2)s} = e^{a-b}e^{-t^2+2xt} = e^{a-b}\sum_{n=0}^{\infty} H_n(x)\frac{t^n}{n!}. \tag{23}$$

In short, the generating function (21) becomes

$$G^*(s,t) = e^{a-b}\sum_{n=0}^{\infty} H_n(x)\frac{t^n}{n!}. \tag{24}$$

Substituting the variables a and b to (24), we have

$$G^*(s,t) = e^{a-b}\sum_{n=0}^{\infty} H_n(\frac{b+2\alpha^2}{2\alpha})\frac{(\alpha s)^n}{n!} = \sum_{n=0}^{\infty} s^n p(n,t). \tag{25}$$

Finally the probability density function is derived as

$$p(n,t) = e^{a-b}\frac{\alpha^n}{n!}H_n(\frac{b+2\alpha^2}{2\alpha}). \tag{26}$$

3.2 Optimization

The model parameters can be uniquely determined by optimizing a proper decision function in such a way that the signal-to-noise ratio determines the degradation rate and the decision boundary determines the production rate. If the degradation rate λ is very small, then in order to have a reasonable distribution over all modification states, the reaction rate has to be very small. In this slow reaction scenario, the effect of temperature dynamics is diluted, which make the reaction hard to capture the useful information in temperatures. On the contrary, if λ is very large and β should be very large to maintain a certain number of active histone modifications. In the fast production and degradation scenario, every single fluctuation in temperatures would drive the reaction in an undesirable manner. Both extreme cases are not practical for plants to rely on. Therefore, λ has to be tuned by the signal-to-noise ratio in temperature to a reasonable level. Meanwhile, the optimal value of β, λ can be determined by setting the decision boundary to the required amount of modified histone sites.

With the obtained parameters $d = 283.3K$, $\tau^{-1} = 0.22$, $\sigma^2 = 12.67$ for the exponential decay in temperature fluctuations and the parameters for the Fourier fitting of temperature dynamics in (2), the probability distribution $p(n,t)$ (26) has only the undetermined reaction rates β, and λ. By optimizing the following objective

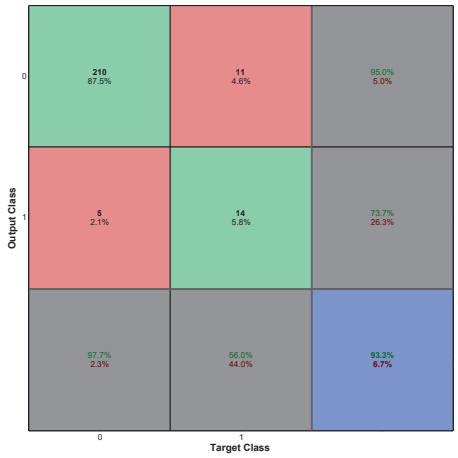
$$F(\beta,\lambda) = \int_{Mar}^{Jun} \sum_{n=N_c}^{N_{max}} p(n,t)dt + \int_{Jul}^{Feb} \sum_{n=0}^{N_c-1} p(n,t)dt, \tag{27}$$

the optimal β and λ were obtained as

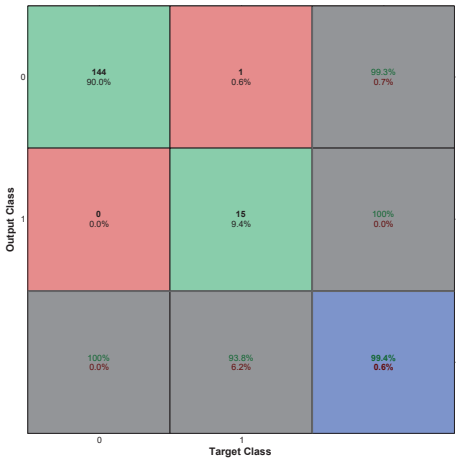
$$\beta = 6.45, \lambda = 2.94.$$

4 Classification for estimating memory length

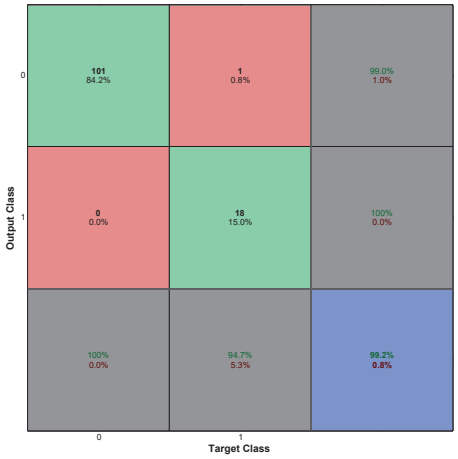
The input features are $(x_{l1}, \cdots, x_{lk}, x_{h1}, \cdots, x_{hk})$, where k is window length, x_{li} and x_{hi}, $i = 1, \cdots, k$, stand for daily lowest and highest temperatures respectively. For example, when the window length is set to be $k = 30$ (i.e. a month), without loss of generality, April is set to be class "1", the rest months are set to be class "0". Then for the temperatures of each year, the corresponding targets are $(0,0,0,1,0,0,0,0,0,0,0,0)$. The same configuration can be applied to other window lengths. We have tested lengths $k \in \{28, 42, 60\}$ by using a repeated cross validation routine, that is, for each model training, 80 years was randomly selected for training and the rest 18 years was used for testing. The averaged testing results were shown in Fig.S6. For length $k = 30$, the false positive rate is 26.3%, although the overall accuracy is 93.3%. When the length is increased to $k = 42$, the false positive rate dropped to zero, indicating that this length provides sufficient information for determining the season, which can be used as the effective memory length of plants for recognizing seasons.



(a) 28 days



(b) 42 days



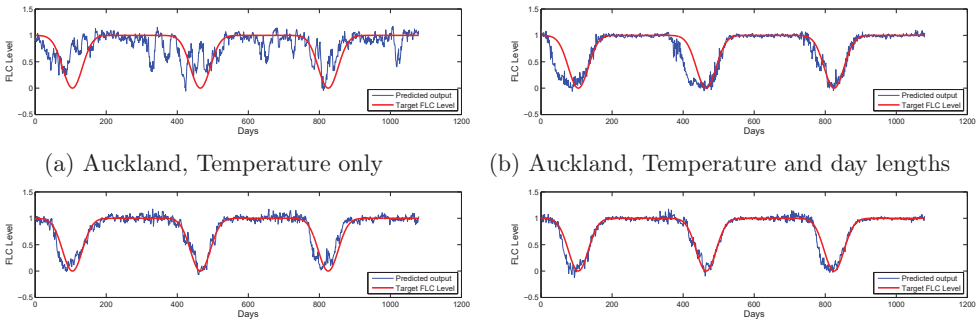
(c) 60 days

Figure S6: The confusion matrices for different window sizes.

Regression of idealized expression patterns for various climates

In this section, different climates information was used to fit the idealized expression pattern of *FLC*. Based on the classification results, the input features and targets for regression were constructed as following. Each day of a year has a corresponding expression value from the idealized expression curve as the target. The features for that day comprised of the temperatures (daily maxima and minima) and/or day lengths of its precedent days. The number days for temperatures was taken as the effective memory

length from the classification. For temperate region, long term memory of temperatures played a key role in the regression and the addition of day lengths significantly improved the regression.



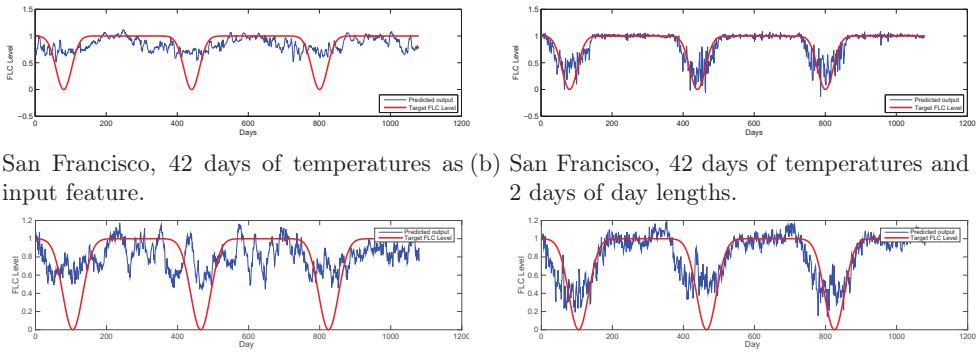
(a) Auckland, Temperature only

(b) Auckland, Temperature and day lengths

(c) Oslo, temperature only

(d) Oslo, temperature and day lengths

Figure S7: The regression results of using different input features combinations of temperate regions.



(a) San Francisco, 42 days of temperatures as input feature.

(b) San Francisco, 42 days of temperatures and 2 days of day lengths.

(c) Kahului, 42 days of temperatures as input features.

(d) Kahului, 42 days of temperatures and 2 days of day lengths.

Figure S8: The regression results by using data from regions with less seasonal changes.

Regions as San Francisco and Kahului have very similar temperature every year, which means very few information can be extracted from their temperature. This also led to the poor generalization in predicting the idealized expression patterns (Fig S8). Moreover the predicting results showed no significant differences for different temperature memory lengths, e.g. 7 days, 20 days and 40 days. The results agree with that plants in tropical region, which do not rely on long term memory of temperature for flowering decision making. In order to better fit *FLC* expression patters of regions such as San Francisco and Kahului, temperature memory was reduced to at most one and longer term of day lengths were added as input features. For both San Francisco and Kahului, it was seen

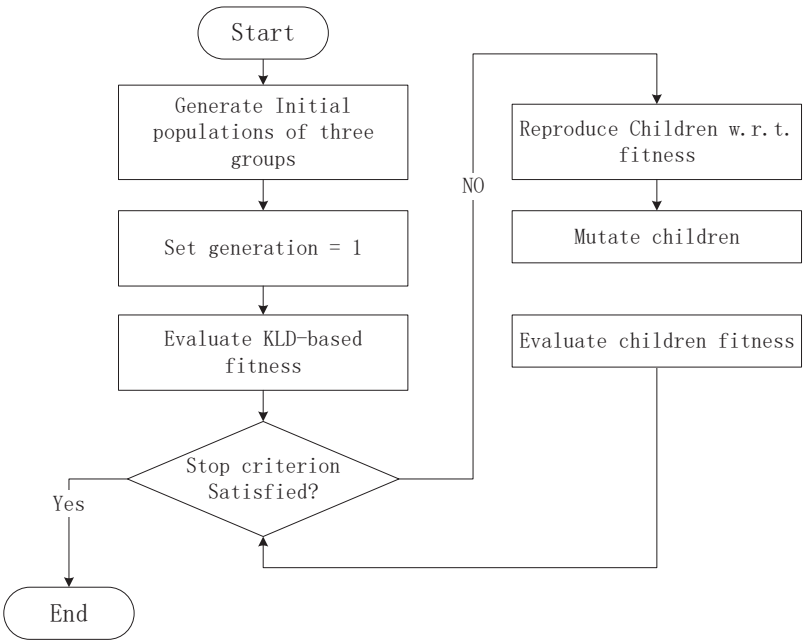


Figure S9: Evolutionary simulations for each group of the population.

that the trained models with only temperatures as features generalized poorly. But the fitting accuracy can be improved by adding day lengths.

5 Evolution Simulation

The simulation flowchart is shown in Fig. S9. And the simulation results are shown in Fig. S10.

References

[1] Rausenberger, J. & Kollmann, M. Quantifying origins of cell-to-cell variations in gene expression. *Biophysical journal* **95**, 4523–4528 (2008).

[2] Walczak, A. M., Mugler, A. & Wiggins, C. H. Analytic methods for modeling stochastic regulatory networks. In *Computational Modeling of Signaling Networks*, 273–322 (Springer, 2012).

[3] Gardiner, C. W. Handbook of stochastic methods: for physics, chemistry and the natural sciences. (2004).

[4] Doob, J. L. The brownian movement and stochastic equations. *Annals of Mathematics* 351–369 (1942).

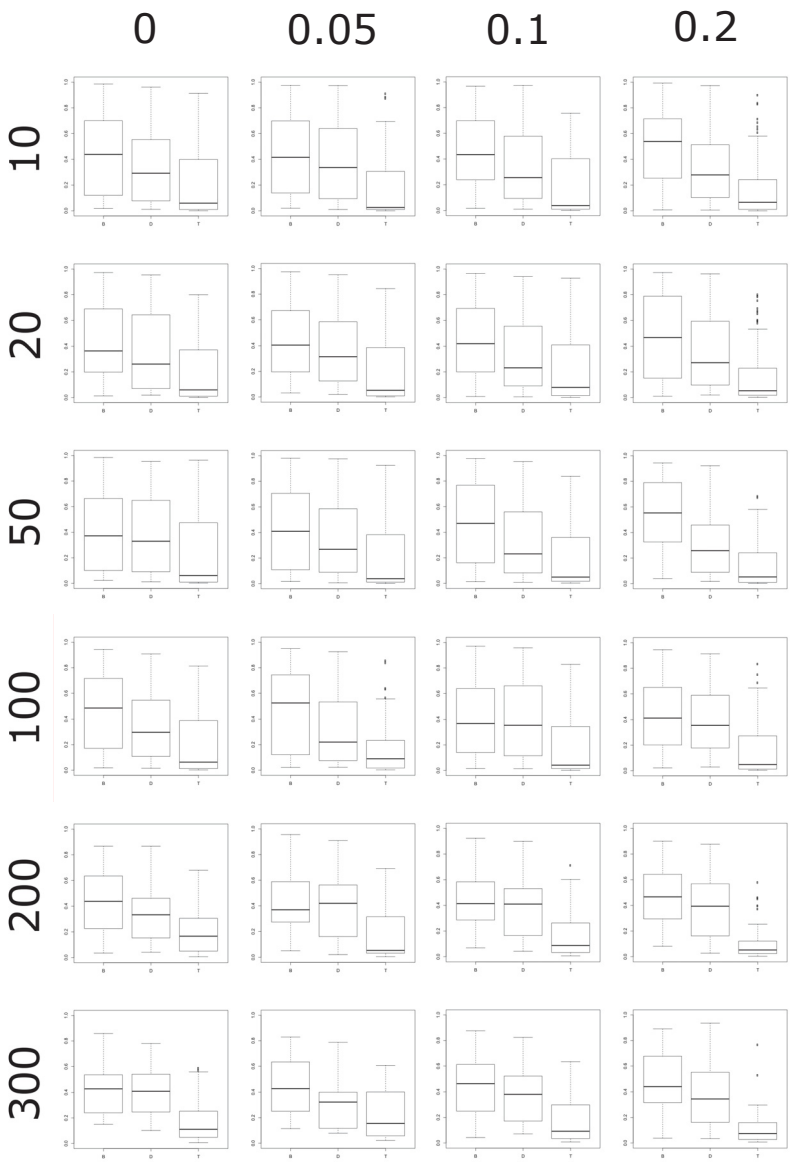


Figure S10: The proportion of survived individuals for different groups. The vertical axis of the plot represents the group sizes, and the horizontal axis represents the mutation rates. Each subplot has three boxplots which corresponding to the distribution of the proportions of survived individuals with accesses to temperatures, day lengths or both.

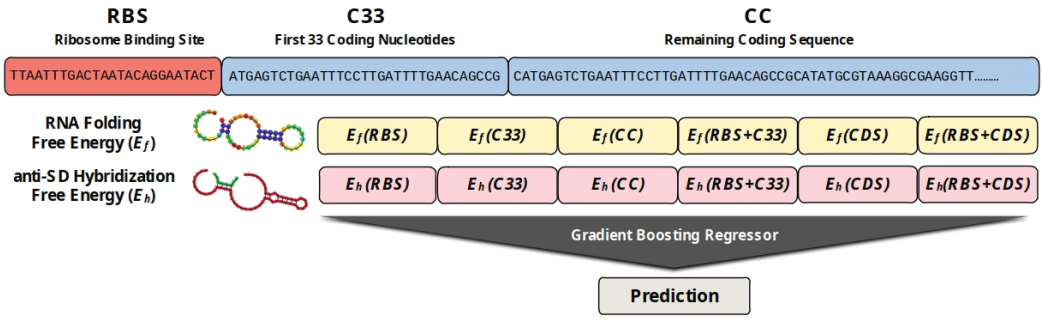
4 Data-driven modeling of the regulation in mRNA translation

4.1 Summary

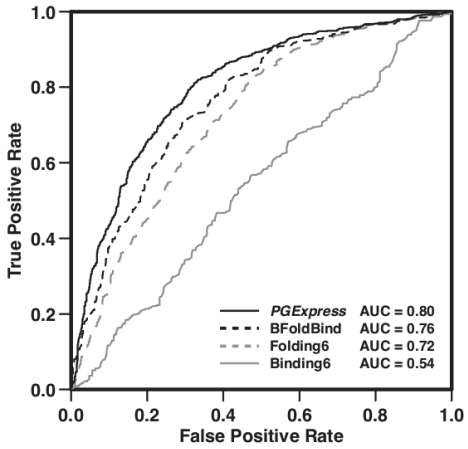
RNA translation is an energy-intensive and resource-demanding process in any cell. During evolution, organisms evolved sophisticated regulatory mechanisms to optimize the translation efficiency and accuracy to maintain the translation fidelity while minimize the cost [CGA18; GP11]. As the blueprint of translation, RNA sequences carry the relevant information for translation regulations. The increasing available data due to the advances in next-generation-sequencing paved the way for machine learning to help investigating of the regulatory patterns underlying sequences. For simplicity, the manuscript in *Section 4.2* focused on the mRNA translation in *E. coli*, which is also the focus of the rest of the summary.

The known regulatory factors mainly include codon bias due to redundant genetic codes, special sequence motifs, secondary folding structures and contents of different nucleotides [CGA18; GP11]. Beside direct influences of these factors on translation, their interplays and their cooperations with cellular resources, such as the abundance of *transfer RNA*, may also play a key role in translation regulations. It was crucial to represent the sequences to incorporate the known factors such that machine learning models can find the correct patterns. Various ways of representations were explored, and eventually it was hypothesized that the global and local secondary structures of mRNA sequences and special motifs like the Shine-Dalgarno motif [SD73] were the key regulatory factors. Other factors such as the codon bias and nucleotide contents may be caused by structural needs. Based on the hypothesis, a gradient boosting trees based predictive model was developed to capture the regulatory patterns based on features generated from mRNA sequences. The framework was shown in Fig. 4.1a. In the consideration of global and local structures, each sequence was divided into ribosome binding sites (RBS), coding sequences near the starting codon (C33), and the rest of coding sequences (CC), where the concatenation of C33 and CC was the whole coding sequence (CDS). The minimum free energies, characterizing the structure signals, and the anti-Shine-Dalgarno hybridization free energies, characterizing the Shine-Dalgarno motifs, were calculated for the sequence segments and their combinations.

The model evaluation was done through gene-based 10 fold cross validation based on the dataset [GCK13] which included 13 variants of 137 essential genes in *E. coli*. The evaluated performance of the model achieved correlation coefficient of 0.57 for regression and area under the curve (AUC) score of 0.80 for classifying genes with high or low expressions (Fig. 4.1b). Moreover, to verify the prediction of the model, from the 10 sequences generated by the model, the predictions of 9 sequences were verified by the in-house experiments.



(a) Representation of the predictive algorithm. The input is a 12-elements vector composed by the predicted RNA secondary structure (E_f) and anti-Shine-Dalgarno hybridization (E_h) free energies per nucleotide. Each sequence is divided into 3 blocks: the Ribosome Binding Site (RBS), the first 33 nucleotides of the coding sequence (C33) and the remaining part of the coding region starting from nucleotide 34 (CC). The whole coding sequence (CDS) is obtained joining C33 and CC.



(b) The performances of models. The final model *PGExpress* was built on all generated features. The *BFoldBind* method was built on most discriminative RNA folding and anti-SD hybridization free energies. The *Folding6* and *Binding6* were built on 6 folding and 6 hybridization energies respectively.

Fig. 4.1

4.2 Predicting translational efficiency from mRNA sequences

Publication status

Linlin Zhao, Nima Abedpour, Christopher Blum, Petra Kolkhof, Mathias Beller, Markus Kollmann and Emidio Capriotti, "Predicting gene expression level in E. coli from mRNA sequence information", accepted by IEEE CIBCB 2019 (International Conference on Computational Intelligence in Bioinformatics and Computational Biology)

Linlin Zhao's contributions

1. Conducted exhaustive literature review for sorting out most relevant features which characterized mRNA sequences.
2. Generated features which represented the secondary structures, binding affinity between ribosomes and ribosomal binding sites and special motifs within coding regions of mRNA sequences. This part has been done with help and discussions from other coauthors.
3. Besides the final model used in the paper, some more complex models were obtained which achieved similar prediction performances. For example, I trained a convolutional neural network based multi-task model on several independent published datasets. According to Occam's Razor principle, we have kept the simplest working model. The final model was obtained in such a way that I first discussed with Emidio Capriotti and then we both trained the model simultaneously to assure the rigor of our results.
4. Generated 10 sequences by the predictive model for in-house experiment.
5. I wrote the introduction, conclusion and the section 2.7 and 3.8 of the manuscript.

Predicting gene expression level in *E. coli* from mRNA sequence information

Linlin Zhao[1], Nima Abedpour[2], Christopher Blum[1], Petra Kolkhof[1], Mathias Beller[3], Markus Kollmann[1,] and Emidio Capriotti[4,*]*

[1] Institute for Mathematical Modeling of Biological Systems, Department of Biology. Heinrich Heine University Düsseldorf. Universitaetsstr. 1, 40225 Düsseldorf, Germany.
[2] Department of Translational Genomics, University of Cologne, Weyertal 115b, 50931 Cologne, Germany.
[3] Systems Biology of Lipid Metabolism, Department of Biology. Heinrich Heine University Düsseldorf. Universitaetsstr. 1, 40225 Düsseldorf, Germany.
[4] BioFolD Unit, Department of Pharmacy and Biotechnology (FaBiT), University of Bologna, Via F. Selmi 3, Bologna, 40126, Italy.
*To whom correspondence should be addressed.

Abstract
Motivation: The accurate characterization of the translational mechanism is crucial for enhancing our understanding of the relationship between genotype and phenotype. In particular, predicting the impact of the genetic variants on gene expression will allow to optimize specific pathways and functions for engineering new biological systems. In this context, the development of accurate methods for predicting the translation efficiency and/or protein expression from the nucleotide sequence is a key challenge in computational biology.
Methods: In this work we present *PGExpress*, a new regression method for predicting the log2-fold-change of the translation efficiency of an mRNA sequence in *E. coli*. *PGExpress* algorithm takes as input 12 features corresponding to the predicted RNA secondary structure and anti-Shine-Dalgarno hybridization free energies. The method was trained on a set of 1,772 sequence variants (WT-High) of 137 essential *E. coli* genes. For each gene, we considered 13 sequence variants of the first 33 nucleotides encoding for the same amino acids followed by the superfolder GFP. Each gene variant is represented sequence blocks that include the Ribosome Binding Site (RBS), the first 33 nucleotides of the coding region (C33), the remaining part of the coding region (CC), and their combinations.
Results: Our gradient-boosting-based tool (*PGExpress*) was trained using a 10-fold gene-based cross-validation procedure on the WT-High dataset. In this test *PGExpress* achieved a correlation coefficient of 0.57, with a Root Mean Square Error (RMSE) of 1.4. When the regression task is cast as a classification problem, *PGExpress* reached an overall accuracy of 0.73 a Matthews correlation coefficient 0.47 and an Area Under the Receiver Operating Characteristic Curve (AUC) of 0.80. When compared with *RBSCalculator, PGExpress* results in better performance in the prediction of the log2-fold-change of the translational efficiency and its variation on the WT-High dataset. Finally, we validated our method by performing in-house experiments on five newly generated mRNA sequence variants. The predictions of the expression level of the new variants are in agreement with our experimental results in *E. coli*.
Availability: http://folding.biofold.org/pgexpress
Contact: markus.kollmann@hhu.de, emidio.capriotti@unibo.it

1 Introduction

The ability to predict translation efficiency in bacteria is important to define the relation between genotype and phenotype, and to engineer new organisms optimized for producing biomaterials (Kyle, et al., 2009), fuels (Toone and de Winde, 2013) and natural products (Krivoruchko and Nielsen, 2015). The information to regulate the translation process is encoded in the mRNA nucleotide sequence. The preference for specific combinations of nucleotides in the coding region, which refers to codon bias, has a strong effect on protein expression and formation (Li, et al., 2012; Mortimer, et al., 2014; Plotkin and Kudla, 2011; Pop, et al., 2014). Changes in the nucleotide sequence and codon usage can affect the mRNA folding process, which is a key determinant of protein expression. The ability of RNA strands to fold and form stable structures influences all the steps of the translation process: the structures at untranslated regions (UTR), especially at ribosome binding sites (RBS), act as a barrier for the ribosome to dock on the transcripts, then slow down the translation initiation (Duval, et al., 2013); the local structures in coding sequences (CDS) interplays with tRNA abundance to smoothen the translation elongation (Gorochowski, et al., 2015); structures also affect mRNA localization and turnover (Mortimer, et al., 2014). The Shine-Dalgarno (SD) sequence encoded in the mRNA is another key factor for translation regulation. Indeed, when the SD sequence is located in untranslated regions (UTRs), it promotes the binding of ribosomes and accelerates translational initiation (Kozak, 2005; Shine and Dalgarno, 1974; Shultzaberger, et al., 2001). Contrarily, its presence in the coding region can reduce the translational elongation rate in bacteria (Li, et al., 2012). Thus, the understanding of the mechanism of bacterial translation will result in accurate predictions of protein expression from mRNA sequence (Gingold and Pilpel, 2011). In this work we primarily considered the measure of translation efficiency, which provides a quantitative estimation of the of translation process, independent from the transcription. The translation efficiency is defined as the ratio of protein to mRNA abundance, which corresponds to the amount of protein produced by a single molecule of mRNA (Tuller, et al., 2010a; Tuller, et al., 2010b).

In the past, many studies and software tools have been developed for predicting protein expression based on mRNA sequence. Tools to tailor the untranslated region (UTR) to achieve a desired protein expression level were also introduced (Na and Lee, 2010; Reeve, et al., 2014; Rodrigo and Jaramillo, 2014; Seo, et al., 2014). For instance, the RBS calculator (Salis, 2011), UTR designer (Seo, et al., 2014), and RBS designer (Na and Lee, 2010) which are statistical models considering free energies for key molecular interactions in translation initiation and the formation of mRNA local structures provided an estimation of the translation efficiency. In general, the predictions from these methods show good correlation with their experimental data respectively. Recently Bonde and colleagues (Bonde, et al., 2016) studied the relationship between SD sequences and protein expression by measuring expression levels of ~3,000 UTRs in the presence of different SD variants. Their empirical method (EMOPEC) was able to predict the protein expression level of newly designed sequences in 91% of the cases. Focusing on the UTR regions, the available tools limit our understanding of the general picture of translational mechanism and our ability to engineer the whole mRNA molecule. Recently, Goodman and colleagues (Goodman, et al., 2013) measured the expression level of more than 14,000 synthetic gene variants in *E. coli* to quantify the effects of N-terminus codons as well as different combinations of promoter and Ribosome Binding Sites (RBSs). They found that rare codons in the N-terminus increased the stability of the RNA structure resulting in decreased gene expression level. The gene variants tested by Kosuri and co-workers (Goodman, et al., 2013) included variations in both UTR and coding sequences, which made the data suitable for investigating the effects from coding sequences as well. We make use of their data to capture regulatory factors from both the UTR and coding region of the mRNA molecule.

For estimating the contributions of different RNA regions on gene expression, we represented the sequences by the predicted global and local RNA folding free energies to define the main features contributing to the translation efficiency. Since mRNA structure impacts each step of translation (Kozak, 2005; Mortimer, et al., 2014), it represents one of the most important features to consider. The RNA folding free energy is a classical scoring function used for the prediction of RNA

secondary structure. Indeed, different tools for predicting RNA secondary structure implement dynamic programming algorithm for minimizing the free energy (Capriotti and Marti-Renom, 2008). Recent experimental studies showed that different regions of mRNA preserve specific structural preferences (Kudla, et al., 2009; Mortimer, et al., 2014). Kudla and colleagues found that the predicted folding free energy of the first ~40 nucleotides of the mRNA significantly correlates with the GFP protein abundance (Kudla, et al., 2009). Furthermore, it was observed that structures at the end of 5′ UTR and the beginning of 3′UTR are well conserved and the coding region is more structured than UTRs (Mortimer, et al., 2014). Thus, the free energy associated to the formation of local structures is also an important predictive feature. Since the SD sequence shows different regulating effects, we also predicted the hybridization free energy (also referred as binding energy) between the anti-SD sequence and different regions of the mRNA. The predicted folding and hybridization free energies were combined to represent the translational features of the mRNA.
In this work we present *PGExpress* (***P***redicting ***G***ene ***Express***ion), a new gradient boosting-based algorithm predicting translation efficiency of mRNA sequences. *PGExpress* is a regression method that predicts the log2-fold-change of translation efficiency with respect to the median value observed experimentally. Our method relies on the calculation of the minimum RNA secondary structure free energy as representations of the local and global mRNA structures and the minimum free energy of hybridization between anti-SD sequence and mRNA, which corresponds to the binding affinity of the ribosome with different strands of mRNA. The performance of *PGExpress* has been tested on previously published datasets and new experimental data generated in-house.

2 Methods

2.1 Datasets

The data used in this work consists of protein expression and/or translation efficiency measures of genes and their variants in *E. coli*. The data was collected both from the literature (Goodman, et al., 2013) and experimental tests in our lab. The data from Kosuri and collaborators (Kosuri-All) is a collection of protein expression (PE) and translation efficiency (TE) measures from ~14,000 gene variants (Goodman, et al., 2013). More information about the gene expression measures considered in this work is reported in section 1 of the Supplementary Materials. Each variant is a combination of the Promoter with high and low strength (High, Low), the Ribosome Binding Site (Wild-Type, Weak, Mid and Strong RBSs) and the first 33 nucleotides of the coding region (C33) of 137 essential *E. coli* genes followed by the superfolder GFP (sfGFP) coding sequence (see Supplementary Materials, section Experimental data). From the Kosuri-All dataset we extracted five subsets (WT-High, WT-Low, Weak-High, Mid-High, Strong-High) with sequence variants composed by four Ribosome Binding Sites (RBS) and two Promoters. The main dataset (WT-High), which has been used for training and testing our method, collects the expression measures of 1,722 sequences formed by the High affinity promoter, the Wild-Type RBSs and 13 variants (including wild-type) of the C33 region of each gene. The Weak-High, Mid-High and Strong-High subsets, which have been used only in the testing phase, differ from the WT-High for the sequence of the Ribosome Binding Site, which has Weak, Mid and Strong binding affinities respectively. The WT-Low and WT-High differ for the sequence of the promoter regions, which have low and high strength respectively. The WT-Low dataset has been used only in the preliminary analysis of the data.
For training and testing the regression algorithm the values of the protein expression and translation efficiency are converted in log2-fold-change with respect to their median values in the WT-High dataset (2,988 and 2,355 respectively). For evaluating the performance of the method as a binary classifier, the previous median values are used as classification thresholds. Finally, to test the performance of *PGExpress*, we measured in our lab the protein expression level of five randomly selected variants from the Kosuri-All dataset (Exp-Set). We used the Exp-Set to check the

agreement between the data in Kosuri-All and our measures. Then, we generated a validation set, namely Exp-Mut, which is composed of new variants derived from the five sequences in Exp-Set. The sequences of the ten gene variants are reported in Table S1.

The Kosuri-All dataset analyzed in this work is provided in Supplementary File 1. A summary of its composition is reported in Table S2.

2.2 Algorithm description

Here we present a regression method (*PGExpress*) to predict the log2-fold-change of the gene translation efficiency (L2TE) from sequence information. *PGExpress* is based on gradient boosting regression algorithm that takes in input a 12-elements vector composed by six predicted RNA folding and six anti Shine-Dalgarno (SD) hybridization free energies per nucleotide. In detail, each gene variant is divided in three sequence blocks: the Ribosome Binding Site (RBS), which consists of ~25 nucleotides preceding the coding sequence, the first 33 nucleotides of the coding region (C33) and the remaining part of the coding sequence starting from nucleotide 34 (CC). Thus, each gene is represented by six sequence fragments including the three blocks previously defined (RBS, C33 and CC), and the combinations of RBS with C33 (RBS+C33), C33 with CC (CDS) and RBS with the whole coding sequence (RBS+CDS). For each block we predicted the RNA secondary structure and the anti-Shine-Dalgarno (anti-SD) hybridization free energies using respectively *RNAfold* and *RNAduplex* tools from the ViennaRNA package (Lorenz, et al., 2011), which automatically replace Thymine (T) with Uracil (U). We used an 8-nucleotides anti Shine-Dalgarno sequence (CCTCCTTA) as reported by Kosuri and coworkers (Goodman, et al., 2013). Both free energies have been rescaled to a temperature of 30°C, which is the temperature at which the experiment in the Kosuri study was carried out. *PGExpress* return in output the predicted log2-fold-change of the translational efficiency with respect its median value on the WT-High subset (2355). A representation of *PGExpress* algorithm and its 12 input features is provided in Fig. 1.

2.3 Feature analysis

To estimate the predictive power of each feature, we calculated the linear regression between the RNA folding and anti-SD hybridization free energies of the five sequence blocks (RBS, C33, RBS+C33, CDS and RBS+CDS) and the log2-fold-change of the translation efficiency in the WT-High dataset. In this analysis we did not consider the C-terminal region of the coding sequence (CC) because it corresponds to the sfGFP for all the variants in the Kosuri-All dataset. Furthermore, we compared the performance of our best approach (*PGExpress*) against five methods including different combinations of the 12 input features. These methods are:

- **BFolding**: the most discriminative RNA folding free energy
- **BBinding**:the most discriminative anti-SD hybridization free energy
- **Folding6:** RNA folding free energies of the six blocks
- **Binding6:** anti-SD hybridization free energies of the six blocks
- **BFoldBind:** the most discriminative RNA folding and anti-SD hybridization free energies

2.4 Algorithm optimization

PGExpress is based on a gradient boosting regression algorithm (GradientBoostingRegressor) implemented in the *scikit-learn* package (Pedregosa, et al., 2011). It has been optimized considering different numbers of estimators (10, 50, 100, 200 and 500) and maximum depth values for the regression estimator (1, 3, 5, and 7). The *scikit-learn* GradientBoostingRegressor class was run using the least squares regression as loss function and the default values for all the remaining parameters.

2.5 Training and testing

To estimate the performance of *PGExpress* and the alternative methods, we performed several tests. First, we tested *PGExpress* using a gene-based 10-fold cross-validation approach on the WT-High dataset to keep all the variants belonging to the same gene in the same subset. For each test we calculated the performance using the evaluation measures defined in section 2 of the Supplementary Materials. The reported scores represent the average values obtained over five 10-fold cross-validation tests. The results obtained on the Kosuri-All (Weak-High, Mid-High and Strong-High) and Experimental (Exp-Set, Exp-Mut) datasets were calculated after removing from the training set all the data related to the genes present in the testing set. This procedure reduced the overfitting due to the presence of data from sequences with high similarity both in training and testing sets. To check for this source of bias, we also performed the all-against-all global alignments (1,558,513) among the RBS+C33 regions of all the gene variants. The global alignments of the nucleotide sequences were calculated using the *align0* algorithm from the *fasta2.0* package (Myers and Miller, 1988).

2.6 Comparison with *RBS Calculator*

For assessing the quality of our predictions we compared our results with those obtained by *RBSCalculator* (Salis, et al., 2009). For the comparison we calculated the performances of the methods both in predicting the value (regression mode) and *sign* function (binary classifier) of the log2-fold-change of the translation efficiency. The predictions of *RBSCalculator* were scaled by calculating the log2-fold-change with respect to the median value of the translation efficiency on WT-High dataset (L2TE). A further comparison of the methods evaluated their performance in predicting the log2-fold-change with respect to the wild-type ($L2TE_{wt}$). For this task we scored the performance of *PGExpress* and *RBSCalculator* as binary classifiers excluding gene variants with absolute $L2TE_{wt}$ close to zero. More details about this test are reported in section 2.4 Supplementary Materials.

2.7 Engineering new testing sequences

For validating our algorithm, we generated new sequences and measured their protein expression level. In this case, considering the protein expression level, we reduced the complexity of the experiment that did not require to measure the mRNA expression. Thus, we selected a subset of gene variants either with positive or negative log2-fold-change of protein expression (L2PE) with respect to its median value of the High-WT dataset (2,988). For checking the similarity between our experiments and those performed by Kosuri and colleagues (Goodman, et al., 2013), we measured the expression level of five randomly selected gene variants (Exp-Set) from High-WT dataset. In the next step, we generated five new sequences not included in the Kosuri-All dataset mutating at most one nucleotide in RBS or three codons in coding region. Finally, we randomly selected a set of five gene variants (Exp-Mut), three of which show a significant variation of the predicted L2PE ($|L2PE_{wt}| \geq 3$) either from positive to negative (*dapB* and *lpxK*) or negative to positive (*zipA*) and two cases (*lgt* and *murF*) where the expression level remains in the same class. The sequences of the ten tested gene variants are reported in Table S1.

2.8 Experimental protein expression measure

DNA sequences consisting of promoter, Ribosome Binding Site (RBS), and 33 coding nucleotides (including ATG start site) of five different genes were synthesized (Genscript, Piscataway, USA) with flanking AscI and NdeI restriction sites. The DNA fragments were excised from the shuttle vector and directionally cloned into the pJ251-GERC vector obtained from Addgene (Kosuri, et al., 2013). A unique EcoRI restriction site was engineered in between the 5′ region of the AscI site and the respective promoter sequence. Using the EcoRI site we identified the positive clones. Final gene variants were verified via Sanger sequencing. The correct variants were transformed in MG165 *E. coli* cells and starter cultures were grown over night at 37 °C. The next day cultures were diluted

1:1000 in 100 µL LB medium in optical quality black walled 96-well plates (PerkinElmer, Waltham, MA, USA) in quadruplicate and overlayed with 40 µL mineral oil. Bacteria were grown at 30 °C. Bacterial growth was followed by measuring the optical density at 600 nm (OD600) as proxy. The different combination of promoter, RBS, and coding region regulate the expression levels of the superfolder green fluorescent protein (sfGFP). Expression of the red fluorescent protein (mCherry) was controlled by a constitutive promoter (PLtetO-1) shared by all gene variants (Kosuri, et al., 2013). sfGFP and mCherry fluorescence levels were measured with a monochromator equipped BioTek Synergy Mx (BioTek, Winooski, USA) plate reader. Every five minutes a fluorescence measurement was performed.

3 Results

3.1 Regression and input features

The selection of the data from Kosuri and co-workers allowed us to develop a machine learning method (*PGExpress*) for predicting the log2-fold-change of the translation efficiency based on sequence information. Before performing our tests, we analyzed the Kosuri-All dataset focusing on the gene variants in the WT-High subset. This set is composed by sequences with promoter with high binding affinity (BBaJ23100) and wild-type RBSs (Ribosome Binding Sites). The choice of WT-High dataset is supported by the observation that the correlation between the level of protein and RNA expression is higher than in WT-Low dataset (Fig. S1). Indeed, the correlation coefficients between RNA and protein expression levels are 0.72 and 0.51 for the WT-High and WT-Low sets respectively. Thus, we selected the WT-High as the main reference set for estimating the predictive power of our machine learning approach. To avoid the overestimation of the performance we performed a gene-based 10-fold cross-validation test. Keeping the variants from the same gene in the same subsets, we excluded the presence of sequences with high level of identity in training and testing. Thus, we calculated the distribution of the percentage of identity (PID) between the first two blocks (RBS+C33) of the different gene variants. The Fig. S2 shows that only ∼4% of the cases the PID achieved a value between 50% and 60%.

To estimate the predictive power of the input features used in *PGExpress*, we performed a linear regression analysis and calculated the correlation coefficients between each feature and the log2-fold-change of the translation efficiency (L2TE). The Tables S3 and S4 report the correlation coefficient (r), the root mean square error (RMSE) and the mean absolute error (MAE) obtained fitting the experimental L2TE with the predicted values of RNA secondary structure and anti-Shine-Dalgarno (anti-SD) hybridization free energies. This analysis revealed that overall the free energies of the RBS+C33 sequence resulted in the highest correlation with the log2-fold-change of translation efficiency (L2TE) while, the anti-SD hybridization free energy of the RBS shows the lowest negative correlation among the binding features (Table S4).

3.2 Performance with different features

In a second step, we calculated the performance of *PGExpress* and five alternative methods including a reduced number of features. The input features for the BFolding, BBinding, Folding6, Binding6 and BFoldBind were described in the section Feature Analysis. In Table 1 we reported the scores of the previous six methods on the WT-High dataset using the gene-based 10-fold cross-validation procedure. The results revealed that the RNA folding free energy corresponding to the RBS+C33 portion of the gene variant is the most informative feature. Indeed the BFolding method with only one feature reached a correlation coefficient (r) of 0.39. When regression values are converted in binary classification predictions BFolding method achieved an overall accuracy (ACC) of 0.67 a Matthews Correlation Coefficient (MC) of 0.35 and Area Under the Receiver Operating Characteristic Curve (AUC) of 0.72.

The discriminative power of the anti-Shine-Dalgarno (anti-SD) binding free energy is much lower. This is evident by measuring the performance of the BBinding method that resulted in lowest *r* and MC. The analysis of the results of the Folding6 and Binding6 methods, which include six features of the same free energy type, do not show any substantial increase in the performances with respect to the BFolding and BBinding methods. An improvement in the performance is obtained combining the RBS+C33 RNA secondary structure free energy with the RBS anti-SD hybridization free energy. Indeed the BFoldBind method, which takes in input two features, reached a correlation coefficient 0.5 and AUC 0.76.

In *PGExpress* we merged the six RNA folding and six anti-SD hybridization free energies. The results in Table 1 show that *PGExpress* achieved a correlation coefficient 0.57, ACC 0.73, MC 0.47 and AUC 0.80 improving the *r* value and the Matthews correlation coefficient of 0.07, and the AUC of 0.04 with respect to BFoldBind. The Receiver Operating Characteristic (ROC) curves for all methods are plotted in Fig. 2. The *PGExpress* method also resulted in lowest values of root mean square error and mean absolute error with are 1.38 and 1.08 respectively. The optimal performance of the *PGExpress* algorithm is obtained considering a maximum depth 5 and 50 estimators (see section 2.4). The results of the optimization procedure are summarized in Table S5.

3.3 Performance on the Kosuri-All subsets

In the next test we focused on the performance of *PGExpress* on three datasets (Weak-High, Mid-High and Strong-High), which contain gene variants with the same 33 starting nucleotides in the coding regions (C33) but three different RBSs (Ribosome Binding Sites). Analyzing the three new datasets, we observed that the distribution of the translation efficiency (TE) in Weak-High and WT-High are similar while Mid-High and Strong-High are strongly unbalanced toward TE values higher than 2,355 (Figure S3). A summary of the performance of *PGExpress* on 4 datasets is reported in Table 2. Thus, comparing the performance on WT-High with those on the three new datasets, we observed that *PGExpress* achieved higher performance in terms of correlation coefficient (r) overall accuracy (ACC) and Matthews correlation coefficient (MC) on the Weak-High. Indeed on this dataset *PGExpress* reached r, MC and AUC of 0.66, 0.55 and 0.85 respectively. Due to the dataset unbalance, lowest *r* and highest ACC are obtained on the Strong-High dataset (Table 2).

3.4 Selecting high-quality predictions

To better characterize the performance of *PGExpress*, we scored our method filtering-out the less reliable training data and predictions in WT-High dataset. We assumed that gene variants with translation efficiency near the median (M(TE)=2,355) constitute the noisy part of the dataset. Thus, we filtered-out progressively the subset of data with absolute log2-fold-change value below a selected threshold (see section 2.3 in Supplementary Materials). The performances of *PGExpress* in binary classification mode after removing the data close to the median value are reported in Fig. 3 and Table S6. We observed that removing 42% of the gene variants with absolute log2-fold-change of the TE lower than 1, *PGExpress* reached an overall accuracy of 0.81 and an AUC of 0.87.

3.5 Comparison with *RBSCalculator*

We compared the performance of *PGExpress* with *RBSCalculator* on the WT-High dataset. The results showed that *PGExpress* reached higher correlation coefficient (*r*) and Matthews correlation (MC) than *RBSCalculator* (see Table 3). Small difference is observed in terms of Area Under the ROC Curve which is ~0.8 for both methods.

PGExpress with *RBSCalculator* were also compared calculating their performance in predicting the log2-fold-change with respect to the wild-type sequence ($L2TE_{wt}$) removing gene variants with absolute $L2TE_{wt}$ value below a given threshold. The results showed that *PGExpress* reaches an higher AUC and Matthews Correlation coefficient than *RBSCalculator* on the subset of gene variants with $|L2TE_{wt}| \leq 0.5$ (see Fig 4).

3.6 Test on in-house experimental dataset

To test the ability of *PGExpress* to predict the gene expression we performed in-house experiments with five gene variants each in the Exp-Set and Exp-Mut datasets (see methods section) and measured the protein expression using the protocol introduced by Kosuri and co-workers (Goodman *et al.*, 2013). In Fig. S4 we plotted the measures of the fluorescence associated to each gene variant normalized by the maximum level of OD600. To make a fair comparison between our results and those reported by Kosuri and collaborators, we used the median value of the protein expression level in Kosuri data as threshold for discriminating between low and high expressed gene variants. Thus, we compared the maximum value of the re-scaled fluorescence (Table S8 and Fig. S5) obtained in our experiment with the median protein expression level in the WT-High dataset (2,998). According to this assumption, we verified that for four gene variants over five (Exp-Set), our experiments match those performed by Kosuri and colleagues (Table 4). The only difference is observed for a gene variant of the *lgt* gene (lgt-23), which is classified to have a protein expression higher than the median value in the Kosuri-All dataset, whereas our experiments revealed a low protein expression level. Nevertheless the prediction of *PGExpress* agrees with the results reported in Kosuri-All dataset. Finally we evaluate the accuracy of *PGExpress* predictions in classification mode on the Exp-Mut dataset, verifying that our predictions are correct for all the five new gene variants.

A dubious prediction is represented by the variant lpxK-Mut, which is predicted to have low protein expression level and, our in-house measure of protein expression (2,996), is only few digits below the median value of protein expression (2,998). Comparing the experimental and predicted value of the log2-fold-change of protein expression, *PGExpress* achieved a correlation coefficient 0.82 and 0.85 when the predictions on Exp-Mut dataset are merged respectively with the predictions on Kosuri-All (Figure 5A) and Exp-Set (Figure 5B) datasets. Our result showed a strong correlation between *PGExpress* prediction and the experimental data in both cases. For this specific task we trained the *PGExpress* algorithm on the log2-fold-change of protein expression with respect to its median valued (L2PE) from Kosuri's dataset. The results of the optimization procedure for the prediction of the log2-fold-change of protein expression are summarized in Table S9.

4 Discussion

In this work we presented *PGExpress*, a gradient boosting regression method for predicting the log2-fold change of the translation efficiency of mRNA from predicted free energy features. The method uses the folding free energy of six sequence blocks, which represent the local and global stability of the mRNA structures. The six sequence blocks include RBS, C33, CC sequence and their combinations. Among them, the predicted folding free energy of the RBS+C33 block is the most informative feature. This is in agreement with previous findings showing that the formation of stable RNA structures around starting codon has a strong effect on translation. Our analysis shows that by adding the folding free energies of the remaining blocks, the performance of the prediction increased. This might indicate that, although other regions of the gene have an impact on translation, the structure of the 5' region constitutes the main contribution to the translation rate. For instance, the presence of a folded SD sequence near a starting codon might slow down the translation process reducing the probability of the ribosomes to bind or elongate. Accordingly, the minimum hybridization free energies were used to represent the effect of the SD sequence predicting the hybridization energy between the mRNA and the anti-SD sequence. Although the minimum hybridization free energy itself shows a weak correlation with the translation efficiency, the combination of all folding and hybridization free energies allowed to improve the performance of our predictor. This indicates that the formation of the mRNA secondary structures and the presence of SD sequences regulate translation process in a cooperating manner.

Thus, our results show that optimized version of *PGExpress* reaches a correlation coefficient of 0.57 in regression mode and an AUC of 0.80 as binary classifier. A comparative assessment of predictions revealed that *PGExpress* achieved better performances than *RBSCalculator* in the prediction of the log2-fold-change of the translation efficiency and its variation with respect to the wild-type,
Finally, we test the sensitivity of *PGExpress* to small changes in the nucleotide sequences. For this purpose we measured the expression level of five gene variants that differ in few nucleotides from the original sequences from Kosuri-All dataset. Our analysis show that *PGExpress* is able to correctly predict the expression level of the new gene variants, most of which (4/5) resulted in an opposite expression level with respect to the original sequence. Strikingly, is the case of the *dapB* variant which achieved >15-fold lower protein expression with only 2 synonymous mutations (see Tables S1 and S8). This observation confirms the robustness of our method, which supports its practical application in biotechnology. Compared with other methods that are merely focusing on the effects of UTRs, we integrated the main effecting factors from the perspective of whole sequence, which enabled us to predict translation efficiency accurately and to engineer new sequences at the whole sequence level.
Future directions of our work will include the analysis of new features to improve the prediction of the translation efficiency of wild-type genes in *E. coli*, and the development of tools for identifying key nucleotides to control protein expressions. We believe that our *in-silico* approach can have strong impact on biotechnological applications reducing the experimental effort to engineer optimized organisms.

Acknowledgements

E.C. acknowledges Genifx - Genome Informatics Service at the University of Alabama at Birmingham, AL (USA) for computational resources. We acknowledge Sriram Kosuri, George Church and collaborators for sharing their experimental data.

Funding

E.C. is supported by the FFABR grant from the Italian Ministry of Research, Education and Universities (MIUR).

Conflict of Interest: none declared.

References

Bonde, M.T.*, et al.* Predictable tuning of protein expression in bacteria. *Nat Methods* 2016;13(3):233-236.
Capriotti, E. and Marti-Renom, M.A. Computational RNA structure prediction. *Curr Bioinform* 2008;3(1):32-45.
Gingold, H. and Pilpel, Y. Determinants of translation efficiency and accuracy. *Mol Syst Biol* 2011;7:481.
Goodman, D.B., Church, G.M. and Kosuri, S. Causes and effects of N-terminal codon bias in bacterial genes. *Science* 2013;342(6157):475-479.
Kosuri, S.*, et al.* Composability of regulatory sequences controlling transcription and translation in Escherichia coli. *Proc Natl Acad Sci U S A* 2013;110(34):14024-14029.
Kozak, M. Regulation of translation via mRNA structure in prokaryotes and eukaryotes. *Gene* 2005;361:13-37.
Krivoruchko, A. and Nielsen, J. Production of natural products through metabolic engineering of Saccharomyces cerevisiae. *Curr Opin Biotechnol* 2015;35:7-15.

Kudla, G., *et al.* Coding-sequence determinants of gene expression in Escherichia coli. *Science* 2009;324(5924):255-258.
Kyle, S., *et al.* Production of self-assembling biomaterials for tissue engineering. *Trends Biotechnol* 2009;27(7):423-433.
Li, G.W., Oh, E. and Weissman, J.S. The anti-Shine-Dalgarno sequence drives translational pausing and codon choice in bacteria. *Nature* 2012;484(7395):538-541.
Lorenz, R., *et al.* ViennaRNA Package 2.0. *Algorithms Mol Biol* 2011;6:26.
Mortimer, S.A., Kidwell, M.A. and Doudna, J.A. Insights into RNA structure and function from genome-wide studies. *Nat Rev Genet* 2014;15(7):469-479.
Myers, E.W. and Miller, W. Optimal alignments in linear space. *Comput Appl Biosci* 1988;4(1):11-17.
Na, D. and Lee, D. RBSDesigner: software for designing synthetic ribosome binding sites that yields a desired level of protein expression. *Bioinformatics* 2010;26(20):2633-2634.
Pedregosa, F., *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 2011;12:2825-2830.
Plotkin, J.B. and Kudla, G. Synonymous but not the same: the causes and consequences of codon bias. *Nat Rev Genet* 2011;12(1):32-42.
Pop, C., *et al.* Causal signals between codon bias, mRNA structure, and the efficiency of translation and elongation. *Mol Syst Biol* 2014;10:770.
Reeve, B., *et al.* Predicting translation initiation rates for designing synthetic biology. *Front Bioeng Biotechnol* 2014;2:1.
Rodrigo, G. and Jaramillo, A. RiboMaker: computational design of conformation-based riboregulation. *Bioinformatics* 2014;30(17):2508-2510.
Salis, H.M. The ribosome binding site calculator. *Methods Enzymol* 2011;498:19-42.
Salis, H.M., Mirsky, E.A. and Voigt, C.A. Automated design of synthetic ribosome binding sites to control protein expression. *Nat Biotechnol* 2009;27(10):946-950.
Seo, S.W., *et al.* Predictive combinatorial design of mRNA translation initiation regions for systematic optimization of gene expression levels. *Sci Rep* 2014;4:4515.
Shine, J. and Dalgarno, L. The 3'-terminal sequence of Escherichia coli 16S ribosomal RNA: complementarity to nonsense triplets and ribosome binding sites. *Proc Natl Acad Sci U S A* 1974;71(4):1342-1346.
Shultzaberger, R.K., *et al.* Anatomy of Escherichia coli ribosome binding sites. *J Mol Biol* 2001;313(1):215-228.
Toone, E. and de Winde, H. Energy biotechnology in 2013: advanced technology development for breakthroughs in fuels and chemicals production. *Curr Opin Biotechnol* 2013;24(3):367-368.
Tuller, T., *et al.* An evolutionarily conserved mechanism for controlling the efficiency of protein translation. *Cell* 2010a;141(2):344-354.
Tuller, T., *et al.* Translation efficiency is determined by both codon bias and folding energy. *Proc Natl Acad Sci U S A* 2010b;107(8):3645-3650.

Tables

Table 1. Performance of the methods using alternative input features.

Method	r	RMSE	MAE	ACC	MC	AUC	N
BFolding	0.39	1.53	1.23	0.67	0.35	0.72	1
BBinding	0.10	1.67	1.40	0.51	0.01	0.52	1
Folding6	0.40	1.54	1.22	0.67	0.35	0.72	6
Binding6	0.08	1.73	1.43	0.52	0.04	0.54	6
BFoldBind	0.50	1.44	1.14	0.70	0.40	0.76	2
PGExpress	0.57	1.38	1.08	0.73	0.47	0.80	12

r, RMSE, MAE, ACC, MC and AUC are defined in Supplementary Materials. N is the number of input features. The input features of BFolding, BBinding, Folding6, Binding6, BFoldBind and *PGExpress* are defined in the section *Features analysis*.

Table 2. Performance of the *PGExpress* on the Kosuri-All subsets.

Dataset	r	RMSE	MAE	ACC	MC	AUC	High
WT-High	0.57	1.37	1.08	0.73	0.47	0.80	0.50
Weak-High	0.66	1.16	0.94	0.77	0.55	0.85	0.53
Mid-High	0.58	1.33	1.02	0.85	0.41	0.84	0.81
Strong-High	0.49	1.40	1.10	0.92	0.47	0.81	0.89

r, RMSE, MAE, ACC, MC and AUC are defined in Supplementary Materials. In the High column is reported the fraction of gene variants with translation efficiency higher than its median value (L2TE>0) on the WT-High dataset.

Table 3. Comparison between *PGExpress* and *RBSCalculator*.

Method	r	RMSE	MAE	ACC	MC	AUC
PGExpress	0.57	1.37	1.08	0.73	0.47	0.80
RBSCalculator	0.53	2.62	2.02	0.71	0.44	0.79

r, RMSE, MAE, ACC, MC and AUC are defined in Supplementary Materials.

Table 4. Prediction of the protein expression level for the gene variants (ID) in the Exp-Set and Exp-Mut datasets.

Dataset	ID	Kosuri-All	Experiment	Prediction	Class
Exp-Set	dapB-28	3.8	0.8	1.7	↑
	lgt-23*	2.3	-0.8	2.0	↑°
	lpxK-30	6.1	3.9	4.2	↑
	murF-21	-0.7	-1.9	-1.1	↓
	zipA-23	-1.5	-3.2	-0.8	↓
Mut-Set	dapB-Mut	-	-3.3	-1.2	↓
	lgt-Mut	-	1.9	3.2	↑
	lpxK-Mut	-	0.0	-0.3	↓*
	murF-Mut	-	-0.9	-1.0	↓
	zipA-Mut	-	0.2	3.0	↑

Kosuri-All: log2-fold-change of protein expression (L2PE) with respect to its median value (2,988) from Kosuri's dataset (Goodman, et al., 2013). Experiment: log2-fold-change calculated from protein expression levels from our in-house experiments. Prediction: predicted L2PE of protein expression from *PGExpress*. Class: Sign function of the log2-fold-change of protein expression. ↑ and ↓ represent respectively the positive and negative values of the L2PE. *Our experimental measure of the protein expression for the lgt-23 gene variant is in disagreement with data from Kosuri dataset. ° The prediction of log2-fold-change of protein expression for the lgt-23 variant is in agreement with the experimental measure from Kosuri's dataset. * The experimental value of L2PE for lpxK-Mut is slightly negative (-1e-4). The sequences of all variants are reported in Table S1. The results of the optimization procedure for predicting L2PE are reported in Table S9.

Figures

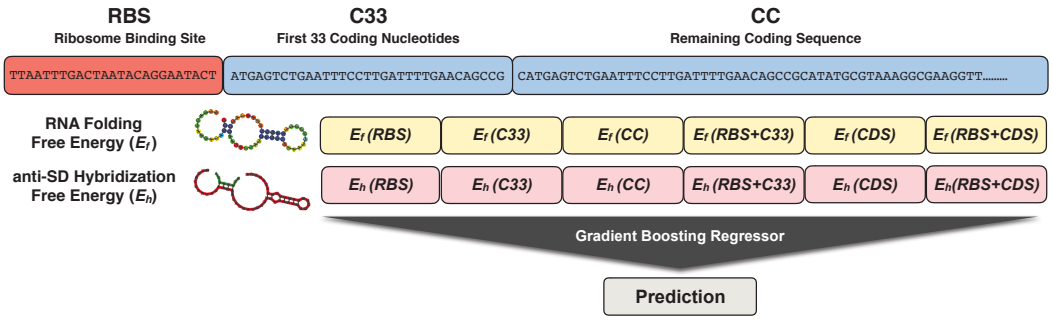


Fig. 1. Representation of the *PGExpress* algorithm. *PGExpress* input is a 12-elements vector composed by the predicted RNA secondary structure (E_f) and anti-Shine-Dalgarno hybridization (E_h) free energies per nucleotide. Each sequence is divided in 3 blocks: the Ribosome Binding Site (RBS), the first 33 nucleotides of the coding sequence (C33) and the remaining part of the coding region starting from nucleotide 34 (CC). The whole coding sequence (CDS) is obtained joining C33 and CC.

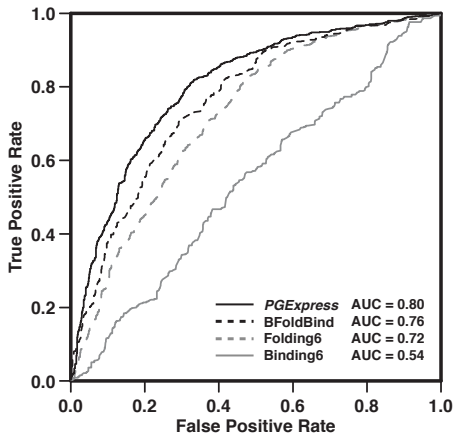


Fig. 2. ROC curves of the predictors. ROC curves *PGExpress* and alternative methods with reduced input features on the WT-High dataset.

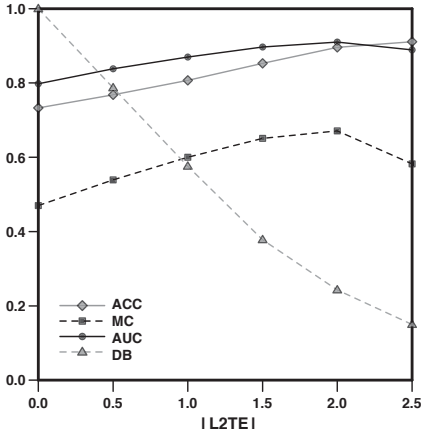


Fig. 3. Performance of the method as a function of the absolute L2TE. |L2TE|, ACC, MC and AUC are defined in Supplementary Materials. DBs the fraction of the dataset after filtering out less reliable training data.

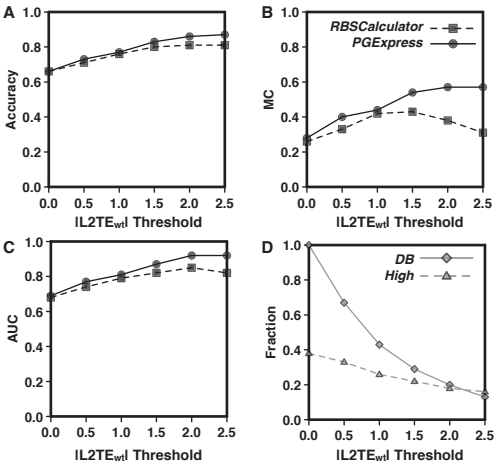


Fig. 4. Comparison of *PGExpress* and *RBSCalculator* in the prediction of the log2-fold-change with respect to the wild-type sequence ($L2TE_{wt}$). We reported the accuracy (A) the Matthews Correlation Coefficient (B) and the Area Under the ROC Curve (C) at different threshold of absolute $L2TE_{wt}$ values. In panel D we plotted the fraction of the dataset (DB) and the variants with positive $L2TE_{wt}$ values (High) at different thresholds. The data of the plot are reported in Table S7.

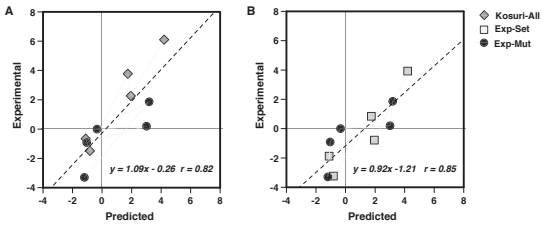


Fig. 5. Comparison of the predicted and experimental protein expression levels. In the panel A are merged the data from Exp-Mut and Kosuri-All datasets. In the panel B are shown the results on our experimental datasets (Exp-Mut and Exp-Set). The values of the root mean square error (RMSE) between predicted and experimental protein expression levels in panels A and B are 1.5 and 1.7 respectively.

SUPPLEMENTARY MATERIALS

Predicting gene expression level in *E. coli* from mRNA sequence information

Linlin Zhao[1], Nima Abedpour[2], Christopher Blum[1], Petra Kolkhof[1], Mathias Beller[3], Markus Kollmann[1,] and Emidio Capriotti[4,*]*

[1] Institute for Mathematical Modeling of Biological Systems, Department of Biology. Heinrich Heine University Düsseldorf. Universitaetsstr. 1, 40225 Düsseldorf, Germany.
[2] Department of Translational Genomics, University of Cologne, Weyertal 115b, 50931 Cologne, Germany.
[3] Systems Biology of Lipid Metabolism, Department of Biology. Heinrich Heine University Düsseldorf. Universitaetsstr. 1, 40225 Düsseldorf, Germany.
[4] BioFolD Unit, Department of Pharmacy and Biotechnology (FaBiT), University of Bologna, Via F. Selmi 3, Bologna, 40126, Italy.
*To whom correspondence should be addressed.

1. Gene expression measures

In this work we referred to different measures of gene expression. In general our method consider the Translation Efficiency (TE) as the main measure for estimating the gene expression level. The Translation Efficiency measures the level of protein expression (PE) for RNA molecule. In particular our method (*PGExpress*) has been trained to predict log2-fold-change of the translational efficiency (L2TE) with respect to its median value on the WT-High dataset (*median(TE)= 2,355*).
For making a fair comparison between *PGExpress* and *RBSCalculator* and reducing the possible differences derived from the training sets, we calculated the log2-fold-change of the translational efficiency with respect to wild-type gene sequence ($L2TE_{wt}$).
Finally, for comparing the prediction of *PGExpress* with the experimental data generated in-house we predicted the log2-fold-change of the protein expression (L2PE) with respect to its median value on the WT-High dataset (*median(PE)= 2,988*). The use of L2PE is justified by the reduced experimental effort that excludes RNA Seq experiments.

2. Performance evaluation measures

2.1 Performance in regression mode

For evaluating the performance of our regression algorithm we compared the predicted and experimental values of the log2-fold-change of the translation efficiency (L2TE) with respect to its median value (2,355) using a regression analysis.

The standard scoring values calculated in our analysis are the correlation coefficient (*r*), the root mean square error (RMSE) and the mean absolute error (MAE). They are defined as follows:

$$r = \frac{n\sum_{i=1}^{n} y_i \bar{y}_i - \left(\sum_{i=1}^{n} y_i\right)\left(\sum_{i=1}^{n} \bar{y}_i\right)}{\sqrt{\left[n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2\right]\left[n\sum_{i=1}^{n} \bar{y}_i^2 - \left(\sum_{i=1}^{n} \bar{y}_i\right)^2\right]}} \quad \text{[Eq. 1]}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(y_i - \bar{y}_i\right)^2}{n}} \quad \text{[Eq. 2]}$$

$$MAE = \frac{\sum_{i=1}^{n} \left|y_i - \bar{y}_i\right|}{n} \quad \text{[Eq. 3]}$$

where y_i and $\bar{y}_i$ are the predicted and experimental values respectively.

2.2 Performance of the binary classifier

Our regression method can be converted in binary classification algorithm dividing the dataset in two classes corresponding to positive and negative log2-fold-change of the translation efficiency with respect to its median value.

Then, we can calculated the standard performance measures for a binary classifier - assuming that positives indicate values of L2TE>0 and negatives measures with L2TE≤0 - TP (true positives) are correctly predicted gene variants with L2TE>0, TN (true negatives) are correctly predicted gene variants with L2TE≤0, FP (false positives) gene variants low translation efficiency predicted with L2TE>0, and FN (false negatives) are gene variants with high translation efficiency predicted with L2TE≤0.

Predictor performance was evaluated using the following metrics: true and false positive rates (*TPR, FPR*) and overall accuracy (*ACC*)

$$FPR = \frac{FP}{FP + TN} \quad TPR = \frac{TP}{TP + FN}$$
$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \quad \text{[Eq. 4]}$$

We computed the Matthew's correlation coefficient *MC* (Eq. 5) as:

$$MC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad \text{[Eq. 5]}$$

We also calculated the area under the receiver operating characteristic (ROC) curve (AUC), by plotting the True Positive Rate as a function of the False Positive Rate at different probability thresholds of predicting a gene variants with L2TE>0.

2.3 Performance of the binary classifier excluding noisy data

The performance of *PGExpress* is calculated removing possible noisy data. Thus, we filtered-out from our dataset the gene variants with absolute L2TE below a selected threshold. The absolute value is calculated as follows

$$|L2TE| = \left| \log_2 \left(\frac{TE}{median(TE)_{High-WT}} \right) \right| \quad \text{[Eq. 6]}$$

where TE is the translation efficiency and its median of the High–WT dataset is 2,355,

2.4 Comparison with RBSCalculator

To compare the performance of *PGExpress* with *RBSCalculator* we converted the output of *RBSCalculator* calculating the log2-fold-change of the translation efficiency (L2TE). The first test consisted in the comparison of the scoring measures reported above (*r, RMSE, MAE, ACC, MC, AUC*). The second test evaluated the performance of *PGExpress* with *RBSCalculator* in predicting the *sign* of variation of L2TE with respect to the wild-type sequence

$$|L2TE_{wt}| = \left| \log_2 \left(\frac{TE}{TE_{wt}} \right) \right| \quad \text{[Eq. 7]}$$

Similar to the analysis described in the previous section, we calculated the performance of the both methods as binary classifiers removing variants with absolute $L2TE_{wt}$ value below a given threshold.

3. Experimental data

Our Exp-Set and Exp-Mut datasets (see Table S2) consist of five gene variants each. The gene variants are composed by the Ribosome Binding Sites (RBSs) and first 33 nucleotides of the coding region (C33) reported in the following table.

ID	RBS	C33
dapB-28	TTAATATTAAAGAGGAGAAATACTAG	ATGCATGATGCCAACATCCGCGTTGCCATCGCC
dapB-Mut	TTAATATTAAAGAGGAGAAATACTAG	ATGCATGATGCCAACATCCACGTTGGCATCGCC
lgt-23	TTAATATTAAAGAGGAGAAATACTAG	ATGACGTCGAGTTATCTGCATTTTCCTGAATTT
lgt-Mut	TTAATATTAAAGAGGAGAAATACTAG	ATGACGTCGAGTTACCTGCATTTTCCTGAATTT
lpxK-30	TTAATATTAAAGAGGAGAAATACTAG	ATGATCGAAAAAATTTGGAGCGGTGAATCTCCG
lpxK-Mut	TTAATATTAAAGAGGAGAAATACTAG	ATGATCGAAAAAATTTGGTCTGGTGAATCTCCG
murF-21	TTAATCGTCTGCTGGGGGTGATTGC	ATGATTAGCGTGACGTTAAGTCAGCTTACCGAT
murF-Mut	TTAATGGTCTGCTGGGGGTGATTGC	ATGATTAGCGTGACGTTAAGTCAGCTTACCGAT
zipA-23	TTAATATTAAAGAGGAGAAATACTAG	ATGATGCAGGATCTCCGCCTGATCCTGATCATC
zipA-Mut	TTAATATTAAAGAGGAGAAATACTAG	ATGATGCAGGATCTCCGGTTAATCTTAATCATC

Table S1. Ribosome Binding Site (RBS) and the first 33 nucleotides of the coding region (C33) of the ten tested gene variants. Identifiers (ID) labeled with Mut (gray rows), which belongs to the Exp-Mut dataset, are variants of the sequences in the Exp-Set dataset (white rows).

All the gene variants are completed with the sfGFP coding sequence reported below

```
>sfGFP|superfolder GFP
CATATGCGTAAAGGCGAAGAGCTGTTCACTGGTTTCGTCACTATTCTGGTGGAACTGGAT
GGTGATGTCAACGGTCATAAGTTTTCCGTGCGTGGCGAGGGTGAAGGTGACGCAACTAAT
GGTAAACTGACGCTGAAGTTCATCTGTACTACTGGTAAACTGCCGGTACCTTGGCCGACT
CTGGTAACGACGCTGACTTATGGTGTTCAGTGCTTTGCTCGTTATCCGGACCACATGAAG
CAGCATGACTTCTTCAAGTCCGCCATGCCGGAAGGCTATGTGCAGGAACGCACGATTTCC
TTTAAGGATGACGGCACGTACAAAACGCGTGCGGAAGTGAAATTTGAAGGCGATACCCTG
GTAAACCGCATTGAGCTGAAAGGCATTGACTTTAAAGAAGACGGCAATATCCTGGGCCAT
AAGCTGGAATACAATTTTAACAGCCACAATGTTTACATCACCGCCGATAAACAAAAAAAT
GGCATTAAAGCGAATTTTAAAATTCGCCACAACGTGGAGGATGGCAGCGTGCAGCTGGCT
GATCACTACCAGCAAAACACTCCAATCGGTGATGGTCCTGTTCTGCTGCCAGACAATCAC
TATCTGAGCACGCAAAGCGTTCTGTCTAAAGATCCGAACGAGAAACGCGATCACATGGTT
CTGCTGGAGTTCGTAACCGCAGCGGGCATCACGCATGGTATGGATGAACTGTACAAATAA
```

4. Supplementary Figures

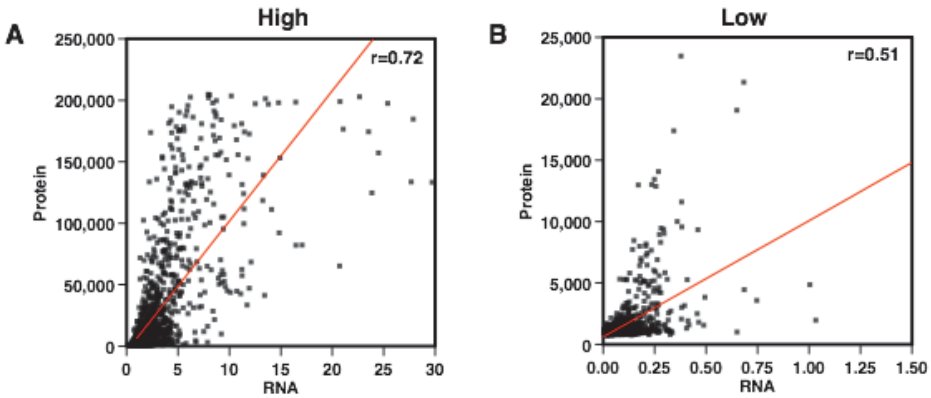


Figure S1. Correlation between RNA and Protein in the subset of gene variants including promoters with High (panel A) and Low (panel B) binding affinity. RNA and Protein levels for high and low binding affinity promoters can be extracted from Supplementary File 1.

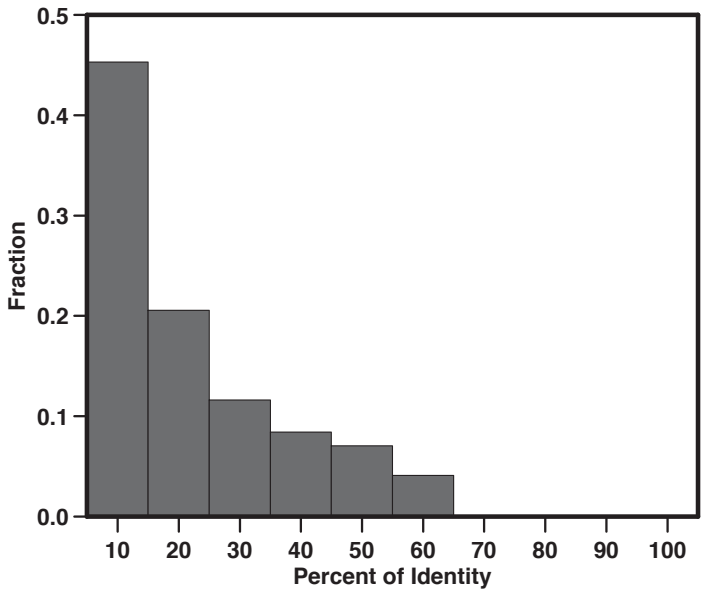


Figure S2. Distribution of the Percent of Identity between RDS+C33 regions from different genes. Total alignments: 1,558,513

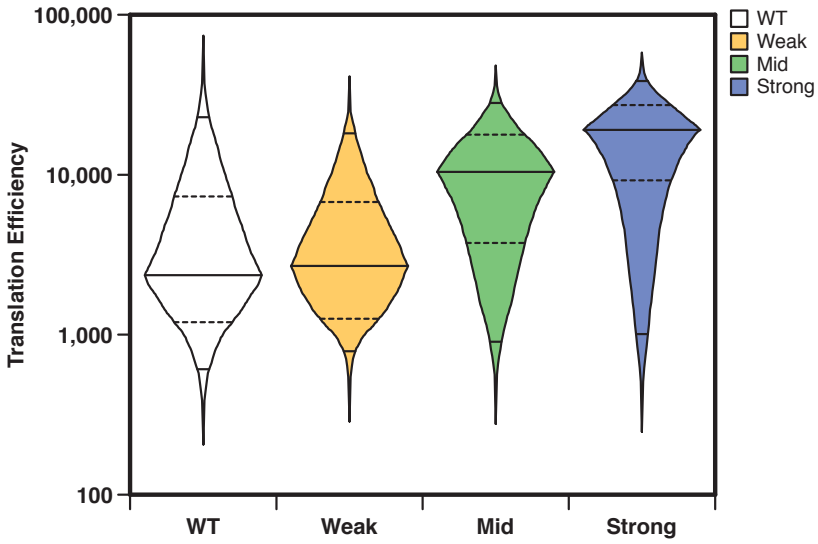


Figure S3. Distribution of the Translation Efficiency for the gene variants with WT, Weak, Mid and Strong Ribosome Binding Sites. The Translation Efficiency values are reported in Supplementary File 1.

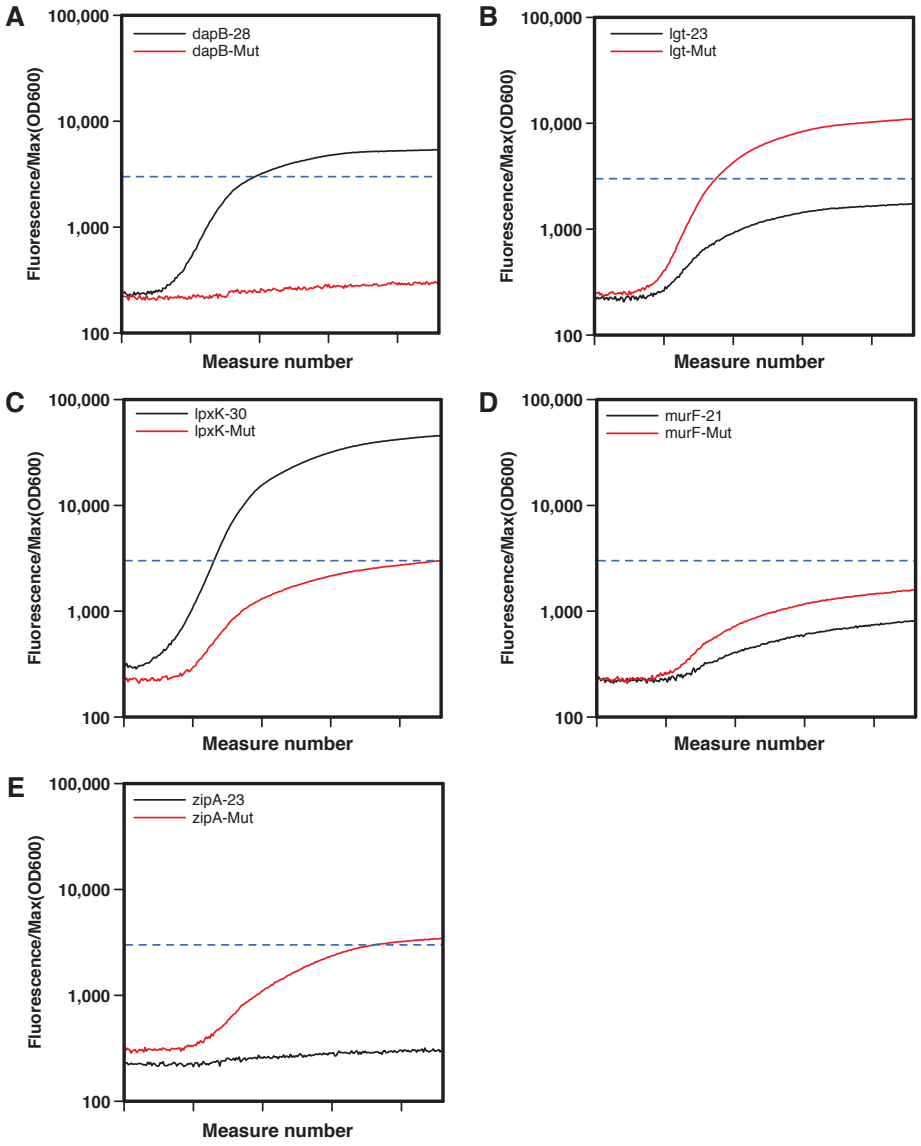


Figure S4. Experimental protein expression measures of the five gene variants in the Exp-Set dataset (black lines) and their mutants in the Exp-Mut dataset (red lines). The blue line, which represents the Fluorescence classification threshold, is set to 2,998. This threshold is the median value of the protein expression in the WT-High dataset. Florescence measures were performed each five minutes.

5. Supplementary Tables

Dataset	RBS	Promoter	N	L2TE>0	L2TE≤0
Kosuri-All	All	High/Low	14,234	12,002	2,232
High-WT	Wild-Type	High	1,772	861	861
High-Weak	Weak	High	1,771	946	825
High-Mid	Mid	High	1,766	1,438	328
High-Strong	Strong	High	1,750	1,562	188
Low-WT	Wild-Type	Low	1,763	1,761	2
Exp-Set*	Wild-Type/Strong	High	5	3(2)	2(3)
Exp-Mut	Wild-Type/Strong	High	5	2	3

Table S2. Datasets composition. The number of gene variants with negative (L2TE≤0) and positive (L2TE>0) log2-fold-change in Kosuri-All dataset (Goodman et al., 2013) and its subset are based on median value of the translation efficiency (2,355). On the remaining datasets the classification is performed using as a threshold the median values of the protein. *For Exp-Set dataset we included in parenthesis the number of variants with high and low protein expression levels - with respect to the median value of the protein expression in the High-WT subset (2,988) - of common mRNA sequences in Kosuri-All dataset

Folding	r	RMSE	MAE
RBS	0.21	1.69	1.36
C33	0.28	1.71	1.36
RBS+C33	0.42	1.71	1.35
C33+CC	0.30	1.79	1.40
RBS+CDS	0.42	1.79	1.39

Table S3. Linear regression between the predicted RNA secondary structure free energy and the log2-fold-change of the experimental translation efficiency. r, RMSE and MAE are defined above in the section *Evaluation measures*.

Binding	r	RMSE	MAE
RBS	-0.21	2.00	1.52
C33	-0.13	1.88	1.46
RBS+C33	-0.03	2.33	1.80
C33+CC	-0.07	2.33	1.80
RBS+CDS	-0.07	2.33	1.80

Table S4. Linear regression between the predicted hybridization free energy with the anti-Shine-Dalgarno RNA sequence and the log2-fold-change of the experimental translation efficiency. r, RMSE and MAE are defined above in the section 2.1.

Depth	Estimators	r	RMSE	MAE	ACC	MC	AUC
1	10	0.45	1.55	1.28	0.62	0.30	0.75
1	50	0.52	1.43	1.16	0.71	0.44	0.78
1	100	0.54	1.39	1.13	0.72	0.46	0.79
1	200	0.54	1.40	1.13	0.72	0.45	0.79
1	500	0.53	1.42	1.14	0.71	0.44	0.78
3	10	0.54	1.44	1.17	0.71	0.43	0.79
3	50	0.55	1.39	1.10	0.73	0.46	0.79
3	100	0.54	1.41	1.11	0.72	0.44	0.79
3	200	0.52	1.44	1.13	0.70	0.41	0.77
3	500	0.49	1.49	1.17	0.69	0.38	0.76
5	10	0.57	1.39	1.12	0.72	0.46	0.80
5	50	0.57	1.38	1.08	0.73	0.47	0.80
5	100	0.56	1.39	1.09	0.73	0.46	0.80
5	200	0.55	1.41	1.11	0.72	0.45	0.79
5	500	0.53	1.44	1.13	0.71	0.42	0.78
7	10	0.54	1.40	1.13	0.72	0.44	0.78
7	50	0.55	1.40	1.10	0.73	0.46	0.79
7	100	0.55	1.41	1.11	0.72	0.45	0.79
7	200	0.54	1.43	1.12	0.72	0.44	0.78
7	500	0.54	1.44	1.13	0.72	0.44	0.78

Table S5. Performances achieved by *PGExpress* in the prediction of the log2-fold-change of translation efficiency obtained with different Depth and Estimators values. r, RMSE, MAE, ACC, MC, and AUC are defined in sections 2.1 and 2.2 of the Supplementary Materials.

|L2TE|	ACC	MC	AUC	DB
0.00	0.73	0.47	0.80	1.00
0.50	0.77	0.54	0.84	0.79
1.00	0.81	0.60	0.87	0.58
1.50	0.85	0.65	0.90	0.38
2.00	0.90	0.67	0.91	0.24
2.50	0.91	0.58	0.89	0.15

Table S6 Performance of the *PGExpress* as a function of the absolute log2-fold-change of the translation efficiency (|L2TE|), which is defined in the section 2.3 of the Supplementary Materials. ACC, MC and AUC are defined in the sections 2.1 and 2.2 of the Supplementary Materials. DB is the fraction of the dataset remaining after filtering-out data close to the median of the translation efficiency.

	RBSCalculator			*PGExpress*				
$\|L2TE_{wt}\|$	ACC	MC	AUC	ACC	MC	AUC	DB	High
0.0	0.66	0.26	0.68	0.66	0.28	0.69	1.00	0.38
0.5	0.71	0.33	0.74	0.73	0.40	0.77	0.67	0.33
1.0	0.76	0.42	0.79	0.77	0.44	0.81	0.43	0.26
1.5	0.80	0.43	0.82	0.83	0.54	0.87	0.29	0.22
2.0	0.81	0.38	0.85	0.86	0.57	0.92	0.20	0.18
2.5	0.81	0.31	0.82	0.87	0.57	0.92	0.13	0.16

Table S7. Predicting the sign function of $L2TE_{wt}$ with *PGExpress* and *RBSCalculator* $|L2TE_{wt}|$: is the threshold of the absolute log2-fold-change of the translation efficiency with respect to the wild-type sequence as described above in section 2.4. ACC, MC and AUC are defined in sections 2.1 and 2.2 of the Supplementary Materials. DB is the fraction of the WT-High dataset. High is the fraction of gene variants increasing translation efficiency ($L2TE_{wt}>0$).

Dataset	ID	MF	MOD600	MF/MOD600
Exp-Set	dapB-28	4820	0.896	5382
	lgt-23	1595	0.917	1739
	lpxK-30	42183	0.927	45530
	murF-21	745	0.915	814
	zipA-23	294	0.923	318
Exp-Mut	dapB-Mut	283	0.929	305
	lgt-Mut	10103	0.925	10928
	lpxK-Mut	2725	0.909	2996
	murF-Mut	1432	0.899	1592
	zipA-Mut	2364	0.688	3438

Table S8. In house experimental measures of the Maximum Fluorescence (MF), maximum OD600 (MOD600) and their ratio (MF/MOD600) for each gene variant (ID).

Depth	Estimators	r	RMSE	MAE	ACC	MC	AUC
1	10	0.51	2.16	1.85	0.54	0.17	0.76
1	50	0.60	1.93	1.60	0.69	0.43	0.80
1	100	0.62	1.87	1.52	0.71	0.46	0.81
1	200	0.62	1.88	1.51	0.72	0.47	0.81
1	500	0.60	1.91	1.53	0.71	0.45	0.81
3	10	0.60	1.98	1.67	0.67	0.41	0.80
3	50	0.62	1.88	1.48	0.73	0.48	0.81
3	100	0.60	1.91	1.49	0.72	0.47	0.81
3	200	0.58	1.96	1.53	0.72	0.45	0.80
3	500	0.55	2.04	1.60	0.70	0.42	0.78
5	10	0.60	1.94	1.61	0.71	0.46	0.81
5	50	0.60	1.91	1.48	0.73	0.47	0.81
5	100	0.59	1.95	1.51	0.72	0.45	0.80
5	200	0.57	2.00	1.54	0.71	0.44	0.80
5	500	0.56	2.03	1.57	0.71	0.42	0.79
7	10	0.59	1.95	1.60	0.71	0.44	0.80
7	50	0.60	1.94	1.49	0.72	0.46	0.81
7	100	0.59	1.97	1.51	0.72	0.45	0.80
7	200	0.58	1.98	1.52	0.72	0.44	0.80
7	500	0.58	1.98	1.52	0.71	0.44	0.80

Table S9. Performances achieved by *PGExpress* in the prediction log2-fold-change of protein expression obtained with different Depth and Estimators values. r, RMSE, MAE, ACC, MC, and AUC are defined in the sections 2.1 and 2.2 of the Supplementary Materials.

6. Supplementary Files

Supplementary File 1: File with all Kosuri datasets. The file include the sequence code (SeqCode), the dataset, the Sequence of the Ribosome Binding Site with 5 upstream nucleotides (RBS), the 33 nucleotides of the coding sequence (CDS33), the RNA, Protein Expression and the Translation Efficiency (TranslationEfficiency).

URL: http://folding.biofold.org/pgexpress/pages/data/supplementary_file_1.txt.gz

REFERENCES

Goodman, D.B., Church, G.M. and Kosuri, S. Causes and effects of N-terminal codon bias in bacterial genes. *Science* 2013;342(6157):475-479.

5 Predicting unconventional protein secretions

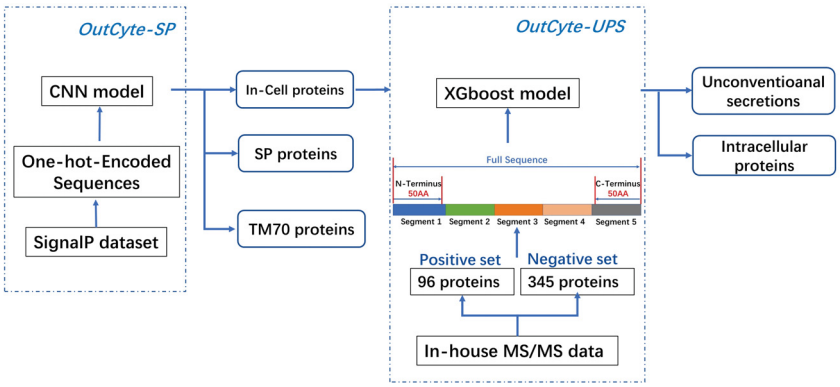
5.1 Summary

Many cells secrete proteins into the extracellular space for fulfilling different biochemical processes. One group of secreted proteins carry a signal peptide as their first part (N-Terminus), which directs the whole proteins to go through the Endoplasmic Reticulum (ER) – Golgi pathway to reach the extracellular space. Those proteins are termed as classical secretory proteins and can be computationally identified with high confidence due to the general patterns of signal peptides. However, some proteins without signal peptides were found to be secreted by different unconventional mechanisms. The lack or unawareness of clear sequence patterns of unconventional protein secretions (UPS) and the low number of known UPS make the building of predictive model difficult.

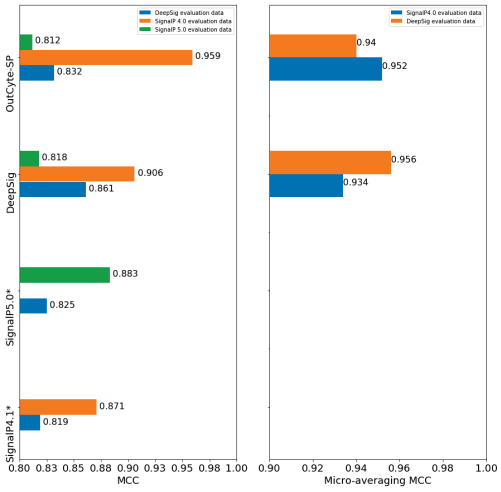
In *Section 5.2*, the existing computational tools for predicting UPS were reviewed. The tools which were dedicated to predict UPS were based on the hypothesis that all secretory proteins share common features, which enabled them to make use of the classical secretory proteins by removing their signal peptides. Other tools which were designed to predict the localizations of all proteins can also be used to predict UPS. That is, if a protein is destined to go to extracellular space and does not have a signal peptide, it is probably an UPS.

In *Section 5.3*, different from the existing tools, data sets from in-house experiments of mass spectrometry based secretomics were used to build the prediction tool *OutCyte*. As shown in Fig. 5.1a, it has two parts: the *OutCyte-SP* was established based on convolutional neural networks to distinguish proteins with or without N-terminus signal (either signal peptides or transmembrane domain); the *OutCyte-UPS* further classified if the proteins without N-terminus signals to be UPS or intracellular proteins. OutCyte-SP and OutCyte-UPS were benchmarked with their existing state-of-the-art counterparts. As shown in Fig. 5.1b and 5.1c, OutCyte-SP reached similar performances with SignalP [Pet+11] and DeepSig [Sav+17] on two independent datasets, and OutCyte-UPS outperformed its counterpart SecretomeP [Ben+04] which was the gold standard in predicting UPS on the known UPS. OutCyte was applied to screen human proteome for potential UPS (Fig. 5.1d). The first step was to filter the proteins which do not possess a N-Terminus signal with OutCyte-SP. Those proteins were further classified by OutCyte-UPS.

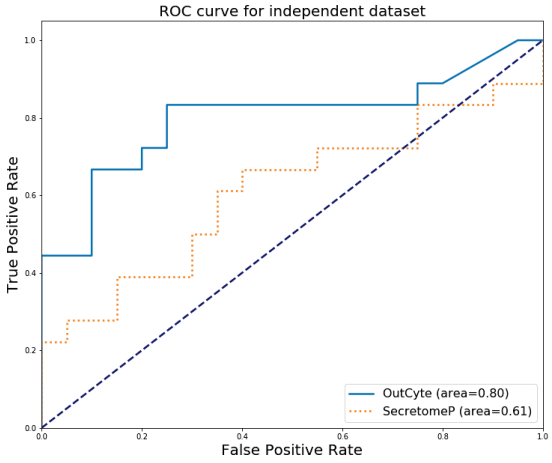
OutCyte is publicly available as a web tool at `outcyte.com`.



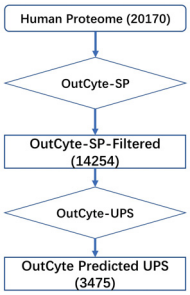
(a) OutCyte Framework



(b) OutCyte-SP VS SignalP and DeepSig.



(c) OutCyte-UPS VS SecretomeP



(d) OutCyte applied to human proteome

Fig. 5.1: The OutCyte framework is an integrated predictive tool for predicting unconventionally secreted proteins. (a). OutCyte-SP classifies input proteins into three categories: with a signal-peptide, with transmembrane-domain in the N-terminus, or none of the two classes. The proteins without N-terminal signals were further analysed by OutCyte-UPS, which has been trained on experimentally determined secreted proteins and classifies input proteins to be intracellular or unconventionally secreted. (b).The Matthews Correlation Coefficients (MCC) for signal peptides identifications of three datasets were shown in the left panel. In the right panel, micro-averaged MCC were calculated for OutCyte-SP and DeepSig on the two evaluation datasets. *SignalP5.0 training dataset overlapped with SignalP4.0's benchmark set, thus two MCCs were not included. (c). An independent data set was used for performance comparison between OutCyte-UPS and SecretomeP; (d).The OutCyte pipeline was applied on all 20170 proteins from the human proteome: OutCyte-SP classified 6077 proteins to contain either an N-teminal signal peptide or transmembrane domain. The remaining 14,254 proteins were passed to OutCyte-UPS prediction of unconventional secreted proteins. Finally, 3,475 human proteins were predicted to be unconventionally secreted.

5.2 Review: Predicting eukaryotic protein secretion without signals

Publication status

Henrik Nielsen, Eirini I. Petsalaki, Linlin Zhao, and Kai Stühler. *"Predicting eukaryotic protein secretion without signals."* Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics (2018).

Linlin Zhao's contributions

1. Reviewed the existing tools SPRED [Kan+10], Sec-GO [Hua12], and [Hun+10].
2. Summarized the data sources and availabilities of the tools dedicated to predict unconventional protein secretion.
3. Benchmarked the available tools on known unconventional protein secretions.

BBA - Proteins and Proteomics xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

BBA - Proteins and Proteomics

journal homepage: www.elsevier.com/locate/bbapap

Predicting eukaryotic protein secretion without signals

Henrik Nielsen[a,*], Eirini I. Petsalaki[a], Linlin Zhao[b], Kai Stühler[b]

[a] Department of Bio and Health Informatics, Technical University of Denmark, Kgs. Lyngby, Denmark
[b] Molecular Proteomics Laboratory, Institute of Molecular Medicine, Universitätsklinikum Düsseldorf, Düsseldorf, Germany

ARTICLE INFO

Keywords:
Protein secretion
Prediction
Machine learning
Artificial neural network
Support vector machine

ABSTRACT

Predicting unconventional protein secretion is a much harder problem than predicting signal peptide-based protein secretion, both due to the small number of examples and due to the heterogeneity and the limited knowledge of the pathways involved, especially in eukaryotes. However, the idea that secreted proteins share certain properties regardless of the secretion pathway used made it possible to construct the prediction method SecretomeP in 2004. Here, we take a critical look at SecretomeP and its successors, and we also discuss whether multi-category subcellular location predictors can be used to predict unconventional protein secretion in eukaryotes. A new benchmark shows SecretomeP to perform much worse than initially estimated, casting doubt on the underlying hypothesis. On a more positive note, recent developments in machine learning may have the potential to construct new methods which can not only predict unconventional protein secretion but also point out which parts of a sequence are important for secretion.

1. Introduction

Prediction of classical signal peptide-based protein secretion has a long history in bioinformatics, with the earliest methods being published in the 1980's [1–3]. The secretory signal peptide is probably the best known and most well-described protein sorting signal, and the large interest in signal peptide prediction is reflected by the high number of citations to the papers describing the SignalP method [4–6], which has been available online since 1996 and is currently in version 4.1 [7].

SignalP is an example of a signal-based method for protein sorting prediction, where the computational model recognizes the actual sorting signal. The two other approaches are global property-based methods and homology-based methods [8]. Global property-based methods exploit the fact that proteins in different compartments have different physicochemical properties, which is reflected in e.g. different amino acid compositions, especially regarding the surfaces of the proteins [9]. The earliest method for distinguishing between intra- and extracellular proteins based on amino acid and amino acid pair compositions was published in 1994 [10]. Homology-based methods, on the other hand, exploit the fact that proteins tend to stay in the same compartment during the course of evolution, meaning that subcellular location can often be inferred by homology to proteins with known location [11].

However, not all secreted proteins follow the “classical” signal peptide-dependent pathway. An increasing number of eukaryotic proteins have been found to be released without passing the endomembrane system, including proteins with very important functions like cytokines [12]. Such proteins will go undetected by signal peptide-dependent prediction methods such as SignalP.

When attempting to predict which proteins are secreted by unconventional “non-classical” signal peptide-independent routes, especially in eukaryotes, one is faced with two obstacles. First, the signal-based approach is not available, since it is generally not known where in the sequence the signals for secretion occur. Second, the number of experimentally confirmed data from which to build a training set is extremely small.

In bacteria, the situation is different, since there are many more examples known of signal peptide-independent secretion (rarely termed “non-classical” in bacteria). In Gram-negative bacteria, the type I, III, IV, and VI secretion pathways function without signal peptides, and in some cases, there is evidence of N-terminal or C-terminal sorting signals [8,13]. In Gram-positive bacteria, there are also a few known pathways (Wss, holin, and SecA2) [13,14]. This paper will discuss prediction of non-classical secretion in eukaryotes only; prediction in bacteria has been described elsewhere [8,14].

2. The SecretomeP method

SecretomeP is a method from 2004 [15] for predicting non-classically secreted proteins from Mammalia. It was published by our former

* Corresponding author.
E-mail address: hnielsen@bioinformatics.dtu.dk (H. Nielsen).

https://doi.org/10.1016/j.bbapap.2018.11.011
Received 1 July 2018; Received in revised form 30 October 2018; Accepted 29 November 2018

colleagues in the Center for Biological Sequence Analysis, which later was transformed into Department of Bio and Health Informatics. SecretomeP 2.0, published in 2005 [16], added the possibility for prediction in Gram-positive and Gram-negative bacteria; the mammalian part was not modified or retrained.

The authors chose a novel way to deal with the two obstacles mentioned in the introduction. The method is built upon the hypothesis that extracellular proteins share certain features regardless of the pathway used to secrete them. If this is true, it must be possible to use the large number of known classically secreted proteins to define these features and use them for prediction. Accordingly, the authors extracted a positive training set of extracellular proteins with annotated signal peptides and removed the signal peptide part of the sequence. A negative training set was extracted with subcellular location annotated as cytoplasm and/or nucleus. Both datasets included mammalian proteins only. In addition, a small additional test set of 13 human proteins known to be secreted without a signal peptide was used to evaluate the prediction.

The features were selected from a set of 16 features that were either directly calculated from the sequence (such as number of atoms, theoretical isoelectric point, or number of positively charged residues) or predicted from the sequence (such as secondary structure or phosphorylation sites). Some degree of position-specific information in features such as secondary structure or phosphorylation sites was preserved by dividing the sequence into a number of equal-sized subsequences (bins) and using the average predicted value within each bin as feature values.

The features were subsequently used as inputs to artificial neural networks, which were constructed in a "bottom-up" fashion, inspired by the ProtFun protein function prediction method [17]. First, one network was trained on each feature in isolation; then, the most promising features were combined in pairs; and again, the most promising feature pairs were selected to build up progressively larger feature combinations, until performance did not improve further. During this process, performance was always measured using five-fold cross-validation on a data set that had been homology partitioned so that no sequence in the test set had > 26% identity to any sequence in the training set.

The final network has six input features: (1) number of atoms, (2) number of positively charged residues, (3) low-complexity regions assigned by SEG [18] (in five bins), (4) transmembrane helices predicted by TMHMM [19] (in five bins), (5) subcellular localization predicted by PSORT [20], and (6) propeptide cleavage sites predicted by ProP [21] (in five bins).

That the number of atoms has predictive information is not surprising, since extracellular proteins are on average shorter than cytoplasmic and nuclear ones (Fig. 1 of the SecretomeP paper [15]). The number of positively charged residues is strongly correlated with the number of atoms; but it makes sense that it was precisely this and not the number of negatively charged residues that was selected by the network training procedure, if you consider the "positive-inside" rule of transmembrane proteins which states that positively charged residues are more frequent in the cytoplasmic loops than in the extracellular loops [22]. Accordingly, the SecretomeP authors report that the Arginine plus Lysine content is higher in intracellular than in secreted proteins.

Concerning the third feature, low-complexity regions seem to be less prevalent in secreted proteins than in intracellular proteins. This was apparently a novel observation by the SecretomeP authors.

The last three input features are more surprising. Proteins with transmembrane helices predicted by TMHMM were explicitly removed from the negative set in order to keep the network from learning the trivial fact that transmembrane proteins are not extracellular, so there should be no positive predictions by TMHMM in the data. However, the network has apparently utilized the probabilities for "inside" and "outside" given by TMHMM to help classify extracellular proteins – even though the TMHMM authors write in the instructions on their website: "Do not use the program to predict whether a non-membrane protein is cytoplasmic or not" [23].

That PSORT should be selected is also surprising, since that method by itself is not able to classify any of the 13 known human examples of non-classical secretion correctly. There are two old signal peptide predictors built into PSORT [2,3], so it is designed to predict classical secretion. But apparently, cytoplasm probability is after all slightly lower for extracellular proteins without their signal peptides than it is for cytoplasmic and nuclear proteins. The PSORT feature showed high correlation to the TMHMM feature.

Finally, the propeptides predicted by ProP are of the type recognized by members of the subtilisin/kexin-like proprotein convertase family, which is active in the secretory pathway. The surprising aspect here is that the number of predicted propeptide cleavage sites is actually *lower* in secretory proteins than in intracellular proteins. This might reflect the fact that the majority of the recognized cleavage sites are dibasic, leading to a higher number of false positive predictions in intracellular proteins due to the higher Lys + Arg content described above.

The predictive performance of SecretomeP is summarized in a Receiver Operating Characteristic (ROC) curve (Fig. 3 of the SecretomeP paper [15] – note that the curve shows false positive rate as a function of sensitivity where the convention is the exact opposite). As is remarked in the text, at a false positive rate of 5%, 40% of the positive examples are predicted. However, it is not clear whether this point on the curve corresponds to the recommended cutoff of 0.6. The reason for choosing 0.6 is unknown, and the false positive rate at this cutoff is not given.

Among the 13 known human examples of non-classical secretion, ten were positively predicted using the recommended cutoff of 0.6. A smoothed curve of the score distribution for these 13 sequences overlaps nicely with the score distribution of the positive training set (Fig. 4 of the SecretomeP paper [15]). These two observations together are taken as a confirmation of the underlying hypothesis that secreted proteins share characteristics regardless of the pathway used to secrete them.

3. Other dedicated methods

Besides SecretomeP, we are aware of five other published methods specifically designed to predict secretion without signal peptides in eukaryotes. These predictive tools have been summarized in Table 1.

Table 1
Summary of predictive tools dedicated for predicting unconventional protein secretion in eukaryotes.

Method	Year	Model	Availability	Link
SecretomeP	2004 [15]	ANN	Web and Standalone	http://www.cbs.dtu.dk/services/SecretomeP/
SRTpred	2008 [24]	SVM, homology	Web	http://crdd.osdd.net/raghava/srtpred/
SecretP (v1)	2010 [26]	SVM	Web (Error)	http://cic.scu.edu.cn/bioinformatics/secretP/ (Internal Server Error)
SPRED	2010 [31]	Random Forest	Standalone	http://www.inb.uni-luebeck.de/tools-demos/spred/spred
Hung et al.	2010 [36]	SVM	No	–
Sec-GO	2012 [34]	SVM, GO-annotations	Web (not accessible)	https://iclab.life.nctu.edu.tw/secgo (404)

Abbreviations used: ANN, Artificial Neural Network; SVM, Support Vector Machine.

Interestingly, all these methods, like SecretomeP, focus on mammalian proteins alone; no method is available for non-mammal eukaryotes. However, none of the papers actually argue for that choice or cite any references showing that non-classical secretion in mammals differs from the process in, e.g., birds, insects, fungi, or plants.

3.1. SRTpred

SRTpred from 2008 [24] used the SecretomeP dataset. In contrast to SecretomeP, the goal was not explicitly to make a predictor for non-classical secretion, but an overall predictor for secretion that did not rely on signal peptides. As the authors correctly remark, large scale genome sequencing projects sometimes assign the 5′-end of coding regions incorrectly, which can easily lead to missed signal peptides. Accordingly, the authors used a set of features that should be independent of signal peptides: 33 physicochemical properties averaged per sequence, amino acid composition, dipeptide (ungapped amino acid pair) composition, and sequence similarity to known proteins from the data set, measured by BLAST or PSI-BLAST [25].

However, the SRTpred authors chose to use the entire sequences instead of cutting off the predicted signal peptides like the SecretomeP authors did. This means that especially for short proteins, the signal peptides are allowed to influence the composition and physicochemical properties, making a direct comparison to SecretomeP performance problematic.

The SRTpred authors first tried artificial neural networks (ANNs), but found support vector machines (SVMs) to perform better. The final SRTpred method is an SVM integrating amino acid composition, dipeptide composition, and PSI-BLAST, so it is partly a homology-based method. The sensitivity of this hybrid method at a 5% false positive rate is 60%. Without the PSI-BLAST input, the corresponding rate is reported to be 44% – only slightly better than SecretomeP.

Keep in mind that when the focus is on predicting non-classically secreted proteins, the PSI-BLAST module is expected to be of little value, since the database of known proteins does not contain such proteins.

SRTpred is available as a web server, but it has the drawback of only being able to process one sequence per submission (where SecretomeP can process up to 100).

3.2. SecretP

Parallel to the development in SecretomeP, SecretP version 1 [26] is for mammalian proteins, while version 2 [27] is for bacteria. SecretP version 1 from 2010 aims to distinguish between three groups of proteins: classically secreted, non-classically secreted and non-secreted. For the first and last groups, the SecretomeP datasets were used. Unfortunately, the description of how the dataset of non-classically secreted proteins were extracted is lacking in detail. Two approaches are mentioned, where the first one is simply described thus: “Firstly, 864 mammalian proteins confirmed to route in non-classical secretory pathways were collected from Swiss-Prot through data mining”. 149 human proteins were put aside as a test set. In the second approach, a “secreted” keyword plus the *absence* of a signal peptide annotation was used in the selection. Using an absence of an annotation as a criterion is always risky, since the absence might simply reflect an incomplete annotation instead of a real absence of the feature.

The two approaches together gave a data set of 1248 non-classically secreted proteins. After homology reduction to 25% identity, there were 230 proteins left in the cross-validation set, and 92 in the exclusively human test set. Unfortunately, it is not clear whether homology reduction was only done *within* the two sets, or also *between* the cross-validation and the test set.

The features used in SecretP are amino acid composition and autocovariance of seven physicochemical properties, fused into what is known as pseudo-amino acid composition [28]. In addition, five more features are used: signal peptides (predicted by SignalP 3.0 [5]), secondary structure content (predicted by SSCP [29] from amino acid composition alone), number of positively charged residues, isoelectric point, and subcellular localization (predicted by WoLF PSORT [30]). No selection process is described; these five features are apparently chosen manually. All the features are then used as inputs to an SVM.

The cross-validated performance of SecretP is reported to be 88.79% correct in the three categories. In the “independent” human test set, 76 out of 92 were correctly predicted to be non-classically secreted (83%). SecretomeP only predicted 50 of these 92 correctly (54%). The reason for the scare quotes around “independent” is that we are not sure whether there were homologous sequences in the human test set with > 25% identity to sequences in the cross-validation set.

Like SRTpred, the SecretP web server can process only one sequence per submission. In addition, SecretP is currently broken, reporting an “internal server error” when a sequence is submitted.

3.3. SPRED

SPRED [31] is a random forest classifier for predicting both conventional and unconventional protein secretion in mammals. The authors extracted 780 extracellular proteins and 1980 intracellular proteins from UniProt by keyword searching and similarity reduction. 180 extracellular and 1380 intracellular proteins were kept as testing data. Just like in SecretomeP, the signal peptides of the proteins in the extracellular group were removed.

In total, 119 features were constructed for representing proteins, which include frequencies of amino acids in 10 functional groups and 7 physicochemical properties (hydrophobic, hydrophilic, neutral, positively charged, negatively charged, polar and non-polar amino acids), frequencies of structural elements and frequencies of short peptides and dipeptides and finally 31 physicochemical features selected from AAIndex [32]. The frequencies of amino acids were calculated both on the whole sequence level and within various structural elements. An information gain-based criterion was used to select top features for building SPRED. The top 10 features achieved 80.38% accuracy on the test data. Adding more features up until a set of 75 increased the accuracy to 82.31%. Due to the limited dataset size, keeping on adding features increased the model complexity and overfitting emerged.

The authors also constructed a set of 19 proteins which were experimentally confirmed to be non-classically secreted for comparing the predictive power of SPRED to that of SRTPred and SecretomeP. SPRED correctly predicted 15 of them to be unconventionally secreted; SecretomeP predicted 13 whereas SRTPred predicted 5 proteins. The 19 proteins are an extension of the 13 proteins used in the SecretomeP paper, however, one of the 19 proteins (CALR_HUMAN) was later found to carry a signal peptide [33]. The remaining 18 proteins and their predictions are listed in Table 2.

SPRED [31] is available as a downloadable program, but in our experience, it does not work “out of the box”. It took some guidance, kindly provided by the first author, before we could make it run.

3.4. Sec-GO

Sec-GO [34] followed a totally different approach for the prediction of unconventional protein secretion for both mammals and Gram-positive bacteria by using gene ontology (GO) annotations [35]. In order to train the GO-based SVM models and benchmark with other existing methods, the author made use of the SPRED dataset with more stringent similarity reduction of 25% identity as training and test data for mammals. Each protein was represented by 60,020 GO terms, which were encoded as a large sparse vector. A dimension reduction of GO feature space was applied due to the small dataset size. The author used the frequency difference of the same GO term between positive and negative datasets as the score for this term. This score stood for the discriminative power of the corresponding term for the positive and

Table 2
The prediction results on 18 experimentally confirmed human non-classically secreted proteins. There were originally 19 proteins in the list in the SPRED paper, but one (CALR_HUMAN) has a signal peptide with experimental evidence in UniProt and has therefore been removed. Numerical output scores are given for SecretomeP and SRTpred. "+" and "−" indicate true positive or false negative predictions of unconventional secretion, respectively.

UniProt ID	UniProt AC	SecretomeP	SRTpred	SPRED	Sec-GO
FGF1_HUMAN	P05230	0.847 (+)	−0.81 (−)	+	+
FGF2_HUMAN	P09038	0.239 (−)	0.80 (+)	+	+
IL1B_HUMAN	P01584	0.610 (+)	0.96 (+)	+	+
IL1A_HUMAN	P01583	0.551 (−)	−0.2 (−)	+	+
LEG3_HUMAN	P17931	0.770 (+)	−1.16 (−)	+	−
MIF_HUMAN	P14174	0.776 (+)	−0.91 (−)	+	+
S10A4_HUMAN	P26447	0.724 (+)	−0.55 (−)	+	+
GSTP1_HUMAN	P09211	0.545 (−)	−0.7 (−)	+	+
PRDX1_HUMAN	Q06830	0.528 (−)	−0.94 (−)	+	+
IL18_HUMAN	Q14116	0.634 (+)	−1 (−)	+	+
H4_HUMAN	P62805	0.408 (−)	−1.12 (−)	+	+
S10A2_HUMAN	P29034	0.324 (−)	−0.48 (−)	+	+
LEG1_HUMAN	P09382	0.345 (−)	−0.62 (−)	+	+
THIO_HUMAN	P10599	0.370 (−)	0.71 (+)	+	+
CNTF_HUMAN	P26441	0.653 (+)	0.02 (+)	−	+
HME2_HUMAN	P19622	0.727 (+)	−1.39 (−)	−	+
THTR_HUMAN	Q16762	0.616 (+)	−1.2 (−)	−	+
HMGB1_HUMAN	P09429	0.068 (−)	−1.2 (−)	−	+

negative datasets. Then all terms were ranked according to their scores and eventually 436 GO terms were used for the mammalian data set. The top scoring GO-terms were vectorized and fed directly to an SVM for optimizing the model. The author reported that by analyzing feature contributions, the GO term "extracellular" was the most important, which was straightforward and intuitive; however, the approach did not give indications of what factors of sequences might lead to the extracellular location, which is of central interest for predicting protein secretion.

Sec-GO achieved for all manually or automatically GO annotated proteins an accuracy of 96.7% of mammal testing data compared to that of 82.2% from SPRED. Furthermore, a benchmark among SecretomeP, SRTpred, SPRED and Sec-GO on the 19 unconventionally secreted proteins from SPRED showed that Sec-GO remained the top one (see Table 2). However, the requirement for already existing GO terms makes it hard to use the approach for novel proteins.

The Sec-GO web server is no longer found at the address given in the paper, which makes it hard to evaluate Sec-GO's efficacy on other datasets.

3.5. Hung et al. [36]

In this study [36], the authors made use of SecretomeP 1.0's dataset for training SVM models on 30 features which were selected from physicochemical properties summarized in AAIndex. The feature selection and model parameter tuning were encoded as binary genes in an inheritable bi-objective genetic algorithm, which made this work different from others. The prediction accuracies for non-secretory proteins and secretory proteins were 90.16% and 76.17%, respectively. However, the authors imposed a more stringent sequence similarity reduction to < 25% identity, which made a direct comparison of performance to SecretomeP difficult. This unnamed method has never been made available as a web server or a downloadable program.

4. Multi-location predictors

Besides methods that predict whether or not a protein is secreted, there are also several methods available which predict a larger number of subcellular locations, including "secreted" or "extracellular". Such multi-location predictors could potentially also be used to predict secretion without signal peptides. However, since the majority of secreted proteins have signal peptides, some kind of signal peptide prediction will usually be built into such methods, either implicitly or explicitly. If signal peptide prediction is essential for predicting the "secreted" or "extracellular" category, chances are not very high that the method will be useful for predicting unconventional protein secretion.

On the other hand, several of the multi-location predictors include some homology-derived features. The simplest approach is taken by the LocTree3 method [37] which directly uses the annotated subcellular location of the best hit in a PSI-BLAST search [25], while other methods use derived features of retrieved database hits such as Gene Ontology (GO) terms [35]. Such approaches could be useful in identifying non-classically secreted proteins, if they have close homologues that are known to be secreted. However, homology-based methods offer no new insights into the secretion signals or specific properties of non-classically secreted proteins, and they will have very limited chance of being able to predict the consequences of mutations affecting sorting signals because the wild-type and the variant would probably pick up the same homologues in a database search.

The predictors mentioned here are summarized in Table 3. In contrast to Table 1, this list is not meant to be complete; we have selected the most important and most used methods.

WoLF PSORT [30] is a successor to PSORT/PSORT II [20] for eukaryotic proteins. It is based on a combination of sequence-derived features and amino acid composition, integrated via a weighted version of the *k* nearest neighbours classifier. There are three features explicitly referring to signal peptides: the two old signal peptide predictors built into PSORT [2,3] and the signal peptide probability from iPSORT [38].

MultiLoc2 [39] is an SVM-based method integrating various sequence-derived features with amino acid composition, GO terms of homologues, and phylogenetic profiles. It also has explicit signal peptide prediction built into the model via the SVMTarget feature. SherLoc2 [40] extends the MultiLoc2 model by integrating also text mining of PubMed abstracts linked to the Swiss-Prot entries of retrieved homologues.

YLoc [41] is based on feature selection from a very large set of initial sequence-derived features. The selected features are subsequently

Table 3
Summary of selected predictors for multi-category protein localization in eukaryotes.

Method	Year	Model	Availability	Link
CELLO	2006 [43]	SVM	Web	http://cello.life.nctu.edu.tw/
WoLF PSORT	2007 [30]	k-NN	Web	https://wolfpsort.hgc.jp/
MultiLoc2	2009 [39]	SVM, homology	Web (not accessible)	https://abi.inf.uni-tuebingen.de/Services (temporarily disabled)
SherLoc2	2009 [40]	SVM, homology	Web (not accessible)	https://abi.inf.uni-tuebingen.de/Services (temporarily disabled)
YLoc	2010 [41]	SVM, homology (optional)	Web (not accessible)	https://abi.inf.uni-tuebingen.de/Services (temporarily disabled)
iLoc-Euk	2011 [42]	k-NN, homology	Web	http://www.jci-bioinfo.cn/iLoc-Euk
LocTree3	2014 [37]	Homology, profile kernel SVM	Web and Standalone	https://rostlab.org/services/loctree3/
SubCons	2017 [45]	Consensus	Web and Standalone	http://subcons.bioinfo.se/
DeepLoc	2017 [46]	Deep ANN	Web and Standalone	http://www.cbs.dtu.dk/services/DeepLoc/

Abbreviations used: ANN, Artificial Neural Network; k-NN, *k* Nearest Neighbours; SVM, Support Vector Machine.

combined via a naïve Bayes model, which makes it possible to indicate for each prediction which features were important. Certain of the selected features are clearly correlated to signal peptides. YLoc can optionally include GO terms of homologues.

iLoc-Euk [42] is a *k* nearest neighbours-based method mainly using GO terms of homologues, with additional evolutionary profiles used only in those cases where no homologues are found, or when the found homologues do not have GO annotation.

CELLO [43] is an SVM-based method that neither uses homology information nor has a built-in signal peptide model. It uses amino acid composition, amino acid pair composition, and n-peptide composition with reduced alphabets. However, it does also use partitioned amino acid composition, where the sequence is divided into a number of subsequences of equal length (like the bins of SecretomeP) and amino acid composition is calculated separately in each partition; a measure which could be influenced by the presence of signal peptides.

LocTree3 [37], as already mentioned, uses a direct transfer of subcellular location annotation from the best PSI-BLAST hit, if the significance of the hit is better than a certain *E*-value threshold. If this is not the case, it reverts to LocTree2 [44], which is an SVM-based method using a so-called profile kernel, a kind of string kernel based on sequence profiles found in a PSI-BLAST search. It is not easy to say whether the profile kernel approach recognizes signal peptides.

SubCons [45] is a consensus method incorporating predictions from CELLO, LocTree2, MultiLoc2 and SherLoc2. Its performance has been optimised on a set of human proteins, but it can be used for other eukaryotes also.

Finally, DeepLoc [46] is a method based on deep learning (convolutional and recurrent neural networks) without using annotation of homologues. It does not have an explicit signal peptide model, but it is apparent from the attention score output that it does seem to look specifically at the signal peptide region when predicting extracellular proteins.

5. A critical re-evaluation of SecretomeP performance

In the years since SecretomeP was first developed a lot more data has become available for protein sequences that are secreted in a non-classical manner. In addition, one common question addressed to the curators of the SecretomeP web service is whether it performs equally well for all eukaryotic sequences as it does for mammalian sequences. As such, an opportunity has presented itself for a critical reevaluation of SecretomeP's performance.

We collected two data sets from UniProt, one with mammalian protein sequences and one with eukaryotic sequences excluding mammalian. For each data set, the positive sub-set consisted of manually reviewed secreted sequences that lacked signal peptide annotation, and the negative sub-set consisted of protein sequences experimentally verified to be located in the cytoplasm or the nucleus, also lacking signal peptide annotation. Additional filtering was performed by excluding protein sequences that appeared to be fragments (not starting with a methionine) and sequences that were predicted to have a signal peptide by SignalP-3.0. The final data sets consisted of 543 non-classically secreted and 5997 non-secreted mammalian protein sequences, and 236 non-classically secreted and 7262 non-secreted eukaryotic (excluding mammalian) protein sequences.

In Table 4 you can see the performance of SecretomeP on these two data sets when using the recommended threshold (0.6) for the NN-score. As expected, SecretomeP performs better for mammalian sequences than other eukaryotic sequences, although the differences are not very significant. The most surprising is the low sensitivity and high false positive rate (> 20%) in both cases.

This is underlined by their Receiving Operating Characteristic (ROC) curves (Fig. 1). A ROC curve is made by varying the threshold for regarding a prediction as positive and plotting the ensuing sensitivity as a function of the false positive rate. The area under the curve (AUC) can then be used as a threshold-independent measure of predictive performance. The AUC can be directly interpreted as the probability that a randomly chosen positive example will score higher than a randomly chosen negative example. A perfect prediction will have AUC = 1,

Table 4
Performance of SecretomeP, SRTpred, and SPRED on two data sets comprising mammalian proteins and other eukaryotic proteins, respectively. AUC could not be calculated for SRTpred and SPRED since they do not provide numeric output values. The value marked by "*" is an estimate based on a subset of the negative data, since we had technical problems running SRTpred on the whole dataset. For the same reason, the TNR could not be calculated for the other eukaryotic proteins (marked "N/A*").

Data	Method	TPR	TNR	AUC
Mammalia	SecretomeP	42.9%	78.7%	0.61
	SRTpred	38.3%	86.7%*	–
	SPRED	48.6%	86.0%	–
Eukaryota (excl. Mammalia)	SecretomeP	35.6%	75.0%	0.60
	SRTpred	35.2%	N/A *	–
	SPRED	52.5%	80.0%	–

Abbreviations used: TPR, True Positive Rate (Sensitivity); TNR, True Negative Rate (Specificity); AUC, Area Under the receiver operating characteristic Curve.

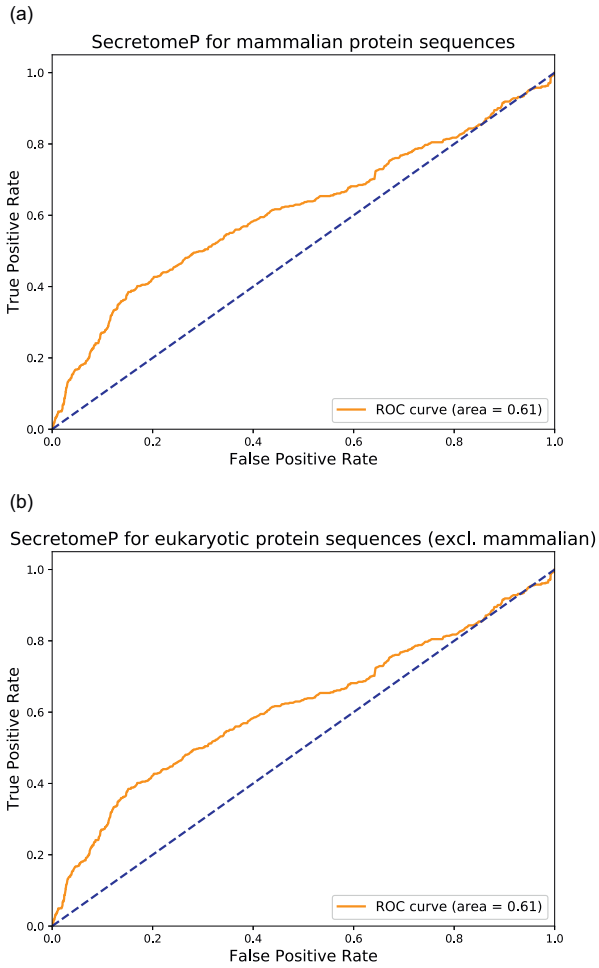


Fig. 1. ROC curves (true positive rate vs. false positive rate) for SecretomeP predictions on new datasets. Panel A: mammalian protein sequences; panel B: eukaryotic protein sequences, excluding Mammalia. The dashed lines represent the theoretical performance of a random guess.

while a random guess will give AUC = 0.5. The AUC of the SecretomeP ROC curves in Fig. 1 is for both around 0.6, indicating a low discriminatory ability, only slightly better than a random classifier.

It should be noted that the performance was significantly better when truncated proteins were included (i.e. proteins not starting with methionine), with the ROC AUC equal to 0.67 and 0.78 for mammalian and other eukaryotic sequences respectively. This suggests that these proteins are in fact secreted in a classical manner, but the signal peptide in the N-terminus has been removed. It also leads to the conclusion that there are faults in the initial hypothesis behind the design of the SecretomeP, and that proteins that are secreted in a non-classical manner do not share as many features with the classically secreted proteins as initially considered.

The performances of SRTpred and SPRED are also shown in Table 4. SRTpred predicts slightly fewer mammalian non-classically secreted proteins than SecretomeP, but at a lower false positive rate. SPRED seems to be a bit better than the other two, detecting around half of the positive examples, but it still has a false positive rate of 14% for mammalian and 20% for non-mammalian proteins. It was not possible to draw ROC curves and calculate AUC values for SRTpred and SPRED, since they don't provide numeric output.

In addition to the three dedicated servers, we also benchmarked three of the multi-location predictors discussed in the previous section. In this analysis, we did not divide the data into Mammalia and other eukaryotes, since the predictors are trained on all eukaryotes together. The results are shown in Table 5.

From the table, it is apparent that CELLO and DeepLoc, using their default output, identify fewer non-classically secreted proteins than SecretomeP at the default threshold, but at a much lower false positive rate. DeepLoc thus identifies 11% of the secreted proteins practically without false positives. If we instead of using the most probable class from the methods use the numerical score for the "Extracellular" category, we can calculate ROC curves (shown in Fig. 2), which show that CELLO and DeepLoc are actually better than SecretomeP in predicting non-classically secreted proteins.

Judged from the table, iLoc-Euk is even better, identifying almost half of the non-classically secreted proteins at a false positive rate of < 2%. This is surprising, since DeepLoc is reported to be better than iLoc-Euk in general [46]. However, it should be kept in mind that iLoc-Euk is a homology-based method, retrieving GO terms of homologues from a UniProt-derived database, and there may be a considerable overlap between that database and our test set. For the same reason, we did not benchmark LocTree3, since many of the predictions from that tool would be simple database retrievals of the annotations contained in our test set.

We would have liked to benchmark more methods, but time constraints and technical difficulties did not allow it. As an example, all the University of Tübingen servers (MultiLoc2, SherLoc2 and YLoc) were temporarily down at the time of writing, and although the text on the website says "We'll be back in a few days", this was still the case at the time of revision.

Table 5
Performance of three multi-category localization predictors on our new set of non-classically secreted proteins. Data were eukaryotic sequences (mammalian and non-mammalian combined). AUC could not be calculated for iLoc-Euk since it does not provide numeric output values. The value marked by "*" is an estimate based on a subset of the negative data, since we had technical problems running iLoc-Euk on the whole dataset.

Method	TPR	TNR	AUC
CELLO	18.0%	97.2%	0.73
DeepLoc	10.9%	99.8%	0.72
iLoc-Euk	48.4%	98.6%*	–

Abbreviations used: TPR, True Positive Rate (Sensitivity); TNR, True Negative Rate (Specificity); AUC, Area Under the receiver operating characteristic Curve.

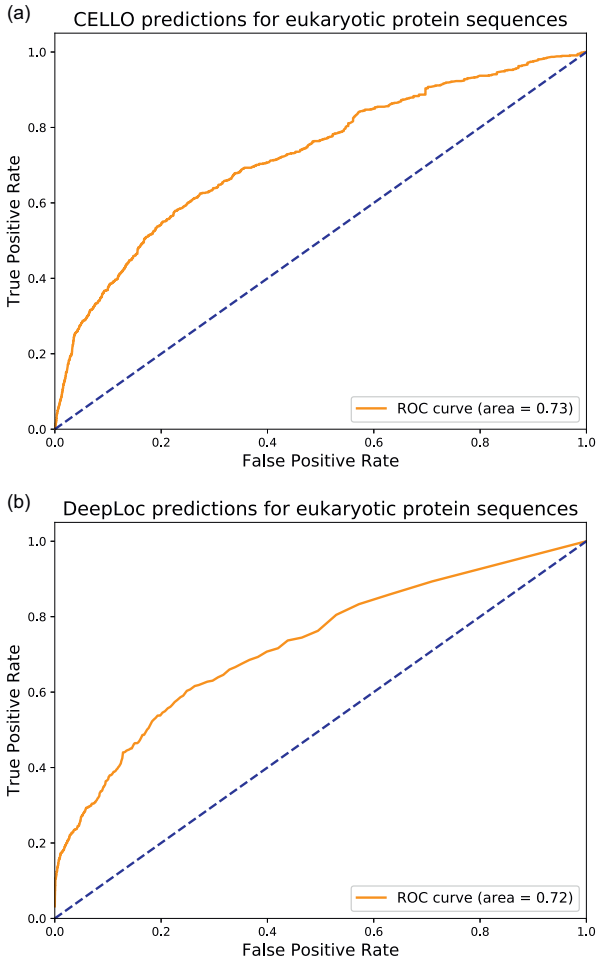


Fig. 2. ROC curves (true positive rate vs. false positive rate) for those multi-location predictors where it was possible to get numeric scores. Data were eukaryotic sequences (mammalian and non-mammalian combined). Panel A shows results for CELLO, while panel B shows results for DeepLoc. The dashed lines represent the theoretical performance of a random guess.

6. Discussion

SecretomeP version 1 was, for its time, a bold and innovative suggestion for how to construct a predictor for secretion without signal peptides. It has been cited > 800 times according to Google Scholar, and it is still being used extensively. However, its performance, measured on new independent data, is not nearly as good as we thought it would be, and the underlying hypothesis that extracellular proteins share features independent of the secretion pathway must be called into question.

SRTpred and SPRED do not represent real alternatives for predicting non-classical secretion, as they are built on the same questionable hypothesis and only perform marginally better. The small performance gain shown by SRTpred may even be attributed to the fact that the signal peptides were not removed in the training. SPRED seems to represent a more genuine performance gain, but is still limited by the constraints of the "common feature" hypothesis. Sec-GO represents an interesting analysis, but is not applicable in practice to the situation where a predictor would be most important, namely newly sequenced genomes.

SecretP, on the other hand, might be significantly better than SecretomeP, SRTpred and SPRED, but it is difficult to say how much

confidence should be put in their data set. Given more time, the set of 864 mammalian proteins, available from the SecretP website, should be critically examined. Unfortunately, it was not possible for us to benchmark SecretP due to the restrictions on the website (1 sequence per submission, 50 sequences per day) and the fact that the webserver reports an error when you attempt to run it.

The three multi-location predictors that we benchmarked performed better than SecretomeP, even though they were not made with non-classical secretion in mind, but the performance is still not high. The best of them, iLoc-Euk, may have an inflated performance due to overlap between its database and our benchmark dataset. All in all, it is fair to say that prediction of non-classical (signal peptide-independent) secretion in eukaryotes is an unsolved problem.

However, the novel deep learning techniques (convolutional and recurrent neural networks) used in DeepLoc [46] has shown promising results for predicting signal peptide-independent secretion in bacteria (E. I. Petsalaki, J. J. Almagro Armenteros, O. Winther and H. Nielsen, unpublished results). In the future, we will apply this approach to eukaryotic non-classical secretion as well. The networks in DeepLoc have, in addition to the convolutional and LSTM (long short-term memory) recurrent layers, a so-called attention layer which calculates a relative weight for each position in the sequence. These attention weights can, for each prediction, point out which parts of the sequence were important for reaching that particular prediction. In this way, a deep neural network trained on non-classically secreted proteins could not only give a prediction of secretion but also help localizing possible signals for non-classical secretion in the sequence.

Acknowledgements

The corresponding author is paid by the Technical University of Denmark. The authors (LZ and KS) thank the research commission of the University Hospital Düsseldorf for funding (FOKO 2018-27). The authors wish to thank Krishna Kumar Kandaswamy from the SPRED team for assistance with running the program.

References

[1] G. von Heijne, Patterns of amino acids near signal-sequence cleavage sites, Eur. J. Biochem. 133 (1983) 17–21, https://doi.org/10.1111/j.1432-1033.1983.tb07424.x.
[2] D.J. McGeoch, On the predictive recognition of signal peptide sequences, Virus Res. 3 (1985) 271–286, https://doi.org/10.1016/0168-1702(85)90051-6.
[3] G. von Heijne, A new method for predicting signal sequence cleavage sites, Nucleic Acids Res. 14 (1986) 4683–4690, https://doi.org/10.1093/nar/14.11.4683.
[4] H. Nielsen, S. Brunak, J. Engelbrecht, G. von Heijne, Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites, Protein Eng. 10 (1997) 1–6, https://doi.org/10.1093/protein/10.1.1.
[5] J.D. Bendtsen, H. Nielsen, G. von Heijne, S. Brunak, Improved prediction of signal peptides: SignalP 3.0, J. Mol. Biol. 340 (2004) 783–795, https://doi.org/10.1016/j.jmb.2004.05.028.
[6] T.N. Petersen, S. Brunak, G. von Heijne, H. Nielsen, SignalP 4.0: discriminating signal peptides from transmembrane regions, Nat. Meth. 8 (2011) 785–786, https://doi.org/10.1038/nmeth.1701.
[7] H. Nielsen, Predicting secretory proteins with SignalP, in: D. Kihara (Ed.), Protein Funct. Predict, Springer, New York, 2017, pp. 59–73, , https://doi.org/10.1007/978-1-4939-7015-5_6.
[8] H. Nielsen, Protein sorting prediction, in: L. Journet, E. Cascales (Eds.), Bact. Protein Secret. Syst, Humana Press, New York, NY, 2017, pp. 23–57, , https://doi.org/10.1007/978-1-4939-7033-9_2.
[9] M.A. Andrade, S.I. O'Donoghue, B. Rost, Adaptation of protein surfaces to subcellular location, J. Mol. Biol. 276 (1998) 517–525, https://doi.org/10.1006/jmbi.1997.1498.
[10] H. Nakashima, K. Nishikawa, Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies, J. Mol. Biol. 238 (1994) 54–61, https://doi.org/10.1006/jmbi.1994.1267.
[11] R. Nair, B. Rost, Sequence conserved for subcellular localization, Protein Sci. 11 (2002) 2836–2847, https://doi.org/10.1110/ps.0207402.
[12] E.H. Duitman, Z. Orinska, S. Bulfone-Paus, Mechanisms of cytokine secretion: a portfolio of distinct pathways allows flexibility in cytokine activity, Eur. J. Cell Biol. 90 (2011) 476–483, https://doi.org/10.1016/j.ejcb.2011.01.010.
[13] M. Desvaux, M. Hébraud, R. Talon, I.R. Henderson, Secretion and subcellular localizations of bacterial proteins: a semantic awareness issue, Trends Microbiol. 17 (2009) 139–145, https://doi.org/10.1016/j.tim.2009.01.004.
[14] H. Nielsen, Predicting subcellular localization of proteins by bioinformatic algorithms, in: F. Bagnoli, R. Rappuoli (Eds.), Protein Export Gram-Posit. Bact, Springer, Cham, 2016, pp. 129–158, , https://doi.org/10.1007/82_2015_5006.
[15] J.D. Bendtsen, L.J. Jensen, N. Blom, G. von Heijne, S. Brunak, Feature-based prediction of non-classical and leaderless protein secretion, Protein Eng. Des. Sel. 17 (2004) 349–356, https://doi.org/10.1093/protein/gzh037.
[16] J.D. Bendtsen, L. Kiemer, A. Fausbøll, S. Brunak, Non-classical protein secretion in bacteria, BMC Microbiol. 5 (2005) 58, https://doi.org/10.1186/1471-2180-5-58.
[17] L.J. Jensen, R. Gupta, N. Blom, D. Devos, J. Tamames, C. Kesmir, H. Nielsen, H.H. Staerfeldt, K. Rapacki, C. Workman, C.A.F. Andersen, S. Knudsen, A. Krogh, A. Valencia, S. Brunak, Prediction of human protein function from post-translational modifications and localization features, J. Mol. Biol. 319 (2002) 1257–1265, https://doi.org/10.1016/S0022-2836(02)00379-0.
[18] J.C. Wootton, S. Federhen, Statistics of local complexity in amino acid sequences and sequence databases, Comput. Chem. 17 (1993) 149–163, https://doi.org/10.1016/0097-8485(93)85006-X.
[19] A. Krogh, B. Larsson, G. von Heijne, E.L.L. Sonnhammer, Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes, J. Mol. Biol. 305 (2001) 567–580, https://doi.org/10.1006/jmbi.2000.4315.
[20] K. Nakai, P. Horton, PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization, Trends Biochem. Sci. 24 (1999) 34–35, https://doi.org/10.1016/S0968-0004(98)01336-X.
[21] P. Duckert, S. Brunak, N. Blom, Prediction of proprotein convertase cleavage sites, Protein Eng. Des. Sel. 17 (2004) 107–112, https://doi.org/10.1093/protein/gzh013.
[22] G. von Heijne, Membrane protein structure prediction: Hydrophobicity analysis and the positive-inside rule, J. Mol. Biol. 225 (1992) 487–494, https://doi.org/10.1016/0022-2836(92)90934-C.
[23] TMHMM 2.0 Guide, https://www.cbs.dtu.dk/services/TMHMM/TMHMM2.0b.guide.php Accessed 10 June 2018, (n.d.).
[24] A. Garg, G.P.S. Raghava, A machine learning based method for the prediction of secretory proteins using amino acid composition, their order and similarity-search, In Silico Biol. 8 (2008) 129–140.
[25] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D.J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25 (1997) 3389, https://doi.org/10.1093/nar/25.17.3389.
[26] L. Yu, Y. Guo, Z. Zhang, Y. Li, M. Li, G. Li, W. Xiong, Y. Zeng, SecretP: a new method for predicting mammalian secreted proteins, Peptides 31 (2010) 574–578, https://doi.org/10.1016/j.peptides.2009.12.026.
[27] L. Yu, Y. Guo, Y. Li, G. Li, M. Li, J. Luo, W. Xiong, W. Qin, SecretP: identifying bacterial secreted proteins by fusing new features into Chou's pseudo-amino acid composition, J. Theor. Biol. 267 (2010) 1–6, https://doi.org/10.1016/j.jtbi.2010.08.001.
[28] K.-C. Chou, Prediction of protein cellular attributes using pseudo-amino acid composition, Proteins 43 (2001) 246–255, https://doi.org/10.1002/prot.1035.
[29] F. Eisenhaber, F. Imperiale, P. Argos, C. Frömmel, Prediction of secondary structural content of proteins from their amino acid composition alone. I. New analytic vector decomposition methods, Proteins Struct. Funct. Bioinforma. 25 (1996) 157–168.
[30] P. Horton, K.-J. Park, T. Obayashi, N. Fujita, H. Harada, C.J. Adams-Collier, K. Nakai, WoLF PSORT: protein localization predictor, Nucleic Acids Res. 35 (2007) W585–W587, https://doi.org/10.1093/nar/gkm259.
[31] K.K. Kandaswamy, G. Pugalenthi, E. Hartmann, K.-U. Kalies, S. Möller, P.N. Suganthan, T. Martinetz, SPRED: a machine learning approach for the identification of classical and non-classical secretory proteins in mammalian genomes, Biochem. Biophys. Res. Commun. 391 (2010) 1306–1311, https://doi.org/10.1016/j.bbrc.2009.12.019.
[32] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, M. Kanehisa, AAindex: amino acid index database, progress report 2008, Nucleic Acids Res. 36 (2008) D202–D205, https://doi.org/10.1093/nar/gkm998.
[33] G. Houen, C. Koch, Human placental calreticulin: purification, characterization and association with other proteins, Acta Chem. Scand. Cph. Den. 48 (1994) (1989) 905–911.
[34] W.-L. Huang, Ranking Gene Ontology terms for predicting non-classical secretory proteins in eukaryotes and prokaryotes, J. Theor. Biol. 312 (2012) 105–113, https://doi.org/10.1016/j.jtbi.2012.07.027.
[35] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, M.A. Harris, D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin, G. Sherlock, Gene Ontology: tool for the unification of biology, Nat. Genet. 25 (2000) 25–29, https://doi.org/10.1038/75556.
[36] C.-H. Hung, H.-L. Huang, K.-T. Hsu, S.-J. Ho, S.-Y. Ho, Prediction of non-classical secreted proteins using informative physicochemical properties, Interdisc. Sci. Comput. Life Sci. 2 (2010) 263–270, https://doi.org/10.1007/s12539-010-0023-z.
[37] T. Goldberg, M. Hecht, T. Hamp, T. Karl, G. Yachdav, N. Ahmed, U. Altermann, P. Angerer, S. Ansorge, K. Balasz, M. Bernhofer, A. Betz, L. Cizmadija, K.T. Do, J. Gerke, R. Greil, V. Joerdens, M. Hastreiter, K. Hembach, M. Herzog, M. Kalemanov, M. Kluge, A. Meier, H. Nasir, U. Neumaier, V. Prade, J. Reeb, A. Sorokoumov, I. Troshani, S. Vorberg, S. Waldraff, J. Zierer, H. Nielsen, B. Rost, LocTree3 prediction of localization, Nucleic Acids Res. 42 (2014) W350–W355, https://doi.org/10.1093/nar/gku396.
[38] H. Bannai, Y. Tamada, O. Maruyama, K. Nakai, S. Miyano, Extensive feature detection of N-terminal protein sorting signals, Bioinformatics 18 (2002) 298–305, https://doi.org/10.1093/bioinformatics/18.2.298.
[39] T. Blum, S. Briesemeister, O. Kohlbacher, MultiLoc2: integrating phylogeny and

Gene Ontology terms improves subcellular protein localization prediction, BMC Bioinformatics. 10 (2009) 274, https://doi.org/10.1186/1471-2105-10-274.
[40] S. Briesemeister, T. Blum, S. Brady, Y. Lam, O. Kohlbacher, H. Shatkay, SherLoc2: a high-accuracy hybrid method for predicting subcellular localization of proteins, J. Proteome Res. 8 (2009) 5363–5366, https://doi.org/10.1021/pr900665y.
[41] S. Briesemeister, J. Rahnenführer, O. Kohlbacher, Going from where to why—interpretable prediction of protein subcellular localization, Bioinformatics 26 (2010) 1232–1238, https://doi.org/10.1093/bioinformatics/btq115.
[42] K.-C. Chou, Z.-C. Wu, X. Xiao, iLoc-Euk: a multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins, PLoS One 6 (2011) e18258, , https://doi.org/10.1371/journal.pone.0018258.
[43] C.-S. Yu, Y.-C. Chen, C.-H. Lu, J.-K. Hwang, Prediction of protein subcellular localization, Proteins 64 (2006) 643–651, https://doi.org/10.1002/prot.21018.
[44] T. Goldberg, T. Hamp, B. Rost, LocTree2 predicts localization for all domains of life, Bioinformatics 28 (2012) i458–i465, https://doi.org/10.1093/bioinformatics/bts390.
[45] M. Salvatore, P. Warholm, N. Shu, W. Basile, A. Elofsson, SubCons: a new ensemble method for improved human subcellular localization predictions, Bioinformatics 33 (2017) 2464–2470, https://doi.org/10.1093/bioinformatics/btx219.
[46] J.J. Almagro Armenteros, C.K. Sønderby, H. Nielsen, O. Winther, DeepLoc: prediction of protein subcellular localization using deep learning, Bioinformatics 33 (2017) 3387–3395, https://doi.org/10.1093/bioinformatics/btx431.

5.3 OutCyte: a novel tool for predicting unconventional protein secretions

Publication status

Linlin Zhao, Gereon Poschmann, Daniel Waldera-Lupa, Nima Rafiee, Markus Kollmann, and Kai Stühler. *"OutCyte: a novel tool for predicting unconventional protein secretion", accepted by Scientific Reports*

Linlin Zhao's contributions

1. Processed the protein data by merging data from different mass spectrometry experiments, homology reduction and preparing training and testing datasets.
2. Generated sequence-based features for the proteins by understanding secretion mechanisms, literature mining, and discussions with Dr. Kai Stühler, Dr. Gereon Poschmann and Dr. Daniel Waldera-Lupa.
3. Developed the model OutCyte-UPS based on the sequence features using XGBoost algorithms.
4. Developed the model OutCyte-SP for predicting proteins with N-terminus signals based on convolutional neural networks.
5. Applied the established models to different independent datasets for benchmarking.
6. Designed the frontend and partly developed the backend of the OutCyte web server.
7. Wrote the manuscript draft.

OutCyte: a novel tool for predicting unconventional protein secretion

Linlin Zhao[1,2], Gereon Poschmann[1], Daniel Waldera-Lupa[1], Nima Rafiee[2], Markus Kollmann[2], Kai Stühler[1,3]*

[1]Institute of Molecular Medicine, Medical Faculty, Heinrich-Heine-University, Düsseldorf, Germany
[2]Mathematical Modelling of Biological Systems, Heinrich-Heine-University, Düsseldorf, Germany
[3] Molecular Proteomics Laboratory, BMFZ, Heinrich-Heine-University, Düsseldorf, Germany
*Correspondence: kai.stuehler@hhu.de (K. Stühler)

Abstract

The prediction of protein localization such as the extracellular space from high-throughput data is essential for functional downstream inferences. It is well accepted that a part of secreted proteins go through the classical Endoplasmic Reticulum – Golgi pathway with the guidance of a signal peptide. However, a large number of proteins have been found to reach the extracellular space following unconventional secretory pathways. Reliable predictions of unconventional protein secretions (UPS) are still demanding. Here, we present OutCyte, a fast and accurate tool for the prediction of UPS, which for the first time has been built upon experimentally determined UPS proteins. OutCyte mediates prediction of protein secretions in two steps: firstly, proteins with N-terminal signals are accurately filtered out, secondly, proteins without N-terminal signals are classified to be UPS or intracellular proteins by physicochemical features directly generated from their amino acid sequences. We are convinced that OutCyte will play a relevant role in the annotation of experimental data and therewith will contribute to further characterize the extracellular nature of proteins by considering the commonly neglected UPS proteins.

OutCyte has been implemented as a web server at www.outcyte.com.

Introduction

A protein's identity is not only determined by its structure but also by its cellular localization which is associated with specific post-translational modification patterns as well as interaction partners. Therefore, the determination of the protein localization by experimental approaches or prediction tools is a relevant task for annotating complex data sets from global approaches like e. g. genomics, transcriptomics or proteomics allowing novel functional inferences. For the transport of proteins into the extracellular space, reliable prediction is already well established for proteins with an N-terminal signal peptide[1,2]. The so-called classical protein secretion is a signal-based process that acts along the Endoplasmic

Reticulum (ER) -Golgi route and is highly conserved in yeast, plant and animal cells[3]. Secretory pathways acting without the involvement of an identified N-terminal signal peptide are classified under the term “unconventional protein secretion” (UPS)[4,5,6]. The lack or unawareness of clear sequence patterns of unconventional protein secretions (UPS) and the low number of known UPS proteins have so far been major obstacles for the generation of meaningful prediction tools. The existing computational tools for predicting UPS, such as SecretomeP[7], SPRED[8], made use of the classical secretory proteins by removing their signal peptides, based on the hypothesis that all secretory proteins share common features regardless of specific pathways.

Here, we introduce OutCyte, an integrated tool featuring two modules for predicting unconventional secreted proteins in eukaryotes. In contrast to existing tools, the module for predicting potential UPS (OutCyte-UPS) was built on the one hand on our in-house experimental secretome data sets, using features directly generated from protein sequences. On the other hand, we decided to filter those proteins by an addition module (OutCyte-SP) to avoid interferences from other secreted proteins with reliable predictable N-terminal signal peptide (classical secretion, ectodomain shedding of membrane proteins). OutCyte-SP is based on convolutional neural networks allowing to filter proteins with N-terminal signals before applying OutCyte-UPS for the prediction of UPS proteins.

Results and Discussion

OutCyte has been designed in a modular fashion (Fig. 1) including two modules: OutCyte-SP classifies proteins exhibiting a signal peptide or transmembrane domain within the first 70 amino acids. Thus, we are able to independently determine N-terminal signal peptides or transmembrane domains. The group of proteins without such sequence motifs are fed into OutCyte-UPS for predicting UPS proteins from this group. In contrast to other tools developed for predicting UPS, OutCyte differs in two main points. Firstly, instead of relying on database information which might be prone to reporting false positives, OutCyte has been trained on experimental data from secretome analysis of several cell lines. Secondly, it provides two layers of predictions to avoid false positive UPS arising from proteins containing signal peptide or transmembrane domains at the N-terminus.

OutCyte-SP: prediction of N-terminal signals

OutCyte-SP allows to annotate proteins with an N-terminal signal peptide or transmembrane domain (Fig. 1). Here, we trained a convolutional neural network model (CNN) with a novel structure (Fig. 2a) for detecting N-terminal signal sequences. The training data was acquired by extracting all eukaryotic protein names from SignalP4.0’s dataset and then downloading the corresponding sequences[1] from UniProt Release 2018-05[10]. The training data contains

proteins of three categories, signal peptide containing (SP), transmembrane domain at N-Terminus (TM70) and intracellular proteins (In-cell), of which the first 70 amino acids were extracted for training the CNN model to distinguish the three groups.

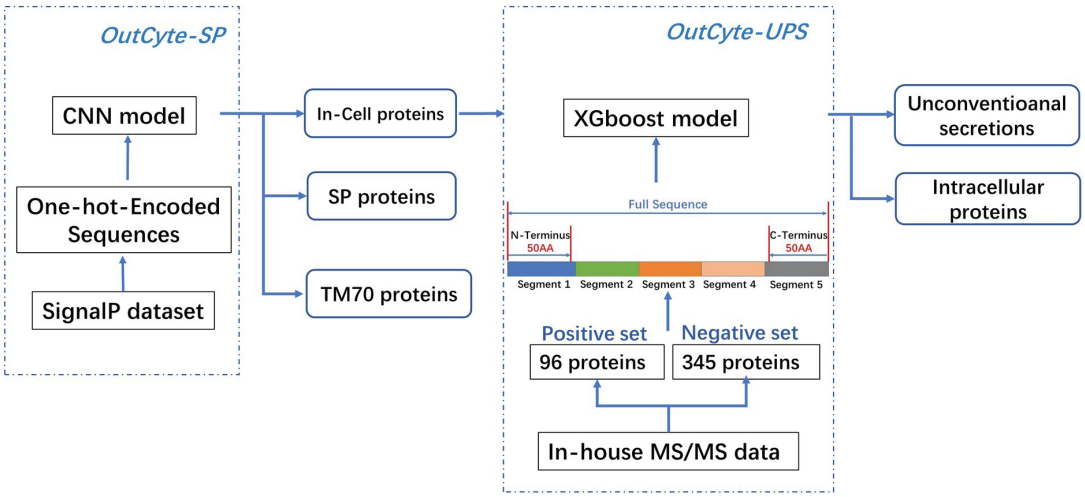


Fig. 1. The OutCyte framework is an integrated predictive tool for signal peptide containing proteins and unconventionally secreted proteins. OutCyte-SP classifies input proteins into three categories: proteins with a signal-peptide, proteins with transmembrane-domain in the N-terminus, or proteins not belonging to those two classes. The latter proteins were further analysed by OutCyte-UPS, which has been trained on experimentally determined secreted proteins and classifies input proteins to be intracellular or unconventionally secreted.

It is well-known that signal peptides carry a charged N-region, hydrophobic H-region and polar C-region with small uncharged residues at -1 and -3 positions[2], and transmembrane domains have a hydrophobic region[11]. In consideration of capturing these motifs underlying the amino acid sequences which were one-hot-coded as 70 x 20 matrices (details in methods section), the first convolutional layer of the CNN performed channel reduction to compress the 20 dimensions by five kernels. The consequent five feature maps may be interpreted as higher level representations of protein sequences, for instance, hydrophobicity, polarity, charges or their combinations. These feature maps were transformed with rectified linear units (ReLU), without pooling for this convolutional layer. In the second layer, one-dimensional convolutional kernels ran along the sequence length dimension to detect motif features. Then followed a max pooling layer and ReLU layer which extracted the maximal feature produced by each one-dimensional kernel. A dense layer then further transformed the learned features to the final softmax layer providing scores separate

for the three classes. Comparing to SignalP and DeepSig, the design of CNN structure made OutCyte-SP a light-weighted model but still efficient in capturing motifs.

The performance comparison with SignalP 4.1[1], SiganlP 5.0[12] and DeepSig[13] on three benchmark data sets showed that our CNN model OutCyte-SP reached comparable or better performance based on the Matthews Correlation Coefficients (MCC) of signal peptides identification and micro-averaging MCC of three-class predictions. As shown by MCC values (Fig. 2b), OutCyte-SP achieved comparable performance to DeepSig and SignalP 5.0 on benchmark sets from SignalP4.0, DeepSig and SignalP5.0. The performances of OutCyte-SP and DeepSig on the benchmark set of SignalP5.0 were close to each other but less accurate than SignalP5.0. Since both DeepSig and OutCyte-SP are three-class models, the micro-averaged MCCs of their predictions on two benchmark sets were also compared. These comparisons showcased OutCyte-SP's ability to identify proteins with N-terminal signal peptides, which achieved comparable performances to other state-of-the-art tools.

To further evaluate OutCyte-SP, we applied it to the human proteome and predicted 3,512 signal peptide containing proteins which is close to former studies (3,102 proteins by DeepSig, 3,556 proteins by SignalP 4.1 and 3,323 proteins in UniProt Release 2018-05). OutCyte-SP exhibited high agreement with the common resources for signal peptide annotation and only 90 proteins were unique to Outcyte-SP (Fig. 2c). Moreover, it seems that our CNN model has a more efficient structure than SignalP5.0 and DeepSig for discriminating TM70, In-cell and SP sequences. Therefore, Outcyte-SP appears optimal for our integrated computational environment to filter proteins for the cascaded module OutCyte-UPS. It is important to mention that until now OutCyte-SP has only been trained and tested on eukaryotic proteins.

4

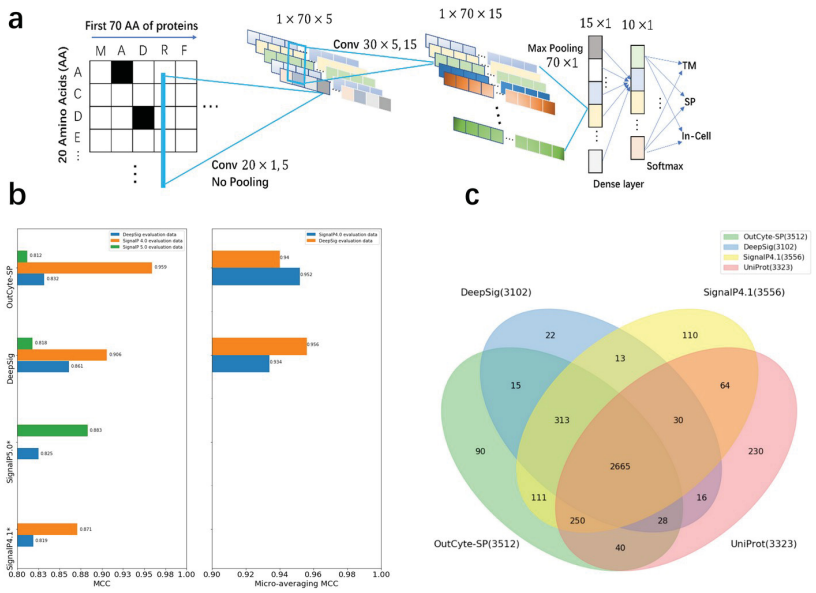


Fig.2. The OutCyte-SP model and its predictions. a). The structure of convolutional neural network for learning the motifs at N-terminus of sequences. It consists of two convolutional layers, which use ReLu transformations and no pooling. A max pooling layer follows to extract the most distinguishing features. Then is the dense and softmax layers. b). The Matthews Correlation Coefficients (MCC) for signal peptides identifications of three datasets were shown in the left panel. In the right panel, micro-averaged MCC were calculated for OutCyte-SP and DeepSig on the two evaluation datasets. *SignalP5.0 training dataset overlapped with SignalP4.0's benchmark set, thus two MCCs were not included. c). The intersections among 4 different annotations for signal-peptide-containing proteins in human proteome from OutCyte-SP, UniProt (with evidences), SignalP 4.1 and DeepSig.

OutCyte-UPS: predicting unconventional secretion

Furthermore, we relied on well-defined and representative data sets for developing OutCyte-UPS – a prediction tool for unconventional protein secretions. For proteins utilizing different UPS routes, only a small group of 18 representative proteins was described[7,8]. Therefore, different strategies like e.g. removing the signal peptide sequence of predicted classically secreted proteins[7] or considering annotated extracellular proteins[8,14] have been performed to virtually extend the number of candidates for training of predictive algorithms. Here, we relied on an in-house data set (157 proteins) of experimentally determined candidate proteins obtained by an integrated secretome and proteome approach[15-18]. The list of candidates was shortened to 96 proteins by reductions of homologues proteins within the candidate set and removing the 18 previously reported proteins of UPS and their homologues (details in methods). The negative data for training comprised 345 proteins which were commonly underrepresented in cells' secretome (details in methods) and therefore less likely to be

secreted. For our independent data set for the evaluation of OutCyte prediction, the 18 representative UPS proteins were considered as well as 20 proteins from our in-house data base which were highly enriched in the intracellular proteome. As we and others[7,8] are convinced that the protein sequence properties affect UPS and thereby the physicochemical properties of the involved amino acids, 61 features were considered (Table S1) as potentially informative for UPS.

To build models from the small and imbalanced data sets, we first performed an effective feature selection and finally kept 8 features (Fig. 3a) and then oversampled the positive set to balance the negative set (details in methods section). Then, the model based on XGBoost[19] was trained by the same nested cross-validation scheme as OutCyte-SP achieving a cross-validation score of 0.73. When applied to the benchmark dataset, OutCyte-UPS classified 14 out of 18 UPS proteins correctly and achieved an AUC of 0.80 by ROC analysis (Fig. 3b, Table S2). The misclassified proteins were H4-Human, FGF2-Human, THTR-Human and HMGB1_Human. In contrast, SecretomeP resulted in an AUC of 0.61 and misclassified 9 UPS proteins.

Next, we were interested in evaluating individual feature contributions for predictions using OutCyte-UPS. Here, the feature ranking for single protein predictions is consistent with the result from independent data set: beside the physicochemical features in C-terminus, the molecular weight and positively charged amino acids, the frequency of arginine within the protein sequence contributes significantly (Fig. 3a, S1, S2). It is interesting to note that the role of arginine in protein transport is well-established. In plants, bacteria and archaea arginine exert its role in signal motif of the *Twin Arginine Translocation* (Tat) system which contains a characteristic twin arginine motif within the N-terminal signal peptide. Moreover, several computational and experimental studies[20-21] reported that arginine have a more mechanistic role in arginine-rich cell-penetrating peptides (CPP), which enhanced CPP's ability to transverse the cell membrane of many mammal cells. The significance of arginine content in a protein shown in our study may indicate its important role in promoting UPS in at least a subgroup of the secretome. By manual inspection we revealed that FGF2-Human and H4-Human misclassified by OutCyte-UPS exhibited higher frequencies of positively charged amino acids and arginine than all other samples in both independent and training positive data sets (Fig. S3, S4), which may contribute to their misclassifications.

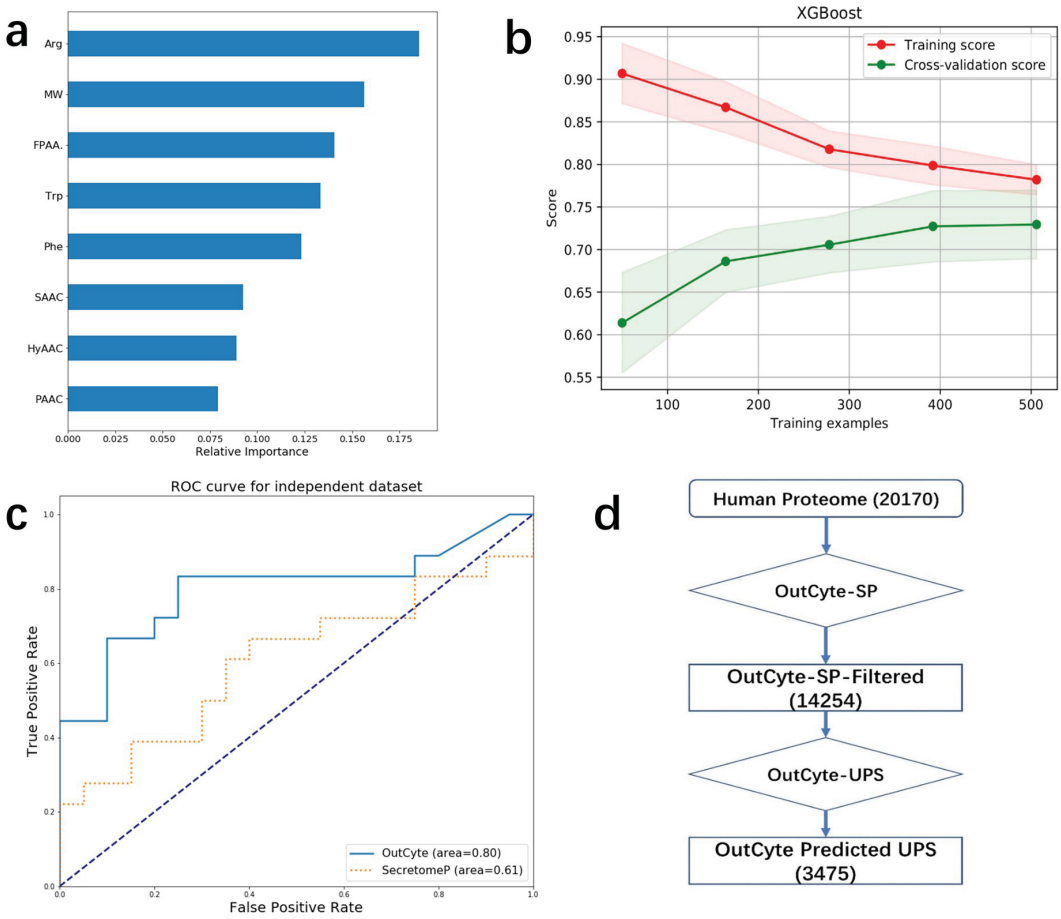


Fig.3 Prediction of unconventional protein secretion by OutCyte-UPS. a). Eight features were identified to be important for the classification of unconventionally secreted proteins. Important features include a high frequency of arginine residues and positively charged amino acids which have already previously associated with the membrane transition of proteins.; b). cross-validated training curve for XGBoost-based OutCyte-UPS; c). An independent data set containing experimentally verified UPS proteins as well as top 20 intracellular proteins from experimental data was used for performance comparison. Here OutCyte-UPS showed improved performance compared to SecretomeP; d). The OutCyte pipeline was applied on all 20170 proteins from the human proteome: OutCyte-SP classified 6077 proteins to contain either an N-teminal signal peptide or transmembrane domain. The remaining 14,254 proteins were passed to OutCyte-UPS prediction of unconventional secreted proteins. Finally, 3,475 human proteins were predicted to be unconventionally secreted.

Annotating human proteome

For a long time, alternative secretion routes were neglected and secreted proteins without a signal peptide were commonly considered as contaminants. Therefore, we were interested in predicting the number classical secreted proteins as well as UPS in the human proteome. Of the reviewed human proteome comprising 20,170 proteins we predicted 1,829 proteins with

a signal peptide for classical secretion by OutCyte-SP. This number is in the range of other prediction tools/database: 1,836 proteins (SignalP 4.1), 1,693 proteins (DeepSig), 1,999 proteins (UniProt) (Fig. S5). From the 14,245 proteins left over after OutCyte-SP filtering, we predicted by OutCyte-UPS 3,475 proteins as human UPS secretome (17.1% of the reviewed human proteins) (Fig. 3d). In contrast, SecretomeP identified 6,688 proteins as UPS, which made roughly one third of the reviewed human proteome (Fig. 3d, S6, S7). However, as the exact number of UPS candidates are still unknown, we cannot exclude a bias toward a specific secretory pathway overrepresented in our secretome data sets.

Conclusion

With our presented experimental data driven approach, we built OutCyte for predicting potential unconventional protein secretions. Its first part - OutCyte-SP - proved its ability to efficiently and accurately identify N-terminal signals such as signal peptides and transmembrane domains. The cascaded part OutCyte-UPS was trained on our experimental data and outperformed SecretomeP on the currently known unconventionally secreted proteins. We are convinced that it will open new perspectives on the long-time hidden processes of UPS and that we have laid the basis to improve the prediction of UPS using upcoming experimentally verified UPS proteins.

Methods

8

The training and benchmark datasets for OutCyte-SP

OutCyte-SP was trained on eukaryotic proteins extracted from SignalP4.0's dataset. To obtain the up-to-date sequences, the eukaryotic protein names were used to retrieve sequences from UniProt Release 2018-5. In the training set, 1361 proteins possess a signal peptide with experimental evidences annotated in UniProt, 913 proteins with transmembrane domains annotated at the first 70 amino acids were extracted, and 4491 proteins from nuclear or cytoplasm were kept for representing proteins without N-terminal signals.

In order to benchmark OutCyte-SP with SignalP 4.0, SignalP 5.0 and DeepSig, we tested all four models on three benchmark datasets (SignalP 4.0 benchmark set, SignalP5.0 benchmarkset and DeepSig benchmark set; due to the overlap between SignalP5.0 training set and SignalP 4.0 benchmark set, two MCC values were not included in the Fig. 2a). The detailed statistics of the benchmark sets were shown in Table S3.

The protein identities were extracted from datasets of SignalP 4.0 and SignalP 5.0 and used for retrieve the sequences from Uniport Release 2018-5.

The datasets for OutCyte-UPS

In a recent approach, we described the comparison of abundances of secreted and cellular proteins as a valuable tool to select proteins which are enriched in the secretome and therefore probably secreted[15]. Using this approach on ten different experimental settings including mouse, rat, and human cell types[16,17], we developed a database containing proteins showing a high likelihood to be secreted in the respective systems (called secreted proteins in this chapter) as well as a very high likelihood not be secreted (called cellular proteins in this chapter). The proteins which showed up to be secreted in at least three different experiments and did not contain a signal peptide and / or transmembrane domain (UniProt Annotation release May 2018) were selected as training set for OutCyte-UPS. It is worthy of noting that the computational annotations of signal peptides and transmembrane domains in UniProt were also included in our data processing. Potential false positives/negatives may present in the annotations due to protein isoforms, false prediction and so on. However, the high accuracy and confidence of the tools[12] used for predicting signal peptides could keep noise level low in our processed data. Grube et al.[15] experimentally validated that for the detection of secretome enriched proteins, e.g. by inhibiting classical secretion by Brefeldin A, 95% proteins whose secretion was inhibited by Brefeldin A indeed has a signal peptide annotated in UniProtKB. Therefore, the noise was kept at low level in our processed data. And a certain level of noise in training data is usually expected when developing a machine learning model[36].

Since the same gene might encode proteins with different names in different organisms, the presences of each protein in different experiments were counted in terms of its encoding gene. If the same gene showed up in multiple organisms with different protein names, its human protein homolog was kept in the positive data. In total, we have obtained 157 unconventional secretory proteins. We further cleaned up the data by removing proteins showing sequence identities above 30% with proteins in the independent data which was consisting of 18 positive proteins from the literature and 20 negative proteins selected from our experiments (explained later). The final UPS training data set contained 96 proteins. Similarly, the intracellular proteins were prepared as negative data set for training. The proteins which were highly abundant in cellular proteome but rare in secretome were kept as candidates. For those candidates, ones with signal peptides or transmembrane domains as

well as proteins showing a sequence identity higher than 30% with other used sequences have been removed.

To generate a trustworthy independent negative data for evaluating OutCyte-UPS, the top 0.5% proteins enriched in cell lysates and underrepresented in secretomes were extracted from our database. As OutCyte-UPS was focused on identifying UPS from proteins without both transmembrane domains and signal peptides, the extracted proteins with either transmembrane domains or signal peptides have been excluded, resulting in finally 20 proteins as negative evaluation data. The sequence length distributions of different datasets were plotted in Fig. S8, which showed that the medians and means were not biased towards either positive or negative training sets.

Preparation of human proteome sequences

To scan for unconventional secretory and signal peptides containing proteins in human proteome, we extracted the reviewed entries in UniprotKB Release 2018-05 for human proteome of 20,170 proteins, which is divided into four categories: proteins with signal peptide and transmembrane domain, proteins with signal peptides but without transmembrane domains, proteins without signal peptide and transmembrane domains and proteins with transmembrane domains but without signal peptides. The statistics of this categorizing is summarized in Fig. S9.

Model metrics

Multiple metrics have been calculated for the model benchmarks. This section explicitly defines all the used metrics, which need these abbreviations, TP for the number of true positives, TN for true negatives, FP for false positives and FN for false negative.

Accuracy (ACC)

$$ACC = \frac{TP + TN}{(FN + TP + TN + FP)}$$

$$ACC = \frac{TP + TN}{(FN + TP + TN + FP)}$$

Sensitivity or true positive rate (TPR)

$$TPR = \frac{TP}{FN + TP}$$

Specificity or true negative rate (TNR)

$$TNR = \frac{TN}{TN + FP}$$

Matthews correlation coefficients

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Micro-averaging for three classes 0, 1, 2:

$$TP = TP_0 + TP_1 + TP_2$$

$$FN = FN_0 + FN_1 + FN_2$$

$$TN = TN_0 + TN_1 + TN_2$$

$$FP = FP_0 + FP_1 + FP_2$$

One-hot coding representations of amino acid sequences

Twenty standard amino acids were considered in this work. After sorting the amino acid letters in alphabetic order, each amino acid letter was encoded by a 20-dimensional vector with its position in the alphabetic order set to 1 and the rest to 0's. Since each encoded vector has only one entry set to 1, it is called one-hot coding scheme. Therefore, for a protein sequence with L amino acids, it is represented by a L x 20 matrix.

Convolutional neural network

Convolutional neural networks (CNN)[23] have the properties of translational invariance and local spatial coherence due to the convolutions between input matrix and filtering kernels where the kernels parameterized by weights are expected to extract features from the input by tuning weight values during learning. CNN models are suitable for learning patterns, e.g. the signal peptide motif and transmembrane domain in amino acid sequences but with varied locations on the sequences.

CNN structures typically have a number of convolution layers for extracting features of different levels from the inputs, and each layer typically consists of operations of convolution, pooling and transformation for its input. The operations have different variants for different tasks. In this work, standard convolution, max pooling, and rectified linear units (ReLu) were used. Max pooling means that for a fixed window from a convolved feature map, its maximal value is used for representing the window. By max pooling, the dimensions of feature maps are reduced (down-sampling) and the most outstanding feature of each window is kept. For example, the max pooling can sharp the edges of blurry items in an image[24]. The same idea was applied to learn features from sequences. The ReLu transformation is defined as $f(x) = max(0, x)$, which provides a simple nonlinear transformation for accelerating the training of neural networks.

The CNN model is implemented and optimized in Theano[25].

Training OutCyte-SP CNN model with nested cross-validation

Cross validation is usually used for optimizing machine learning models, for instance, the *k*-fold cross-validation divides the whole dataset into *k* partitions, where *k-1* partitions are used for training and validating the model while one partition is left out for testing the model performance. As discussed in the paper of SignalP 4.0[1], the standard k-fold procedure is sufficient if the test data has been kept as blind for the model during training procedure, i.e. it should not be used for either hyperparameters tuning or model selection. To overcome this problem, nested cross validation is applied to tune the models, which further performs inner *n*-fold cross validation on the k-1 partitions. In order to benchmark with SignalP 4.1 and DeepSig, we used the same cross-validation setup to tune hyperparameters and select models: one partition of a 5-fold is kept out and an inner 4-fold cross-validation on the rest

four partitions is performed to optimize the CNN hyperparameters, for example, the learning rate, the mini-batch size and so on, and the CNN structures, for example, the number of convolution layers, number of kernels, kernel sizes and so on. After the nested cross validation procedure, 20 CNN models with the same structures trained with the same hyperparameters but with different inner training partitions are obtained for constructing the final ensembled model in application.

Feature generation and selection for OutCyte-UPS

In the context of classifying UPS, given the dataset of 96 positive and 345 negative examples, proteins need to be represented by informative characteristics in terms of secreted proteins, unlike predicting signal peptides where one-hot-coded raw sequences can be directly used as input features for convolutional neural networks for extracting the clear motif and the dataset size is much larger than UPS dataset. Moreover, the same as reported by Bendtsen et.al[7], we did not find any clear motif in the 18 reported unconventional secretory proteins.

To generally characterize the proteins, amino acid compositions are represented the individual amino acid frequencies over the entire sequences. For example, the frequency of amino acid *i* (AA_i) is calculated as

$$AA_i = \frac{Numberof AA_i}{Totalsequencelengt\,h}.$$

Many studies have reported the molecular weights of proteins had influenced proteins' secretions. We also plotted the histograms of human classical secretory proteins and human proteome in Fig. S10. which showed that relatively small proteins are favored for secretions. To this end, molecular weights are calculated by the *molecular_weight()* function in Biopython[26] as protein features.

Physicochemical features are widely believed in playing a critic role in proteins secretion, such as hydrophobicity, polarity, positively or negative charged residues. Small amino acids are also considered for generating features as they also affect protein functions[27] and we hypothesize their content of a protein sequence might influence the secretion of proteins. For characterizing the positional physicochemical features of protein sequences, the frequencies of different groups amino acids (hydrophobic, polar, positively charged, negatively charged and small) are calculated for segments as shown in Fig. S11.

We did not account for protein folding structures within our list of features as even for classical secretion in most cases the exact process of folding and maturation during the ER Golgi passage is not clearly defined. In bacteria, the two major modes of protein secretion follow either the Sec or Tat (*Twin Arginine Translocation*) pathway. , Proteins are folded after secretion when secretion is mediated by the Sec machinery, whereas the Tat machinery carries folded proteins to the outside of bacteria. Both Sec and Tat pathways have been found in eukaryotes as well[28].

In total, 61 features have been generated for individual proteins, which are summarized in Table S1. An exploratory data analysis for all the features of UPS dataset showes the correlation of features (Fig. S12). Due to the limited size of the available UPS dataset, we performed an extensively feature selection to keep only the most representative features. Feature importance ranking shown in Fig. S1. was obtained from averaging 500 single

rankings using Random Forest classifier in Scikit-learn 0.19 on the merged dataset from both training data and independent data. It is worthy of noting that the real population of UPS in even human proteome remains mystery, therefore both our 96 positive example and 18 reported proteins are merely samples from the population, which are highly likely to suffer from the sample selection bias. Due to the bias, many features could be identified as drifting features which have a strong discriminative power for training dataset and independent dataset. The ranking of drifting features is shown in Fig. S2. Using features that are important and less drifting, we further performed best-one search[29] to keep the top feature combinations which resulted our final features.

The subsequent feature selection was based on feature importance ranking and feature drifting analysis. To avoid a bias due to different sample sources we considered both training and test data from the same distribution[30,31]. Further, we selected the features which are top in importance ranking meanwhile less drifting in the drifting ranking. Finally, the features considered by OutCyte-UPS include the molecular weight, frequencies of small, hydrophobic and positively charged residues in C-terminus, frequency of positively charged residues over the entire sequence, and the frequencies of tryptophan, phenylalanine and arginine.

Another common challenge in machine learning tasks is the unbalanced training dataset. We have 345 negative examples but only 96 positive samples. To handle the imbalanced dataset, we chose oversampling the minority rather than downsampling the majority due to small size of datasets. The repeated oversampling, synthetic minority oversampling technique (SMOTE)[32] and adaptive synthetic oversampling approaches (ADASYN)[33] have been applied to oversample the positive dataset and to balance it with negative datasets. Repeated oversampling is simply duplicating positive examples to match the number of negative examples. Both SMOTE and ADASYN oversampled the positives by generating new synthetic example with the assumption that the individual feature values are continuous such that similar features can be generated with values next to a given example.

Model training and selection for OutCyte-UPS

Due to the small UPS dataset, logistic regression, random forest and gradient boosting trees have been extensively tested and compared. In terms of binary classification with classes 0 and 1, logistic regression performs a sigmoidal transform of linear combination of input features to values falling in the range of [0, 1] which can be interpreted as the probabilities of being in class 1. Random forest and gradient boosting are both tree-based ensemble learning algorithms. Trees in random forest are parallelly grew and uncorrelated to each other, each tree is trained on boostrapped samples of the original training data and the output of a random forest is obtained by averaging outputs from all trees[34]. Gradient boosting trees grow CART (Classification and Regression Tree) sequentially to fit each tree to the current residue given by its preceding tree, in other words, it additively ensembles weaker learners (CART trees) in a sequential manner to have powerful models[19].

As we have an independent dataset for evaluating the final model predictions, a routine of nested cross validation with grid search of hyperparameters was used for training models in order to make full use of the limited dataset and avoid information leaking from the training data to test data. The same as training OutCyte-SP, the cross validation leads to 20 runs for each hyperparameter setting. Then averaged metrics for training and testing are obtained from 20 runs. The averaged metrics MCC are used for selecting models.

Probability calibration for the tree-based models is often needed for the reason that we intend to not only predict the class for the given data point but also obtain a well calibrated probability as the confidence of being in a certain class. For example, it is difficult for methods like random forest to make predictions with scores of 0 or 1 because the variance from individual trees drag the actual predicted scores from zero or one as they should be[35]. We made use of the parametric "sigmoid" and nonparametric "isotonic" methods implemented in Scikit-Learn 0.19 for calibrating the final predictions scores.

Data and code availability

All datasets and code used in this work for developing models are publicly available on our web server (www.outcyte.com).

Competing interests statement

The authors have declared that no competing interests exist.

ACKNOWLWDGEMENT

We would like to thank research commission (Foko) of University Hospital Düsseldorf (LZ, KS) and CRC1208 (DWL, GP) for support.

AUTHOR CONTRIBUTIONS

LZ: Data analysis, webtool, manuscript writing

GP: Project planning, Data analysis

DWL: Data analysis

NR: webtool, data analysis

MK: Project planning

KS: Concept, Project planning, manuscript writing

References

1. Petersen, T. N., Brunak, S., Von Heijne, G., & Nielsen, H. SignalP 4.0: discriminating signal peptides from transmembrane regions. Nat. methods., 8(10), 785 (2011).
2. Nielsen, H., Engelbrecht, J., Brunak, S., & von Heijne, G. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein. Eng.*, 10(1), 1-6 (1997).
3. Blobel, G., & Dobberstein, B. Transfer of proteins across membranes. I. Presence of proteolytically processed and unprocessed nascent immunoglobulin light chains on membrane-bound ribosomes of murine myeloma. J. Cell Biol., 67(3):835–851 (1975).
4. Pompa, A.et al. Unconventional transport routes of soluble and membrane proteins and their role in developmental biology. *Int. J. Mol. Sci.*, 18(4), 703 (2017).

5. Rabouille, C. Pathways of Unconventional Protein Secretion. *Trends Cell Biol.*, 27(3):230–240 (2017).

6. Rabouille, C., Malhotra, V. & Nickel, W. Diversity in unconventional protein secretion. J. Cell Sci. 125: 5251-5255 (2012).

7. Bendtsen, J. D., Jensen, L. J., Blom, N., Von Heijne, G., & Brunak, S. Feature-based prediction of non-classical and leaderless protein secretion. *Protein. Eng. Des. Sel.*, 17(4), 349-356 (2004).

8. Kandaswamy, K.K. et al. SPRED: A machine learning approach for the identification of classical and non-classical secretory proteins in mammalian genomes. *Biochem Bioph Res Co*, 391(3):1306, 11(2010).

9. Arribas, J. & Borroto, A. Protein ectodomain shedding. *Chem. Rev.* 102, no. 12: 4627-4638 (2002).

10. The UniProt Consortium, UniProt: the universal protein knowledgebase. *Nucleic Acids Res.*, 45: D158-D169 (2017).
11. Sharpe, H. J., Stevens, T. J., & Munro, S. A comprehensive comparison of transmembrane domains reveals organelle-specific properties. *Cell*, 142(1), 158-169. 2010

12. Armenteros, J.J., et al. SignalP 5.0 improves signal peptide predictions using deep neural networks. *Nature biotechnol. 37*(4), 420. (2019).
13. Savojardo, C., Martelli, P. L., Fariselli, P., & Casadio, R. DeepSig: deep learning improves signal peptide detection in proteins. *Bioinformatics*, *34*(10), 1690-1696 (2017).

14. Yu, L., et al. SecretP: A new method for predicting mammalian secreted proteins. *Peptides*, 31(4):574–578 (2010).

15. Grube, et al. Mining the secretome of C2C12 muscle cells: data dependent experimental approach to analyze protein secretion using label-free quantification and peptide based analysis. *J. Proteome Res*. 17. 10.1021(2018).
16. Baberg, F., et al. Secretome analysis of human bone marrow derived mesenchymal stromal cells. *BBA-Proteins Proteom.* 1867, no. 4: 434-441 (2019).

17. Schira, J., et al. Secretome analysis of nerve repair mediating Schwann cells reveals Smad-dependent trophism. *FASEB J.* 33, no. 4: 4703-4715(2018).

18. Lupa-Waldera, D., et al. Characterization of Skin Aging–Associated Secreted Proteins (SAASP) Produced by Dermal Fibroblasts Isolated from Intrinsically Aged Human Skin. *J. Invest. Dermatol.* 135, no. 8: 1954-1968 (2015).

19. Chen, T. & Guestrin, C. Xgboost: A scalable tree boosting system. CoRR, bs/1603.02754, (2016).

20. Manavalan, B., Subramaniyam, S., Shin, T. H., Kim, M. O., & Lee, G. Machine-learning-based prediction of cell-penetrating peptides and their uptake efficiency with improved accuracy. *J. proteome res.*, *17*(8), 2715-2726 (2018).

21. Schmidt, N., Mishra, A., Lai, G. H., & Wong, G. C. Arginine-rich cell-penetrating peptides. *FEBS letters*, *584*(9), 1806-1813(2010).

22. El-Sayed, A., Futaki, S., & Harashima, H. Delivery of macromolecules using arginine-rich cell-penetrating peptides: ways to overcome endosomal entrapment. *AAPS J.*, *11*(1), 13-22(2009).

23. LeCun, Y., et al. Handwritten digit recognition with a back-propagation network. *Adv. Neur. In.*, pp. 396-404(1990).

24. Stanford CS231n: http://cs231n.stanford.edu/

25. Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688(2016).

26. Cock P.A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics,* 25*,* 1422-1423(2009).

27. Hastoy, B., et al. A central small amino acid in the VAMP2 transmembrane domain regulates the fusion pore in exocytosis. *Sci. Rep. 7*(1), 2835 (2017)

28. Natale, P., Brüser, T. & Driessen, A.M. Sec- and Tat-mediated protein secretion across the bacterial cytoplasmic membraneDistinct translocases and mechanisms. *BBA–Biomem.*, 1778(9):1735– 1756(2008).

29. Dechter, R. & Pearl, J. Generalized best-first search strategies and the optimality. *J. ACM*, 32(3):505(1985).

30. Bickel, P. J., Ritov, Y. A., & Tsybakov, A. B. Simultaneous analysis of Lasso and Dantzig selector. *Ann. Stat.*, *37*(4), 1705-1732 (2009).

31. Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., & Smola, A. J. Correcting sample selection bias by unlabeled data. *Adv. Neur. In.* 601-608 (2007).

32. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res*., 16, 321-357(2002).

33. He, H., Bai, Y., Garcia, E. A., & Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *IEEE World Congr. Comput. Intell*. 1322-1328(2008).

34. Breiman, L. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

35. Niculescu-Mizil, A., & Caruana, R. Predicting good probabilities with supervised learning. *ICML.* pp. 625-632. ACM (2005).

36. Atla, A., Tada, R., Sheng, V., & Singireddy, N. Sensitivity of different machine learning algorithms to noise. *J. Comp. Sci. Col.*, 26(5), 96-103(2011).

OutCyte: a novel tool for predicting unconventional protein secretion-

Supplementary information

Linlin Zhao[1,2], Gereon Poschmann[1], Daniel Waldera-Lupa[1], Nima Rafiee[2], Markus Kollmann[2], Kai Stühler[1,3 *]

[1]Institute of Molecular Medicine, Medical Faculty, Heinrich-Heine-University, Düsseldorf, Germany

[2]Mathematical Modelling of Biological Systems, Heinrich-Heine-University, Düsseldorf, Germany

[3] Molecular Proteomics Laboratory, BMFZ, Heinrich-Heine-University, Düsseldorf, Germany

*Correspondence: kai.stuehler@hhu.de (K. Stühler)

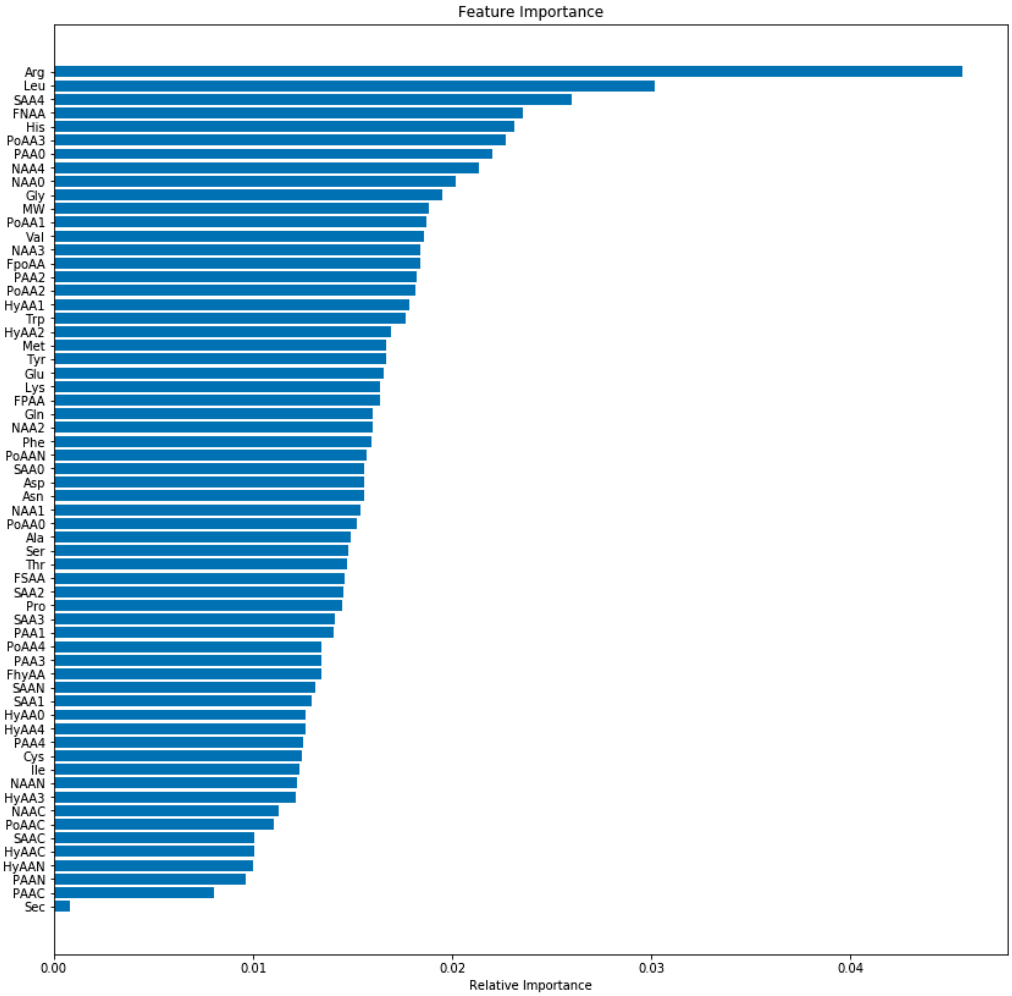


Fig. S1 Feature importance ranking

1

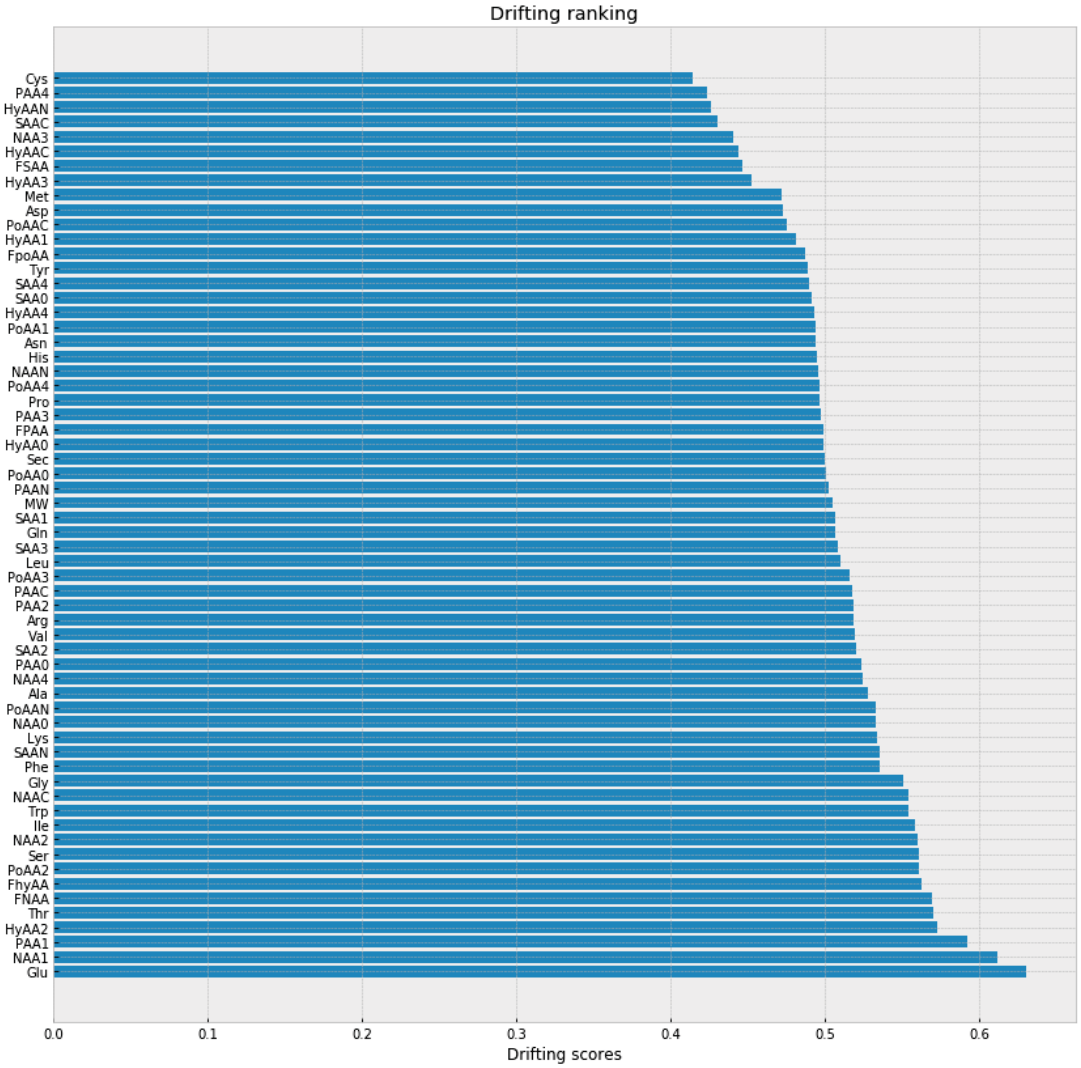


Fig. S2 Drifting ranking for features

2

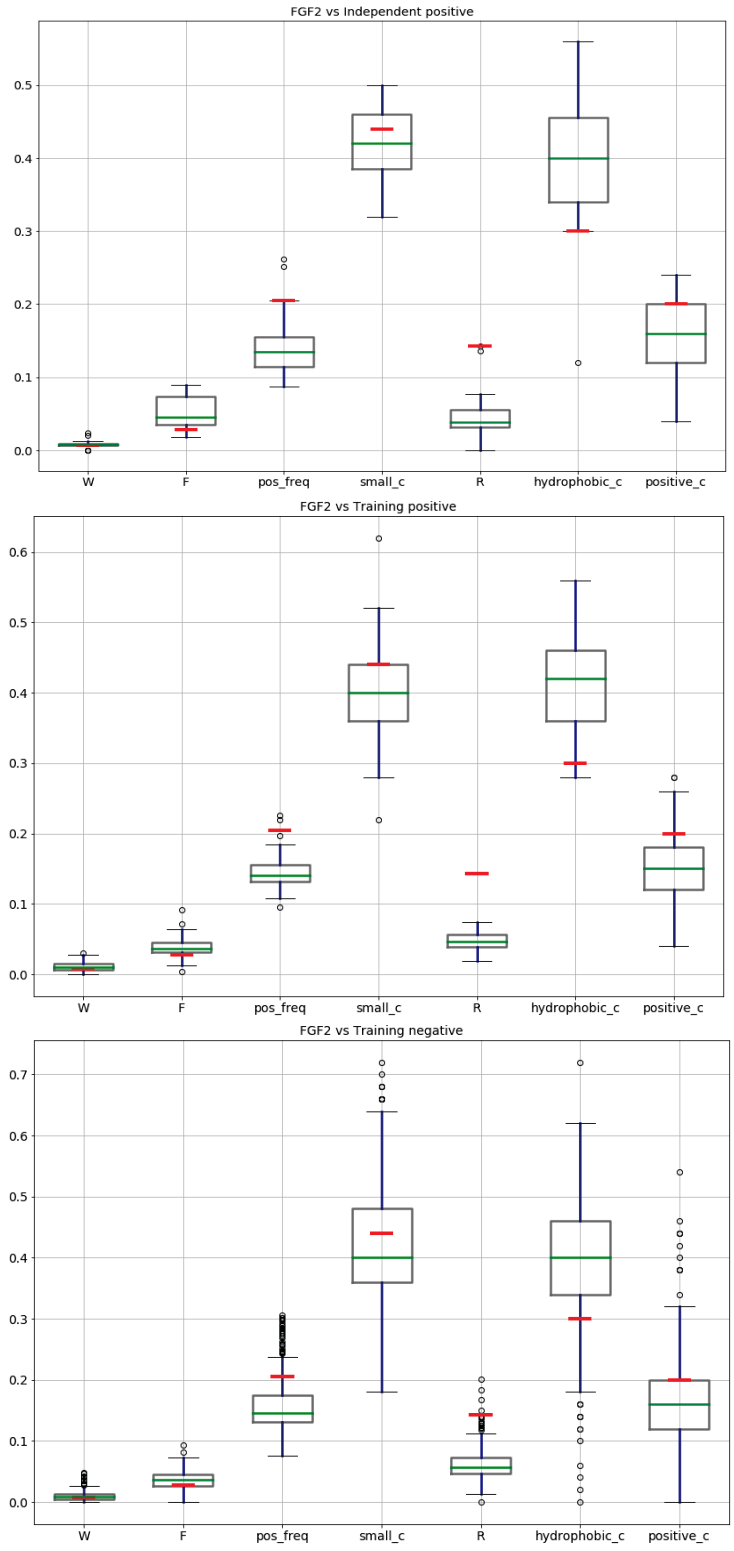


Fig. S3 The FGF2-Human's features (the red horizontal line) compare to the boxplot of features in different data sets. The y-axes stand for the values of the features on x-axes.

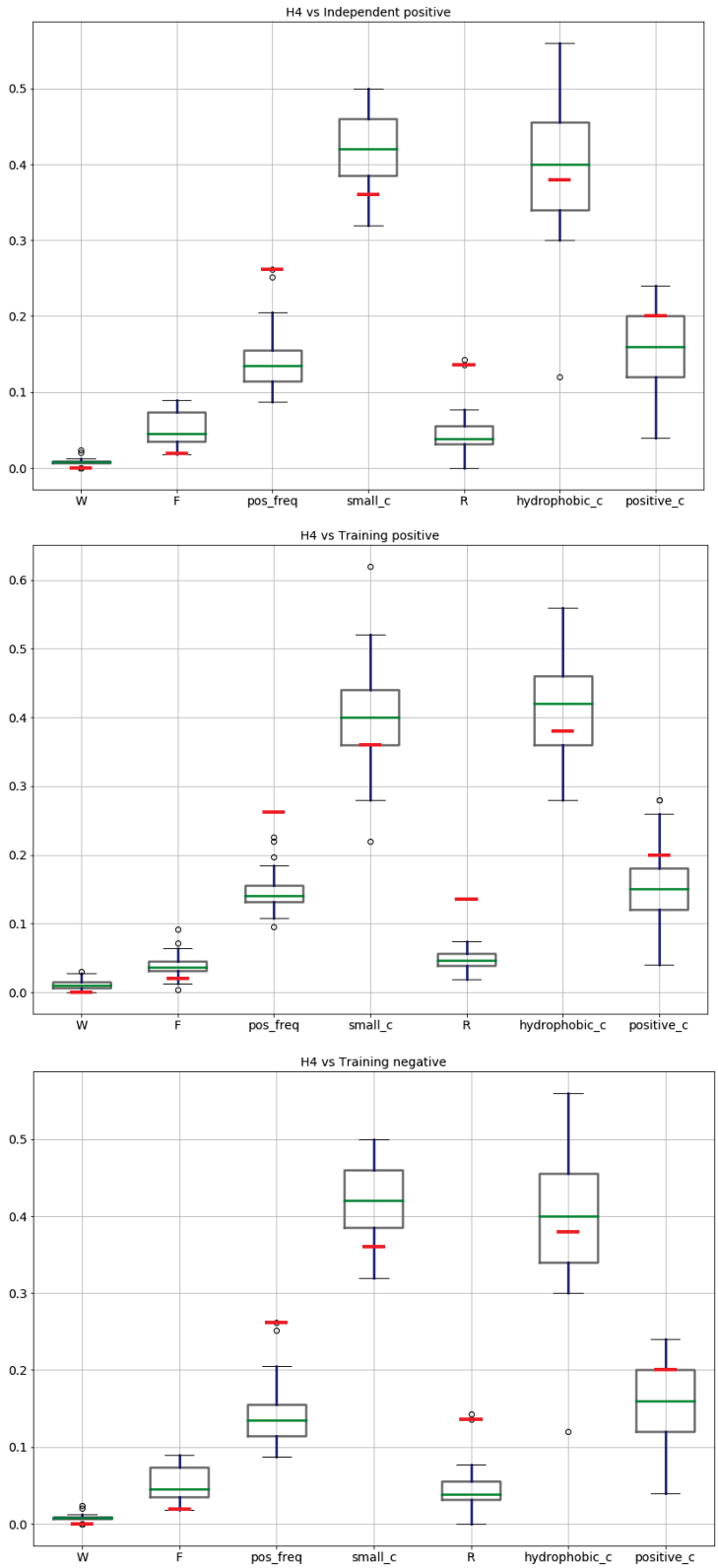


Fig. S4 The H4-Human's features (the red horizontal line) compare to the boxplot of features in different data sets

4

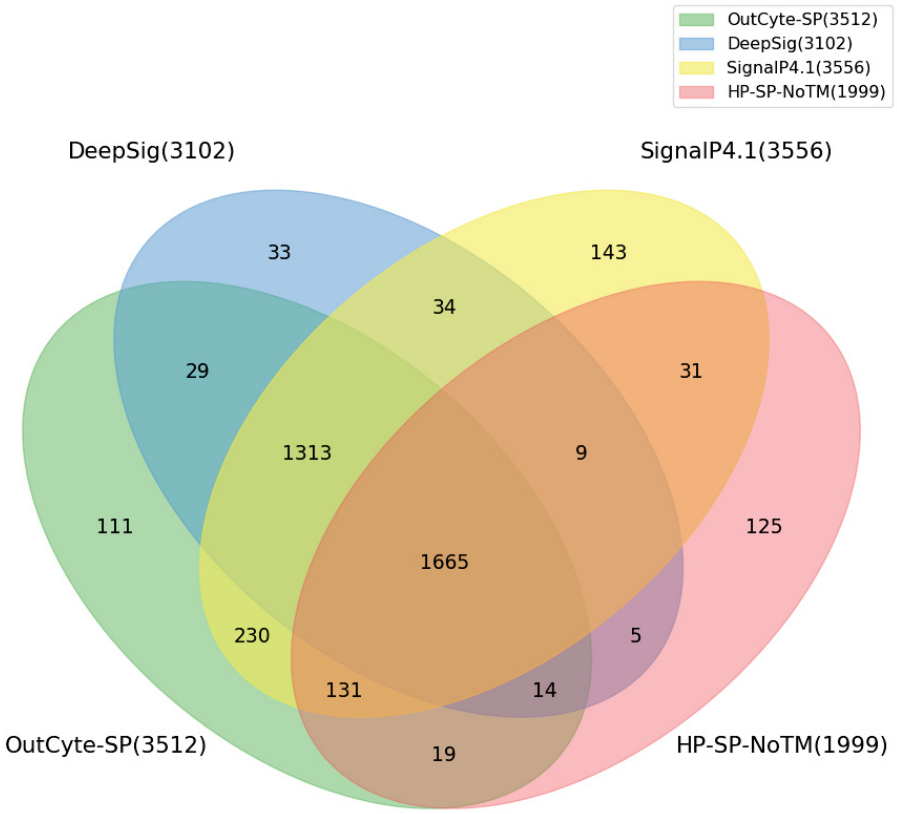


Fig. S5 Predictions of signal peptides within human proteins using different tools and databases (HP-SP-NoTM = signal peptide annotated in the UniProt database)..

5

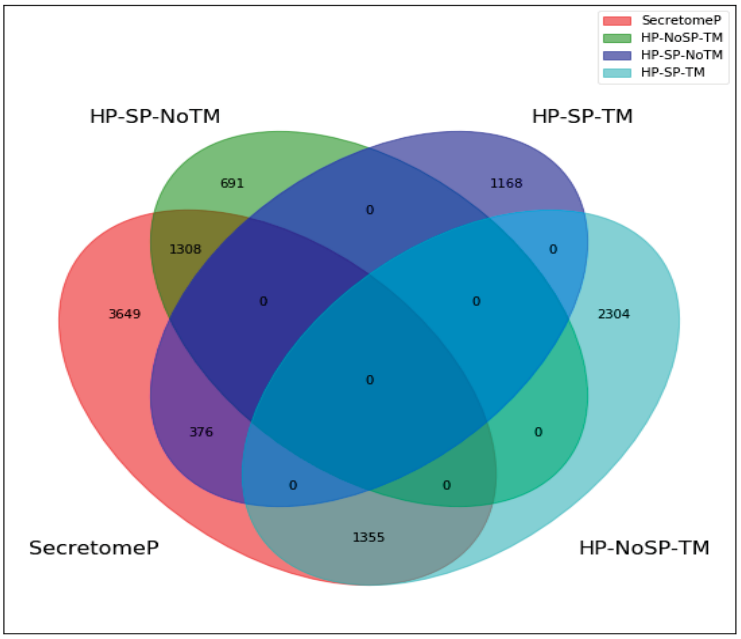


Fig. S6. The Venn diagram for SecretomeP prediction's intersection with three human proteome subgroups.

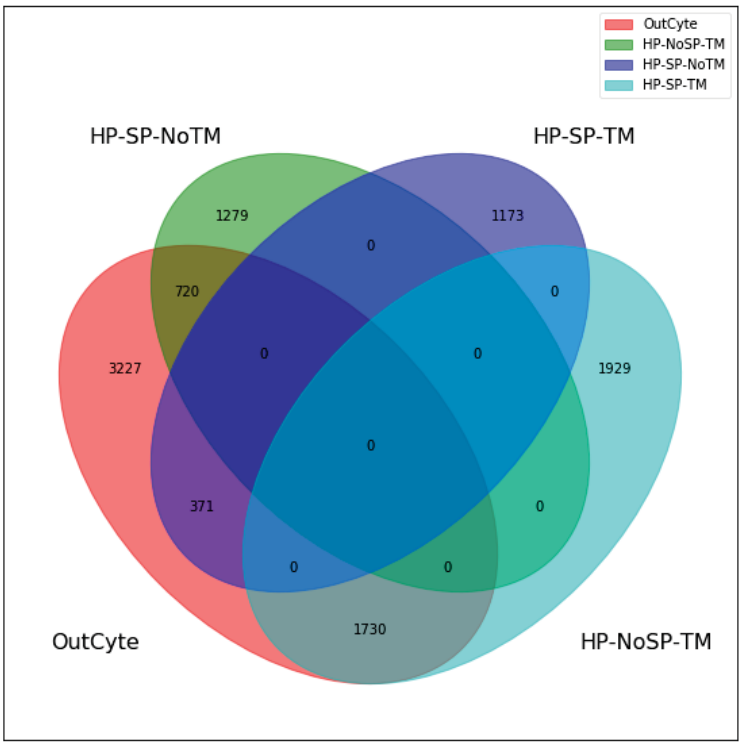


Fig. S7 The Venn diagram for OutCyte prediction's intersection with three human proteome subgroups.

6

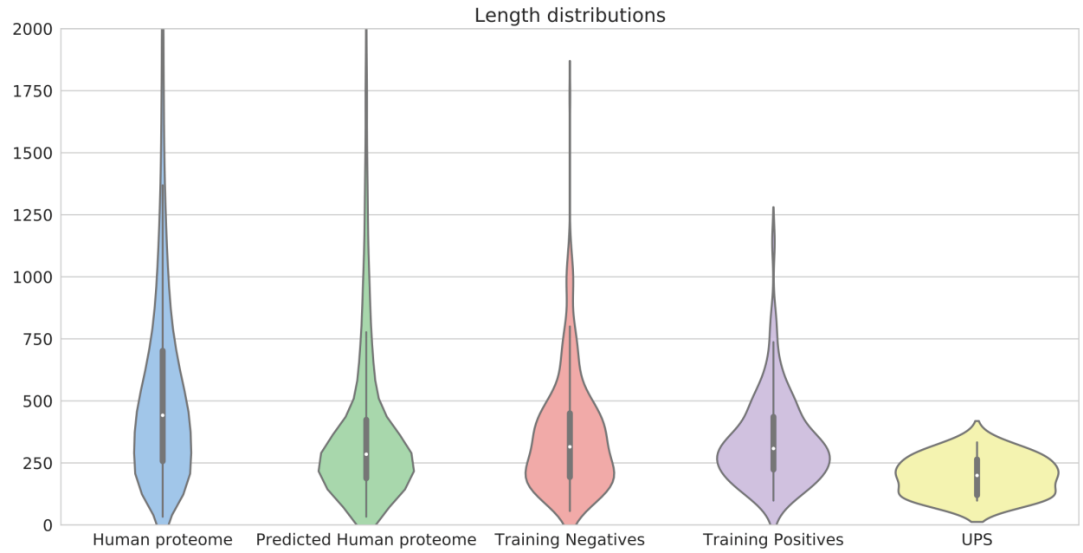


Fig. S8. Length distributions for datasets related to OutCyte-UPS

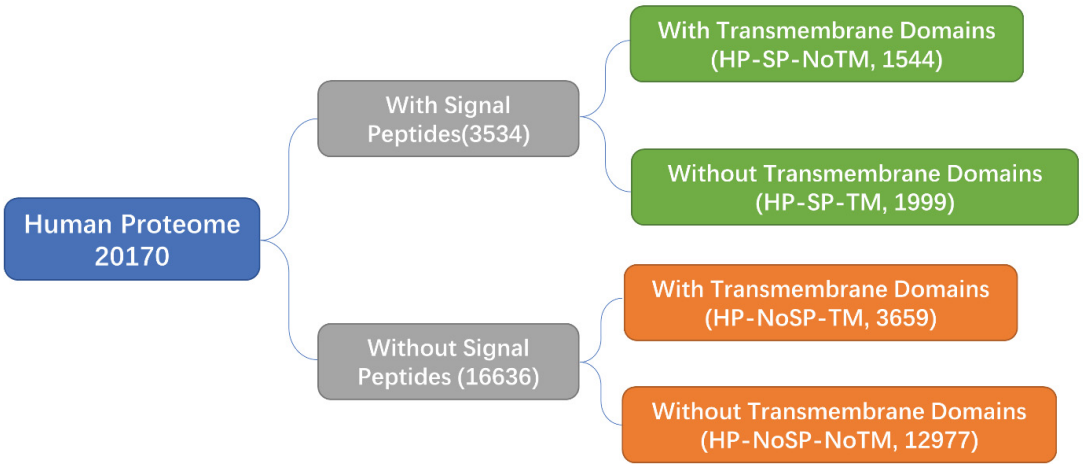


Fig. S9. Human proteome subgroups.

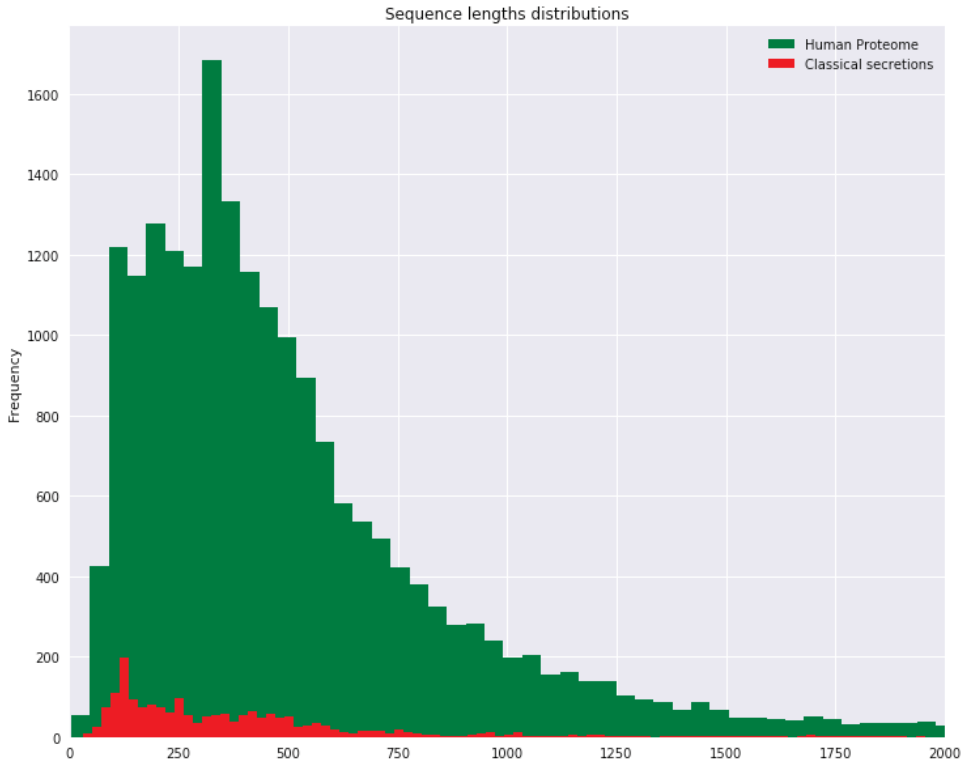


Fig. S10. The length distribution of human classical secretory proteins and human proteome, which showed the favor of smaller molecular in terms of secretion.

Fig. S11 Segmentations of sequences for generating positional physicochemical features

8

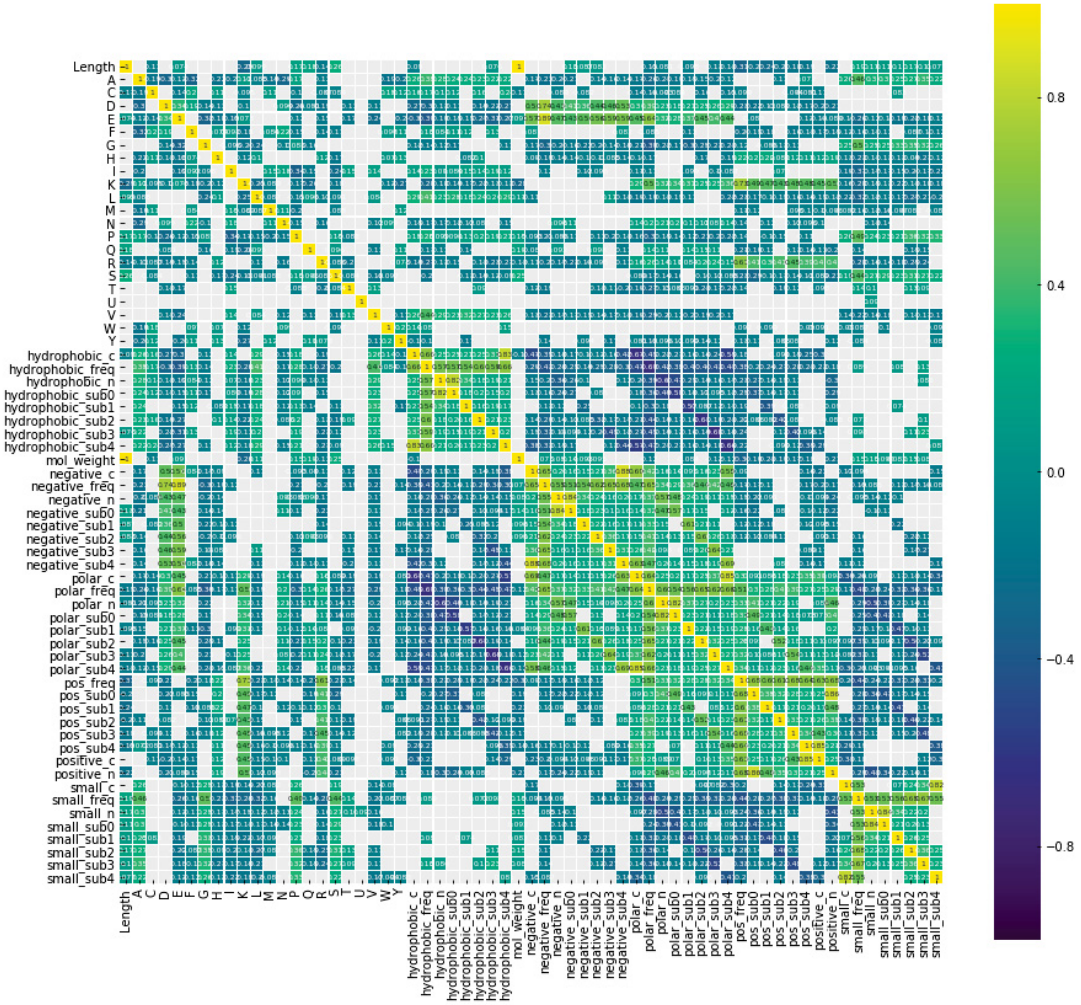


Fig. S12. The correlations of 61 features generated for building OutCyte-UPS.

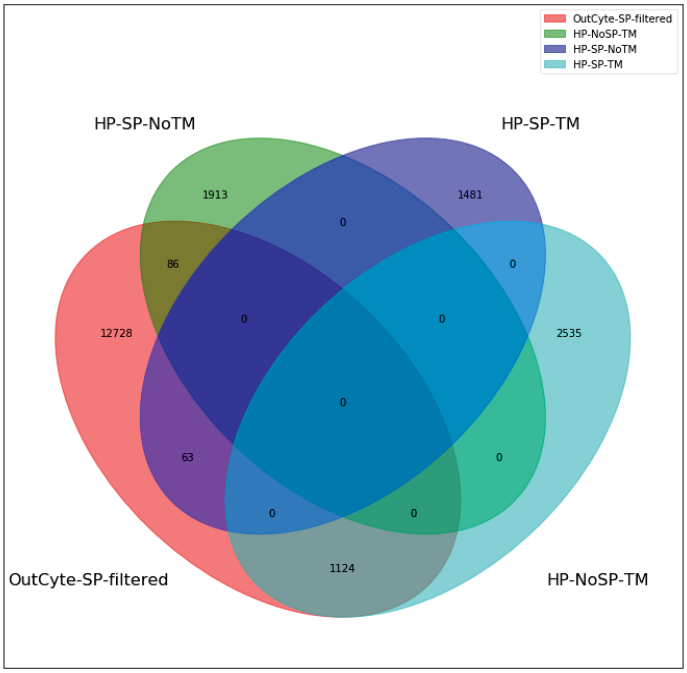


Fig. S13. The proteins without an N-terminal signal predicted by OutCyte-SP intersect with other human proteome subgroups. It shows the ability of OutCyte-SP to filter away proteins with N-terminal signals.

Table S1 List of 61 features for representing sequences

Size	Feature Names	Abbreviations
1	Molecular Weights	MW
20	Amino acid frequencies of entire sequence	Met, Cys, Trp, Phe …
3	Small amino acid frequencies of entire sequence, N- and C-terminus	FSAA, SAAN, SAAC
3	Positively charged amino acid frequencies of entire sequence, N- and C-terminus	FPAA, PAAN, PAAC
3	Negatively charged amino acid frequencies of entire sequence, N- and C-terminus	FNAA, NAAN, NAAC
3	Polar amino acid frequencies of entire sequence, N- and C-terminus	FPoAA, PoAAN, PoAAC
3	Hydrophobic amino acid frequencies of entire sequence, N- and C-terminus	FHyAA, HyAAN, HyAAC
5	Positively charged amino acid frequencies of 5 sequence segments	PAA1, PAA2, …, PAA5
5	Negatively charged amino acid frequencies of 5 sequence segments	NAA1, NAA2, …, NAA5
5	Polar amino acid frequencies of 5 sequence segments	PoAA1, PoAA2, …, PoAA5
5	Hydrophobic amino acid frequencies of 5 sequence segments	HyAA1, HyAA2, …, HyAA5
5	Small amino acid frequencies of 5 sequence segments	SAA1, SAA2, …, SAA5

Table S2 Predictions on known UPS

Protein	UniProt ID	OutCyte-UPS	SecretomeP	SRTpred
FGF1-Human	P05230	0.616(+)	0.847(+)	-0.81(-)
FGF2-Human	P09038	0.066(-)	0.239(-)	0.8(+)
IL1B- Human	P01584	0.598(+)	0.610(+)	0.96(+)
IL1A- Human	P01583	0.615(+)	0.551(-)	-0.2(-)
LEG3-Human	P17931	0.618(+)	0.770(+)	-1.16(-)
MIF-Human	P14174	0.584(+)	0.776(+)	-0.91(-)
S10A4-Human	P26447	0.614(+)	0.724(+)	-0.55(-)
GSTP1-Human	P09211	0.598(+)	0.545(-)	-0.7(-)
PRDX1-Human	Q06830	0.618(+)	0.528(-)	-0.94(-)
IL18-Human	Q14116	0.614(+)	0.634(+)	-1(-)
H4-Human	P62805	0.065(-)	0.408(-)	-1.12(-)
S10A2-Human	P29034	0.614(+)	0.324(-)	-0.48(-)
LEG1-Human	P09382	0.598(+)	0.345(-)	-0.62(-)
THIO-Human	P10599	0.617(+)	0.370(-)	0.71(+)
CNTF-Human	P26441	0.571(+)	0.653(+)	0.02(+)
HME2-Human	P19622	0.525(+)	0.727(+)	-1.39(-)
THTR-Human	Q16762	0.066(-)	0.616(+)	-1.2(-)
HMGB1-Human	P09429	0.499(-)	0.068(-)	-1.2(-)

12

Table S3 Statistics of datasets for training and evaluating OutCyte-SP

	SP	TM	N/C	Globular
Training	1361	913	4491	
Evaluation-SignalP4	609	939	1001	
Evaluation-DeepSig	46	323	688	
Evaluation-SignalP5	211			7248

Table S4 Signal peptide prediction benchmarks

	OutCyte-SP	DeepSig	SignalP4.0	UniProt
OutCyte-SP	3512	3021	3339	2983
DeepSig		3102	3021	2739
SignalP 4.0			3556	3009
UniProt				3323

14

6 Data-driven Automatic Annotations for Honeybee Behavior

6.1 Summary

Honeybees are social insects, forming societies with a queen, thousands of workers and a few male drones. The thousands of members collectively function as a single unit, of which the collaborative features are regulated by individual behaviours. The systematic understanding of honeybee societies requires information of simultaneous and continuous annotations of individual behaviours. Manual tracking of various behaviours for thousands of bees is not feasible. Therefore, automatic annotations of behaviours are needed to enable deep insights of collaborative features of honeybee colonies.

The paper in this chapter established *Bee Behavior Annotation System (BBAS)*, a system that can automatically classify stereotypical behaviours of individual workers in a group of honeybees. The system first makes use of a tracking device [MCK13] to obtain the continuous information on workers' positions and orientations over time by simultaneously tracking 100 bees. The tracking device as shown in Fig. 6.1a consists of a high resolution camera (Cam), an infrared lighting system (LS) and the observation hive (OH). Each each bee is attached with a 2D barcode for the tracking device to recognize (Fig. 6.1b,c,d). The obtained information is then used to calculate behavioural and social features which are in turn used to train behaviour classifiers by an interactive machine learning framework JAABA [Kab+13].

BBAS accurately classified encounter behaviours between worker bees, which are head-to-head quick contacts among bees (Fig. 6.1d). As is shown in Table 6.1, of the encounter behaviours that were manually annotated, 93% were accurately detected. Even though the trained classifier may not detect 7% of the encounter behaviours, the large number of behaviours of the many worker bees that can be detected over multiple days of observation produces a reliable test sensitivity. This statistical power will support the identification of even tiny differences between internal physiological states or the effects of experimental manipulation. According to the manual annotations, the system falsely classified other behaviours as encounter behaviours. Of these false detections, 13% had no similarity to encounter behaviours, whereas 15% had a close similarity to encounter behaviours, possibly suggesting that the classifier can detect a broader spectrum of encounter and encounter-related behaviours than can be manually annotated. These borderline cases may have a similar biological function and require further investigation.

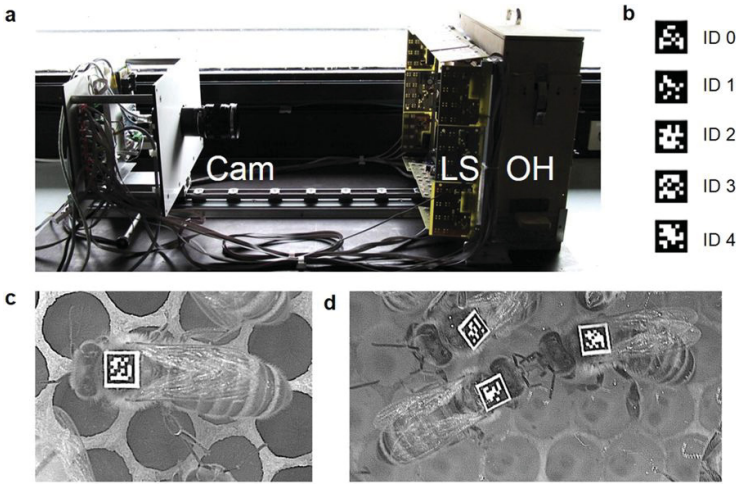


Fig. 6.1: Setup of the tracking device. (a) The tracking device consisted of a high-resolution camera (Cam), an infrared lighting system (LS) and the observation hive holding one "Deutsch Normal" comb (OH). The entire device was placed under a cardboard box in a laboratory. (b) Examples of 2D barcodes. (c) Bee marked with a tag bearing a 2D barcode. (d) Encounter behaviour between two worker bees defined by the head to head orientation and the antennal contact of the interacting bees.

Tab. 6.1: The result of the automatic detection of encounter behaviours by BBAS.

		Automatic annotations by "encounter classifier"	
		Encounter (%)	Non-encounter (%)
Training set	Encounter	100	0
	Non-encounter	0	100
Test set	Encounter	93	7
	Non-encounter	28[(1)]	n.d. [(2)]

[(1)] Percentage of falsely annotated encounter behaviors to all annotations

[(2)] Not determined (n.d.) since non-encounter behaviors were not manually labeled.

6.2 Automated computer-based detection of encounter behaviours in groups of honeybees

Publication status

Blut, Christina, Alessandro Crespi, Danielle Mersch, Laurent Keller, Linlin Zhao, Markus Kollmann, Benjamin Schellscheidt, Carsten Fülber, and Martin Beye. *"Automated computer-based detection of encounter behaviours in groups of honeybees."* Scientific reports 7, no. 1 (2017): 17663.

Linlin Zhao's contributions

1. Adapted the published program JAABA [Kab+13] originally for flies to be capable of applying to bees.
2. Processed the tracking data from the tracking device to generate input data for JAABA.
3. Generated extra features which were unique for bee societies.
4. Processed the continuous annotation results from JAABA.

SCIENTIFIC REPORTS

OPEN

Automated computer-based detection of encounter behaviours in groups of honeybees

Received: 26 May 2017
Accepted: 1 December 2017
Published online: 15 December 2017

Christina Blut[1], Alessandro Crespi[2], Danielle Mersch[3], Laurent Keller[4], Linlin Zhao[5], Markus Kollmann[5], Benjamin Schellscheidt[6], Carsten Fülber[6] & Martin Beye[1]

Honeybees form societies in which thousands of members integrate their behaviours to act as a single functional unit. We have little knowledge on how the collaborative features are regulated by workers' activities because we lack methods that enable collection of simultaneous and continuous behavioural information for each worker bee. In this study, we introduce the Bee Behavioral Annotation System (BBAS), which enables the automated detection of bees' behaviours in small observation hives. Continuous information on position and orientation were obtained by marking worker bees with 2D barcodes in a small observation hive. We computed behavioural and social features from the tracking information to train a behaviour classifier for encounter behaviours (interaction of workers via antennation) using a machine learning-based system. The classifier correctly detected 93% of the encounter behaviours in a group of bees, whereas 13% of the falsely classified behaviours were unrelated to encounter behaviours. The possibility of building accurate classifiers for automatically annotating behaviours may allow for the examination of individual behaviours of worker bees in the social environments of small observation hives. We envisage that BBAS will be a powerful tool for detecting the effects of experimental manipulation of social attributes and sub-lethal effects of pesticides on behaviour.

Honeybees, like other eusocial insects, form societies in which their members integrate their behaviours to form a single functional unit (often described as 'superorganisms')[1]. In honeybee colonies, for example, the brood is collectively reared by the worker bees under constant temperature conditions in worker-made and well-structured wax combs[2]. We still have little knowledge on how the collaborative features are regulated within the colony by single workers' task engagements, worker-worker interactions and environmental cues.

A honeybee may engage in many behavioural tasks, for example, cell cleaning, brood feeding, comb building, pollen and nectar storing, and foraging[3]. The many in-hive tasks are usually performed within the first three weeks of their life, whereas foraging tasks are performed later[3]. Individual task engagements are flexible and are adapted according to the colony's needs[4,5]. Differences in individuals' internal response thresholds for task-specific stimuli (response threshold model)[6–8], actively seeking for tasks (foraging for work model)[9], repeatedly performing the same task when being successful at it (self-reinforcement models)[8,10] and information transferred by social partners through direct contact[11] may play an important role in the organisation of task engagements within the colony.

Gaining continuous behavioural information on each single worker, their direct contacts (encounters) to other worker bees and their interactions with the local environment would facilitate the further characterization of the underlying mechanisms of colony organization. However, we currently lack methods that enable the collection of simultaneous and continuous behavioural information for each individual worker bee in the environment of a colony[12]. In current methods, behaviours are manually detected by an observer either from video recordings of small observation hives or from direct observations[3,13–15]. These manually detected behaviours represent only

[1]Evolutionary Genetics, Heinrich-Heine University, Düsseldorf, Germany. [2]Biorobotics Laboratory (BioRob), École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland. [3]Neurobiology, MRC Laboratory of Molecular Biology, University of Cambridge, Cambridge, United Kingdom. [4]Department of Ecology and Evolution, Université de Lausanne, Lausanne, Switzerland. [5]Mathematical Modelling of Biological Systems, Heinrich-Heine University, Düsseldorf, Germany. [6]Faculty of Electrical Engineering & Information Technology, University of Applied Sciences, Düsseldorf, Germany. Correspondence and requests for materials should be addressed to C.B. (email: blut@hhu.de) or M.B. (email: martin.beye@hhu.de)

a fraction of the behaviours that the many worker bees can display in a colony, especially when the behaviour is frequently performed, for example, in the case of encounter behaviours.

In honeybees, encounter behaviours between workers are characterized by the following: the two worker bees face each other head to head and their moving antennae are repeatedly in contact. Encounter behaviours summarize different worker-worker interaction behaviours that display constant antennal contact and can be further grouped into the following behaviours: (i) antennation behaviour, which is required to initialize and maintain a contact[16], whereby the antennae of two worker bees are in constant contact but no other features of the following behaviours are displayed; (ii) begging behaviour, in which a worker bee begs for food from another nestmate;[16,17] (iii) offering behaviour, in which a worker bee offers food to another nestmate;[17] and, (iv) trophallaxis behaviour, in which nectar from the crop is exchanged between two bees[18,19].

Worker bees may perform begging behaviour to gain information about the quality and source of nectar offered by the incoming forager bees[18,20–22]. Incoming forager bees perform offering behaviour to unload the collected nectar to a recipient in-hive worker bee via trophallaxis[20,23–25]. Returning foragers presenting high-quality nectar show increased offering behaviour as well as increased dancing behaviour[26]. They more often find a recipient bee and will more often return with nectar to the colony. Effects of different nectar qualities on worker-worker interaction establish a control mechanism for the workers' foraging engagement, performance and the influx of high-quality nectar[27]. Despite their role in regulating workers' foraging engagement and performance[23,28], we have little knowledge on other possible roles that these encounter behaviours may have in task engagements and colony organization.

In this study, we introduce the Bee Behavioral Annotation System (BBAS), which enables the automated classification of worker-worker encounters within a group of honeybees. We obtained continuous information on workers' positions and orientations over time by simultaneously tracking 100 bees tagged with a 2D barcode by adapting a tracking device that was developed for ants[29]. From this tracking information, behavioural and social features were computed, and a behaviour classifier was trained based on machine learning using the Janelia Automatic Animal Behavior Annotator (JAABA) program[30]. Our study demonstrates that we can automatically and accurately classify encounter behaviours within a group of bees. This system has the prospect of automatically obtaining quantitative and continuous behavioural information on hundreds of bees at once in small colonies.

Results

Automatic classification of encounter behaviours in a group of worker bees. To automatically classify worker behaviours in a small observation hive, we developed the BBAS. We obtained tracking information from individual worker bees in a small group and computed behavioural features (per-frame features), which were utilized to classify behaviours. Per-frame features represented parameters calculated from the tracking information that provided information on the bees' behavioural properties in each frame. Such properties included, for example, a bee's speed or orientation towards a nestmate (see Kabra *et al.*[30] for a detailed listing of per-frame features). We applied the per-frame features to manually labelled behaviour classes to train a machine learning-based system and thus generate an automatic behaviour classifier.

First, we adapted a tracking device developed for ants[29] to obtain information on the position and orientation of individually tagged bees at a rate of four frames per second. In our setting, we tracked 100 newly emerged worker bees for two days. Bees were individually tagged with 2D barcodes from the AprilTags library[31] printed on 2×2 mm tags and housed in a small observation hive on a single comb providing food (Fig. 1a–c). We chose a rate of four frames per second to ensure that we obtained sufficient information on the bees' position and orientation for subsequent use in automatic behaviour classification. To test whether the chosen rate captured sufficient information we determined the average change in posiotion and orientation of bees (see Supplementary information online). On average, bees' positions changed by 0.9 mm (SD $\pm$ 0.9 mm) from one frame to another, which corresponds to ~0.06% of an *Apis mellifera* worker size. Bees' average change in orientation from one frame to another was 6° (SD $\pm$ 4°). These small changes in position and orientation suggest that we can capture sufficiently detailed information on the bees' movements with the chosen rate. The AprilTag system was chosen because it is an actively maintained open source project and provides a robust system to minimize inter-tag confusion. It also has better performance on images taken under non-uniform lighting conditions as compared to several other similar systems[31].

The results of the detection rate and positional accuracy of the tracking device of immobile tags glued to a comb and tags attached to moving and resting worker bees are summarized in Table 1. On average, resting bees were detected in 98.2% of the frames, whereas moving bees were detected in 90.8% of the frames. The orientation accuracy of immobile tags glued to a comb was 1.5° and the positional accuracy was 0.04 mm. The high detection rate and positional accuracy suggest that we can obtain a considerable amount of detailed information on the movement of each single worker in a group of bees.

Second, to generate an automatic behaviour classifier, we computed per-frame features from the tracking information using the JAABADetect program[30]. Computing the per-frame features for the tracking information on 100 worker bees required a high-performance computing cluster. We used the social per-frame features to train a classifier for honeybee encounter behaviours[30]. The social per-frame features are a set of per-frame features providing information on an individual's state in each frame in relation to its nearest nestmates. For example, the distance, orientation and speed towards another worker may be described by these features (see Kabra *et al.*[30] for a detailed listing of social per-frame features).

Third, we determined whether we could automatically classify encounter behaviours between workers using an automatic behaviour classifier generated with the JAABA program. The automatic behaviour classifier was expected to classify the four different behaviours - antennation, begging, offering and trophallaxis - as a single class, which have the behavioural features of head to head orientation and antennal contact of two worker bees in common (Fig. 1d). To train the automatic behaviour classifier, we manually labelled 76 encounter behaviours

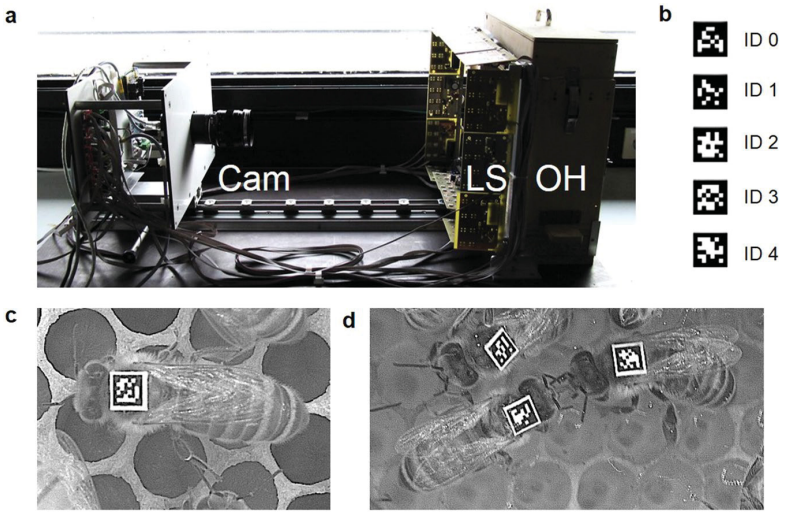


Figure 1. Setup of the tracking device. (**a**) The tracking device consisted of a high-resolution camera (Cam), an infrared lighting system (LS) and the observation hive holding one "Deutsch Normal" comb (OH). The entire device was placed under a cardboard box in a laboratory. (**b**) Examples of 2D barcodes from the AprilTags library. (**c**) Bee marked with a tag bearing a 2D barcode. (**d**) Encounter behaviour between two worker bees defined by the head to head orientation and the antennal contact of the interacting bees. This specific encounter shown is trophallaxis.

		No. of tracked tags	No. of frames analysed (sequence duration)[3]	Detection rate[4] (%)	x/y position accuracy[5] (mm ± SD)	Orientation accuracy[5] (degrees ± SD)
Tags glued to a comb	immobile	100	1200 (5 min)	99.9	0.04[6] ± 0.03	1.5 ± 0.8
Tags glued to a bee	resting[1]	10	240 (1 min)	98.2	n.d.[7]	n.d.[7]
	moving[2]	30	240 (1 min)	90.8	n.d.[7]	n.d.[7]

Table 1. Detection rate and positional accuracy of the tracking device. [1]Bee sits in one position without moving for ≥ 5 seconds. [2]Bee walks across the comb without interacting with other bees, inspecting cells or performing any other task. [3]Duration of the tracking. [4]The percentage of frames in which tags were detected. [5]Accuracy of the tracking device for the detected x/y centre position and the orientation. [6]i.e., ~0.003% of an *Apis mellifera* worker size. [7]Not determined (n.d.) because changes could result from the bees' behaviours.

and 77 non-encounter behaviours from 105 minutes of video recording and corresponding tracking information of the 100 tracked bees. We only labelled encounter behaviours of which we were highly confident that encounter behaviour was truly displayed. The 76 encounter behaviours (EBs) comprised a sample of 28 antennation, 8 begging, 6 offering and 34 trophallaxis behaviours (see Supplementary Videos V1-V4 online for examples of the four encounter behaviours). The non-encounter behaviours (NEBs) represented a sample of 46 sitting, 20 walking, 7 self-grooming, 1 social grooming and 3 sitting with subsequent walking behaviours. We trained the classifier by entering the 76 EBs and 77 NEBs (training set) bit by bit into the JAABA program in five training rounds until we observed no further improvement in the cross-validation estimates (see Supplementary information online for details on cross-validation). Cross-validation estimates were obtained by randomly splitting the EBs and NEBs from the training set into testing and training subsets. The training subset was used to train the classifier while the testing subset was used to subsequently estimate the classifier's error rate on classifications[30]. Table 2 presents the final cross-validation estimates from 10 cross-validation rounds for our trained 'encounter classifier'. The estimates represent the percentage of frames automatically classified as EB* and NEB* by the 'encounter classifier' (asterisks indicate automatically classified behaviours; see Supplementary information online for details on calculation of estimates). Of the EB frames, 77.3% were correctly classified by our 'encounter classifier' (SD ± 1.3%, Table 2), whereas 73.7% of the NEB frames were correctly classified (SD ± 1.2%, Table 2). The false positive rate was 26.3% (NEB frames falsely classified as EBs*), whereas the false negative rate was 22.7% (EB frames falsely classified as NEBs*; Table 2).

Next, we examined whether our classifier was able to correctly classify all 76 manually labelled EBs from our training set. Since the training set included the different behaviour classes - antennation, begging, offering and trophallaxis - we examined whether the classifier could correctly classify these four different behaviours as encounter behaviour. We determined the classification rate and observed that all manually labelled encounter behaviours of the training set were correctly detected by our classifier (training set in Table 3; Supplementary Table S1).

		Automatically detected by the 'encounter classifier'	
		Encounter (EB*)[6] (±SD) (%)[2]	Non-encounter (NEB*)[6] (±SD) (%)[2]
Manually annotated[1]	Encounter (EB)	77.3 (±1.3)[3]	22.7 (±1.3)[5]
	Non-encounter (NEB)	26.3 (±1.3)[4]	73.7 (±1.2)[3]

Table 2. The accuracy of the trained 'encounter classifier' estimated through cross-validation on the labelled frames for EBs and NEBs. [1]The manually labelled high-confidence behaviours (EBs and NEBs) used to train the classifier. [2]Mean estimates with standard deviation (SD) of the 10 rounds of cross-validation. Estimate values are given as percentage of frames correctly or falsely classified as EBs or NEBs using the classifier. [3]Frames correctly classified as EB or NEB (true positives). [4]NEB frames falsely classified as EB* (false positives). [5]EB frames falsely classified as NEB* (false negatives). [6]Asterisks indicate automatically classified behaviours.

		Automatically detected by the 'encounter classifier'	
		Encounter (EB*) (%)	Non-encounter (NEB*) (%)
Training set[1]	Encounter (EB)	100	0
	Non-encounter (NEB)	0	100
Testing set[2]	Encounter (EB)	93	7
	Non-encounter (NEB)	28[3]	n.d.[4]

Table 3. Comparison of manually annotated behaviours (EBs and NEBs) and automatically classified behaviours (EBs* and NEBs*). [1]The manually labelled high-confidence behaviours (EBs and NEBs) used to train the classifier [2]Manually annotated behaviours not used to train the classifier [3]Automatically detected behaviours falsely classified as EB* by the 'encounter classifier' [4]not determined (n.d.) because we did not manually annotate NEBs for the testing set and thus could not determine the automatic classification rate.

We then determined the accuracy of our 'encounter classifier' by comparing manual annotations and automatic classifications of behaviours that were not included in our initial training set. We manually annotated 43 encounter behaviours comprising 4 trophallaxis, 8 begging, 12 offering and 19 antennation behaviours (testing set; see Supplementary Table S1). Our 'encounter classifier' detected 93% of the manually annotated encounter behaviours in this testing set. The false negative rate was 7%, whereas 28% of the automatically detected behaviours were falsely classified as EBs* (testing set in Table 3; Supplementary Table S1). We re-examined the falsely classified EBs* and found that 15% of the 28% falsely classified EBs* displayed similar features to those of encounter behaviours, i.e. head to head orientation and proximity of two bees. However, these falsely classified EBs* collectively lacked antennal contact. Of the behaviours falsely classified as encounters, 13% were unrelated to encounter behaviour, i.e. displayed no features characterizing encounter behaviours. The results on the high classification rates suggest that the BBAS can be used to automatically and accurately annotate encounter behaviours in groups of honeybees.

Classification of trophallaxis behaviour based on the duration of the encounter behaviour. We demonstrated that we could automatically classify the different encounter behaviours, antennation, begging, offering and trophallaxis, as a single behavioural class with our 'encounter classifier'. Next, we considered whether we could use the duration of the different encounter behaviours to distinguish these from each other. In 105 minutes of the 22 hours of video recording, we measured the frequency and duration of antennation, begging, offering and trophallaxis behaviours in the group of 100 worker bees.

We manually detected 658 encounter behaviours from which 57% were antennation behaviours, 26% were offering behaviours, 9% were begging behaviours and 8% trophallaxis behaviours (Table 4; Supplementary Videos V1-V4 online). The median duration of the trophallaxis behaviours was 8 seconds (75% percentile: 13 seconds; range of duration: 5–30.5 seconds; Table 4; Fig. 2a). The median duration of antennation, offering and begging behaviours was much shorter, ranging from 1 to 2 seconds with a considerable overlap in the 75% percentile (range of durations: antennation: 0.25–9.25 seconds, offering: 0.25–4.5 seconds, begging: 0.75–6.75 seconds; Table 4; Fig. 2b-d). There was a significant difference between the duration of the four different encounter behaviours (One Way ANOVA on Ranks: $N = 658$, $\alpha = 0.05$, $H = 175$, d.f. $= 3$, $P = < 0.001$). Post hoc tests showed that pairwise comparisons were significantly different except for begging vs. antennation behaviours (Dunn's Method, $\alpha = 0.05$: trophallaxis vs. offering: $N = 222$, $Q = 13$, $P < 0.001$; trophallaxis vs. antennation: $N = 427$, $Q = 10.7$, $P < 0.001$; trophallaxis vs. begging: $N = 109$, $Q = 6.7$, $P < 0.001$; begging vs. offering: $N = 231$, $Q = 5.3$, $P < 0.001$; antennation vs. offering: $N = 549$, $Q = 5.2$, $P < 0.001$; begging vs. antennation: $N = 436$, $Q = 2.3$, $P = 0.138$). This result suggests that the duration of encounter behaviours could be utilized to distinguish the different encounter behaviours from each other.

Next, we tested whether encounter behaviours could be accurately classified as antennation, begging, offering or trophallaxis based solely on their duration. Therefore, we analysed the ranges of duration of the 658 encounters from the four behaviour classes to determine whether duration thresholds could be used as classifier for the different

Encounter behaviour	No. of encounters	Relative proportion (%)	Min. duration (sec)	Max. duration (sec)	Median (sec)	75% percentile (sec)
Antennation	377	57	0.25	9.25	1.8	2.5
Offering	172	26	0.25	4.5	1	1.9
Begging	59	9	0.75	6.75	2	3
Trophallaxis	50	8	5	30.5	8.4	12.9

Table 4. Frequency and duration of the different manually detected encounter behaviours.

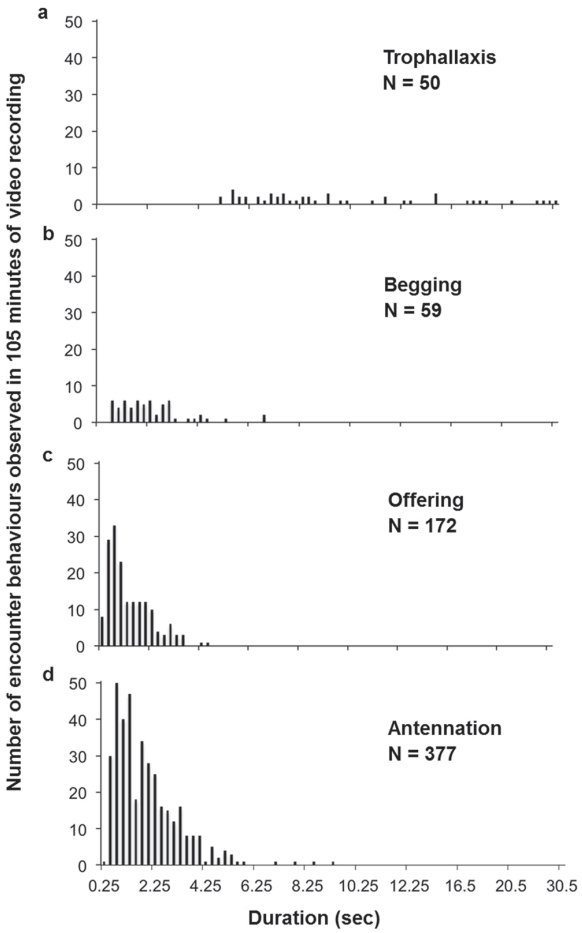


Figure 2. Number of encounter behaviours observed for the different duration of encounter behaviours from the four behaviour classes. (**a**) Trophallaxis, (**b**) Begging, (**c**) Offering, (**d**) Antennation.

encounter behaviours. Hereby, we attempted to find thresholds above which behaviours could be reliably classified as one of the four behaviour classes. We observed that duration thresholds could not be utilized as classifiers for begging, offering and antennation behaviours since their ranges of duration overlapped too strongly (Table 4; Fig. 2). When considering only behaviours with duration of 5 and more seconds, we observed that all trophallaxis behaviours could be correctly classified (100%; Table 5). Non-trophallaxis behaviours (i.e. begging and antennation behaviours), however, were falsely classified as trophallaxis behaviours with a false positive rate of 8% (Table 5).

We then tested whether trophallaxis behaviours could be automatically classified based on their duration. We applied the duration threshold of ≥ 5 seconds to the automatically classified EBs* from the testing set comprising 43 encounter behaviours. We observed that 100% of the trophallaxis behaviours were automatically classified (Table 5). However, 28% of the detected behaviours were falsely classified as trophallaxis (false positive rate; Table 5). These classification rates suggest that we can automatically classify the vast majority of trophallaxis behaviours in a group of worker honeybees using our 'encounter classifier' together with the duration threshold of ≥ 5 seconds.

	Manually classified by duration among the 658 manually detected behaviours[1]		Automatically classified by duration among the EBs* from the testing set[2]	
	Trophallaxis (%)[3]	Non-trophallaxis (%)[3]	Trophallaxis* (%)[4]	Non-trophallaxis* (%)[4]
Trophallaxis[5]	100	0	100	0
Non-trophallaxis[5]	8	92	28	72

Table 5. The classification of trophallaxis behaviours of manually detected and automatically detected encounter behaviours using the duration threshold of ≥ 5 seconds. [1]We manually classified trophallaxis behaviours from the 658 manually detected encounter behaviours using the duration threshold of ≥ 5 seconds. [2]We applied the 'encounter classifier' with the duration threshold of ≥ 5 seconds to the 43 manually annotated encounter behaviours not used for training. [3]Percentage of the manually detected trophallaxis and non-trophallaxis behaviours that were manually classified as trophallaxis using the duration threshold of ≥ 5 seconds. [4]Percentage of the manually annotated trophallaxis and non-trophallaxis behaviours from the testing set that were automatically classified as trophallaxis* and non-trophallaxis* (asterisks indicate automatic classification) using the duration threshold of ≥ 5 seconds. [5]Trophallaxis and non-trophallaxis behaviours that were manually annotated by the observer.

Discussion

We introduced the BBAS, a system that can automatically classify stereotypical behaviours of individual workers in a group of honeybees. Our results show that the BBAS can be reliably used to automatically detect encounter behaviours.

Current behavioural observation methods usually require the manual detection of behaviours by an observer[12]. Manual detection limits the number of observable behaviours, especially when the behaviour is frequently displayed by the many worker bees in a colony, as is the case for encounter behaviours. In this study, we accurately classified encounter behaviours between worker bees using automatic classification. Of the encounter behaviours that were manually annotated, 93% were accurately detected. Even though our classifier may not detect 7% of the encounter behaviours, the large number of behaviours of the many worker bees that can be detect over multiple days of observation produces a reliable test sensitivity. This statistical power will support the identification of even tiny differences between internal physiological states or the effects of experimental manipulation. According to the manual annotations, our classifier falsely classified other behaviours as encounter behaviours. Of these false detections, 13% had no similarity to encounter behaviours, whereas 15% had a close similarity to encounter behaviours, possibly suggesting that our classifier can detect a broader spectrum of encounter and encounter-related behaviours than can be manually annotated. These borderline cases may have a similar biological function and require further investigation.

In this study, the duration of the four different classes of encounter behaviours – trophallaxis, begging, offering and antennation - was obtained from 100 same-aged bees kept in a one-frame observation hive without a queen and brood. Our results showed that trophallaxis behaviours lasted between 5 and 30.5 seconds. The duration of offering and begging behaviours ranged from 0.25 to 6.75 seconds while antennation lasted 0.25 to 9.25 seconds. These measurements correspond to previous reports on the duration of trophallaxis, begging and offering behaviour that were obtained under more natural conditions (queenright colonies in one - or two-frame observation hives[17,19,26]). Trophallaxis behaviours of different aged worker bees in these small queenright colonies lasted 4 to 30 seconds while begging and offering lasted less than 0.5 to 10 seconds[17,19,26]. This constancy under different conditions suggests that duration can possibly be used as a predictive parameter to distinguish among the behavioural classes of encounters.

Our survey of manually annotated encounter behaviours suggests that a duration threshold of ≥ 5 seconds for an encounter behaviour can be used to accurately separate trophallaxis behaviour from the other encounter behaviours (begging, offering and antennation). When we applied our 'encounter classifier' together with the duration threshold, we were able to classify 100% of the manually annotated trophallaxis behaviours. However, the false positive rate was relatively high (28%), suggesting that we may need further adjustments of the behaviour duration parameter to reduce false classifications.

It has been proposed that encounter behaviours and the transmission of food are ways for worker bees to gather information about their colony's state and thus can adjust their behaviours according to the colony's needs[32–35]. So far, we have detailed knowledge on the role of trophallaxis, begging and offering behaviours between incoming foragers and worker bees inside the colony in accessing information about the quality and source of nectar and the honey stores of the colony. Foraging worker bees usually unload the nectar from the honey crop to the in-hive worker bees via trophallaxis[18,23,36]. The recipient worker bees store the nectar within the wax cells or further reduce the water content. Offering behaviour is performed by the returning nectar foragers, which are willing to unload their crop contents to a recipient worker bee[17]. Inside the nest, worker bees beg incoming forager bees to receive nectar[17,22,23,37]. The rate of begging behaviour is affected by the colony's state and the amount of stored honey in the colony[38]. Antennation behaviour is essential in making and maintaining the contact between encountering bees[16,20]. We envisage that with more classifiers trained for other behaviours, we can further examine the possible effects of encounter behaviours on subsequent task engagement.

For training the classifier and for measuring the accuracy of detection, we used 100 tagged worker bees in this study. However, with the current setup the BBAS can track up to 1000 worker bees on a brood comb in a small observation hive (preliminary data). It can be further scaled up to over 2000 worker bees by adding an additional camera, lighting system and cluster of five computers. Hence, we suggest that the BBAS will enhance our ability to gather knowledge on worker bees' individual and collective behaviours. With more classifiers trained to detect

different behaviours in honeybees, the BBAS can be used to examine single-worker behavioural phenotypes and worker-worker interactions within small observation hives. We envisage that the BBAS will be a powerful tool to detect the experimental effects of genetic and physiological manipulations on single workers[39,40]. Additionally, we propose that the BBAS can be an accurate method for measuring the sub-lethal effects of pesticides on behaviour[41]. The key to understanding the effects of pesticides on honeybee colonies is gaining knowledge on how these influence individual behaviour. With the BBAS we will be able to analyse the effects of pesticides on individual behaviour because we can continuously and simultaneously quantify the in-hive behaviours of hundreds of worker bees under standardized conditions with computer-based classifiers. For encounter behaviours, for example, we can analyse the effects of pesticides on the duration of encounters or their quantity.

In conclusion, we foresee that the BBAS will be beneficial in various research areas for honeybee researchers who need to obtain detailed behavioural information of hundreds of individual bees.

Methods

Tracking device and procedure. Video recordings of worker bees on a comb and tracking information were obtained with a tracking device that was developed for ants by Mersch *et al.*[29] and modified for tracking honeybees (see Supplementary information online). The honeybee tracking device consisted of a monochrome high-resolution camera, a cluster of five desktop computers, an infrared lighting system and an observation hive holding a single "Deutsch Normal" comb (Fig. 1a). The infrared light was provided in flashes synchronized with the images taken every quarter second (4 frames per second) by the camera. To omit daylight exposure, both the observation hive and the camera stood in a laboratory covered by a cardboard box that was lined with infrared-reflecting foil, which intensified the infrared illumination of the comb area. The cardboard box was equipped with a ventilation device that kept the temperature at approximately 29 °C (±1 °C).

We used 2 × 2 mm tags bearing 2D barcodes from the AprilTags library (Fig. 1b)[31] to tag and track honeybee workers. These 2D barcodes consisted of a square outline with a 36-bit code word encoded in the interior, which could generate up to 2320 unique identification (ID) numbers. An experiment on mortality and behavioural observations of tagged bees showed that bees bearing tags survived and behaved as untagged bees did (see Introductory experiments and observations in Supplementary information online). The tracking information obtained by the tracking software[29] contained (after postprocessing) the tag's ID number, the x- and y-coordinates of its centre and its orientation with the corresponding frame number and timestamp in UNIX time (with a precision of 1/100 seconds).

Automatic behaviour classification using the tracking information. From the tracking information, we used the JAABADetect program[30] to compute social per-frame features to provide information on the bees' properties in relation to their nearest nestmate in each frame (for example, the distance, speed, and orientation to the closest bee; see Kabra *et al.*[30] for a detailed listing of social per-frame features).

To produce the 'encounter classifier', we labelled examples of encounter and non-encounter behaviour in 105 minutes of tracking information and video material using the graphical user interface of the JAABA program[30]. We only labelled encounter and non-encounter behaviours for which we had high confidence in classification. Thus, for encounter behaviours we only labelled those for which we could confidently identify that behavioural features characterizing encounter behaviours were displayed. Information about the social per-frame features that were computed from the tracking information was used to train the 'encounter classifier' via machine learning implemented in the JAABA program[30].

The classifier's accuracy was determined using the cross-validation method implemented in the JAABA program[30]. We used JAABA's default settings for the cross-validation and performed 10 cross-validation rounds to obtain an average estimate on the classifier's accuracy (see Supplementary information online for more details on calculation of accuracy and cross-validation).

Manual annotation of encounter behaviours and further analysis. We manually examined the video recordings to detect all encounter behaviours. We determined the duration in seconds and the type of encounter behaviour: i) antennation behaviour, ii) begging behaviour, iii) offering behaviour, iv) trophallaxis behaviour.

Statistical analyses were performed using the SigmaPlot 13 software.

Bee handling. We used newly emerged honeybees that originated from a colony of western honeybee *Apis mellifera* from our bee yard at the Heinrich-Heine University of Düsseldorf, Germany. A sealed brood comb was taken from the source colony and incubated at 34 °C. Emerging worker bees were collected when they were 0–24 hours old. One hundred bees were marked with hand-cut tags by gluing these centrally on the thorax of the bees with glue ("Opalith Zeichenleim", Heinrich Holtermann KG, Brockel, Germany).

The bees were tracked from May 3rd to May 4th, 2016 on a comb comprising 40 capped cells filled with honey. Bees were restricted to one side of the comb without a queen. As worker-worker encounters were the interest of this study, neither a queen nor drones were included in the group. The comb did not contain brood because we used newly emerged worker bees for tracking, and it is known that brood rearing first begins at an age of two to three days[3,24].

To ensure that sufficient encounter behaviours occurred during the tracking process, a proportion of the bees were either fed ad libitum with a sugar solution (Ambrosia Bienenfutter-Sirup, Nordzucker AG, Braunschweig, Germany) or starved before tracking was started. On the first day of tracking, 16 bees were fed with the sugar solution before starting the tracking experiment, whereas the remaining bees were starved for approximately an hour. For sustenance, we provided the bees with a sugar pastry (Apifonda Futterteig, Südzucker AG, Mannheim, Germany) two hours after tracking was started. On the second day of tracking, we removed the sugar pastry and

fed 15 of the 100 bees again with the sugar solution. The other 85 bees were starved for three hours. The 15 bees were reintroduced into the observation hive before tracking began. In total, information from 22 hours of tracking was generated for 96 bees. Four bees lost their tags during tracking.

Data availability. The datasets generated and analysed during the current study are available from the corresponding author on reasonable request. Programs developed for this study will be shared and can be requested from the corresponding author.

References

1. Hölldobler, B. & Wilson, E. O. *The superorganism: the beauty, elegance, and strangeness of insect societies*. 1st edn, (W. W. Norton, 2009).
2. Winston, M. L. *The biology of the honey bee*., (Harvard University Press, 1987).
3. Seeley, T. D. Adaptive significance of the age polyethism schedule in honeybee colonies. *Behav Ecol Sociobiol* **11**, 287–293, https://doi.org/10.1007/Bf00299306 (1982).
4. Page, R. E. & Erber, J. Levels of behavioral organization and the evolution of division of labor. *Naturwissenschaften* **89**, 91–106, https://doi.org/10.1007/s00114-002-0299-x (2002).
5. Gordon, D. M. From division of labor to the collective behavior of social insects. *Behav Ecol Sociobiol* **70**, 1101–1108, https://doi.org/10.1007/s00265-015-2045-3 (2016).
6. Page, R. E., Robinson, G. E., Fondrk, M. K. & Nasr, M. E. Effects of worker genotypic diversity on honey bee colony development and behavior (*Apis mellifera* L). *Behav Ecol Sociobiol* **36**, 387–396, https://doi.org/10.1007/Bf00177334 (1995).
7. Bonabeau, E., Theraulaz, G. & Deneubourg, J. L. Fixed response thresholds and the regulation of division of labor in insect societies. *B Math Biol* **60**, 753–807, https://doi.org/10.1006/bulm.1998.0041 (1998).
8. Beshers, S. N. & Fewell, J. H. Models of division of labor in social insects. *Annu Rev Entomol* **46**, 413–440, https://doi.org/10.1146/annurev.ento.46.1.413 (2001).
9. Tofts, C. Algorithms for task allocation in ants - (a study of temporal polyethism-theory). *B Math Biol* **55**, 891–918, https://doi.org/10.1007/Bf02460691 (1993).
10. Theraulaz, G., Bonabeau, E. & Deneubourg, J. L. Response threshold reinforcement and division of labour in insect societies. *P Roy Soc B-Biol Sci* **265**, 327–332, https://doi.org/10.1098/rspb.1998.0299 (1998).
11. Beshers, S. N., Huang, Z. Y., Oono, Y. & Robinson, G. E. Social inhibition and the regulation of temporal polyethism in honey bees. *J Theor Biol* **213**, 461–479, https://doi.org/10.1006/jtbi.2001.2427 (2001).
12. Scheiner, R. *et al*. Standard methods for behavioural studies of *Apis mellifera*. *J Apicult Res* **52**, 1–58, https://doi.org/10.3896/Ibra.1.52.4.04 (2013).
13. Lindauer, M. Ein Beitrag zur Frage der Arbeitsteilung im Bienenstaat. *Z. vergl Physiol* **34**, 299–345, https://doi.org/10.1007/Bf00298048 (1952).
14. Frisch, K. v. Über die "Sprache" der Bienen, eine tierpsychologische Untersuchung. *Zool Jb Physiol* **40**, 1–186 (1923).
15. Gempe, T., Stach, S., Bienefeld, K. & Beye, M. Mixing of honeybees with different genotypes affects individual worker behavior and transcription of genes in the neuronal substrate. *Plos One* **7**(2), e31653, https://doi.org/10.1371/journal.pone.0031653 (2012).
16. Free, J. B. A study of the stimuli which release the food begging and offering responses of worker honeybees. *Br J Anim Behav* **4**, 94–101, https://doi.org/10.1016/S0950-5601(56)80129-9 (1956).
17. De Marco, R. J. & Farina, W. M. Trophallaxis in forager honeybees (*Apis mellifera*): resource uncertainty enhances begging contacts? *J Comp Physiol A* **189**, 125–134, https://doi.org/10.1007/S00359-002-0382-Y (2003).
18. Goyret, J. & Farina, W. M. Trophallactic chains in honeybees: a quantitative approach of the nectar circulation amongst workers. *Apidologie* **36**, 595–600, https://doi.org/10.1051/apido:2005050 (2005).
19. Korst, P. J. A. M. & Velthuis, H. H. W. The nature of trophallaxis in honeybees. *Insect Soc* **29**, 209–221, https://doi.org/10.1007/Bf02228753 (1982).
20. Goyret, J. & Farina, W. M. Descriptive study of antennation during trophallactic unloading contacts in honeybees *Apis mellifera carnica*. *Insect Soc* **50**, 274–276, https://doi.org/10.1007/s00040-003-0678-0 (2003).
21. Gil, M. & De Marco, R. J. Olfactory learning by means of trophallaxis in *Apis mellifera*. *J Exp Biol* **208**, 671–680, https://doi.org/10.1242/Jeb.01474 (2005).
22. Bozic, J. & Valentincic, T. Attendants and followers of honey bee waggle dances. *J Apicult Res* **30**, 125–131, https://doi.org/10.1080/00218839.1991.11101246 (1991).
23. Frisch, K. v. *Tanzsprache und Orientierung der Bienen*. (Springer, 1965).
24. Rösch, A. G. Untersuchungen über die Arbeitsteilung im Bienenstaat. 1. Teil: Die Tätigkeiten im normalen Bienenstaate und ihre Beziehungen zum Alter der Arbeitsbienen. *Z vergl Physiol* **2**, 571–631 (1925).
25. Nixon, H. L. & Ribbands, C. R. Food transmission within the honeybee community. *Proc R Soc Ser B-Bio* **140**, 43–50, https://doi.org/10.1098/rspb.1952.0042 (1952).
26. De Marco, R. J. & Farina, W. M. Changes in food source profitability affect the trophallactic and dance behavior of forager honeybees (*Apis mellifera* L.). *Behav Ecol Sociobiol* **50**, 441–449, https://doi.org/10.1007/s002650100382 (2001).
27. Seeley, T. D., Camazine, S. & Sneyd, J. Collective decision-making in honey bees: how colonies choose among nectar sources. *Behav Ecol Sociobiol* **28**, 277–290, https://doi.org/10.1007/BF00175101 (1991).
28. Farina, W. M. & Nunez, J. A. Trophallaxis in the honeybee, *Apis mellifera* (L) as related to the profitability of food sources. *Anim Behav* **42**, 389–394, https://doi.org/10.1016/S0003-3472(05)80037-5 (1991).
29. Mersch, D. P., Crespi, A. & Keller, L. Tracking individuals shows spatial fidelity is a key regulator of ant social organization. *Science* **340**, 1090–1093, https://doi.org/10.1126/science.1234316 (2013).
30. Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S. & Branson, K. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nature Methods* **10**, 64–67, https://doi.org/10.1038/nmeth.2281 (2013).
31. Olson, E. AprilTag: A robust and flexible visual fiducial system. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 3400–3407, https://doi.org/10.1109/ICRA.2011.5979561 (2011).
32. Ribbands, C. R. *The behaviour and social life of honeybees*. (Bee Research Association, 1953).
33. Farina, W. M. Food-exchange by foragers in the hive - A means of communication among honey bees? *Behav Ecol Sociobiol* **38**, 59–64, https://doi.org/10.1007/S002650050217 (1996).
34. Seeley, T. D. Social foraging by honeybees - How colonies allocate foragers among patches of flowers. *Behav Ecol Sociobiol* **19**, 343–354, https://doi.org/10.1007/Bf00295707 (1986).
35. Lindauer, M. Über die Einwirkung von Duft- und Geschmacksstoffen sowie anderer Faktoren auf die Tänze von Bienen. *Z vergl Physiol* **31**, 348–412 (1948).
36. Seeley, T. D. *The wisdom of the hive: The social physiology of honey bee colonies*. (Harvard University Press, 1995).
37. Farina, W. M. & Wainselboim, A. J. Trophallaxis within the dancing context: a behavioral and thermographic analysis in honeybees (*Apis mellifera*). *Apidologie* **36**, 43–47, https://doi.org/10.1051/Apido:2004069 (2005).

38. Schulz, D. J., Vermiglio, M. J., Huang, Z. Y. & Robinson, G. E. Effects of colony food shortage on social interactions in honey bee colonies. *Insect Soc* **49**, 50–55, https://doi.org/10.1007/s00040-002-8279-x (2002).
39. Schulte, C., Theilenberg, E., Muller-Borg, M., Gempe, T. & Beye, M. Highly efficient integration and expression of piggyBac-derived cassettes in the honeybee (*Apis mellifera*). *Proc Natl Acad Sci USA* **111**, 9003–9008, https://doi.org/10.1073/pnas.1402341111 (2014).
40. Liang, Z. Z. S. *et al.* Molecular determinants of scouting behavior in honey bees. *Science* **335**, 1225–1228, https://doi.org/10.1126/science.1213962 (2012).
41. Charreton, M. *et al.* A locomotor deficit induced by sublethal doses of pyrethroid and neonicotinoid insecticides in the honeybee *Apis mellifera*. *Plos One* **10**, e0144879, https://doi.org/10.1371/journal.pone.0144879 (2015).

Acknowledgements

Computational support and infrastructure were provided by the "Centre for Information and Media Technology" (ZIM) at the University of Düsseldorf (Germany). We thank Mayank Kabra for advice on use of JAABA, Andreas Behrend for programming and technical support, Andre Buschhausen for the manufacture and assembly of the tracking device's electronics, Stephan Raub for assistance with the high performance-computing cluster, Eva Theilenberg and Marion Müller-Borg for assistance with bee handling and Chantal Brauer for assisting with data collection and training in JAABA.

Author Contributions

C.B. designed and performed the experiments and conducted the behaviour analyses. C.B. and M.B. conceived the study, supervised its design and its coordination, and wrote the manuscript. A.C., D.M. and L.K. developed the tracking software. L.Z. and M.K. worked on programming required for use of JAABA with our tracking data. C.F. and B.S. developed and manufactured the electronics for the tracking device. All authors reviewed the manuscript.

Additional Information

Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-017-17863-4.

Competing Interests: The authors declare that they have no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

7 Summary

This thesis is devoted to studying various biological systems through different modeling strategies, from mechanistic modeling to machine learning. The complex nature of biological systems imposes difficulties on both experimental data collection and theoretical modeling. The main contribution is twofold. First, with emphasis on the system components and their interconnections, it was shown that mechanistic modeling led to significant understandings of investigated systems. Second, with emphasis on learning directly from data, machine learning was deployed to study biological problems from mRNA translation to automated annotations of high throughput biological data.

The biological systems studied in the thesis include networks of general biological oscillators, flowering regulations in plants, mRNA translation regulations and bee behavior annotations. Both mechanistic and machine learning modelings played in a central role in describing the systems and deriving results. According to the modeling approaches, the systems are categorized into these groups:

1. Systems were mechanistically modeled.

   In *Chapter 2*, a network system was modeled to derive conditions for the synchronization of its subsystems based on the understanding of quorum sensing and diffusive systems and assumptions to simplify the system but keep essential properties. The mechanism of quorum sensing involves the sensing molecules released by individual subsystems connected by a common medium. Subsystems in the network were mathematically described by input-output operators and then were interconnected by the common medium. The successive deductions and analyses led to the conclusion that synchronization of the subsystems can be achieved if they were stable oscillators and the interconnection strength cooperates with input-output properties of the subsystems.

2. Systems were modeled by both approaches.

   In *Chapter 3*, the flowering-time decision making in plants *Arabidopsis Thaliana* is modeled. The modeling aim was to understand the flowering decision-making procedure from the perspective of available information from the climates which plants should adapt to.

   Firstly, a mechanistic model was developed based on the understanding of a core mechanism *vernalization* which involved a biochemical reaction to turn off the expression of the gene *FLOWERING LOCUS C (FLC)*, the repressor of flowering. The model described the biochemical reaction using a master equation to incorporate stochasticity in environmental signals like temperature. The master equation is a differential equation over time of the probabilities of the reaction states [Wikd]. The successive deductions and analyses concluded that the reaction was able to capture useful information from temperature to robustly determine flowering seasons. And moreover, it required certain assumptions on the temperature properties to reach the conclusions.

Artificial neural networks as the machine learning model was employed to approximate the processing of climate information as the plants can be regarded as climate information processing units in terms of making flowering decisions. Neural networks (NN) were trained on datasets of the temperature and day length from different climates to learn the idealized expression patterns of FLC. The NN models did not impose any assumptions on the systems or data properties such that they were applicable to model all climates types. And due to fact that the target expressions of FLC were suitable only for temperate plants, the NN models could fit well on the temperate climates. The learning results complemented the results from mechanistic modeling by that extra signals from day length can make the detection of flowering seasons more robust.

The modelings in this chapter showed although machine learning models limits our ability to reveal the causality among system components, they are more broadly applicable due to general-purpose nature.

3. Systems were exclusively modeled by machine learning.

   In *Chapter 4*, the correlation between mRNA sequences and their translational efficiencies were modeled by gradient boosting trees, another machine learning model, in order to investigate the translational mechanism and to make predictions. As general-purpose learning algorithms take only numerical values as input, the mRNA sequences which consists of a series of "AUCG" letters need to be represented by numerical features. Therefore, the primary task of the modeling procedure was to incorporate biological understandings of mRNA regulations to numerically represent the sequences. In this sense, the features generation procedure did require a high level of understandings on the biological systems. Once the features were available, the actual building of learning models did not require any systems details or assumptions. Based on a published dataset [GCK13], a predictive model was built using the gradient boosting trees for predicting translational efficiencies of unknown sequences. Nine predictions on ten new sequences were experimentally verified to be correct. However, the settings of the new sequences followed the sequences in the dataset [GCK13], which limits the generalizability of the predictive power of the model to more general sequences. Due to the limited understanding of the complex dynamics of translations and limited variability in existing data, the picture of translational regulation was far from complete. More quantitative data and more powerful models such as deep learning can further promote the understanding of the translational mechanism.

   In *Chapter 5*, a machine learning framework *OutCyte* was developed to predict *unconventional protein secretions (UPS)*. The term "unconventional" is used to include the protein secretions not following the classical secretion mechanism [Rab17; DN18]. The classical secretions require a *signal peptide* which is a short amino acid segment at the starting part of the protein sequences (N-terminus). The signal peptides direct the whole protein molecules going through the Endoplasmic Reticulum (ER) – Golgi pathway to reach the extracellular space. For decades it had been believed that all secreted proteins followed the ER-Golgi pathway [Rab17; DN18]. However, since 1980s, a dozen of proteins going through alternative secretion routes have been reported [Cha+97; MS+16; Nic11; MSS15; Kli+16]. Following the success of computationally identifying signal peptides [Nie17; Sav+17], efforts have also been paid to build predictive tools for UPS [Nie+18; Kan+10; Yu+10; Ben+04]. The central challenge of predicting UPS is the imbalance between small number of known UPS and the problem complexity. The existing tools for predicting UPS relied on the hypothesis that all

secreted proteins share common features to make use of the large amount classical secretions. Different from that, OutCyte has relied on the experimental datasets from studying secretome by *mass spectrometry*. OutCyte has achieved the state of the art accuracy for predicting a set of known UPS. Further a model based on convolutional neural network for identifying signal peptides was integrate to OutCyte for annotating proteomes.

In *Chapter 6*, an interactive machine learning framework was set up to automatically annotate behaviors in bee societies. A bee society typically has thousands of members, to manually annotate all behaviors even for one minute is a tedious task for humans. The interactive framework consisted of a tracking device, adapted from an ant tracking device [MCK13], for recording the continuous position and orientation information of a group of bees and a published interactive machine learning program JAABA [Kab+13]. Then for annotating each behavior, JAABA requires moving trajectories of individual bees and the monitoring video as input. A short clip of the video then needs to manually annotated to make JAABA be able to learn general patterns of that behavior. The framework was able to correctly annotating 93% of the encounter behaviors which are head-to-head quick contacts among bees. Though annotating all of bee's behaviors was a complex problem, the annotations can lead to significant understandings of social behaviors within a bee community.

Through the course of study and research by modeling biological systems, the highly interdisciplinary area has brought me both great challenges and joys, with challenges from applying mathematics and computation techniques to biological systems and understanding complex biological phenomena, with joys from thoroughly understanding biological systems and successfully solving relavant problems. As a final remark, in the case of modeling, the most elegant research result would be to obtain simple models which can both fit the present data best and more importantly can predict new data best.

Bibliography

[AC12] Fernando Andrés and George Coupland. „The genetic basis of flowering responses to seasonal cues“. In: *Nature Reviews Genetics* 13.9 (2012), p. 627 (cit. on p. 39).

[Ack15] Martin Ackermann. „A functional perspective on phenotypic heterogeneity in microorganisms“. In: *Nature Reviews Microbiology* 13.8 (2015), p. 497 (cit. on p. 4).

[Ald+97] John Aldrich et al. „RA Fisher and the making of maximum likelihood 1912-1922“. In: *Statistical science* 12.3 (1997), pp. 162–176 (cit. on p. 16).

[Ang+11] Andrew Angel, Jie Song, Caroline Dean, and Martin Howard. „A Polycomb-based switch underlying quantitative epigenetic memory.“ In: *Nature* 476.7358 (Aug. 2011), pp. 105–8 (cit. on p. 39).

[Ang+15] Andrew Angel, Jie Song, Hongchun Yang, et al. „Vernalizing cold is registered digitally at FLC“. In: *Proceedings of the National Academy of Sciences* 112.13 (2015), pp. 4146–4151 (cit. on p. 39).

[AW10] Steven J Altschuler and Lani F Wu. „Cellular heterogeneity: do differences make a difference?“ In: *Cell* 141.4 (2010), pp. 559–563 (cit. on p. 4).

[BAK18] Danilo Bzdok, Naomi Altman, and Martin Krzywinski. „Points of significance: statistics versus machine learning“. In: *Nature Methods* (2018), pp. 1–7 (cit. on pp. 5, 6).

[BC11] N Bellomo and B Carbonaro. „Toward a mathematical theory of living systems focusing on developmental biology and evolution: a review and perspectives“. In: *Physics of Life Reviews* 8.1 (2011), pp. 1–18 (cit. on pp. 3, 4).

[BDJ15] Martina Bluemel, Nadine Dally, and Christian Jung. „Flowering time regulation in crops—what did we learn from Arabidopsis?“ In: *Current opinion in biotechnology* 32 (2015), pp. 121–129 (cit. on p. 39).

[Ben+04] Jannick Dyrløv Bendtsen, Lars Juhl Jensen, Nikolaj Blom, Gunnar Von Heijne, and Søren Brunak. „Feature-based prediction of non-classical and leaderless protein secretion“. In: *Protein Engineering Design and Selection* 17.4 (2004), pp. 349–356 (cit. on pp. 119, 174).

[Bis06] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006 (cit. on p. 30).

[BL16] Ezio Bartocci and Pietro Lió. „Computational modeling, formal analysis, and tools for systems biology“. In: *PLoS computational biology* 12.1 (2016), e1004591 (cit. on p. 2).

[Bla+97] Frederick R Blattner, Guy Plunkett, Craig A Bloch, et al. „The complete genome sequence of Escherichia coli K-12“. In: *science* 277.5331 (1997), pp. 1453–1462 (cit. on pp. 37, 40).

[Bre97] Leo Breiman. „Arcing the edge“. In: *Statistics* (1997) (cit. on p. 26).

[BT15] Fabian Bratzel and Franziska Turck. *Molecular memories in the regulation of seasonal flowering: From competence to cessation.* 2015 (cit. on p. 39).

[Bur17] Thomas Burger. „Gentle introduction to the statistical foundations of false discovery rate in quantitative proteomics". In: *Journal of proteome research* 17.1 (2017), pp. 12–22 (cit. on p. 40).

[CG16] Tianqi Chen and Carlos Guestrin. „XGBoost : Reliable Large-scale Tree Boosting System". In: *arXiv* (2016). arXiv: 1603.02754 (cit. on pp. 20, 28, 29).

[CGA18] Guillaume Cambray, Joao C Guimaraes, and Adam Paul Arkin. „Evaluation of 244,000 synthetic sequences reveals design principles to optimize translation in Escherichia coli". In: *Nature biotechnology* 36.10 (2018), p. 1005 (cit. on p. 87).

[Cha+97] Hsiao C Chang, Felipe Samaniego, Bala C Nair, Luigi Buonaguro, and Barbara Ensoli. „HIV-1 Tat protein exits from cells via a leaderless secretory pathway and binds to extracellular matrix-associated heparan sulfate proteoglycans through its basic region". In: *Aids* 11.12 (1997), pp. 1421–1431 (cit. on p. 174).

[Cle99] Maurice Clerc. „The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization". In: *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999.* 1999. arXiv: arXiv:1011.1669v3 (cit. on p. 36).

[Col14] David Colquhoun. „An investigation of the false discovery rate and the misinterpretation of p-values". In: *Royal Society open science* 1.3 (2014), p. 140216 (cit. on p. 40).

[Con+01] International Human Genome Sequencing Consortium et al. „Initial sequencing and analysis of the human genome". In: *Nature* 409.6822 (2001), p. 860 (cit. on p. 40).

[Cso+14] Tibor Csorba, Julia I Questa, Qianwen Sun, and Caroline Dean. „Antisense COOLAIR mediates the coordinated switching of chromatin states at FLC during vernalization". In: *Proceedings of the National Academy of Sciences* 111.45 (2014), pp. 16160–16165 (cit. on p. 39).

[DB11] Richard C Dorf and Robert H Bishop. *Modern control systems.* Pearson, 2011 (cit. on pp. 8, 9).

[DN18] Eleni Dimou and Walter Nickel. „Unconventional mechanisms of eukaryotic protein secretion". In: *Current Biology* 28.8 (2018), R406–R410 (cit. on p. 174).

[DS12] Morris H DeGroot and Mark J Schervish. *Probability and statistics.* Pearson Education, 2012 (cit. on p. 14).

[DZK12] Peter M Dower, Huan Zhang, and Christopher M Kellett. „Nonlinear L2-gain verification for nonlinear systems". In: *Systems & Control Letters* 61.4 (2012), pp. 563–572 (cit. on p. 13).

[EL00] Michael B Elowitz and Stanislas Leibler. „A synthetic oscillatory network of transcriptional regulators". In: *Nature* 403.6767 (2000), p. 335 (cit. on pp. 8, 43, 44).

[ESI01] R C Eberhart, Y H Shi, and Ieee. „Particle swarm optimization: Developments, applications and resources". In: *Proceedings of the 2001 Congress on Evolutionary Computation, Vols 1 and 2* (2001), pp. 81–86 (cit. on pp. 35, 36).

[FB09] Paul Flicek and Ewan Birney. „Sense from sequence reads: methods for alignment and assembly". In: *Nature methods* 6.11s (2009), S6 (cit. on p. 40).

[Fri01] Jerome H. Friedman. „Greedy function approximation: A gradient boosting machine". In: *Annals of Statistics* (2001). arXiv: arXiv:1011.1669v3 (cit. on p. 26).

[Fri02] Jerome H. Friedman. „Stochastic gradient boosting“. In: *Computational Statistics and Data Analysis* (2002). arXiv: 0607324v2 [arXiv:hep-ph] (cit. on p. 26).

[GCC00] Timothy S Gardner, Charles R Cantor, and James J Collins. „Construction of a genetic toggle switch in Escherichia coli“. In: *Nature* 403.6767 (2000), p. 339 (cit. on p. 8).

[GCK13] Daniel B Goodman, George M Church, and Sriram Kosuri. „Causes and effects of N-terminal codon bias in bacterial genes“. In: *Science* (2013), p. 1241934 (cit. on pp. 87, 174).

[GP11] Hila Gingold and Yitzhak Pilpel. „Determinants of translation efficiency and accuracy“. In: *Molecular systems biology* 7.1 (2011), p. 481 (cit. on p. 87).

[Gru+18] Leonie Grube, Rafael Dellen, Fabian Kruse, et al. „Mining the Secretome of C2C12 Muscle Cells: Data Dependent Experimental Approach To Analyze Protein Secretion Using Label-Free Quantification and Peptide Based Analysis“. In: *Journal of proteome research* 17.2 (2018), pp. 879–890 (cit. on p. 40).

[Hai10] Anna-Bettina Haidich. „Meta-analysis in medical research“. In: *Hippokratia* 14.Suppl 1 (2010), p. 29 (cit. on p. 5).

[Hen03] Victor Henri. *Lois générales de l'action des diastases*. Librairie Scientifique A. Hermann, 1903 (cit. on p. 2).

[Hiy+11] Akira Hiyoshi, Kohji Miyahara, Chiaki Kato, and Yasumi Ohshima. „Does a DNA-less cellular organism exist on Earth?“ In: *Genes to Cells* 16.12 (2011), pp. 1146–1158 (cit. on p. 37).

[HK11] Richard P Horgan and Louise C Kenny. „'Omic'technologies: genomics, transcriptomics, proteomics and metabolomics“. In: *The Obstetrician & Gynaecologist* 13.3 (2011), pp. 189–195 (cit. on p. 40).

[HM80] D Hill and P Moylan. „Connections between finite-gain and asymptotic stability“. In: *IEEE Transactions on Automatic Control* 25.5 (1980), pp. 931–936 (cit. on p. 13).

[Hof15] Daniel Sander Hoffmann. „The dawn of mathematical biology“. In: *arXiv preprint arXiv:1511.01455* (2015) (cit. on p. 3).

[Hop95] Frank Hoppensteadt. „Getting started in mathematical biology“. In: *Notices of the AMS* 42.9 (1995), pp. 969–975 (cit. on p. 3).

[Hua12] Wen-Lin Huang. „Ranking gene ontology terms for predicting non-classical secretory proteins in eukaryotes and prokaryotes“. In: *Journal of theoretical biology* 312 (2012), pp. 105–113 (cit. on p. 121).

[Hun+10] Chiung-Hui Hung, Hui-Ling Huang, Kai-Ti Hsu, Shinn-Jang Ho, and Shinn-Ying Ho. „Prediction of non-classical secreted proteins using informative physicochemical properties“. In: *Interdisciplinary Sciences: Computational Life Sciences* 2.3 (2010), pp. 263–270 (cit. on p. 121).

[Ide+01] Trey Ideker, Vesteinn Thorsson, Jeffrey A Ranish, et al. „Integrated genomic and proteomic analyses of a systematically perturbed metabolic network“. In: *Science* 292.5518 (2001), pp. 929–934 (cit. on p. 3).

[JM15] Michael I Jordan and Tom M Mitchell. „Machine learning: Trends, perspectives, and prospects“. In: *Science* 349.6245 (2015), pp. 255–260 (cit. on p. 19).

[K. 09] R.M. Murry K. J. Astrom. *Feedback systems*. 2009 (cit. on pp. 8, 9).

[Kab+13] Mayank Kabra, Alice A Robie, Marta Rivera-Alba, Steven Branson, and Kristin Branson. „JAABA: interactive machine learning for automatic annotation of animal behavior“. In: *nature methods* 10.1 (2013), p. 64 (cit. on pp. 161, 163, 175).

[Kan+10] Krishna Kumar Kandaswamy, Ganesan Pugalenthi, Enno Hartmann, et al. „SPRED: A machine learning approach for the identification of classical and non-classical secretory proteins in mammalian genomes“. In: *Biochemical and biophysical research communications* 391.3 (2010), pp. 1306–1311 (cit. on pp. 121, 174).

[Kha] Hassan K Khalil. *Nonlinear systems*. Vol. 3 (cit. on pp. 11, 13).

[Kit02a] Hiroaki Kitano. „Computational systems biology“. In: *Nature* 420.6912 (2002), p. 206 (cit. on pp. 3, 40).

[Kit02b] Hiroaki Kitano. „Systems biology: a brief overview“. In: *Science* 295.5560 (2002), pp. 1662–1664 (cit. on p. 4).

[Kli+16] Daniel J Klionsky, Kotb Abdelmohsen, Akihisa Abe, et al. „Guidelines for the use and interpretation of assays for monitoring autophagy“. In: *Autophagy* 12.1 (2016), pp. 1–222 (cit. on p. 174).

[LBH15a] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. „Deep learning“. In: *nature* 521.7553 (2015), p. 436 (cit. on p. 5).

[LBH15b] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. „Deep learning“. In: *nature* 521.7553 (2015), p. 436 (cit. on p. 20).

[Le+11] Quoc V Le, Marc'Aurelio Ranzato, Rajat Monga, et al. „Building high-level features using large scale unsupervised learning“. In: *arXiv preprint arXiv:1112.6209* (2011) (cit. on p. 5).

[Lee+08] D-S Lee, Juyong Park, KA Kay, et al. „The implications of human metabolic network topology for disease comorbidity“. In: *Proceedings of the National Academy of Sciences* (2008) (cit. on p. 3).

[LH16] Teck Yew Low and Albert JR Heck. „Reconciling proteomics with next generation sequencing“. In: *Current opinion in chemical biology* 30 (2016), pp. 14–20 (cit. on pp. 37, 40).

[Mar17] Florian Markowetz. „All biology is computational biology“. In: *PLoS biology* 15.3 (2017), e2002050 (cit. on pp. 2, 3, 40).

[Mas+99] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. „Boosting algorithms as gradient descent“. In: *NIPS* (1999). arXiv: `0607324v2 [arXiv:hep-ph]` (cit. on p. 26).

[May04] Robert M May. „Uses and abuses of mathematics in biology“. In: *Science* 303.5659 (2004), pp. 790–793 (cit. on p. 3).

[MB01] Melissa B Miller and Bonnie L Bassler. „Quorum sensing in bacteria“. In: *Annual Reviews in Microbiology* 55.1 (2001), pp. 165–199 (cit. on pp. 4, 43).

[MCK13] Danielle P Mersch, Alessandro Crespi, and Laurent Keller. „Tracking individuals shows spatial fidelity is a key regulator of ant social organization“. In: *Science* 340.6136 (2013), pp. 1090–1093 (cit. on pp. 161, 175).

[Mil+09] Ron Milo, Paul Jorgensen, Uri Moran, Griffin Weber, and Michael Springer. „BioNumbers—the database of key numbers in molecular and cell biology“. In: *Nucleic acids research* 38.suppl_1 (2009), pp. D750–D753 (cit. on p. 3).

[Mit17] Tom Mitchell. *Chapter 14, Machine learning 2nd Edition*. McGraw Hill, 2017 (cit. on p. 5).

[MK10] Jon McClellan and Mary-Claire King. „Genetic heterogeneity in human disease“. In: *Cell* 141.2 (2010), pp. 210–217 (cit. on p. 4).

[MM13] Leonor Michaelis and Maude L Menten. „The kinetics of the inversion effect“. In: *Biochem. Z* 49 (1913), pp. 333–369 (cit. on p. 2).

[MP12] Santo Motta and Francesco Pappalardo. „Mathematical modeling of biological systems“. In: *Briefings in Bioinformatics* 14.4 (2012), pp. 411–422 (cit. on pp. 2–4).

[MS+16] Fatima Martín-Sánchez, Catherine Diamond, Marcel Zeitler, et al. „Inflammasome-dependent IL-1β release depends upon membrane permeabilisation“. In: *Cell death and differentiation* 23.7 (2016), p. 1219 (cit. on p. 174).

[MSS15] Mercedes Monteleone, Jennifer L Stow, and Kate Schroder. „Mechanisms of unconventional secretion of IL-1 family cytokines“. In: *Cytokine* 74.2 (2015), pp. 213–218 (cit. on p. 174).

[Mue79] Fritz Mueller. „Ituna and Thyridia: a remarkable case of mimicry in butterflies“. In: *Proclamations of the Entomological Society of London* 1879 (1879), pp. 20–29 (cit. on p. 3).

[Mur12] Kevin P Murphy. „Machine learning: a probabilistic perspective“. In: (2012) (cit. on pp. 5, 19).

[MW15] Federico Marini and Beata Walczak. „Particle swarm optimization (PSO). A tutorial“. In: *Chemometrics and Intelligent Laboratory Systems* 149 (2015), pp. 153–165 (cit. on p. 35).

[Nic11] Walter Nickel. „The unconventional secretory machinery of fibroblast growth factor 2“. In: *Traffic* 12.7 (2011), pp. 799–805 (cit. on p. 174).

[Nie15] Michael Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015 (cit. on p. 23).

[Nie17] Henrik Nielsen. „Predicting secretory proteins with SignalP“. In: *Protein Function Prediction: Methods and Protocols* (2017), pp. 59–73 (cit. on p. 174).

[Nie+18] Henrik Nielsen, Eirini I Petsalaki, Linlin Zhao, and Kai Stühler. „Predicting eukaryotic protein secretion without signals“. In: *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics* (2018) (cit. on p. 174).

[Nob02] Denis Noble. „The rise of computational biology“. In: *Nature Reviews Molecular Cell Biology* 3.6 (2002), p. 459 (cit. on p. 3).

[Org+06] World Health Organization et al. „Definition and diagnosis of diabetes mellitus and intermediate hyperglycaemia: report of a WHO/IDF consultation“. In: (2006) (cit. on p. 21).

[Orz12] Steven Hecht Orzack. „The philosophy of modelling or does the philosophy of biology have any use?“ In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 367.1586 (2012), pp. 170–180 (cit. on p. 4).

[Pea96] Karl Pearson. „Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity and Panmixia“. In: *Philosophical Transactions of the Royal Society of London* 187 (1896), pp. 251–318 (cit. on p. 3).

[Ped+11] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 20).

[Pet+11] Thomas Nordahl Petersen, Søren Brunak, Gunnar von Heijne, and Henrik Nielsen. „SignalP 4.0: discriminating signal peptides from transmembrane regions“. In: *Nature methods* 8.10 (2011), p. 785 (cit. on p. 119).

[PV02] K. E. Parsopoulos and M. N. Vrahatis. „Recent approaches to global optimization problems through Particle Swarm Optimization“. In: *Natural Computing* 1.2-3 (2002), pp. 235–306 (cit. on p. 35).

[Qin+18] Zhuwei Qin, Funxun Yu, Chenchen Liu, and Xiang Chen. „How convolutional neural network see the world-A survey of convolutional neural network visualization methods“. In: *arXiv preprint arXiv:1804.11191* (2018) (cit. on p. 5).

[Rab17] Catherine Rabouille. „Pathways of unconventional protein secretion“. In: *Trends in cell biology* 27.3 (2017), pp. 230–240 (cit. on p. 174).

[Ree04] Michael C Reed. „Why is mathematical biology so hard“. In: *Notices of the AMS* 51.3 (2004), pp. 338–342 (cit. on p. 4).

[RHS07] Dean Rickles, Penelope Hawe, and Alan Shiell. „A simple guide to chaos and complexity“. In: *Journal of Epidemiology & Community Health* 61.11 (2007), pp. 933–937 (cit. on p. 3).

[Rie+10] Karin Ried, Thomas Sullivan, Peter Fakler, Oliver R Frank, and Nigel P Stocks. „Does chocolate reduce blood pressure? A meta-analysis“. In: *BMC medicine* 8.1 (2010), p. 39 (cit. on p. 5).

[Ros+00] Robert Ross, Damon Dagnone, Peter JH Jones, et al. „Reduction in obesity and related comorbid conditions after diet-induced weight loss or exercise-induced weight loss in men: a randomized, controlled trial“. In: *Annals of internal medicine* 133.2 (2000), pp. 92–103 (cit. on p. 3).

[Ros62] Frank Rosenblatt. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Spartan Books, 1962 (cit. on p. 23).

[Rub90] Harry Rubin. „The significance of biological heterogeneity“. In: *Cancer and Metastasis Reviews* 9.1 (1990), pp. 1–20 (cit. on p. 4).

[Rud16] Sebastian Ruder. „An overview of gradient descent optimization algorithms“. In: *arXiv preprint arXiv:1609.04747* (2016) (cit. on p. 34).

[SAS10] Luca Scardovi, Murat Arcak, and Eduardo D Sontag. „Synchronization of interconnected systems with applications to biochemical networks: An input-output approach“. In: *IEEE Transactions on Automatic Control* 55.6 (2010), pp. 1367–1379 (cit. on pp. 43, 45).

[Sav+17] Castrense Savojardo, Pier Luigi Martelli, Piero Fariselli, and Rita Casadio. „DeepSig: deep learning improves signal peptide detection in proteins“. In: *Bioinformatics* 34.10 (2017), pp. 1690–1696 (cit. on pp. 119, 174).

[Sch15] Jürgen Schmidhuber. „Deep learning in neural networks: An overview“. In: *Neural networks* 61 (2015), pp. 85–117 (cit. on p. 23).

[SD18] Ying Sun and José R Dinneny. „Q&A: How do gene regulatory networks control environmental responses in plants?“ In: *BMC biology* 16.1 (2018), p. 38 (cit. on p. 38).

[SD73] J Shine and L Dalgarno. „Occurrence of heat-dissociable ribosomal RNA in insects: the presence of three polynucleotide chains in 26 S RNA from cultured Aedes aegypti cells“. In: *Journal of molecular biology* 75.1 (1973), pp. 57–72 (cit. on p. 87).

[SJ80] John Shore and Rodney Johnson. „Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy“. In: *IEEE Transactions on information theory* 26.1 (1980), pp. 26–37 (cit. on p. 22).

[Sti07] Stephen M Stigler. „The epic story of maximum likelihood“. In: *Statistical Science* (2007), pp. 598–620 (cit. on p. 16).

[Str18] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC Press, 2018 (cit. on pp. 7, 10, 11).

[TFC08] Franziska Turck, Fabio Fornara, and George Coupland. „Regulation and Identity of Florigen: FLOWERING LOCUS T Moves Center Stage". In: *Annual Review of Plant Biology* 59.1 (2008), pp. 573–594 (cit. on p. 39).

[VMO15] Eberhard O Voit, Harald A Martens, and Stig W Omholt. „150 years of the mass action law". In: *PLoS computational biology* 11.1 (2015), e1004012 (cit. on p. 2).

[WA04] George H Wadhams and Judith P Armitage. „Making sense of it all: bacterial chemotaxis". In: *Nature reviews Molecular cell biology* 5.12 (2004), p. 1024 (cit. on p. 4).

[Wil05] Herbert S Wilf. *generatingfunctionology*. AK Peters/CRC Press, 2005 (cit. on p. 17).

[Yu+10] Lezheng Yu, Yanzhi Guo, Yizhou Li, et al. „SecretP: identifying bacterial secreted proteins by fusing new features into Chou's pseudo-amino acid composition". In: *Journal of Theoretical Biology* 267.1 (2010), pp. 1–6 (cit. on p. 174).

[Zha+16] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. „Understanding deep learning requires rethinking generalization". In: *CoRR* abs/1611.03530 (2016). arXiv: 1611.03530 (cit. on p. 20).

Websites

[Aca] Khan Academy. *Overview of transcription*. URL: https://www.khanacademy.org/science/biology/gene-expression-central-dogma/transcription-of-dna-into-rna/a/overview-of-transcription (visited on 2019) (cit. on p. 38).

[Bod] Eberhard Bodenschatz. *Complex Systems*. URL: https://www.mpg.de/36885/cpt08_ComplexSystems-basetext.pdf (visited on 2018) (cit. on p. 3).

[Deu] Andreas Deutsch. *Mathematical and Theoretical Biology: A European Perspective*. URL: http://www.sciencemag.org/careers/2004/03/mathematical-and-theoretical-biology-european-perspective (visited on 2018) (cit. on p. 3).

[FR] Scott Fortmann-Roe. *Understanding the Bias-Variance Tradeoff*. URL: http://scott.fortmann-roe.com/docs/BiasVariance.html (visited on 2018) (cit. on p. 29).

[Isi] Alberto Isidori. *Robust Stability via H-Infinity Methods*. URL: http://www.eeci-institute.eu/GSC2012/Photos-EECI/EECI-GSC-2012-M9/Handout_1.pdf (visited on 2011) (cit. on p. 13).

[Mat] Math24. *Method of Lyapunov Functions*. URL: https://www.math24.net/method-lyapunov-functions/ (visited on 2019) (cit. on p. 11).

[Mel] C. Melchiorri. *Stability analysis of dynamic systems*. URL: http://www-lar.deis.unibo.it/people/cmelchiorri/Files_ACST/05_Stability.pdf (visited on 2013) (cit. on p. 9).

[Nyk] Nykamp. *The idea of a dynamical system*. URL: http://mathinsight.org/dynamical_system_idea (visited on 2019) (cit. on p. 7).

[Ola] Chris Olah. *Conv Nets: A Modular Perspective*. URL: http://colah.github.io/posts/2014-07-Conv-Nets-Modular/ (visited on 2018) (cit. on p. 25).

[Pro] Proofwiki. *Derivation of variance of Poisson distribution*. URL: https://proofwiki.org/wiki/Variance_of_Poisson_Distribution (visited on 2018) (cit. on p. 18).

[Ros] Jonathan Roseblatt. *Translator*. URL: http://www.john-ros.com/translator/ (visited on 2018) (cit. on p. 5).

[Sha] Shane. *The Two Cultures: statistics vs. machine learning?* URL: https://stats.stackexchange.com/q/6 (visited on 2018) (cit. on p. 5).

[Uni] The Open University. *Models and Modeling*. URL: https://www.open.edu/openlearn/science-maths-technology/computing-and-ict/models-and-modelling/content-section-2.1 (visited on 2018) (cit. on p. 2).

[Was] Larry Wasserman. *Statistics versus machine learning*. URL: https://normaldeviate.wordpress.com/2012/06/12/statistics-versus-machine-learning-5-2/ (visited on 2019) (cit. on p. 5).

[Wei] E. Weisstein. *Z-Transform*. URL: http://mathworld.wolfram.com/Z-Transform.html (visited on 2018) (cit. on p. 17).

[Wika] Wikipedia. *DNA*. URL: https://en.wikipedia.org/wiki/DNA (visited on 2019) (cit. on p. 37).

[Wikb] Wikipedia. *Gene regulatory network*. URL: https://en.wikipedia.org/wiki/Gene_regulatory_network (visited on 2019) (cit. on p. 38).

[Wikc] Wikipedia. *Genomics*. URL: https://en.wikipedia.org/wiki/Genomics (visited on 2019) (cit. on p. 40).

[Wikd] Wikipedia. *Master equations*. URL: https://en.wikipedia.org/wiki/Master_equation (visited on 2018) (cit. on p. 173).

[Wike] Wikipedia. *Michaelis Menten Kinetics*. URL: https://en.wikipedia.org/wiki/Michaelis%E2%80%93Menten_kinetics (visited on 2018) (cit. on p. 2).

[Wikf] Wikipedia. *Moment-generating function*. URL: https://en.wikipedia.org/wiki/Moment-generating_function (visited on 2018) (cit. on p. 17).

[Wikg] Wikipedia. *Müllerian mimicry*. URL: https://en.wikipedia.org/wiki/M%C3%BCllerian_mimicry (visited on 2018) (cit. on p. 3).

[Wikh] Wikipedia. *Research fields in mathematical and theoretical biology*. URL: https://en.wikipedia.org/wiki/Mathematical_and_theoretical_biology#Areas_of_research (visited on 2018) (cit. on p. 3).