RaPIDシステムを利用してCypDを認識する特殊ペプチドの創成を行った。

図7はRaPIDシステムにより同定された、CypD結合環状ペプチドRI-M24とRI-N6のCypD及びサイクロフィリンA（CypA）に対する結合を表面プラズモン共鳴測定（SPR）により評価したものである。いずれの環状ペプチドにおいても、CypDに対しては強く（それぞれKd = 480及び900 nM）結合したのに対し、CypAに対しては定量ができないほど極めて弱い結合活性しか示さない。この結果は、これらのペプチドがアイソフォーム選択性を持っていることを示唆している。

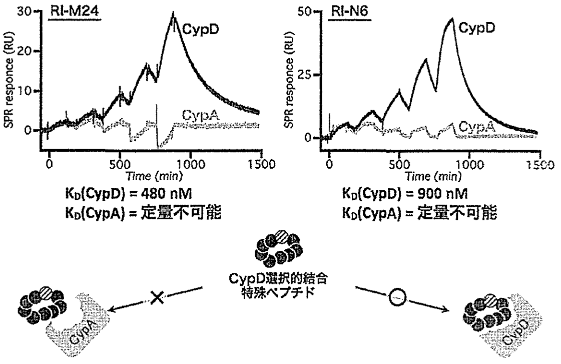

図7　RaPIDシステムで単離されたCypD選択的に結合する特殊ペプチド

8. おわりに

近年の生命科学研究の進歩により、創薬のターゲットとなり得るタンパク質が次々と同定されるなかにあって、化合物ライブラリーの数、及びそこからのリード化合物のスクリーニングは、その増加に追いついていない現状がある。本稿ではCypD結合ペプチドの例を示したが、用いる標的タンパク質を変更すれば、RaPIDシステムはいかなるタンパク質にも適用可能である。また、特殊ペプチドもチオエーテル環状ペプチドに限らず、これまでに我々のグループで翻訳合成に成功している多彩な特殊骨格を含むペプチドを探索できる。さらに、RaPIDシステムにおけるセレクションの条件を変え（たとえば、標的タンパク質の濃度を下げる・アイソフォームタンパクを競合させるなど）、より強いKdや選択性を持った特殊ペプチドを指向した人工進化を行えば、得られる特殊ペプチドの機能を自在にコントロールすることも原理的に可能だ。

本手法により、これまで薬剤シーズは天然物に頼らざるを得なかった特殊ペプチドというモチーフにおいて、大規模ライブラリーのなかから、de novoなリード化合物を発見する道を開くことができた。現在、当研究室ではさまざまな疾病関連タンパク質に結合し阻害活性を示す特殊ペプチドを精力的に開発しており、実際に有望な生理活性を示す化合物も見つかっている。今後、新機軸の創薬アプローチとして、多くの薬剤候補を生み出していければ幸いである。

英文タイトル：A new drug discovery strategy powered by RaPID system

参考文献

1) Grunewald, J. *et al. Microbiol. Mol. Biol. Rev.* **2006**, *70*, 121-146.
2) Forster, A. C. *et al. Proc. Natl. Acad. Sci. U.S.A.* **2003**, *100*, 6353-6357.
3) Hecht, S. M. *et al. J. Biol. Chem.* **1978**, *253*, 4517-4520.
4) Wang, L. *et al. Science* **2001**, *292*, 498-500.
5) Murakami, H. *et al. Chem. Biol.* **2003**, *10*, 655-662.
6) Murakami, H. *et al. Nat. Methods* **2006**, *3*, 357-359.
7) Goto, Y. *et al. ACS Chem. Biol.* **2008**, *3*, 120-129.
8) Goto, Y. *et al. RNA* **2008**, *14*, 1390-1398.
9) Ohta, A. *et al. Chem. Biol.* **2007**, *14*, 1315-1322.
10) Kawakami, T. *et al. Chem. Biol.* **2008**, *15*, 32-42.
11) Kawakami, T. *et al. J. Am. Chem. Soc.* **2008**, *130*, 16861-16863.
12) Sako, Y. *et al. ACS Chem. Biol.* **2008**, *3*, 241-249.
13) Goto, Y. *et al. Chem. Commun.* **2009**, 3419-3421.
14) Nakajima, E. *et al. Chembiochem* **2009**, *10*, 1186-1192.
15) Kawakami, T. *et al. Nat. Chem. Biol.* **2009**, *5*, 888-890.
16) Sako, Y. *et al. J. Am. Chem. Soc.* **2008**, *130*, 7232-7234.
17) Yamagishi, Y. *et al. Chembiochem* **2009**, *10*, 1469-1472.
18) Nemoto, N. *et al. FEBS Lett.* **1997**, *414*, 405-408.
19) Roberts, R. W. *et al. Proc. Natl. Acad. Sci. U.S.A.* **1997**, *94*, 12297-12302.

A General Procedure for Accurate Parameter Estimation in Dynamic Systems Using New Estimation Errors

Masahiko Nakatsui
AIST
Tokyo, Japan
m.nakatsui@aist.go.jp

Alexandre Sedoglavic
University Lille I. LIFL
Lille, France
alexandre.sedoglavic@lifl.fr

François Lemaire
University Lille I. LIFL
Lille, France
Francois.Lemaire@lifl.fr

François Boulier
University Lille I. LIFL
Lille, France
Francois.Boulier@lifl.fr

Asli Ürgüplü
University Lille I. LIFL
Lille, France
Asli.Urguplu@lifl.fr

Katsuhisa Horimoto
AIST
Tokyo, Japan
k.horimoto@aist.go.jp

Abstract

The investigation of network dynamics is a major issue in systems and synthetic biology. One of the essential steps in a dynamics investigation is the parameter estimation in the model that expresses biological phenomena. Indeed, various techniques for parameter optimization have been devised and implemented in both free and commercial software. While the computational time for parameter estimation has been greatly reduced, due to improvements in calculation algorithms and the advent of high performance computers, the accuracy of parameter estimation has not been addressed.

We previously proposed an approach for accurate parameter optimization by using Differential Elimination, which is an algebraic approach for rewriting a system of differential equations into another equivalent system. The equivalent system has the same solution as the original system, and it includes high-order derivatives, which contain information about the form of the observed time-series data. The introduction of an equivalent system into the numerical parameter optimizing procedure resulted in the drastic improvement of the estimation accuracy, since our approach evaluates the difference of not only the values but also the forms between the measured and estimated data, while the classical numerical approach evaluates only the value difference. In this report, we describe the detailed procedure of our approach for accurate parameter estimation in dynamic systems. The ability of our approach is illustrated in terms of the parameter estimation accuracy, in comparison with classical methods.

1 Introduction

The investigation of network dynamics is a major issue in systems and synthetic biology[1]. In general, a network model for describing the kinetics of constituent molecules is first constructed with reference to the biological knowledge, and then the model is mathematically expressed by differential equations, based on the chemical reactions underlying the kinetics. Finally, the kinetic parameters in the model are estimated by various parameter optimization techniques[2], from the time-series data measured for the constituent molecules. While the computational time for parameter estimation has been greatly reduced, due to the improvements in calculation algorithms and the advent of high performance computers, the accurate numerical estimation of parameter values for a given model remains a limiting step. Indeed, the parameter values estimated by various optimization techniques are frequently quite variable, due to the conditions for parameter estimation, such as the initial values. In particular, we cannot always obtain the data measured for all of the constituent molecules, due to limitations of measurement techniques and ethical constraints. In this case, one of the issues we should resolve is that the parameters are estimated from the data for only some of the constituent molecules. Unfortunately, it is quite difficult to estimate the parameters in such a network model including unmeasured variables.

Differential elimination was applied[3] to improve the parameter estimation methods, especially in the model dynamics including unmonitored variables. The idea consisted of computing differential equations from the input system, from which the unmonitored variables were eliminated. These differential equations could then be used to guess the initial values for the Newton-type numerical parameter optimization scheme. The overall method was implemented over the BLAD libraries[4]. Differential elimination theory is a branch of the differential algebra of Ritt and Kolchin[5, 6]. Its basis was developed by Ritt, who founded the theory of characteristic sets. Ritt's ideas were subsequently developed by Seidenberg [7], Wu[8], Boulier et al.[9, 10] and many other researchers. The Rosenfeld-Gröbner algorithm[9, 10] is the first complete algorithm for differential elimination ever implemented. It relies on Ritt and Seidenberg's ideas, on the Rosenfeld Lemma, which reduces differential problems to non-differential polynomial ones, and on the Gröbner bases theory for solving non-differential polynomial systems (although recent implementations completely avoid Gröbner bases computations). The Rosenfeld-Gröbner algorithm was implemented in 1996 in the diffalg package of the MAPLE computer algebra software. Starting from MAPLE 14, it should be replaced by the MAPLE DifferentialAlgebra package, which relies on the BLAD libraries[11].

Recently, we proposed a new procedure for optimizing the parameters, by using differential elimination. Our procedure partially utilizes a technique from a previous study[12, 13], regarding the introduction of differential elimination into parameter optimization in a network. Instead of using differential elimination for estimating the initial values for the following parameter optimization, the equations derived by differential elimination are directly introduced as the constraints into the objective function for the parameter optimization[14, 15, 16, 17]. Here, we will describe the detailed procedure of our approach, by using a simple model represented as non-linear differential equations. We also discuss the merits and pitfalls of our procedure, in terms of its extension to more realistic and complex models.

2 Procedure

2.1 Overview of Present Procedure

The key point of this study is the introduction of new constraints obtained by differential elimination into the objective function, to improve the parameter accuracy. This section outlines our new procedure for estimating the parameters, using constraints built from differential elimination, and compared it with the classical constraints based on the total relative error. For clarity, the method is described using an academic example.

We first present the example. We then show how to build our new constraints using differential elimination, and how to optimize the evaluation of those new constraints over numeric values. Subsequently, we present our genetic algorithm for estimating the parameter values, and finish with the results. All Maple commands used for computing the expressions described in the following subsections are provided in appendix A.

2.2 Example

Differential algebra aims at studying differential equations from a purely algebraic point of view[5, 6]. Differential elimination theory is a sub theory of differential algebra, based on Rosenfeld-Gröbner[9]. Differential elimination rewrites the inputted system of differential equations to another equivalent system, according to (order of terms). Here, we provide an example of differential elimination, as shown below, according to Boulier[12].

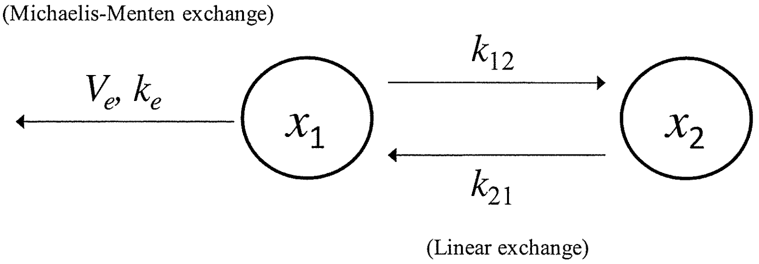

Figure 1: Schematic representation of the model

The model is composed of two state variables, x_1 and x_2. We assumed that the time-series data for one of the variable, x_1, are obtained.

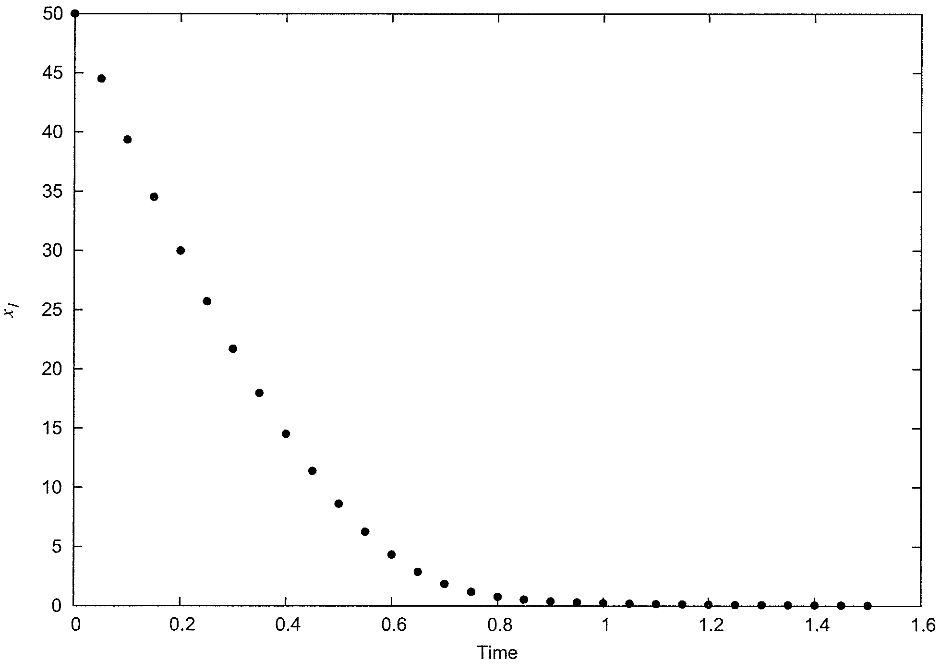

Figure 2: Reference curve

According to the kinetics of the model (Eqn. (1)), a reference curve of one variable, x_1, was generated for $0 \leq t \leq 1.5$ with intervals of 0.05, under the following conditions: $x_1(0) = 50.0, x_2(0) = 0.0, Ve = 101.0, k_{12} = 0.5, k_{21} = 3.0$ and $k_e = 7.0$.

Assume a model of two variables, x_1 and x_2, as schematically depicted in Fig. 1, with the corresponding system of differential equations expressed as follows:

$$\begin{cases} \dot{x_1} = -k_{12}x_1 + k_{21}x_2 - \frac{V_e x_1}{k_e + x_1} \\ \dot{x_2} = k_{12}x_1 - k_{21}x_2 \end{cases} \tag{1}$$

where k_{12}, k_{21}, k_e and V_e are some constants. Two molecules are assumed to bind according to Michaelis-Menten kinetics.

Here we assume that the time-series of only one variable, x_1, can be observed. x_2 is assumed to be non-observed; however, we assumed that $x_2(0) = 0$ was known. According to the model in Fig. 1, a reference curve of one variable, x_1, was generated in Fig. 2. Among the parameters in the model, the values of three parameters, k_{12}, k_{21}, and V_e, were estimated, and the values of the remaining parameters were set to the same values as those used in the generation of the reference curve of Fig. 2.

2.3 Differential Elimination

The differential elimination then produces the following two equations equivalent to the above system.

$$\begin{cases} \dot{x_1}(k_{21}+x_1)+k_{21}x_1^2+(k_{12}+V_e)x_1-k_{21}(k_e+x_1)x_2=0 \\ \ddot{x_1}(x_1+k_e)^2+(k_{12}+k_{21})\dot{x_1}(x_1+k_e)^2+V_e\dot{x_1}k_e+k_{21}V_ex_1(x_1+k_e)=0 \end{cases} \tag{2}$$

As a consequence, the latter two equations should be zero for any solution of (1). The latter to equations, respectfully, called $C_{1,t}$ and $C_{2,t}$ in the following, will be used to define our error estimation, based on the evaluation of $|C_{1,t}|+|C_{2,t}|$.

System (2) can be computed in Maple 14, using the following commands:

```
> with(DifferentialAlgebra):
> sys := [
>     x1[t] - ( -k12*x1 + k21*x2 - Ve*x1/(ke+x1)),
>     x2[t] - ( k12*x1 - k21*x2)
>        ];
                                         Ve x1
      sys := [x1[t] + k12 x1 - k21 x2 + -------, x2[t] - k12 x1 + k21 x2]
                                        ke + x1

> R := DifferentialRing(blocks=[x2,x1,k12(),k21(),Ve(),ke()], derivations=[t]);
                             R := differential_ring

> Ids := RosenfeldGroebner( numer(sys), denom(sys), R,
>                            basefield=field(generators=[k12,k21,Ve,ke]));
                      Ids := [regular_differential_chain]

> eqs := Equations(Ids[1]);
eqs := [
                                                        2
    k21 x2 x1 + k21 x2 ke - x1[t] x1 - x1[t] ke - k12 x1  - k12 x1 ke - Ve x1,


               2                                 2           2
    x1[t, t] x1  + 2 x1[t, t] x1 ke + x1[t, t] ke  + x1[t] x1  k12


               2                                                           2
     + x1[t] x1  k21 + 2 x1[t] x1 k12 ke + 2 x1[t] x1 k21 ke + x1[t] k12 ke


                   2                   2
     + x1[t] k21 ke  + x1[t] Ve ke + x1  k21 Ve + x1 k21 Ve ke]
```

2.4 Simplification

In general, the problem of reducing the evaluation complexity (additions, multiplications) is difficult and requires a large number of computer operations (a.k.a. a high algorithmic complexity). Moreover, the evaluation complexity of the Rosenfeld-Gröbner output tends to be exponential in the evaluation complexity of the input, especially when using elimination rankings, as in this case. Consequently, before directly applying techniques such as factorization, Horner schemes, common sub expression detection, etc. for reducing the evaluation complexity, we try to use the knowledge we already have on the initial ODE system.

We now describe a preprocessing step that facilitates the evaluation of $\bar{C}_{DE} = |C_{1,t}| + |C_{2,t}|$.

The expressions of $C_{1,t}$ and $C_{2,t}$ given in (2) are not the expressions originally computed by the Rosenfeld-Gröbner algorithm. Indeed, the Rosenfeld-Gröbner algorithm outputs expanded expressions.

Thus, using the Rosenfeld-Gröbner outputs, one has to evaluate the following expression, $\bar{C}_{DE}$:

$$\begin{aligned}\bar{C}_{DE} \quad = \quad & \left|-k_{21}x_2k_e - k_{21}x_2x_1 + \dot{x}_1k_e + \dot{x}_1x_1 + k_{12}x_1k_e + k_{12}x_1^2 + V_ex_1\right| \\ & + \left|k_{21}k_eV_ex_1 + 2k_ek_{12}x_1\dot{x}_1 + 2k_{21}k_e\dot{x}_1x_1 + 2k_e\ddot{x}_1x_1 + k_{12}\dot{x}_1k_e^2\right. \\ & \left. + k_eV_e\dot{x}_1 + k_{12}x_1^2\dot{x}_1 + k_{21}k_e^2\dot{x}_1 + k_{21}x_1^2\dot{x}_1 + k_{21}x_1^2V_e + \ddot{x}_1k_e^2 + \ddot{x}_1x_1^2\right|\end{aligned} \qquad (3)$$

requiring 18 additions + 46 multiplications (+2 function evaluations for the absolute value). These operations represent the evaluation complexity of the expression $\bar{C}_{DE}$.

Since the expressions of $C_{1,t}$ and $C_{2,t}$ were computed from an ODE system involving the denominator $k_e + x_1$, from a Michaelis-Menten factor, the expression $k_e + x_1$ can be likely be factorized. By introducing a new variable, $d_e = k_e + x_1$, and applying the substitution $k_e \longrightarrow d_e - x_1$ in the previous expression of $\bar{C}_{DE}$, one gets

$$\begin{aligned}\bar{C}_{DE} \quad = \quad & \left|-k_{21}x_2d_e + \dot{x}_1d_e + k_{12}x_1d_e + V_ex_1\right| \\ & + \left|k_{21}V_ex_1d_e + k_{12}\dot{x}_1d_e^2 + V_e\dot{x}_1d_e - V_e\dot{x}_1x_1 + k_{21}\dot{x}_1d_e^2 + \ddot{x}_1d_e^2\right|\end{aligned} \qquad (4)$$

requiring 9 additions + 21 multiplications.

Please note that the last expression of $\bar{C}_{DE}$ does not involve k_e anymore, which shows that the variable k_e only appears in $\bar{C}_{DE}$ in the term $k_e + x_1$.

This trick with the denominators has divided the number of operations by 2. On more complex systems, the benefit can be much greater. It is worth noting that the trick works quite similarly if several denominators are involved and if each denominator linearly involves a parameter that is not involved in the other denominators. More precisely, if one has n denominators of the form $k_i + f_i$, and if k_i is not involved in any f_i, then one performs n substitutions $k_i \longrightarrow f_i - d_i$.

Further computations using a Horner scheme can now be accomplished. For example, applying a recursive Horner scheme with decreasing priority on the variables $d_e, x_1, x_2, \dot{x}_1, \ddot{x}_1$ yields:

$$\begin{aligned}\bar{C}_{DE} \quad = \quad & |V_ex_1 - (k_{21}x_2 - \dot{x}_1 - k_{12}x_1)d_e| \\ & + |-V_e\dot{x}_1x_1 + (k_{21}V_ex_1 + V_e\dot{x}_1 + (\ddot{x}_1 + (k_{12} + k_{21})\dot{x}_1)d_e)d_e|\end{aligned} \qquad (5)$$

requiring 9 additions + 12 multiplications.

To finish, further simplification can be achieved using the optimize command of the optimize package in the Computer Algebra software Maple. This last command tries to recognize common expressions in order to compute common subexpressions only once. This command is not very costly, since it is based on easy heuristics. In our case, it yields the sequence of commands:

$$\begin{aligned} t7 &= |V_e x_1 - (k_{21}x_2 - \dot{x_1} - k_{12}x_1)d_e|, \\ t8 &= V_e\dot{x_1}, \\ t19 &= |-t8x_1 + (k_{21}V_e x_1 + t8 + (\ddot{x_1} + (k_{12}+k_{21})\dot{x_1})d_e)d_e|, \\ C_{DE} &= t7 + t19 \end{aligned} \tag{6}$$

requring 9 additions + 11 multiplications + 4 assignments. Note that the last gain here is only 1 multiplication, but can be higher on larger systems.

All previous operations can be automated in Maple (see appendix A for the complete set of Maple commands); the *C* command of the optimize package yields the C code as

```
t7 = fabs(Ve*x1-(k21*x2-x1t-k12*x1)*de);
t8 = Ve*x1t;
t19 = fabs(-t8*x1+(k21*Ve*x1+t8+(x1tt+(k12+k21)*x1t)*de)*de);
E = t7+t19;.
```

2.5 Introduction of Constraints

The objective function for parameter optimization in this study is composed of two terms: one is the standard error function between the estimated and monitored data, and the other is the constraints obtained by differential elimination. The error function is defined as follows: Suppose $x^c_{i,t}$ is the time-series data at time t of x_i, simulated by using the estimated parameter values and the model equations by integration, and $x^m_{i,t}$ represents the monitored data at time t. The sum of the absolute values of the relative error between $x^c_{i,t}$ and $x^m_{i,t}$ gives the averaged relative error over the numbers of monitored variables and time points, E, as a standard error function, i.e.,

$$E = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left|\frac{x^c_{i,t} - x^m_{i,t}}{x^m_{i,t}}\right| \tag{7}$$

where N and T are the numbers of monitored variables and time points, respectively.

Next we define the DE constraints obtained by the differential elimination and simplification procedure. The simplified equivalent system (Eqn. (6)) is composed of x_1, its derivatives ($\dot{x_1}$ and $\ddot{x_1}$), x_2, and the parameters (k_{12}, k_{21}, V_e and k_e). Note that x_2 in Eqn. (6) can be estimated by x_1, the parameters, and $x_2(0)$. The derivatives of variable x_1 can be estimated numerically by the following procedure. First, we obtain two equations by a Taylor expansion of $x_1(t)$,

$$x_1(t+h) = x_1(t) + hx_1'(t) + \frac{h^2}{2}x_1''(t) + \frac{h^3}{6}x_1'''(t) + \cdots, \tag{8}$$

$$x_1(t-h) = x_1(t) - hx_1'(t) + \frac{h^2}{2}x_1''(t) - \frac{h^3}{6}x_1'''(t) + \cdots. \tag{9}$$

Second, we subtract Eqn. (9) from (8),

$$\begin{aligned} x_1(t+h) - x_1(t-h) &= 2hx_1'(t) + \frac{1}{3}h^3x_1'''(t) + \cdots, \\ 2hx_1'(t) &= x_1(t+h) - x_1(t-h) - \frac{1}{3}h^3x_1'''(t) + \cdots, \\ x_1'(t) &= \frac{x_1(t+h) - x_1(t-h)}{2h} - \frac{h^2}{6}x_1'''(t) + \cdots. \end{aligned} \tag{10}$$

Finally, we obtain following approximation, under the assumption of $0 < h < 1$,

$$x_1'(t) = \frac{x_1(t+h) - x_1(t-h)}{2h} + O(h^2). \tag{11}$$

We are able to obtain higher-order derivatives from lower-order derivatives in same way, as mentioned above. For instance, we can estimate second order derivatives of x_1 by using following equation,

$$x_1''(t) = \frac{x_1'(t+h) - x_1'(t-h)}{2h} + O(h^2). \tag{12}$$

The value of the simplified equivalent system (Eqn. (6)) can be calculated by the substitution of the observed x_1, its numerically the estimated derivatives, estimated x_2, and the parameter values estimated by the numerical parameter optimizing procedure. In general, Differential Elimination rewrites the original system of differential equations into an equivalent system, which means both systems have the same solutions. This clearly shows that the evaluated values of the equivalent system will be zero with exactly estimated parameter sets, time-series data without noise, and derivatives. Thus, the equivalent system can be regarded as a kind of objective function that expresses the difference between the monitored and estimated data. In this study, we express DE Constraint (C_{DE}), as the average of the linear combination of the equation in the equivalent system over the number of equations and time points, as follows:

$$C_{DE} = \frac{1}{LT} \sum_{l=1}^{L} \sum_{t=1}^{T} |C_{l,t}| \tag{13}$$

where L and T are the numbers of equivalent equations and time points, respectively. Finally, we introduce C_{DE} $\bar{C}_{DE}$, which is simplified as C_{DE}, into the objective function, F, in combination with E, as:

$$F = \alpha E + (1-\alpha) C_{DE} \tag{14}$$

where $\alpha (0 \leq \alpha \leq 1)$ is the weight of the two functions. As a result, our computational task is to find a set of parameter values that minimize F. When we apply the simplification procedure (see 2.4), then $\frac{1}{LT}\bar{C}_{DE}$ is used instead of C_{DE}.

The weighting factor α in the objective function F is estimated from the slope of the Pareto-optimal solutions. First, we obtained some parameter sets (in the case study, we obtained 200 kinds of parameter sets) by the compute_parameter_set function, under the conditions of $\delta = 1.0$ and the tentative value of α $ta = 1$ (this means we used the classical objective function, i.e. $F = E$). Second, we selected the Pareto-optimal solutions from the list of estimated parameter sets, by the select_pareto_optimal_solutions function. By fitting the linear function $C = aE + b$ to the selected the Pareto-optimal solutions, we obtained the slope of Pareto-optimal solutions, a. Finally, we estimated the value of α from the slope a. The detailed algorithms for estimating the value of α are shown in Algorithm 1 and 2. Fig. 3 represents a part of the estimated parameter sets in the case study (the detailed algorithms for the parameter optimization we used for the case study are shown in 2.6), the Pareto-optimal solutions, and the fitted line for the Pareto-optimal solutions. We obtained the slope $a = 20.7653$ for the case study, and the value of α was estimated as $\alpha = 0.95406$.

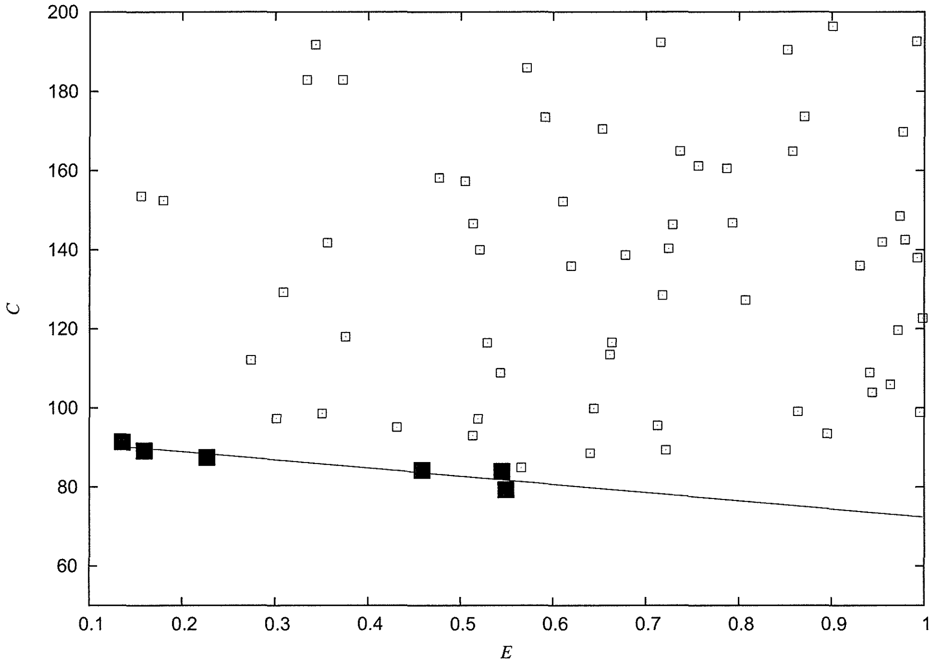

Figure 3: Slope of Pareto-optimal solutions

The empty squares (□) indicate the set of evaluated values, E and C_{DE}. The filled squares (■) show the Pareto-optimal solutions, and the line represents the fitted line for Pareto-optimal solutions.

Algorithm 1 Estimate value of weighting factor α

Function : estimate_alpha(δ, n, ta)

Input : error tolerance δ, number of trials n, and tentative value of α ta

Return : estimated value of weighting factor α

```
$RES \leftarrow$ compute_parameter_set($ta$, $\delta$)
$P \leftarrow$ select_pareto_optimal_solutions($RES$)
$EV \leftarrow \phi$
$CV \leftarrow \phi$
$n$ size of $P$
for $i = 0$ to $n$ do
    $EV \leftarrow EV$ union $E(R_i)$
    $CV \leftarrow CV$ union $C(R_i)$
end for
fit $CV_i = -aEV_i + b$ from $EV$ and $CV$ by using least square method
return $a/(a+1)$
```

Algorithm 2 Select Pareto-optimal solutions

Function : select_pareto_optimal_solutions(R)
Input : R set of estimated parameters
Return : Pareto-optimal solutions (P)

```
$P \leftarrow \phi$
$EV \leftarrow \phi$
$CV \leftarrow \phi$
$n$ size of $R$
for $i = 0$ to $n$ do
    $EV \leftarrow EV$ union $E(R_i)$
    $CV \leftarrow CV$ union $C(R_i)$
end for
for $i = 0$ to $n$ do
    Flag $lp$ = true
    for $j = 0$ to $n$ do
        if $!(EV_i \leq EV_j$ and $CV_i \leq CV_j)$ then
            $lp \leftarrow$ false
        end if
    end for
    if $lp$ then
        $P \leftarrow P$ union $R_i$
    end if
end for
return $P$
```

2.6 Optimization Algorithm

Our approach can be applied to many kinds of parameter optimizing procedures, such as the Gradient-based method and the evolutionary optimizing method, including the Modified Powell method[18, 19], Genetic Algorithms[20, 21], and Particle Swarm Optimization[22, 23], by modifying the objective function (cost function) only[16].

Here, we applied our approach to Real-coded Genetic Algorithms[24, 25, 26], to demonstrate its ability. The detailed algorithms used to analyze the case study (Fig. 1 and 2) are shown in Algorithm 3 to 5.

Let us explain the differences between our procedure and the classical constraint E. First of all, by using $\alpha = 1$, one obtains a classical genetic algorithm using the relative error E, since we have $F = E$ when $\alpha = 1$. Second, when using $\alpha < 1$, each parameter set k returned by the compute_parameter_sets satisfies $E(k) \leq \delta$, as in the classical procedure. However, the manner in which the population evolves (in the compute_next_generation) depends on the function F. To summarize, the objective function F is only used to direct the evolution of the population, by not using the objective function F to select the final candidates, and thus it makes sense to compare the parameter sets computed in the classical procedure and in our procedure.

2.7 Results

To evaluate the ability of our procedure, we performed a simulation study by using the objective function with and without the newly developed DE constraints, by estimating the kinetic parameters in Eqn. (1).

Algorithm 3 Modify the parameter set K by computing the next generation

Function compute_next_generation(α, K)

Input : the weighting factor α, a parameter set K

```
$n$ size of $K$
denote $K = \{k_1,\dots,k_n\}$
compute $1 \le s \le n$ such that $k_s$ is the one best element according to the $F$ function (i.e. $F(k_s)$ is the minimum of $F(k_1),\dots,F(K_n)$)
pick a random number $r$ such that $1 \le r \le n$, and $r$ is different from $s$
mix $k_s$ and $k_r$ and compute a new set $k' = \{K'_1,\dots,k'_n\}$
$K' \leftarrow K'$ union $\{k_s\}$
modify $k$ by replacing $k_s$ and $k_r$ by the two best elements of $K'$ according to the $F$ function
```

Algorithm 4 Optimization process

Function : compute_one_parameter_set(α, δ, *pop*, *gen*)

Input : the weighting factor α, the error tolerance *delta* for function F, the population size of GA *pop*, the maximum generation counts *gen*

Return : a set containing zero or one parameter set

```
create a set $K$ containing $pop$ random parameter sets
for $i = 1$ to $gen$ do
  compute_next_generation($alpha$, $K$)
  if an element $k$ in $K$ satisfies $E(k) \le \delta$ then
    return $k$
  end if
end for
return $\phi$
```

Algorithm 5 generate a list of estimated parameter sets

Function : compute_parameter_sets(α, δ, *pop*, *gen*, *trials*)

Input : the weighting factor α, the error tolerance δ for function F, the population size of GA *pop*, the maximum generation counts *gen*, the trial number *trials*

Return : a list of parameter sets

```
$RES \leftarrow \phi$
for $i = 1$ to $trials$ do
  $RES \leftarrow RES$ union compute_one_parameter_set($alpha$, $delta$, $pop$, $gen$)
end for
return $RES$
```

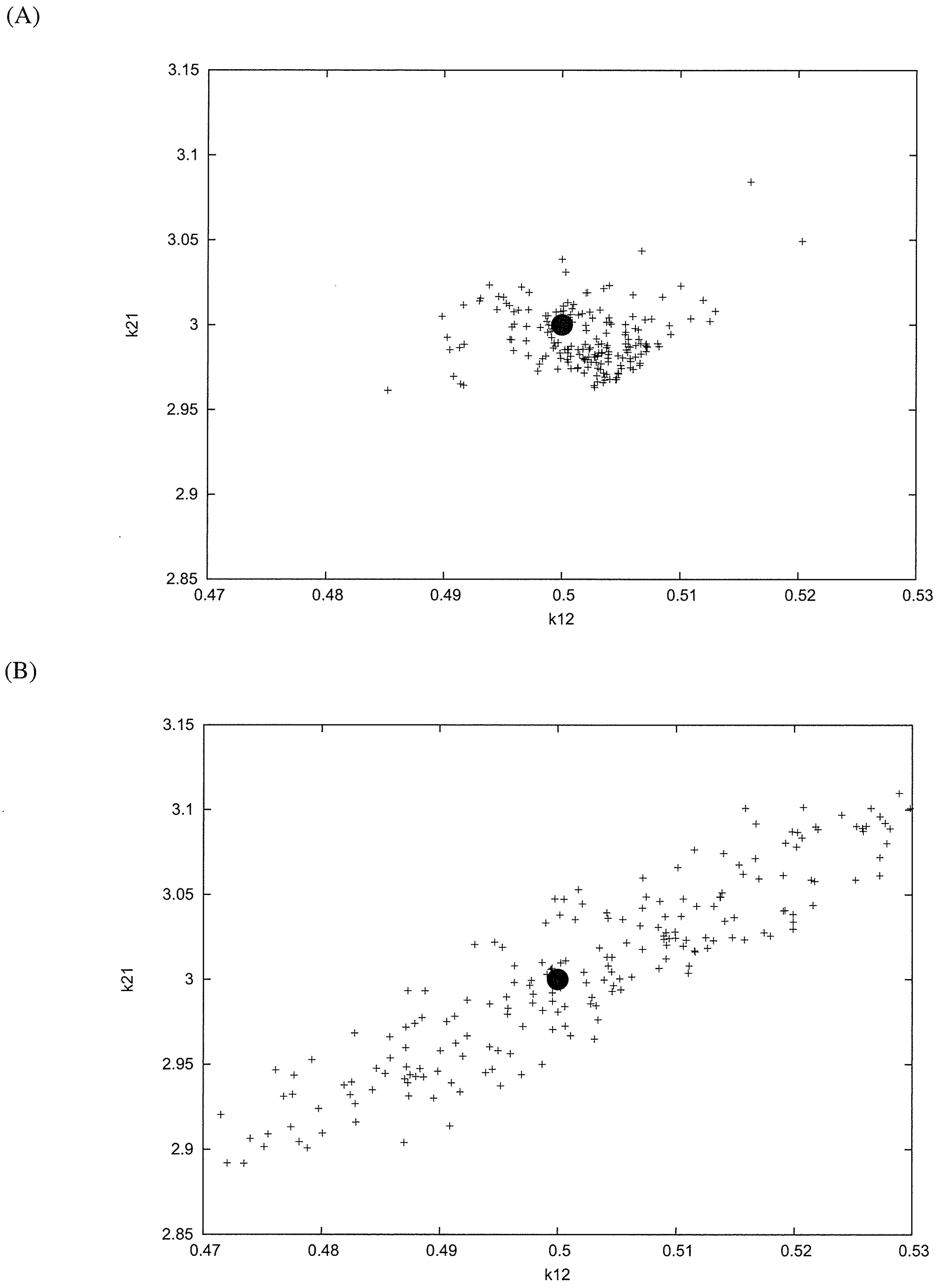

Figure 4: Comparison of parameter value clouds estimated by the classical or our proposed procedure, (A) with and (B) without DE constraints

The given values are as follows: $x_2(0) = 0.0$ and $k_e = 7.0$. The black circles indicate the correct parameter set.

Here, we assume that the time-series of only one variable, x_1, can be observed. According to the model, the reference curve of one variable, x_1, was generated in Fig. 2. Among the parameters in the model, the values of three parameters, k_{12}, k_{21}, and V_e, were estimated, and the values of the remaining parameters were set to the same values as those used in the generation of the reference curve.

The introduction of DE constraints into the objective function was quite effective, in the comparison with the distributions of the parameter values estimated with and without DE constraints (see Fig. 4). Indeed, the distribution of the estimated k_{12} and k_{21} values was highly concentrated around the correct values by the estimation with the introduction (Fig. 4 (A)), while the estimated parameters were widely distributed by the estimation without the introduction of DE constraints (Fig. 4 (B)).

3 Discussion

The accuracy of parameter estimation was clearly improved by the introduction of DE constraints into the objective function of the numerical parameter optimizing method. Indeed, the parameter value sets estimated with the introduction of DE constraints into the objective function were sharply distributed near the correct values, in contrast to the wide distribution without the introduction. In general, the derivatives included the information on the curve form of the observed time-series data, such as slope, extremal point and inflection point. This indicates that the new objective function we proposed estimates the difference of not only the values but also the forms between the measured and estimated data, while the classical objective function estimates only the value difference. Note that the DE constraint is rationally reduced from the original system of differential equations for a given model, in a mathematical sense. Thus, our approach is expected to become a general approach for parameter optimization to improve the parameter accuracy.

As expected, the new objective function requires more computational time, in comparison with an objective function with only a standard error function, due to the increase of the function in the DE constraints. In equivalent systems derived by Differential Elimination, the number of terms and operators frequently increases, and this may make the application of our procedure to a complex or large system difficult, without simplification of the equivalent system. To overcome the difficulty in the complex system, we applied simplification by symbolic computation (see 2.4). For instance, we could estimate the kinetic parameters in the negative feed-back oscillator model[27, 28, 29] by using the simplification procedure[17], while the estimation without the simplification failed, due to the immense computational time.

Another possible way to overcome the difficulty in complex models is to approximate the DE constraint. In the DE constraint, the terms with a higher order of derivatives in the differential equations generally appeared in the equivalent system. The magnitudes of the estimated values of the higher order derivatives were relatively smaller than those of the lower order derivatives. Although our procedure was useful, even for noisy data in a simple model[15], the estimated values of the higher order derivatives for noisy data may become large in this case. However, some techniques are frequently used for smoothing noisy data, and after smoothing, the values of the higher order derivatives may be smaller. If the terms with higher order derivatives can be neglected in the estimation, then the computational time may be reduced. Further studies to improve the computational time by approximation of the DE constraint will be reported in the near future.

A Implementation of Simplification

The following commands use the new DifferentialAlgebra package, and thus require Maple 14 to work.

```
    |\^/|     Maple 14 (X86 64 LINUX)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2010
 \  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
  libname := "/home/calforme/lemaire/Triade/src/lib", "/usr/local/maple14/lib"

> with(DifferentialAlgebra):
> with(CodeGeneration):
> with(codegen):
> sys := [
>     x1[t] - ( -k12*x1 + k21*x2 - Ve*x1/(ke+x1)),
>     x2[t] - ( k12*x1 - k21*x2)
>        ];
                                         Ve x1
      sys := [x1[t] + k12 x1 - k21 x2 + -------, x2[t] - k12 x1 + k21 x2]
                                        ke + x1

>
> R := DifferentialRing(blocks=[x2,x1,k12(),k21(),Ve(),ke()], derivations=[t]);
                             R := differential_ring

> Ids := RosenfeldGroebner( numer(sys), denom(sys), R,
>                           basefield=field(generators=[k12,k21,Ve,ke]));
                      Ids := [regular_differential_chain]

> eqs := Equations(Ids[1]);
eqs := [

                                                        2
    k21 x2 x1 + k21 x2 ke - x1[t] x1 - x1[t] ke - k12 x1  - k12 x1 ke - Ve x1,

               2                                  2            2
    x1[t, t] x1  + 2 x1[t, t] x1 ke + x1[t, t] ke  + x1[t] x1  k12

               2                                                           2
     + x1[t] x1  k21 + 2 x1[t] x1 k12 ke + 2 x1[t] x1 k21 ke + x1[t] k12 ke

                   2                   2
     + x1[t] k21 ke  + x1[t] Ve ke + x1  k21 Ve + x1 k21 Ve ke]

# One performs some necessary renaming
> eqs := subs(x1[t,t]=x1tt, x1[t]=x1t, x1[]=x1, x2[t]=x2t, x2[]=x2, eqs);
                                                        2
eqs := [k21 x2 x1 + k21 x2 ke - x1t x1 - x1t ke - k12 x1  - k12 x1 ke - Ve x1,

           2                         2         2             2
    x1tt x1  + 2 x1tt x1 ke + x1tt ke  + x1t x1  k12 + x1t x1  k21

                                                      2             2
     + 2 x1t x1 k12 ke + 2 x1t x1 k21 ke + x1t k12 ke  + x1t k21 ke

                     2
     + x1t Ve ke + x1  k21 Ve + x1 k21 Ve ke]

> toTransform := [ result = abs(eqs[1]) + abs(eqs[2]) ];
toTransform := [result =

                                                       2
    | -k21 x2 x1 - k21 x2 ke + x1t x1 + x1t ke + k12 x1  + k12 x1 ke + Ve x1 |

                2                         2         2             2
     + | x1tt x1  + 2 x1tt x1 ke + x1tt ke  + x1t x1  k12 + x1t x1  k21

                                                     2             2
     + 2 x1t x1 k12 ke + 2 x1t x1 k21 ke + x1t k12 ke  + x1t k21 ke

                         2
     + x1t Ve ke + x1  k21 Ve + x1 k21 Ve ke |]

> cost(toTransform);
         18 additions + 2 functions + 46 multiplications + assignments

# One guesses that the denominator ke+x1 appears in many places.
# To make it appear, one introduces de = ke + x1
# and performs the substitution ke -> de - x1
> toTransform2 := subs(ke = de - x1, toTransform):
> toTransform2 := simplify(toTransform2);
toTransform2 := [result = | -k21 x2 de + x1t de + k12 x1 de + Ve x1 | + |

           2             2             2
    x1tt de  + x1t k12 de  + x1t k21 de  + x1t Ve de - x1t Ve x1 + x1 k21 Ve de

    |]

> cost(toTransform2);
          9 additions + 2 functions + 21 multiplications + assignments

> eqs2 := subs(ke = de - x1, eqs):
> eqs2 := simplify(eqs2);
eqs2 := [k21 x2 de - x1t de - k12 x1 de - Ve x1,

           2             2             2
    x1tt de  + x1t k12 de  + x1t k21 de  + x1t Ve de - x1t Ve x1 + x1 k21 Ve de

    ]

# One remarks that ke does not appear anymore.
# Using horner and optimization.
> eqs3 := convert(eqs2, horner, [de,x1,x2,x1t,x1tt]);
eqs3 := [-Ve x1 + (k21 x2 - x1t - k12 x1) de,

    -x1t Ve x1 + (x1 k21 Ve + x1t Ve + (x1tt + (k12 + k21) x1t) de) de]

> toTransform := [ result = abs(eqs3[1]) + abs(eqs3[2]) ];
toTransform := [result = | Ve x1 - (k21 x2 - x1t - k12 x1) de |

     + | -x1t Ve x1 + (x1 k21 Ve + x1t Ve + (x1tt + (k12 + k21) x1t) de) de |]

> cost( toTransform);
          9 additions + 2 functions + 12 multiplications + assignments

> out := optimize( toTransform );
out := t7 = | Ve x1 - (k21 x2 - x1t - k12 x1) de |, t8 = x1t Ve,

    t19 = | -t8 x1 + (x1 k21 Ve + t8 + (x1tt + (k12 + k21) x1t) de) de |,

    result = t7 + t19
```

```
> cost([out]);
         2 functions + 11 multiplications + 9 additions + 4 assignments

# One generates the C code
> C( [ out ]);
      t7 = fabs(Ve*x1-(k21*x2-x1t-k12*x1)*de);
      t8 = x1t*Ve;
      t19 = fabs(-t8*x1+(x1*k21*Ve+t8+(x1tt+(k12+k21)*x1t)*de)*de);
      result = t7+t19;
> quit
memory used=32.7MB, alloc=28.4MB, time=0.18
```

Acknowledgments

This work was partially supported by a project grant, 'Development of Analysis Technology for Induced Pluripotent Stem (iPS) Cell', from The New Energy and Industrial Technology Development Organization (NEDO).

References

[1] Kitano, H.: System Biology: A Brief Overview, *Science* Vol. 295, no. 5560, March (2002) 1662 - 1664.

[2] Nocedal, J., Wright, S. J.: Numerical Optimization, *Springer-Verlag, New York*, (1999).

[3] Denis-Vidal, L., Joly-Blanchard, G., and Noiret, C.: System identifiability (symbolic computation) and parameter estimation (numerical computation). In *Numerical Algorithms*, volume 34, pages 282–292, 2003.

[4] Boulier, F., Denis-Vidal, F., Henin, T., and Lemaire, F.: LÉPISME. In *Proceedings of the ICPSS conference*, 2004. `http://hal.archives-ouvertes.fr/hal-00140368`.

[5] Ritt, J.F.: Differential Algebra, *Dover Publications Inc., New York*, (1950)

[6] Kolchin, E.R.: Differential Algebra and Algebraic Groups, *Academic Press, New York*, (1973)

[7] Seidenberg, A.: An elimination theory for differential algebra. *Univ. California Publ. Math. (New Series)*, 3:31–65, 1956.

[8] Wu, W.T.: On the foundation of algebraic differential geometry. *Mechanization of Mathematics, research preprints*, 3:2–27, 1989.

[9] Boulier, F., Lazard, D., Ollivier, F., Petitot, M.: Representation for the radical of a finitely generated differential ideal, *Proceedings of ISSAC 1995*, (1995), 158-166.

[10] Boulier, F., Lazard, D., Ollivier, F., and Petitot, M.: Computing representations for radicals of finitely generated differential ideals. *Journal of AAECC*, 20(1):73–121, 2009. (1997 Techrep. IT306 of the LIFL).

[11] Boulier, F.: The BLAD libraries. `http://www.lifl.fr/~boulier/BLAD`, 2004.

[12] Boulier, F.: Differential Elimination and Biological Modeling, *Johann Radon Institute for Computational and Applied Mathematics (RICAM) Book Series*, Vol 2, (2007) 111-139.

[13] Boulier, F.: Differential algebra and system modeling in cellular biology, *Proceedings of Algebraic Biology 2008*, (2008) 22-39.

[14] Nakatsui, M. and Horimoto K.: Parameter Optimization in the network dynamics including unmeasured variables by the symbolic-numeric approach, *Proc. of the Third International Symposium on Optimization and Systems Biology (OSB'09)*, pp. 245-253, (2009).

[15] Nakatsui, M. and Horimoto, K.: Improvement of Estimation Accuracy in Parameter Optimization by Symbolic Computation, *Proceedings of IEEE Multi-conference on Systems and Control*, in press.

[16] Nakatsui, M., Horimoto, K., Okamoto, M., Tokumoto, Y. and Miyake, J.: Parameter Optimization by Using Differential Elimination: a General Approach for Introducing Constraints into Objective Functions, *BMC Systems Biology*, in press.

[17] Nakatsui, M., Horimoto, K., Lemaire, F., Ürgüplü, A., Sedoglavic, F., Boulier, F.:Brute force meets Bruno force in parameter optimization: Introduction of novel constraints for parameter accuracy improvement by symbolic computation, *IET Systems Biology*, in press.

[18] Powell M.J.D.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, (1954), 7:142-162.

[19] Powell M.J.D.: On the calculation of orthogonal vectors. *Computer Journal*, (1968), 11:302-304.

[20] Holland, J. H.: Adaptation in Natural and Artificial Systems, *The University of Michigan Press, Ann Arbor, MI* (1975).

[21] Goldberg, D. D.: Genetic Algorithms in Search, Optimization and Machine Learning, *Addison-Wesley Longman Publishing Co., Inc., Boston. MA.* (1989).

[22] Eberhart, R., and Kennedy, J.: A New Optimizer Using Particle Swarm Theory. em Proc. of Sixth International Symposium on Micro Machine and Human Science (Nagoya Japan), IEEE Service Center, Piscataway, NJ, (1995), 39-43.

[23] Kennedy J and Eberhart R: Particle swarm optimization. *Proc. IEEE International Conference on Neural Networks (Perth, Australia)*, IEEE Service Center, Piscataway, NJ, (1995), pp. IV: 1942-1948.

[24] Jonikow, C. Z. and Michalewicz, Z.: An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms, *Proceedings of the Fourth International Conference on Genetic Algorithms*, (1991) 31-36.

[25] Ono, I. and Kobayashi, S.: A real-coded genetic algorithm for function optimization using unimodal distribution crossover, *Proc 7th ICGA*, (1997) 249–253.

[26] Satoh, H., Ono, I. and Kobayashi, S.: A new generation alternation model of genetic algorithm and its assessment, *J. of Japanese Society for Artificial Intelligence*, 15(2) (1997) 743-744.

[27] Novák, B., Tyso,n J.J.: Design principles of biochemical oscillators, *Nat. Rev. Mol. Cell Biol.*, (2008), 9(12) 981-991.

[28] Kwon, Y. K. and Cho, K. H.: Quantitative analysis of robustness and fragility in biological networks based on feedback dynamics, *Bioinformatics*, (2008), 24 (7), 987-994.

[29] Tyson, J. J., Chen, K. C., and Novák, B.: Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell, *Curr Opin Cell Biol.* Apr;15(2):221-31 (2003)

HEPATOLOGY
Official Journal of the American Association for the Study of Liver Diseases
AMERICAN ASSOCIATION FOR THE STUDY OF LIVER DISEASES
AASLD

Discrete Nature of EpCAM$^+$ and CD90$^+$ Cancer Stem Cells in Human Hepatocellular Carcinoma

Taro Yamashita,[1] Masao Honda,[1] Yasunari Nakamoto,[1] Masayo Baba,[1] Kouki Nio,[1] Yasumasa Hara,[1] Sha Sha Zeng,[1] Takehiro Hayashi,[1] Mitsumasa Kondo,[1] Hajime Takatori,[1] Tatsuya Yamashita,[1] Eishiro Mizukoshi,[1] Hiroko Ikeda,[1] Yoh Zen,[1] Hiroyuki Takamura,[1] Xin Wei Wang,[2] and Shuichi Kaneko[1]

Recent evidence suggests that hepatocellular carcinoma (HCC) is organized by a subset of cells with stem cell features (cancer stem cells; CSCs). CSCs are considered a pivotal target for the eradication of cancer, and liver CSCs have been identified by the use of various stem cell markers. However, little information is known about the expression patterns and characteristics of marker-positive CSCs, hampering the development of personalized CSC-targeted therapy. Here, we show that CSC markers EpCAM and CD90 are independently expressed in liver cancer. In primary HCC, EpCAM$^+$ and CD90$^+$ cells resided distinctively, and gene-expression analysis of sorted cells suggested that EpCAM$^+$ cells had features of epithelial cells, whereas CD90$^+$ cells had those of vascular endothelial cells. Clinicopathological analysis indicated that the presence of EpCAM$^+$ cells was associated with poorly differentiated morphology and high serum alpha-fetoprotein (AFP), whereas the presence of CD90$^+$ cells was associated with a high incidence of distant organ metastasis. Serial xenotransplantation of EpCAM$^+$/CD90$^+$ cells from primary HCCs in immune-deficient mice revealed rapid growth of EpCAM$^+$ cells in the subcutaneous lesion and a highly metastatic capacity of CD90$^+$ cells in the lung. In cell lines, CD90$^+$ cells showed abundant expression of c-Kit and *in vitro* chemosensitivity to imatinib mesylate. Furthermore, CD90$^+$ cells enhanced the motility of EpCAM$^+$ cells when cocultured *in vitro* through the activation of transforming growth factor beta (TGF-β) signaling, whereas imatinib mesylate suppressed *TGFB1* expression in CD90$^+$ cells as well as CD90$^+$ cell-induced motility of EpCAM$^+$ cells. *Conclusion*: Our data suggest the discrete nature and potential interaction of EpCAM$^+$ and CD90$^+$ CSCs with specific gene-expression patterns and chemosensitivity to molecular targeted therapy. The presence of distinct CSCs may determine the clinical outcome of HCC. (HEPATOLOGY **2012;00:000–000)**

The cancer stem cell (CSC) hypothesis, which suggests that a subset of cells bearing stem-cell–like features is indispensable for tumor development, has recently been put forward subsequent to advances in molecular and stem cell biology. Liver cancer, including hepatocellular carcinoma (HCC), is a leading cause of cancer death worldwide.[1] Recent studies have shown the existence of CSCs in liver cancer cell lines and primary HCC specimens using various stem cell markers.[2-7] Independently, we have identified novel HCC subtypes defined by the hepatic stem/progenitor cell markers,

Abbreviations: 5-FU, fluorouracil; Abs, antibodies; AFP, alpha-fetoprotein; CK-19, cytokeratin-19; CSC, cancer stem cell; DNs, dysplastic nodules; EMT, epithelial mesenchymal transition; EpCAM; epithelial cell adhesion molecule; FACS, fluorescent-activated cell sorting; HBV, hepatitis B virus; HCC, hepatocellular carcinoma; HCV, hepatitis C virus; HSCs, hepatic stem cells; IF, immunofluorescence; IHC, immunohistochemistry; IR, immunoreactivity; MDS, multidimensional scaling; NBNC, non-B, non-C hepatitis; NOD/SCID, nonobese diabetic, severe combined immunodeficient; NT, nontumor; OV-1, ovalbumin 1; qPCR, quantitative real-time polymerase chain reaction; SC, subcutaneous; Smad3, Mothers against decapentaplegic homolog 3; TECs, tumor epithelial cells; TGF-β, transforming growth factor beta; T/N, tumor/nontumor; VECs, vascular endothelial cells; VM, vasculogenic mimicry; VEGFR, vascular endothelial growth factor receptor.

From the [1]Liver Center, Kanazawa University Hospital, Kanazawa, Ishikawa, Japan; and [2]Laboratory of Human Carcinogenesis, Center for Cancer Research, National Cancer Institute, Bethesda, MD.

Received July 9, 2012; revised October 22, 2012; accepted November 6, 2012.

This study was supported by a Grant-in-Aid from the Ministry of Education, Culture, Sports, Science, and Technology of Japan (23590967), a grant from the Japanese Society of Gastroenterology, a grant from the Ministry of Health, Labor, and Welfare, and a grant from the National Cancer Center Research and Development Fund (23-B-5) of Japan. X.W.W. is supported by the Intramural Research Program of the Center for Cancer Research, U.S. National Cancer Institute.

epithelial cell adhesion molecule (EpCAM) and alpha-fetoprotein (AFP), which correlate with distinct gene-expression signatures and prognosis.[8,9] EpCAM$^+$ HCC cells isolated from primary HCC and cell lines show CSC features, including tumorigenicity, invasiveness, and resistance to fluorouracil (5-FU).[10] Similarly, other groups have shown that CD133$^+$, CD90$^+$, and CD13$^+$ HCC cells are also CSCs, and that EpCAM, CD90, and CD133 are the only markers confirmed to enrich CSCs from primary HCCs thus far.[3-5,10]

Although EpCAM$^+$, CD90$^+$, and CD133$^+$ cells show CSC features, such as high tumorigenicity, an invasive nature, and resistance to chemo- and radiation therapy, it remains unclear whether these cells represent an identical HCC population and whether they share similar or distinct characteristics. In this study, we used fluorescent-activated cell sorting (FACS), microarray, and immunohistochemistry (IHC) techniques to investigate the expression patterns of the representative liver CSC markers CD133, CD90, and EpCAM in a total of 340 HCC cases and 7 cases of mesenchymal liver tumors. We further explored gene- and protein-expression patterns as well as tumorigenic capacity of sorted cells isolated from 15 primary HCCs and 7 liver cancer cell lines in an attempt to identify the molecular portraits of each cell type.

Materials and Methods

Clinical Specimens. HCC samples were obtained with informed consent from patients who had undergone radical resection at the Liver Center in Kanazawa University Hospital (Kanazawa, Japan), and tissue acquisition procedures were approved by the ethics committee of Kanazawa University. A total of 102 formalin-fixed and paraffin-embedded HCC samples, obtained from 2001 to 2007, were used for IHC analyses. Fifteen fresh HCC samples were obtained between 2008 and 2012 from surgically resected specimens and an autopsy specimen and were used immediately to prepare single-cell suspensions and xenotransplantation (Table 1). Seven hepatic stromal tumors (three cavernous hemangioma, two hemangioendothelioma, and two angiomyolipoma) were formalin fixed and paraffin embedded and used for IHC analyses.

Table 1. Clinicopathological Characteristics of HCC Cases Used for Xenotransplantation

ID	Age/ Sex	Etiology	Tumor Size (cm)	Histological Grade	AFP (ng/mL)	DCP (IU/mL)
P1	77/M	Alcohol	12.0	Moderate	198	322
P2	61/F	NBNC	11.0	Moderate	12	3,291
P3	66/M	NBNC	2.2	Moderate	13	45
P4	65/M	HCV	4.2	Poor	13,700	25,977
P5	52/M	HBV	6.0	Moderate	29,830	1,177
P6	60/M	HCV	2.7	Poor	249	185
P7	79/F	HBV	4.0	Poor	46,410	384
P8	77/F	NBNC	5.5	Moderate	17,590	562
P9	71/M	Alcohol	7.0	Poor	3,814	607
P10	51/M	HBV	2.2	Well	<10	21
P11	71/M	Alcohol	2.1	Well	<10	11
P12	60/M	HBV	10.8	Poor	323	2,359
P13	66/M	HCV	2.8	Moderate	11	29
P14	71/M	HCV	7.2	Moderate	235,700	375,080
P15	75/M	HBV	5.5	Poor	<10	97

Abbreviation: DCP, des-gamma-carboxy prothrombin.

Additional details of experimental procedures are available in the Supporting Information.

Results

EpCAM, CD133, and CD90 Expression in HCC. We first evaluated the frequencies of three representative CSC markers (EpCAM$^+$, CD90$^+$, and CD133$^+$ cells) in 12 fresh primary HCC cases surgically resected by FACS (representative data shown in Fig. 1A). Clinicopathological characteristics of primary HCC cases are shown in Table 1. We noted that frequency of EpCAM$^+$, CD90$^+$, and CD133$^+$ cells varied between individuals. Abundant CD90$^+$ (7.0%), but almost no EpCAM$^+$, cells (0.06%, comparable to the isotype control) were detected in P2, whereas few CD90$^+$ (0.6%), but abundant EpCAM$^+$, cells (17.5%) were detected in P4. Very small populations of EpCAM$^+$ (0.09%), CD90$^+$ (0.04%), and CD133$^+$ cells (0.05%) were found in P12, but they were almost nonexistent in P8, except for CD90$^+$ cells (0.08%) (Fig. 1A). We further evaluated the expression of EpCAM, CD90, and CD133 in xenografts obtained from surgically resected samples (P13 and P15) and an autopsy sample (P14). As a whole, compared to the isotype control, 7 of 15 HCCs contained definite EpCAM$^+$ cells (46.7%), whereas only 3 HCCs

Address reprint requests to: Taro Yamashita, M.D., Ph.D., Department of General Medicine, Kanazawa University Hospital, 13-1 Takara-Machi, Kanazawa, Ishikawa 920-8641, Japan. E-mail: taroy@m-kanazawa.jp; fax: +81-76-234-4250.

View this article online at wileyonlinelibrary.com.
DOI 10.1002/hep.26168

Potential conflict of interest: Nothing to report.

Additional Supporting Information may be found in the online version of this article.

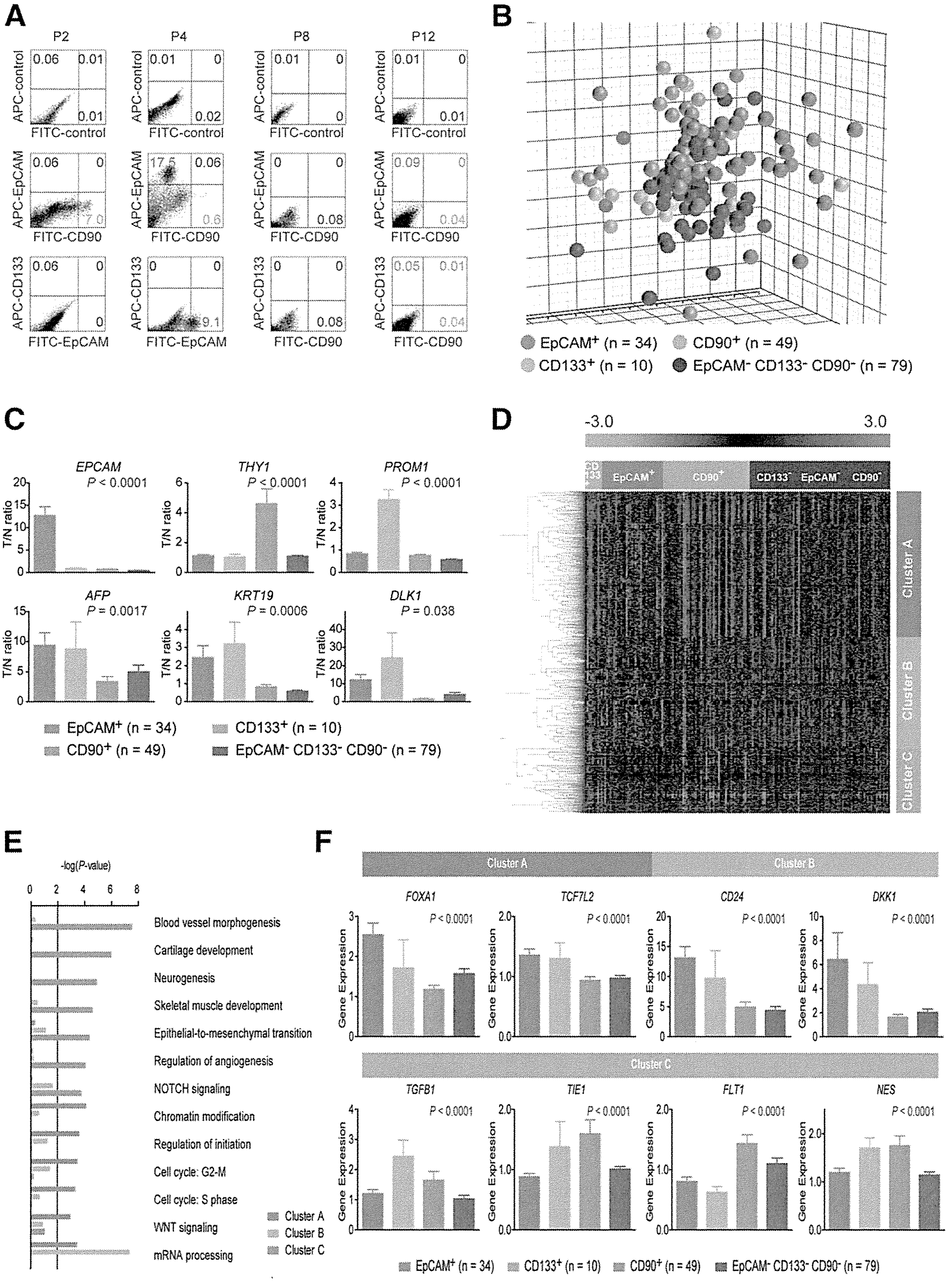

Fig. 1. Gene-expression profiles of CSC marker-positive HCCs. (A) FACS analysis of primary HCCs stained with fluorescent-labeled Abs against EpCAM, CD90, or CD133. (B) Multidimensional scaling analysis of 172 HCC cases characterized by the expression patterns of EpCAM, CD133, and CD90. Red, EpCAM$^+$ CD90$^-$ CD133$^-$ (n = 34); orange, EpCAM$^-$ CD90$^-$ CD133$^+$ (n = 10); light blue, EpCAM$^-$ CD90$^+$ CD133$^-$ (n = 49); blue, EpCAM$^-$ CD90$^-$ CD133$^-$ (n = 79). HCC specimens were clustered in specific groups with statistical significance ($P < 0.001$). (C) Expression patterns of well-known hepatic stem/progenitor markers in each HCC subtype, as analyzed by microarray. Red bar, EpCAM$^+$; orange bar, CD133$^+$; light blue bar, CD90$^+$; blue bar, EpCAM$^-$ CD90$^-$ CD133$^-$. (D) Hierarchical cluster analysis based on 1,561 EpCAM/CD90/CD133-coregulated genes in 172 HCC cases. Each cell in the matrix represents the expression level of a gene in an individual sample. Red and green cells depict high and low expression levels, respectively, as indicated by the scale bar. (E) Pathway analysis of EpCAM/CD90/CD133-coregulated genes. Canonical signaling pathways activated in cluster A (red bar), cluster B (orange bar), or cluster C (light blue bar) with statistical significance ($P < 0.01$) are shown. (F) Expression patterns of representative genes differentially expressed in EpCAM/CD90/CD133 HCC subtypes. Red bar, EpCAM$^+$; orange bar, CD133$^+$; light blue bar, CD90$^+$; blue bar, EpCAM$^-$ CD133$^-$ CD90$^-$.